

1 Создание проекта и разметка MainWindow

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

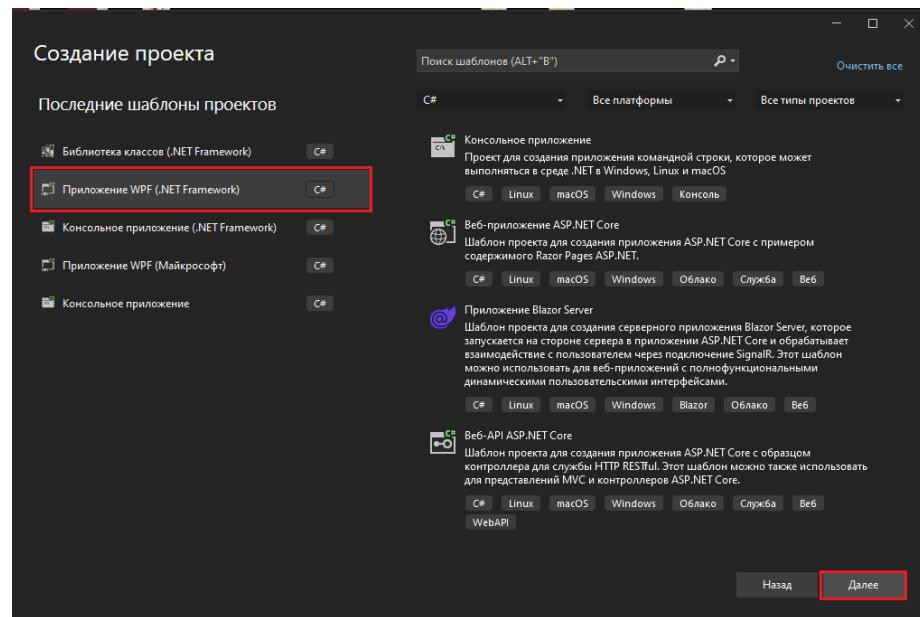


Рисунок 1.1 – Создание проекта

В окне «Настроить новый проект» заполняем поля:

- Имя проекта;
- Расположение;
- Выбираем платформу.

После заполнения всех полей нажимаем на кнопку «Создать»

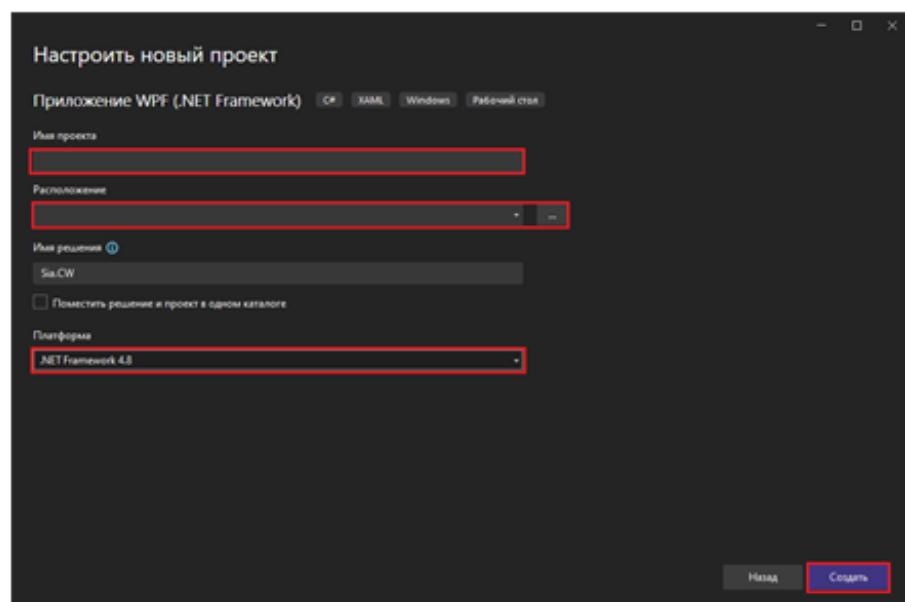


Рисунок 1.2 – Настройка нового проекта

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».

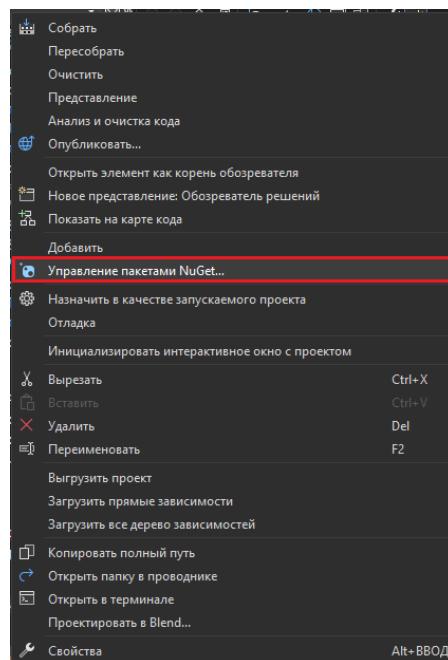


Рисунок 1.3 – Контекстное меню «Управление пакетами NuGet...»

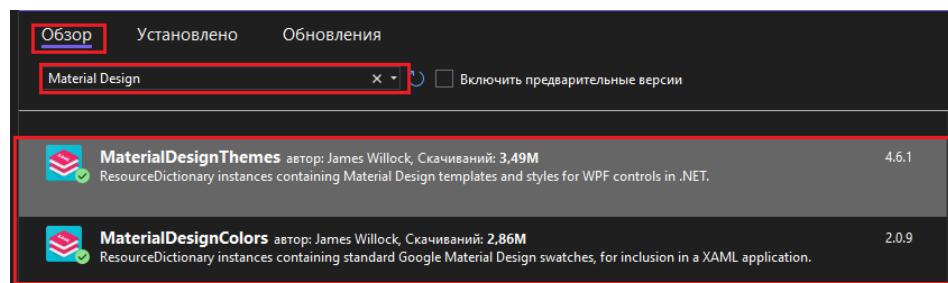


Рисунок 1.4 – Установка «MaterialDesignThemes» и «MaterialDesignColors».

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».

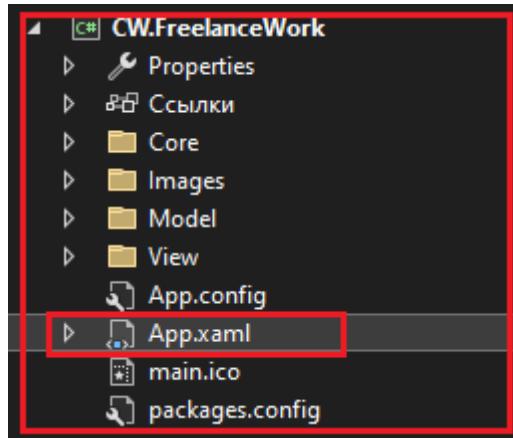


Рисунок 1.5 – Выбор файла «App.xaml»

```

<Application x:Class="CW.FreelanceWork.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    xmlns:local="clr-namespace:CW.FreelanceWork"
    StartupUri="View/MainWindow.xaml">
    <Application.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <materialDesign:BundledTheme BaseTheme="Light" PrimaryColor="Brown" SecondaryColor="Lime" />
                <ResourceDictionary Source="pack://application:///MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Defaults.xaml" />
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Application.Resources>
</Application>

```

Рисунок 1.6 – Редактирование файла «App.xaml»

Далее переходим в директорию «Model». В данную директорию импортируем, используя «EntityFramework» объектно-ориентированную модель базы данных (более детально данная тема разбирается в «Практической работе №1 – Работа с EntityFramework»). После импорта переходим к разметке «MainWindow»

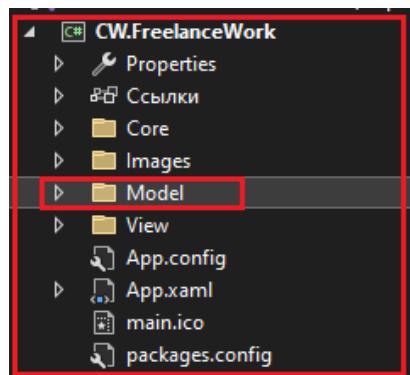


Рисунок 1.7 – Выбор директории файла «Model»

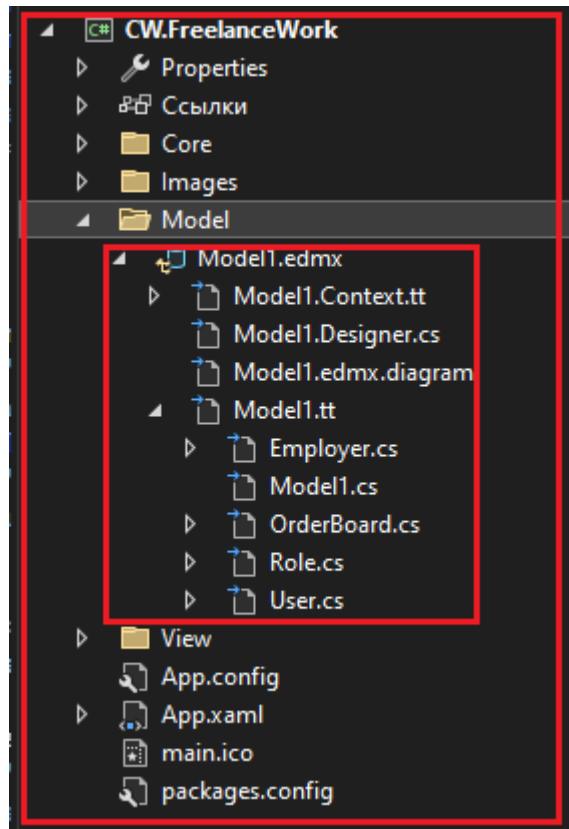


Рисунок 1.8 – Импорт данных

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» представлена на рисунках 1.10 – 1.).

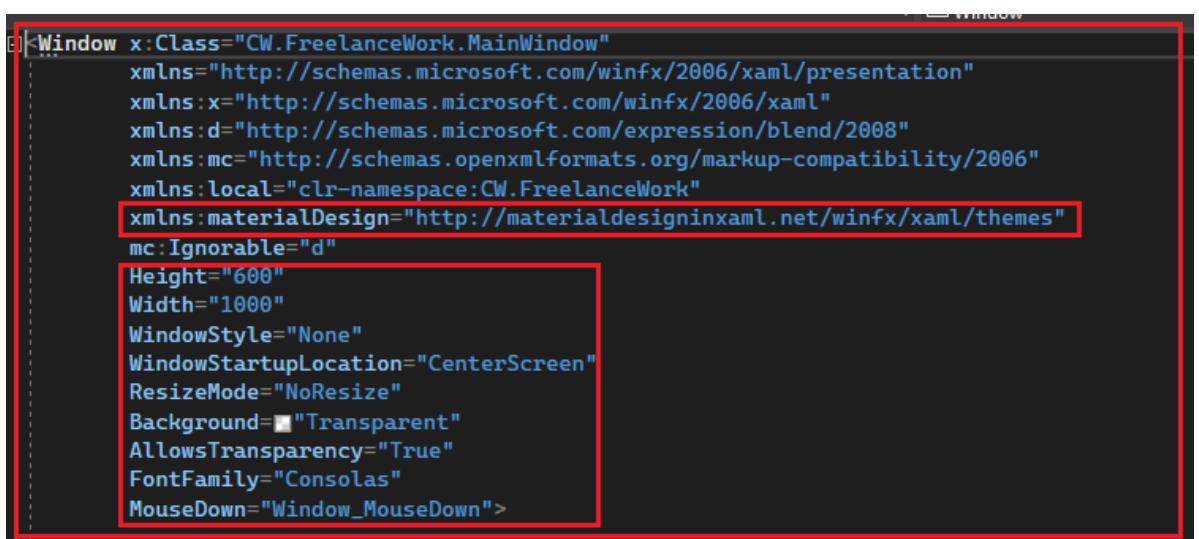


Рисунок 1.9 – Импорт данных

```

<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="50"/>
        <RowDefinition/>
    </Grid.RowDefinitions>
    <Image Grid.RowSpan="2"
        Source="/Images/main_bg.jpg"
        Stretch="UniformToFill"/>
    <StackPanel Background="#4D3825">
        <TextBlock HorizontalAlignment="Left"
            VerticalAlignment="Center"
            Margin="10 0 0 0"
            FontSize="20"
            FontWeight="SemiBold">
            <Run Foreground="White">Курсовая работа | Фриланс Биржа |</Run>
            <Run Foreground="#D4A84F"
                TextDecorations="Underline"
                Cursor="Hand"
                ToolTip="Курсовая работа по ТРПО">Работа Всем</Run>
        </TextBlock>
        <StackPanel HorizontalAlignment="Right"
            VerticalAlignment="Center"
            Margin="0 0 10 0">
            <Button Style="{StaticResource MaterialDesignIconForegroundButton}"
                ToolTip="Закрыть приложение"
                x:Name="BtnClose"
                Click="BtnClose_Click">
                <materialDesign:PackIcon Kind="Power"
                    Foreground="#D4A84F" />
            </Button>
        </StackPanel>
    </StackPanel>

```

Рисунок 1.10 – Разметка «MainWindow»

```

<Frame Grid.Row="1"
    x:Name="MainWindowFrame"
    NavigationUIVisibility="Hidden" />
</Grid>
</Window>

```

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.

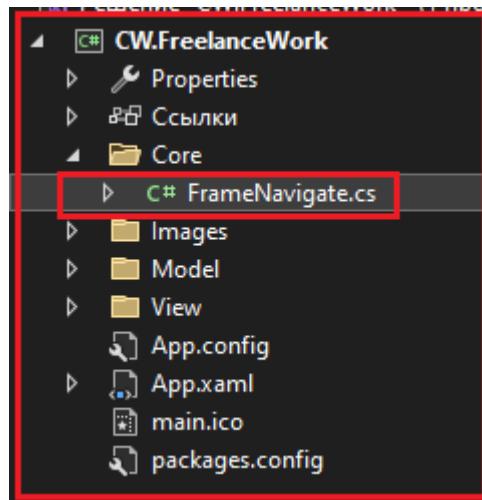


Рисунок 1.13 – Директория «Core»

```
Ссылок: 36
public static class FrameNavigate
{
    Ссылок: 13
    public static Frame FrameObject { get; set; }
    Ссылок: 23
    public static FreelanceWorkDBEntities DB { get; set; }
}
```

Рисунок 1.14 – Реализация класса «FrameNavigate»

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

- UserPage – будет содержать страницы для роли пользователя;
- LoginPage – будет содержать страницы для входа и регистрации;
- EmployerPage – будет содержать страницы для роли работодателя;
- AdministratorPage – будет содержать страницы для роли администратора.

В директории «LoginPage» создаем две страницы «MainWindowLoginPage» и «MainWindowRegistrationPage». 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.

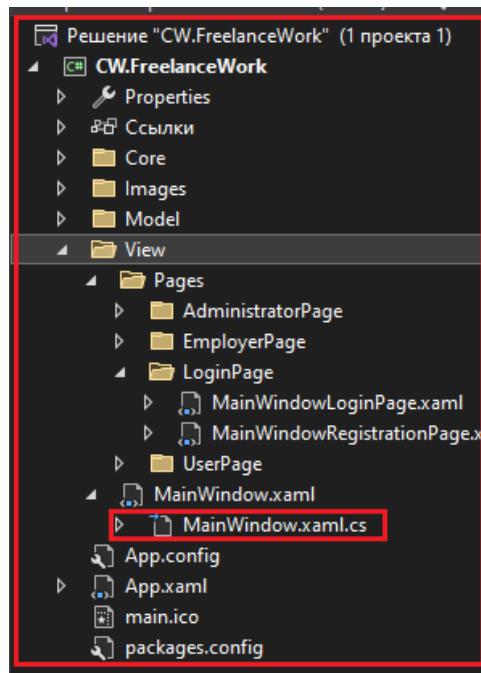


Рисунок 1.15 – Файл «MainWindow.xaml.cs»

```
Ссылок: 0
public MainWindow()
{
    InitializeComponent();

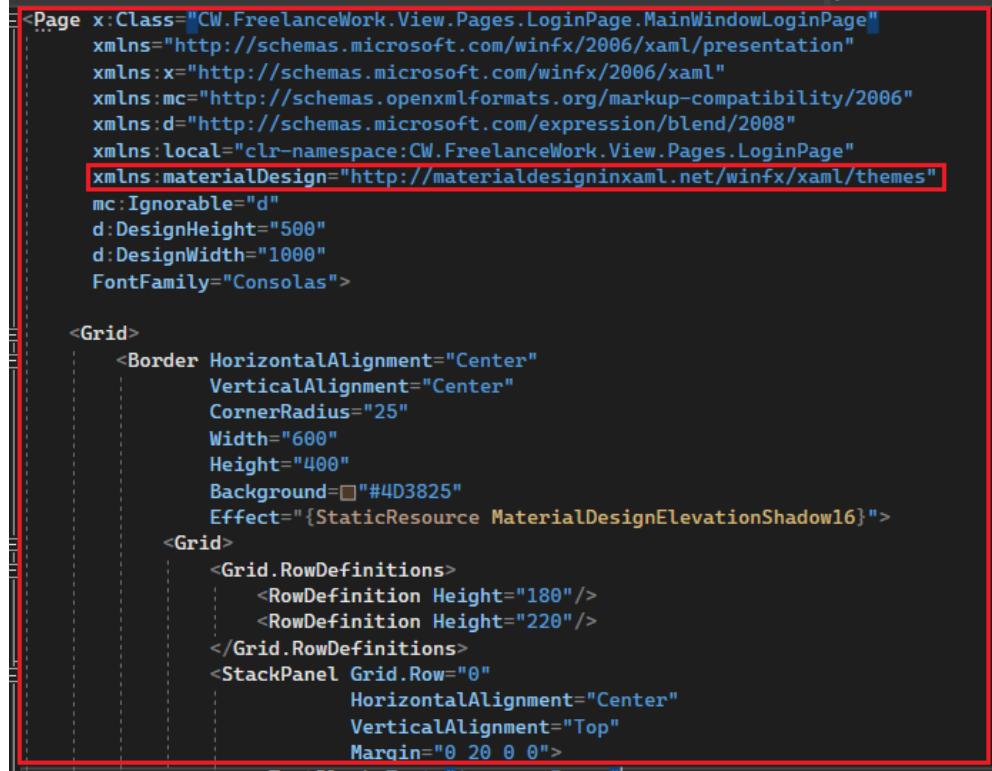
    FrameNavigate.FrameObject = MainWindowFrame;
    FrameNavigate.DB = new FreelanceWorkDBEntities();
    MainWindowFrame.Navigate(new MainWindowLoginPage());
}

Ссылок: 1
private void Window_MouseDown(object sender, MouseButtonEventArgs e)
{
    DragMove();
}

Ссылок: 1
private void BtnClose_Click(object sender, RoutedEventArgs e)
{
    Application.Current.Shutdown();
}
```

Рисунок 1.16 – Реализация логики в файле «MainWindow.xaml.cs»

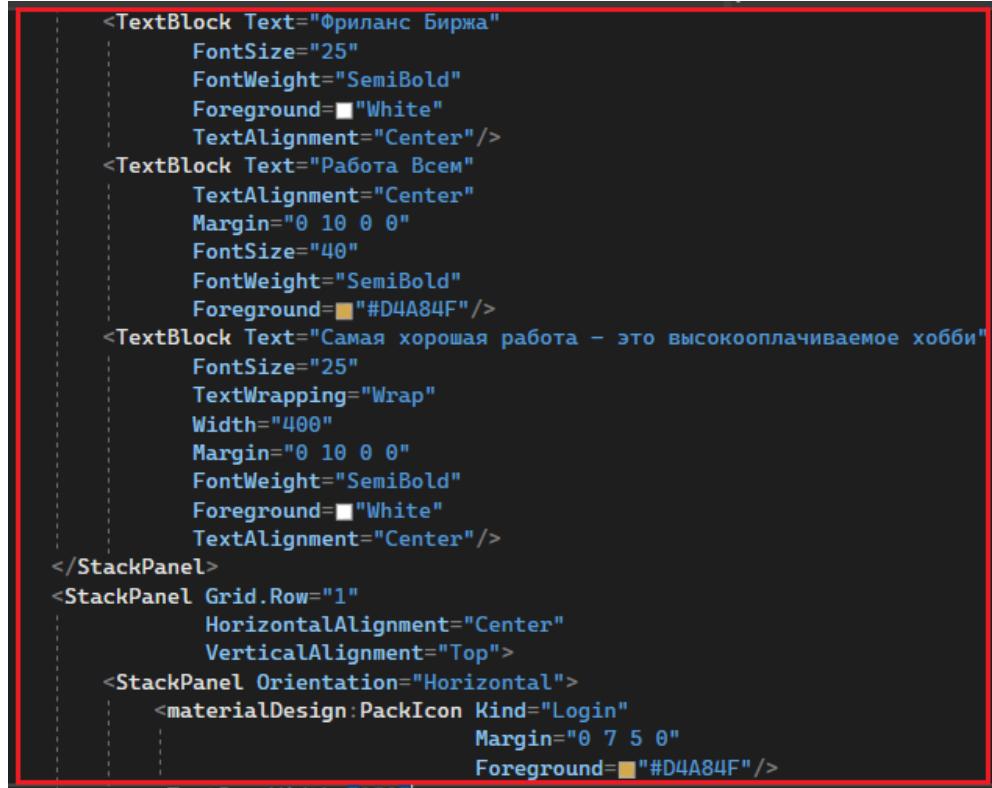
Переходим на страницу «MainWindowLoginPage». Импортируем пространство имен «materialDesign» и переходим к разметке (Разметка «MainWindowLoginPage» приведена на рисунках 1.17 – 1.22)



```
<Page x:Class="CW.FreelanceWork.View.Pages.LoginPage.MainWindowLoginPage"
    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/expressionblend/2008"
    xmlns:local="clr-namespace:CW.FreelanceWork.View.Pages.LoginPage"
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    mc:Ignorable="d"
    d:DesignHeight="500"
    d:DesignWidth="1000"
    FontFamily="Consolas">

    <Grid>
        <Border HorizontalAlignment="Center"
            VerticalAlignment="Center"
            CornerRadius="25"
            Width="600"
            Height="400"
            Background="#4D3825"
            Effect="{StaticResource MaterialDesignElevationShadow16}">
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="180"/>
                    <RowDefinition Height="220"/>
                </Grid.RowDefinitions>
                <StackPanel Grid.Row="0"
                    HorizontalAlignment="Center"
                    VerticalAlignment="Top"
                    Margin="0 20 0 0">
```

Рисунок 1.17 – Разметка «MainWindowLoginPage»



```
<TextBlock Text="Фриланс Биржа"
    FontSize="25"
    FontWeight="SemiBold"
    Foreground="White"
    TextAlignment="Center"/>
<TextBlock Text="Работа Всем"
    TextAlignment="Center"
    Margin="0 10 0 0"
    FontSize="40"
    FontWeight="SemiBold"
    Foreground="#D4A84F"/>
<TextBlock Text="Самая хорошая работа – это высокооплачиваемое хобби"
    FontSize="25"
    TextWrapping="Wrap"
    Width="400"
    Margin="0 10 0 0"
    FontWeight="SemiBold"
    Foreground="White"
    TextAlignment="Center"/>
</StackPanel>
<StackPanel Grid.Row="1"
    HorizontalAlignment="Center"
    VerticalAlignment="Top">
    <StackPanel Orientation="Horizontal">
        <materialDesign:PackIcon Kind="Login"
            Margin="0 7 5 0"
            Foreground="#D4A84F"/>
```

Рисунок 1.18 – Разметка «MainWindowLoginPage»

```

<TextBox Width="150"
        x:Name="TbLogin"
        FontSize="15"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Введите логин"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>

<StackPanel Orientation="Horizontal"
            Margin="0 10 0 0">
    <materialDesign:PackIcon Kind="Key"
        Margin="0 5 5 0"
        Foreground="#D4A84F"/>
    <PasswordBox Width="150"
        x:Name="PsbPassword"
        FontSize="15"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Введите пароль"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>

<StackPanel Margin="0 10 0 0">
    <Button Background="#D4A84F"
            Foreground="White"
            x:Name="BtnLogin"
            Click="BtnLogin_Click"
            BorderThickness="0"
            Style="{StaticResource MaterialDesignRaisedButton}"
            materialDesign:ButtonProgressAssist.Value="-1"
            materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
            materialDesign:ButtonProgressAssist.IsIndeterminate="True"
            IsEnabled="{Binding DataContext.ControlsEnabled, RelativeSource={RelativeSource FindAncestor, AncestorType=Window}}">
        <TextBlock Text="Вход в систему"
            Margin="0 0 0 10"/>
    </Button>
</StackPanel>

```

Рисунок 1.19 – Разметка «MainWindowLoginPage»

```

<StackPanel Orientation="Horizontal">
    <materialDesign:PackIcon Kind="User"
        Margin="0 0 5 0"/>
    <TextBlock Text="Выйти на работу"
        Margin="0 0 5 0"
        FontWeight="SemiBold"/>
</StackPanel>
</Button>

<Button Background="#D4A84F"
        Foreground="White"
        x:Name="BtnWork"
        Click="BtnWork_Click"
        BorderThickness="0"
        Margin="0 10 0 0"
        Style="{StaticResource MaterialDesignRaisedButton}"
        materialDesign:ButtonProgressAssist.Value="-1"
        materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
        materialDesign:ButtonProgressAssist.IsIndeterminate="True"
        IsEnabled="{Binding DataContext.ControlsEnabled, RelativeSource={RelativeSource FindAncestor, AncestorType=Window}}">
    <StackPanel Orientation="Horizontal">
        <materialDesign:PackIcon Kind="UserAlert"
            Margin="0 0 5 0"/>
        <TextBlock Text="Предложить работу"
            Margin="0 0 5 0"
            FontWeight="SemiBold"/>
    </StackPanel>
</Button>

<Button Background="#D4A84F"
        Foreground="White"
        BorderThickness="0"
        x:Name="BtnRegistration"
        Click="BtnRegistration_Click"
        Style="{StaticResource MaterialDesignFlatButton}">
    <TextBlock Text="Зарегистрироваться"
        Margin="0 0 0 10"/>
</Button>

```

Рисунок 1.20 – Разметка «MainWindowLoginPage»

```

<Button Background="#D4A84F"
        Foreground="White"
        BorderThickness="0"
        x:Name="BtnRegistration"
        Click="BtnRegistration_Click"
        Style="{StaticResource MaterialDesignFlatButton}"
        materialDesign:ButtonProgressAssist.Value="-1"
        materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
        materialDesign:ButtonProgressAssist.IsIndeterminate="True"
        IsEnabled="{Binding DataContext.ControlsEnabled, RelativeSource={RelativeSource FindAncestor, AncestorType=Window}}"
        Margin="0 10 0 0">
    <StackPanel Orientation="Horizontal">
        <materialDesign:PackIcon Kind="Register"
            Margin="0 0 5 0"/>
        <TextBlock Text="Начать работать"
            Margin="0 0 5 0"
            FontWeight="SemiBold"/>
    </StackPanel>
</Button>
</StackPanel>
</Grid>
</Border>
</Grid>
</Page>

```

Рисунок 1.21 – Разметка «MainWindowLoginPage»

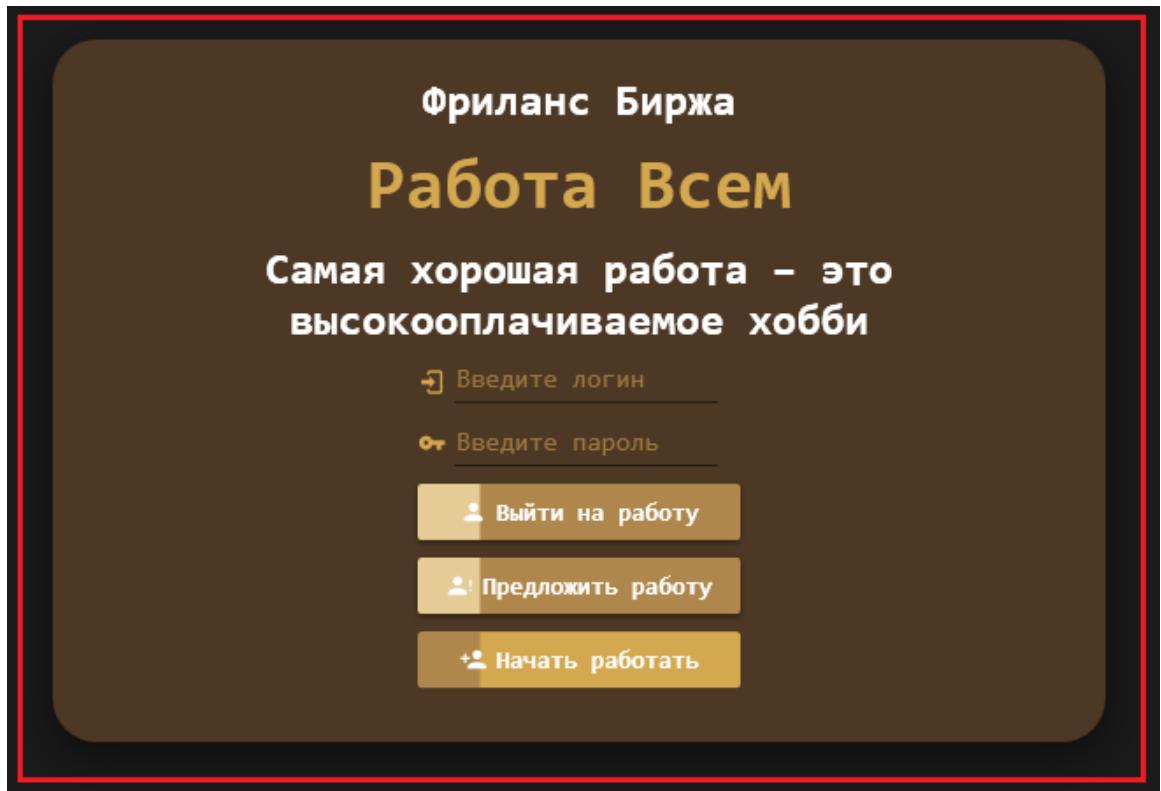


Рисунок 1.22 – Результат разметки «MainWindowLoginPage»

Далее переходим в файле «MainWindowLoginPage.xaml.cs». Этот файл будет содержать логику авторизации при клике на кнопку «Выйти на работу» (Реализация логики приведена на рисунках 1.23 – 1.24). Также в директории «EmployerPage» создадим страницу «MainEmployerPage» на которую мы будем переходить при клике на кнопку «Предложить работу», а по клику на кнопку «Начать работать» мы будем переходить на «MainWindowRegistrationPage».

```
Ссылка: 6
public MainWindowLoginPage()
{
    InitializeComponent();
}

Ссылка: 1
private void BtnLogin_Click(object sender, RoutedEventArgs e)
{
    try
    {
        User userModel = FrameNavigate.DB.Users.FirstOrDefault(u =>
            u.UserMail == TbLogin.Text && u.UserPhone == PsbPassword.Password);

        if (userModel == null)
        {
            MessageBox.Show("Ошибка данных",
                "Системное сообщение",
                MessageBoxButton.OK,
                MessageBoxImage.Error);
        }
    }
}
```

Рисунок 1.22 – Реализация логики файла «MainWindowLoginPage.xaml.cs»

```
else
{
    switch (userModel.RoleID)
    {
        case 1:
            FrameNavigate.FrameObject.Navigate(new MainAdministratorPage());
            break;
        case 2:
            FrameNavigate.FrameObject.Navigate(new MainUserPage());
            break;
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message.ToString(),
                    "Системная ошибка",
                    MessageBoxButton.OK,
                    MessageBoxImage.Error);
}
```

Рисунок 1.23 – Реализация логики файла «MainWindowLoginPage.xaml.cs»

```
Ссылка: 1
private void BtnWork_Click(object sender, RoutedEventArgs e)
{
    FrameNavigate.FrameObject.Navigate(new MainEmployerPage());
}

Ссылка: 1
private void BtnRegistration_Click(object sender, RoutedEventArgs e)
{
    FrameNavigate.FrameObject.Navigate(new MainWindowRegistrationPage());
}
```

Рисунок 1.24 – Реализация логики файла «MainWindowLoginPage.xaml.cs»

Теперь запустим и проверим работоспособность приложения.

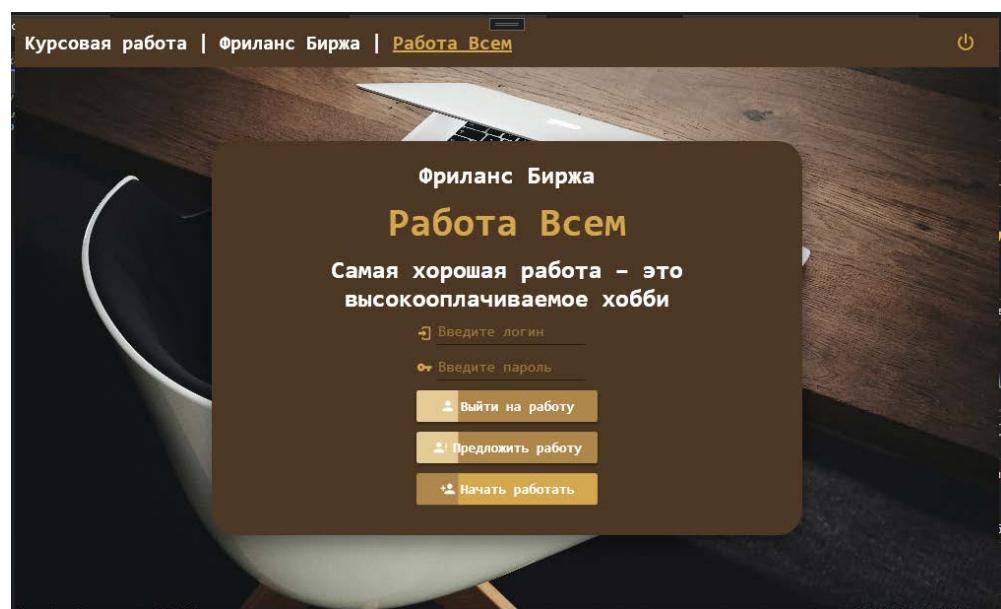


Рисунок 1.25 – Результат работы приложения

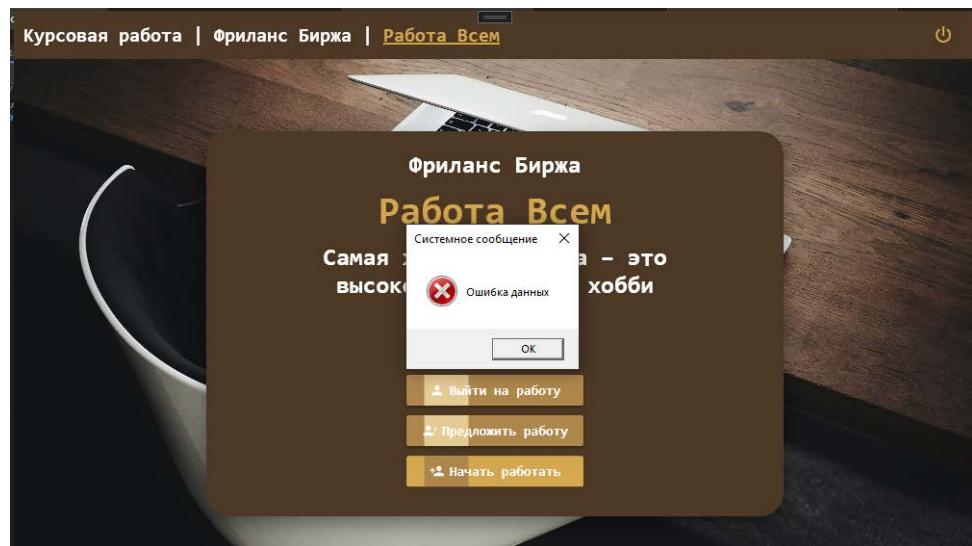


Рисунок 1.26 – Введены некорректные данные

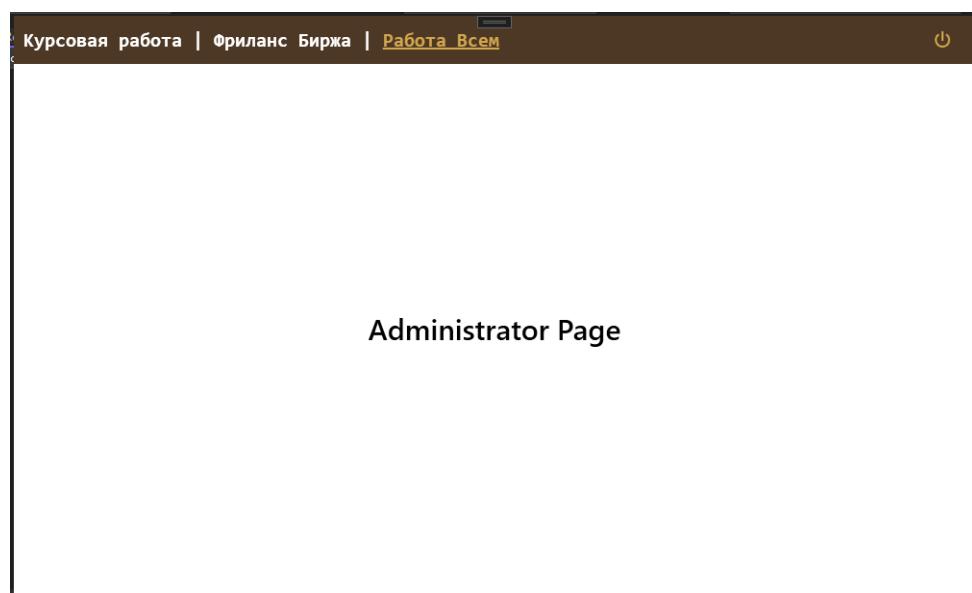


Рисунок 1.27 – Введены данные учетной записи администратора

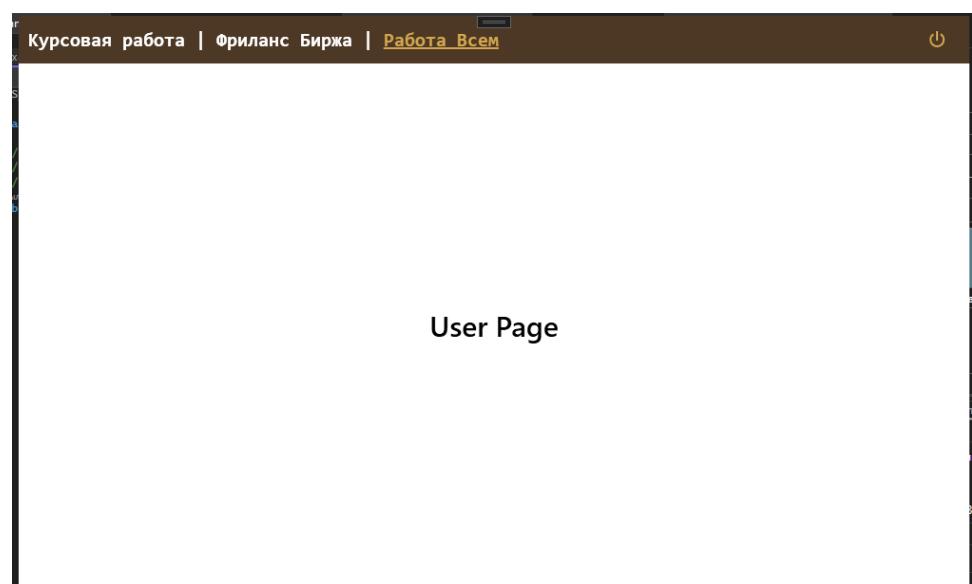


Рисунок 1.28 – Введены данные учетной записи пользователя

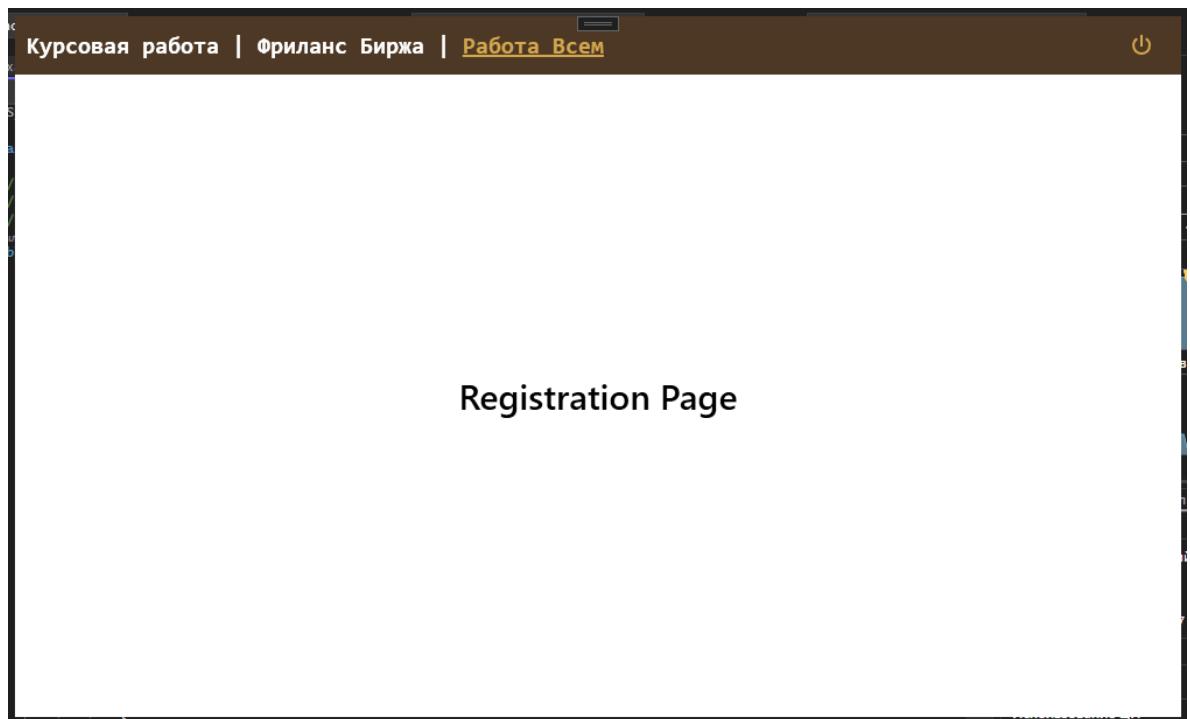


Рисунок 1.29 – Переход на страницу регистрации

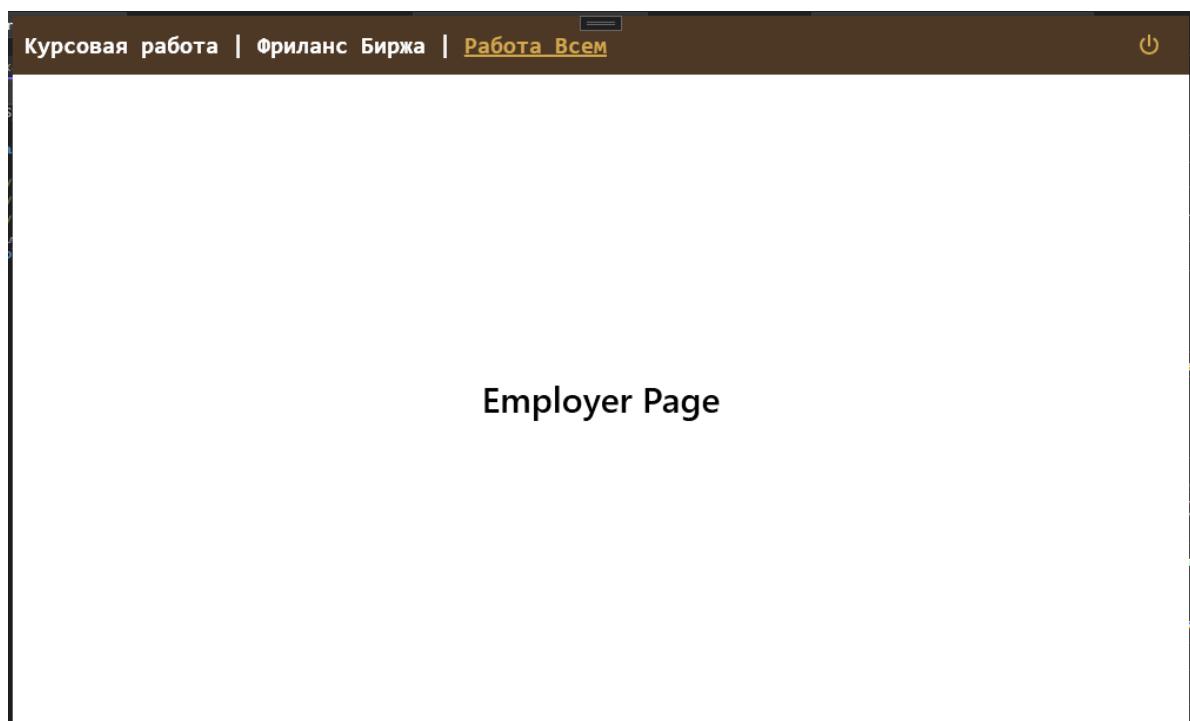


Рисунок 1.30 – Переход на страницу работодателя

2 Реализация MainWindowRegistrationPage

Перейдем к разметке «MainWindowRegistrationPage» (разметка представлена на рисунках 2.1 – 2.5).

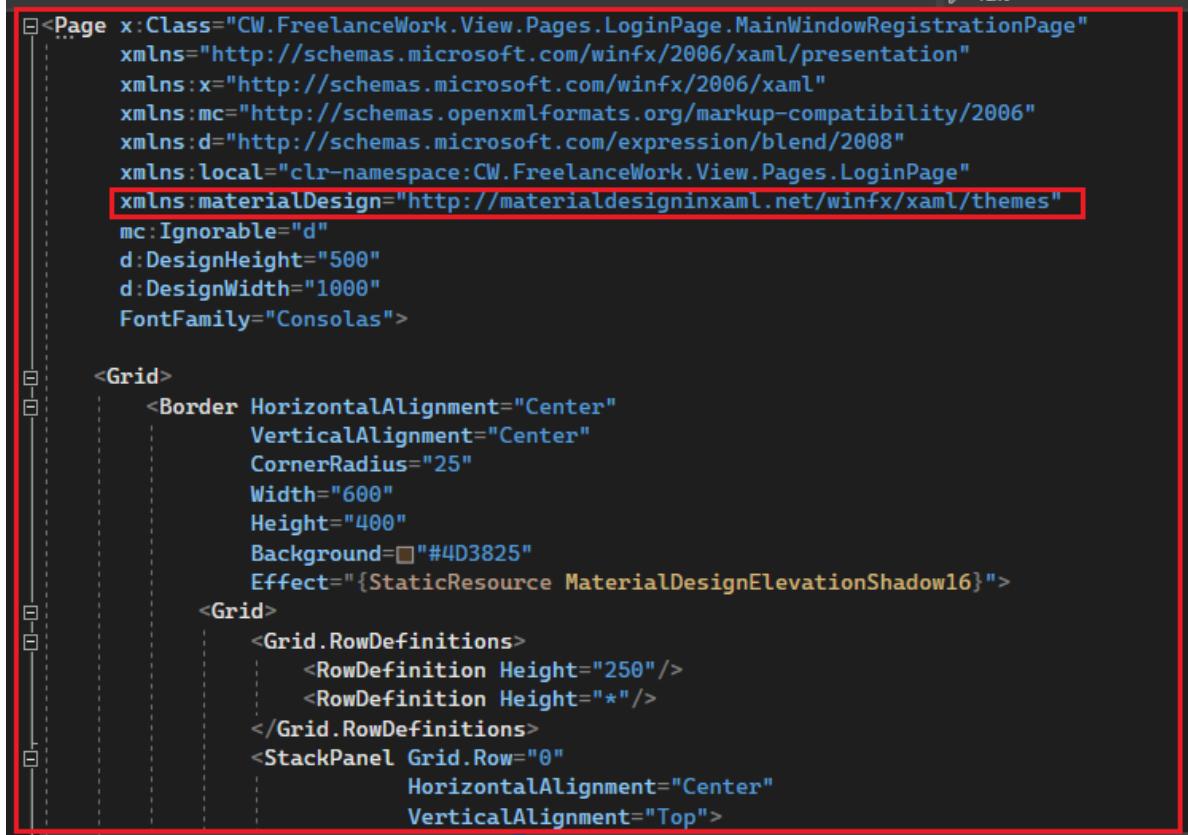


Рисунок 2.1 – Разметка «MainWindowRegistrationPage»

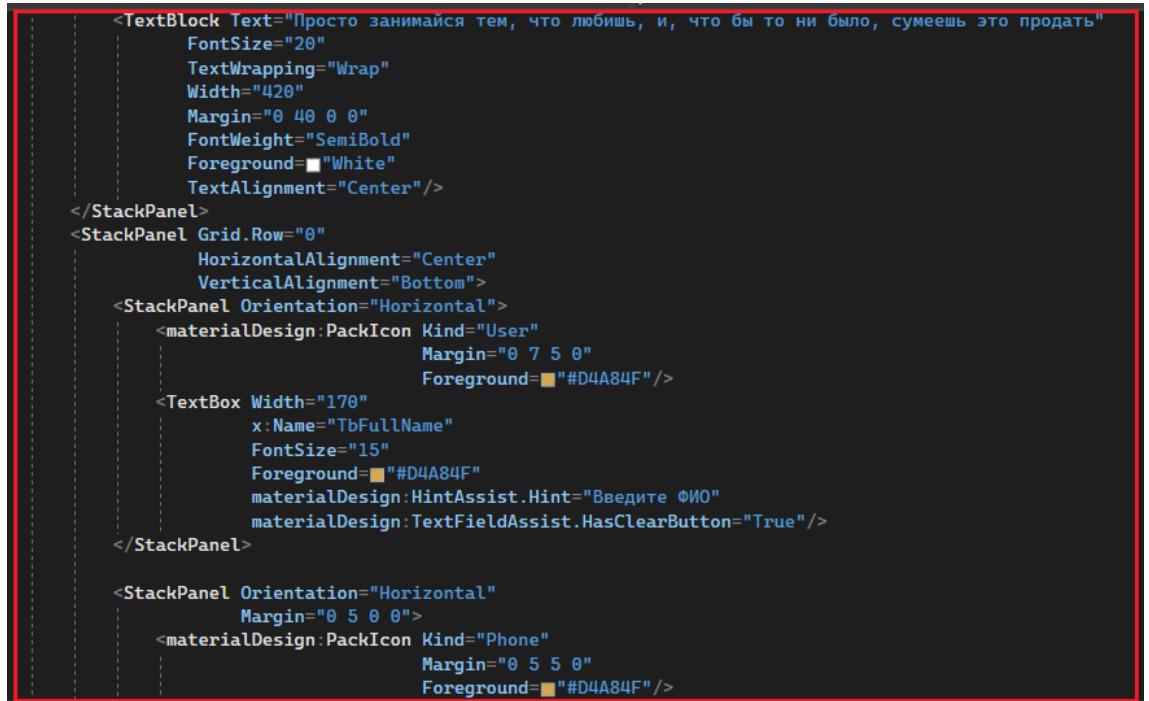


Рисунок 2.2 – Разметка «MainWindowRegistrationPage»

```

<StackPanel>
    <TextBox Width="170"
        x:Name="TbPhone"
        FontSize="15"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Введите моб. номер"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>

<StackPanel Orientation="Horizontal"
    Margin="0 5 0 0">
    <materialDesign:PackIcon Kind="Email"
        Margin="0 5 5 0"
        Foreground="#D4A84F"/>
    <TextBox Width="170"
        x:Name="TbEmail"
        FontSize="15"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Введите email"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>

<StackPanel Orientation="Horizontal"
    Margin="0 5 0 0">
    <materialDesign:PackIcon Kind="VennDiagram"
        Margin="0 5 5 0"
        Foreground="#D4A84F"/>
    <TextBox Width="170"
        x:Name="TbSkills"
        FontSize="15"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Мои навыки"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>

```

Рисунок 2.3 – Разметка «MainWindowRegistrationPage»

```

</StackPanel>

<StackPanel Orientation="Horizontal"
    Margin="0 15 0 0"
    Grid.Row="1"
    HorizontalAlignment="Center"
    VerticalAlignment="Top">
    <Button Background="#D4A84F"
        Foreground="White"
        x:Name="BtnRegistration"
        Click="BtnRegistration_Click"
        BorderThickness="0"
        Style="{StaticResource MaterialDesignRaisedButton}"
        materialDesign:ButtonProgressAssist.Value="-1"
        materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
        materialDesign:ButtonProgressAssist.IsIndeterminate="True">
        <StackPanel Orientation="Horizontal"
            HorizontalAlignment="Center"
            VerticalAlignment="Center">
            <materialDesign:PackIcon Kind="Bitcoin"
                Margin="0 2 5 0"/>
            <TextBlock Text="Начать работать!"
                FontWeight="SemiBold"/>
        </StackPanel>
    </Button>

```

Рисунок 2.4 – Разметка «MainWindowRegistrationPage»

```

<Button Background="#D4A84F"
    Foreground="White"
    x:Name="BtnBack"
    Click="BtnBack_Click"
    BorderThickness="0"
    Margin="10 0 0 0"
    Style="{StaticResource MaterialDesignRaisedButton}"
    materialDesign:ButtonProgressAssist.Value="-1"
    materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
    materialDesign:ButtonProgressAssist.IsIndeterminate="True"
    IsEnabled="(Binding DataContext.ControlsEnabled, RelativeSource={RelativeSource FindAncestor, AncestorType=Window})">
    <StackPanel Orientation="Horizontal"
        HorizontalAlignment="Center"
        VerticalAlignment="Center">
        <materialDesign:PackIcon Kind="Logout"
            Margin="0 2 5 0"/>
        <TextBlock Text="Я передумал"
            FontWeight="SemiBold"/>
    </StackPanel>
</Button>
</Grid>
</Border>
</Grid>
</Page>

```

Рисунок 2.5 – Разметка «MainWindowRegistrationPage»

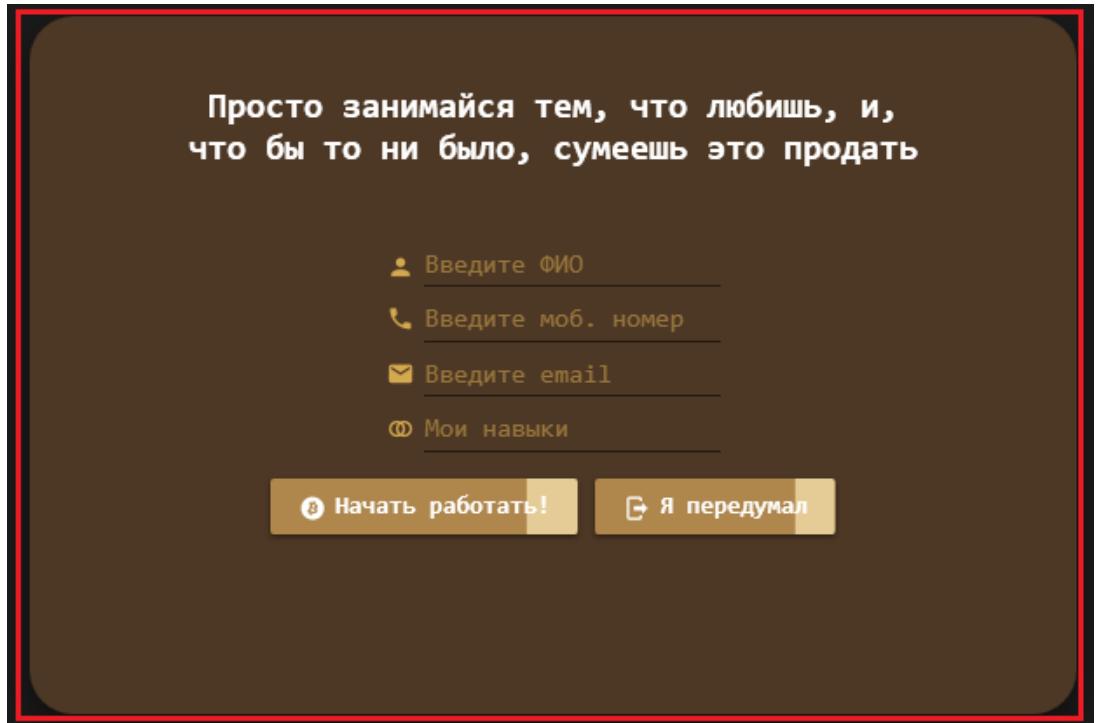


Рисунок 2.6 – Результат разметки «MainWindowRegistrationPage»

Теперь перейдем в файл «MainWindowRegistrationPage.xaml.cs» для реализации логики регистрации. Сделаем проверку заполнения полей, проверку на дубликат ФИО, проверку на ошибку соединения с базой данных, также при успешной регистрации будет выводить диалоговое окно и перенаправлять на страницу «MainWindowLoginPage» (реализация логики представлена на рисунках 2.7 – 2.8).

```

Ссылка: 1
private async void BtnRegistration_Click(object sender, RoutedEventArgs e)
{
    if (string.IsNullOrEmpty(TbFullName.Text) ||
        string.IsNullOrEmpty(TbPhone.Text) ||
        string.IsNullOrEmpty(TbEmail.Text) ||
        string.IsNullOrEmpty(TbSkills.Text))
    {
        MessageBox.Show("Все поля должны быть заполнены!",
                        "Системное сообщение",
                        MessageBoxButton.OK,
                        MessageBoxIcon.Error);
    }
    else
    {
        if (FrameNavigate.DB.Users.Count(u => u.FIO == TbFullName.Text) > 0)
        {
            MessageBox.Show("Пользователь с такими инициалами уже зарегистрирован!",
                            "Системное сообщение",
                            MessageBoxButton.OK,
                            MessageBoxIcon.Error);
        }
    }
}

```

Рисунок 2.7 – Реализация файла «MainWindowRegistrationPage.xaml.cs»

```

FrameNavigate.DB.Users.Add(new User
{
    FIO = TbFullName.Text,
    UserPhone = TbPhone.Text,
    UserMail = TbEmail.Text,
    UserSkills = TbSkills.Text,
    RoleID = 2
});

await FrameNavigate.DB.SaveChangesAsync();
MessageBox.Show("Учетная запись создана!",
                "Системное сообщение",
                MessageBoxButton.OK,
                MessageBoxIcon.Information);
FrameNavigate.FrameObject.Navigate(new MainWindowLoginPage());
}

}

Ссылка: 1
private void BtnBack_Click(object sender, RoutedEventArgs e)
{
    FrameNavigate.FrameObject.Navigate(new MainWindowLoginPage());
}

```

Рисунок 2.8 – Реализация файла «MainWindowLoginPage.xaml.cs»

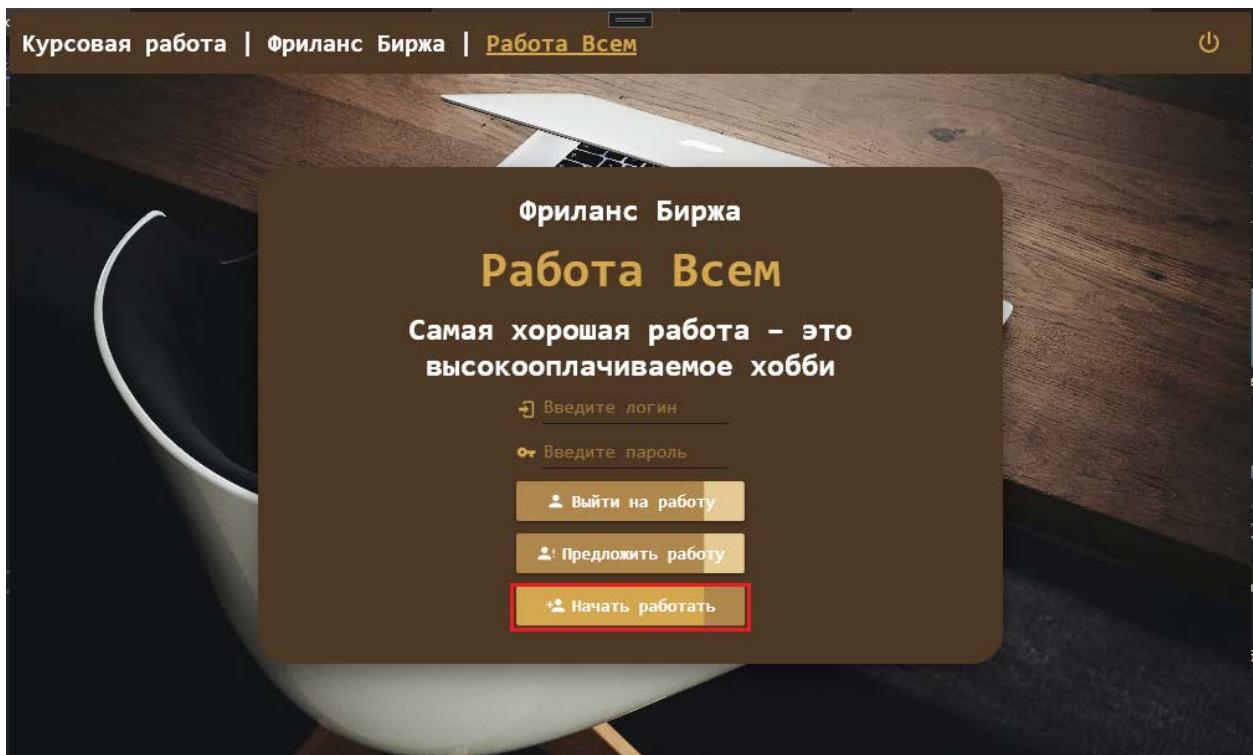


Рисунок 2.9 – Переход на «MainWindowRegistrationPage»

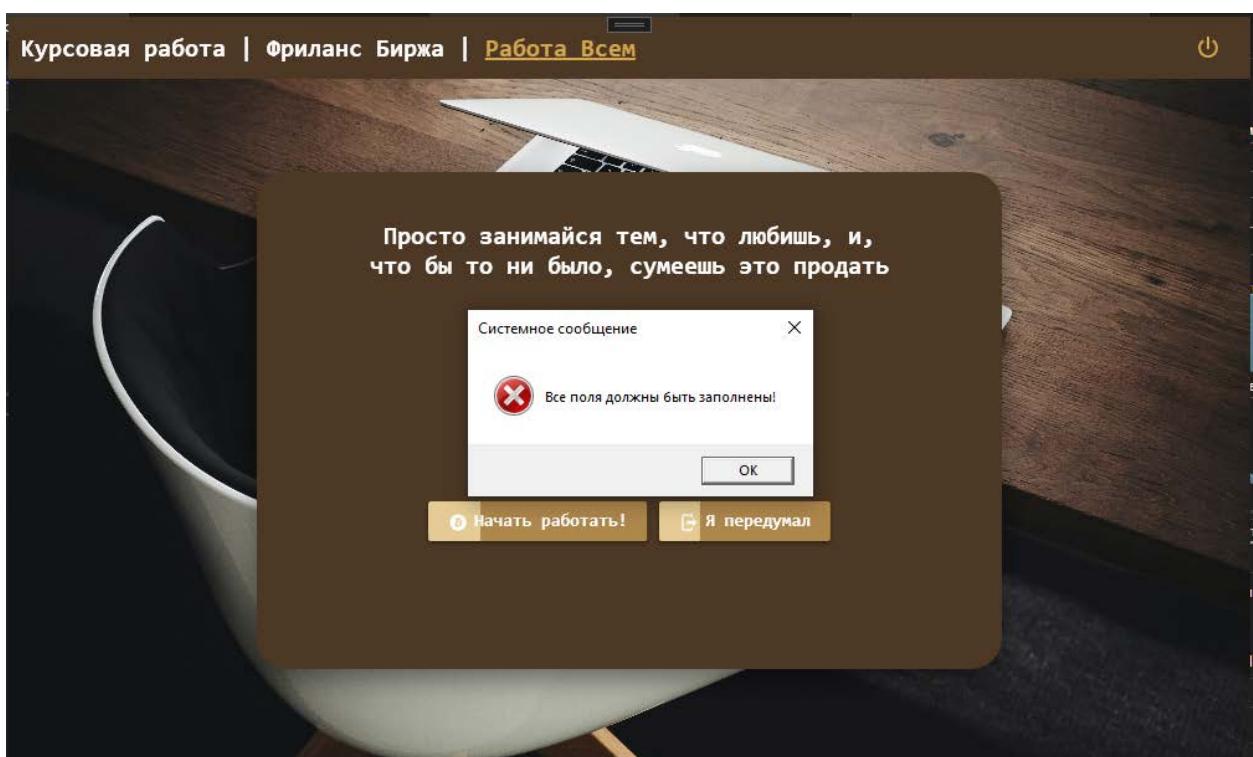


Рисунок 2.10 – Ошибка заполнения полей

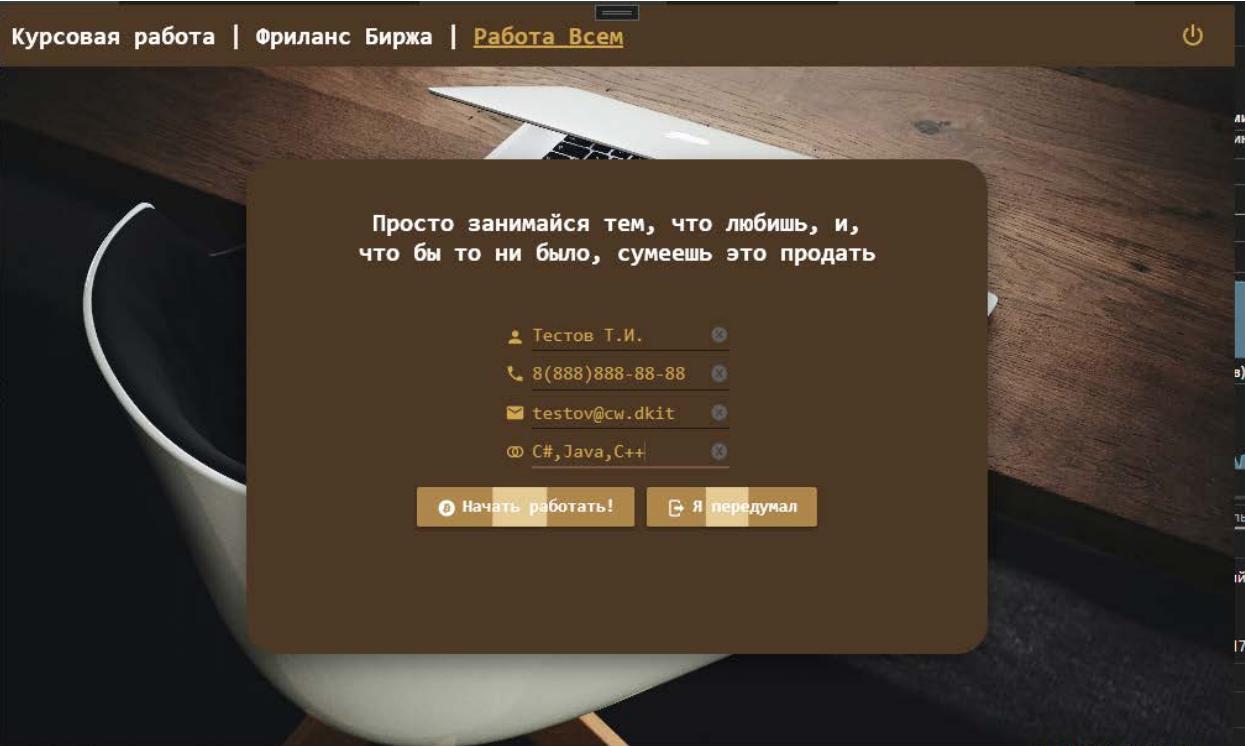


Рисунок 2.11 – Все поля заполнены

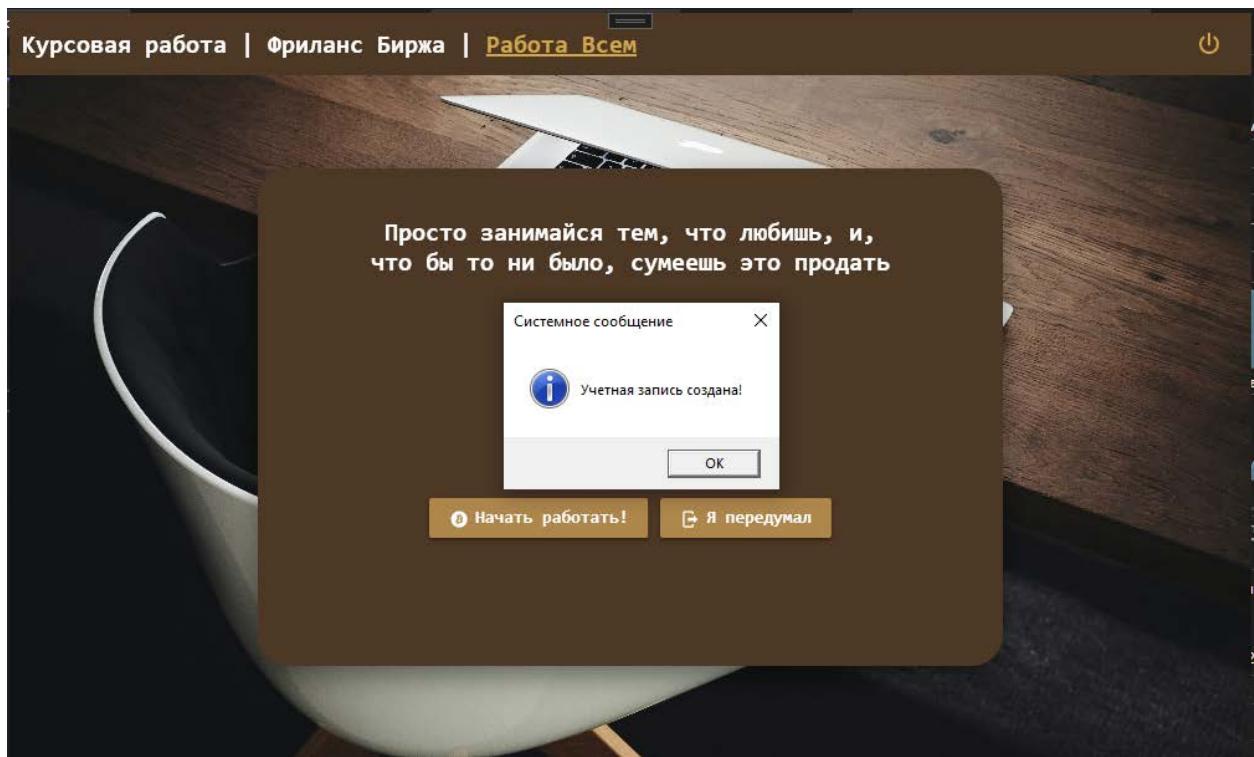


Рисунок 2.12 – Успешное создание учетной записи

	UserID	FIO	UserPhone	UserMail	UserSkills	RoleID
5	5	Мясников Г.М.	8(415)235-24-76	test4@cw.dkit	C#, WPF	2
6	6	Андреева Ю.А.	8(105)192-75-11	test5@cw.dkit	C#, WPF	2
7	7	Воронцова И.Ю.	8(279)887-24-55	test6@cw.dkit	C#, WPF	2
8	8	Лихачёва А.М.	8(037)894-33-63	test7@cw.dkit	C#, WPF	2
9	9	Паринова К.И.	8(497)778-57-33	test8@cw.dkit	C#, WPF	2
10	10	Тихонова М.К.	8(160)187-91-53	test9@cw.dkit	C#, WPF	2
11	11	Мартынова С.П.	8(495)003-84-45	test10@cw.dkit	C#, WPF	2
12	12	Сидорова Н.В.	8(495)674-00-74	test11@cw.dkit	C#, WPF	2
13	13	Волкова В.Д.	8(495)312-42-34	test12@cw.dkit	C#, WPF	2
14	14	Ершов К.М.	8(495)355-63-18	test13@cw.dkit	C#, WPF	2
15	15	Тестов Т.И.	8(888)888-88-88	testov@cw.dkit	C#,Java,C++	2

Рисунок 2.13 – Запись информации в базу данных

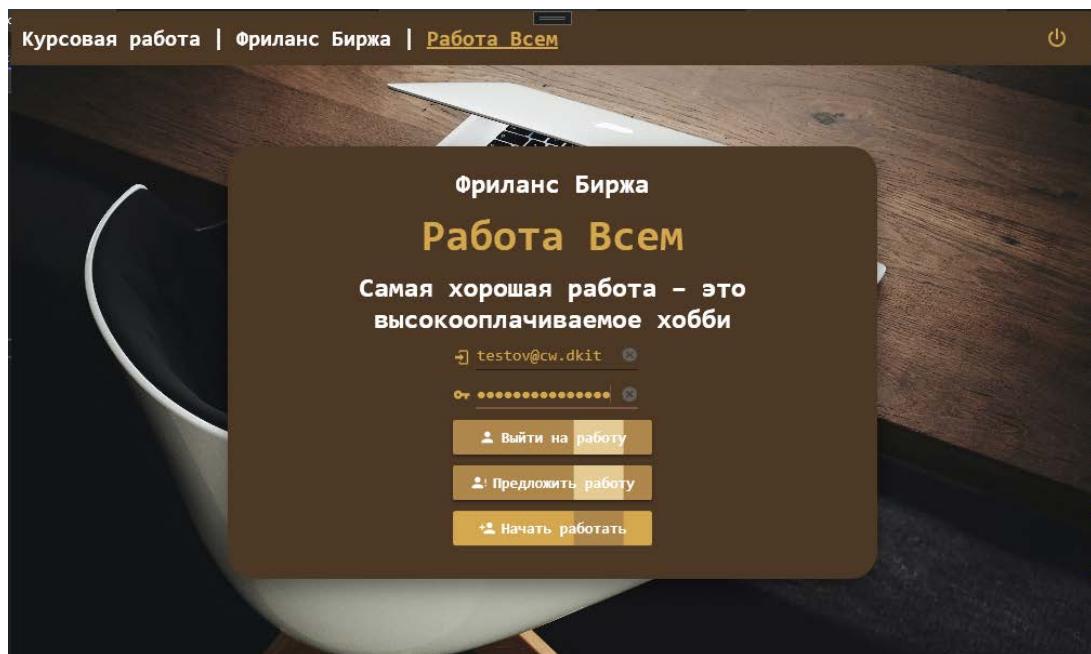


Рисунок 2.14 – Проверка созданного аккаунта пользователя



Рисунок 2.15 – Успешная авторизация нового пользователя

3 Реализация MainEmployerPage, LoginEmployerPage, DetailEmployerPage

Теперь перейдем к реализации страниц работодателя. Логика будет состоять из трех страниц «MainEmployerPage, LoginEmployerPage, DetailEmployerPage». «MainEmployerPage» будет производится проверка работодателя (по ИНН, названию компании и ФИО генерального директора.), если проверка прошла успешна становится доступна страница «LoginEmployerPage» где представитель компании может авторизироваться. Далее становится доступна «DetailEmployerPage» где работодатель может заполнить форму размещения заказа. Перейдем к более детальной реализации.

В директории «EmployerPage» создаём три страницы «MainEmployerPage, LoginEmployerPage, DetailEmployerPage».

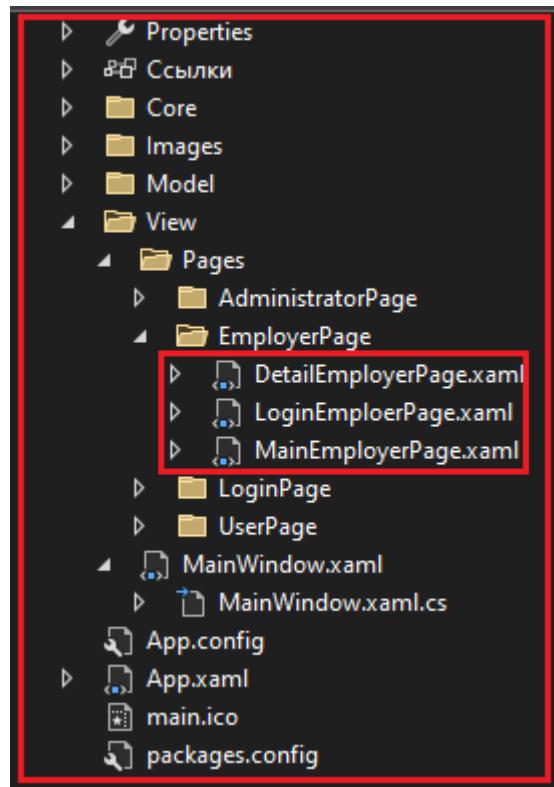


Рисунок 3.1 – Создание страниц

Выбираем «MainEmployerPage». Данная страница будет проверять подлинность работодателя (реализация разметки и логики представлена) на рисунках 3.2 – 3.14).

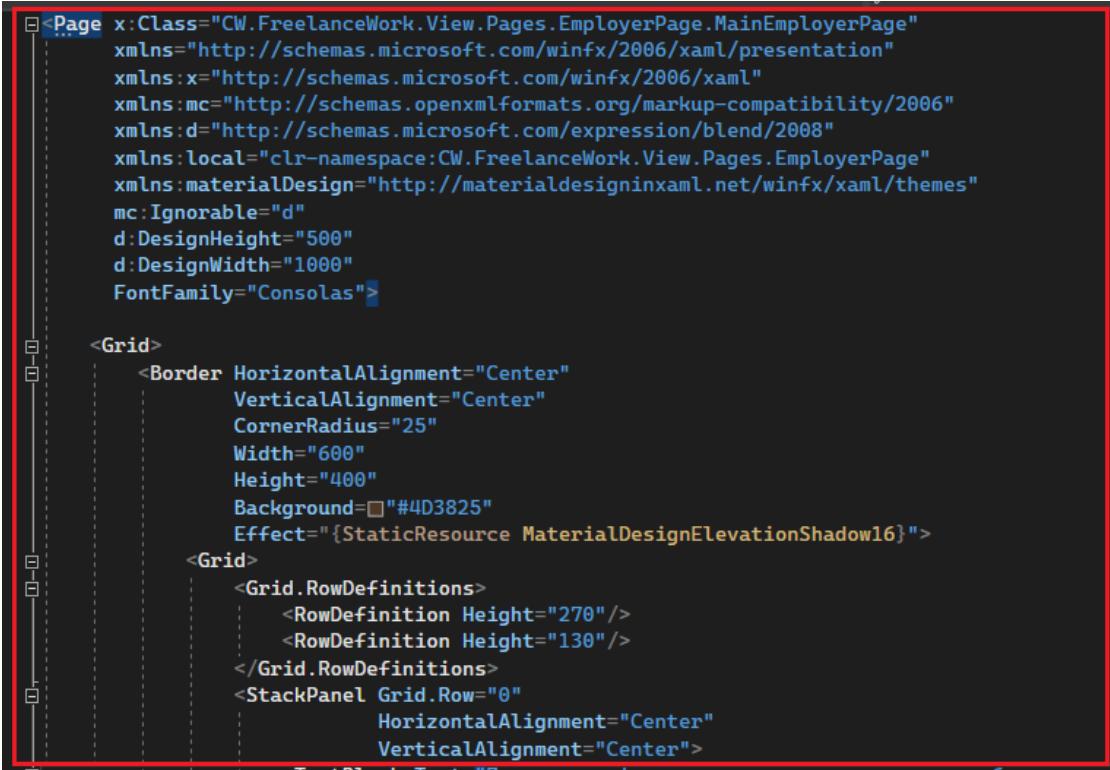


Рисунок 3.2 – Разметка «MainEmployerPage»

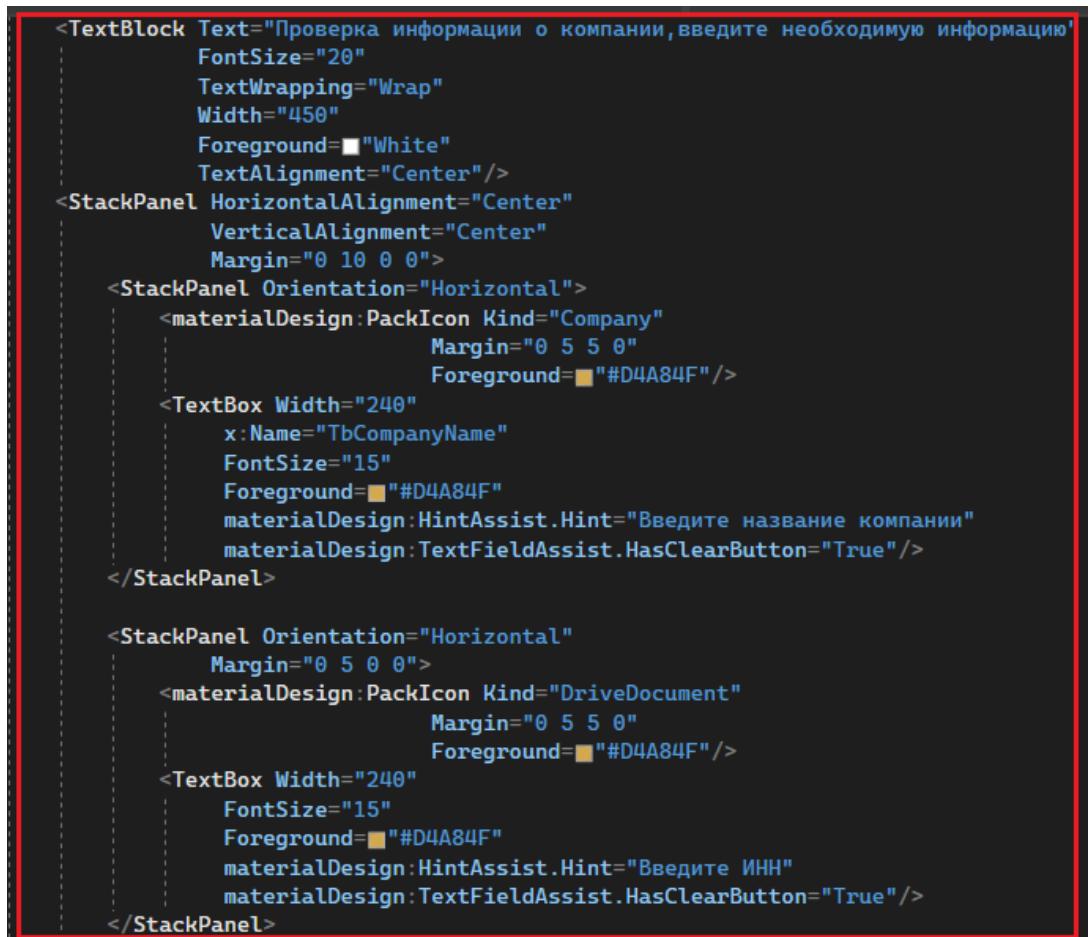


Рисунок 3.3 – Разметка «MainEmployerPage»

```

<StackPanel Orientation="Horizontal"
    Margin="0 5 0 0">
    <materialDesign:PackIcon Kind="Email"
        Margin="0 5 5 0"
        Foreground="#D4A84F"/>
    <TextBox Width="240"
        FontSize="15"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Введите ФИО ген. директора"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>

<StackPanel HorizontalAlignment="Center"
    VerticalAlignment="Center"
    Margin="0 20 0 0">
    <Button Background="#D4A84F"
        Foreground="White"
        x:Name="BtnCheck"
        Click="BtnCheck_Click"
        BorderThickness="0"
        Style="{StaticResource MaterialDesignRaisedButton}"
        materialDesign:ButtonProgressAssist.Value="-1"
        materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
        materialDesign:ButtonProgressAssist.IsIndeterminate="True">
        <StackPanel Orientation="Horizontal"
            HorizontalAlignment="Center"
            VerticalAlignment="Center">
            <materialDesign:PackIcon Kind="Check"
                Margin="0 1 5 0"/>
            <TextBlock Text="Проверить"
                FontWeight="SemiBold"/>
        </StackPanel>
    </Button>
</StackPanel>

```

Рисунок 3.4 – Разметка «MainEmployerPage»

```

</StackPanel>
<StackPanel Orientation="Horizontal"
    Margin="0 15 0 0"
    Grid.Row="1"
    HorizontalAlignment="Center"
    VerticalAlignment="Top">
    <Button Background="#D4A84F"
        Foreground="White"
        x:Name="BtnLoginEmployer"
        Click="BtnLoginEmployer_Click"
        BorderThickness="0"
        IsEnabled="False"
        Style="{StaticResource MaterialDesignRaisedButton}"
        materialDesign:ButtonProgressAssist.Value="-1"
        materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
        materialDesign:ButtonProgressAssist.IsIndeterminate="True">
        <StackPanel Orientation="Horizontal"
            HorizontalAlignment="Center"
            VerticalAlignment="Center">
            <materialDesign:PackIcon Kind="LoginVariant"
                Margin="0 1 5 0"/>
            <TextBlock Text="Войти"
                FontWeight="SemiBold"/>
        </StackPanel>
    </Button>

```

Рисунок 3.5 – Разметка «MainEmployerPage»

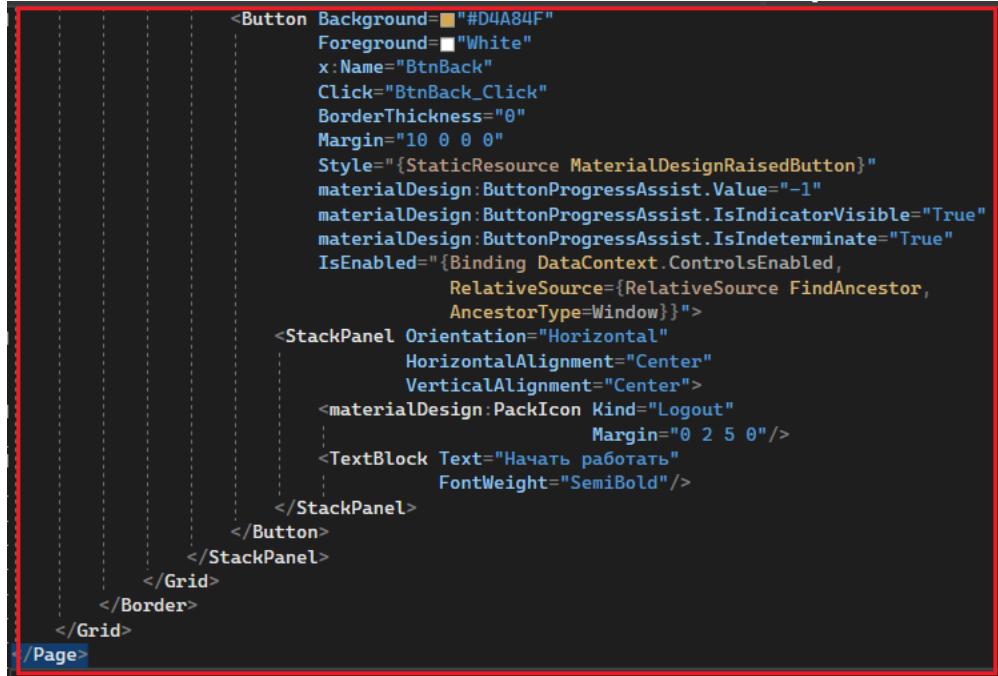


Рисунок 3.6 – Разметка «MainEmployerPage»

```
private void BtnLoginEmployer_Click(object sender, RoutedEventArgs e)
{
    FrameNavigate.FrameObject.Navigate(new LoginEmployerPage());
}

private void BtnBack_Click(object sender, RoutedEventArgs e)
{
    FrameNavigate.FrameObject.Navigate(new MainWindowLoginPage());
}

private void BtnCheck_Click(object sender, RoutedEventArgs e)
{
    try
    {
        Employer employerModel = FrameNavigate.DB.Employers.FirstOrDefault(m => m.FIO == TbCompanyName.Text);

        if (employerModel == null)
        {
            MessageBox.Show("Ошибка данных",
                           "Системное сообщение",
                           MessageBoxButton.OK,
                           MessageBoxIcon.Error);
        }
        else
        {
            MessageBox.Show("Данные проверены, доступ разрешен",
                           "Системное сообщение",
                           MessageBoxButton.OK,
                           MessageBoxIcon.Information);
            BtnCheck.IsEnabled = false;
            BtnLoginEmployer.IsEnabled = true;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(),
                       "Системная ошибка",
                       MessageBoxButton.OK,
                       MessageBoxIcon.Error);
    }
}
```

Рисунок 3.7 – Реализация логики в файле «MainEmployerPage.xaml.cs»

```
else
{
    MessageBox.Show("Данные проверены, доступ разрешен",
                   "Системное сообщение",
                   MessageBoxButton.OK,
                   MessageBoxIcon.Information);
    BtnCheck.IsEnabled = false;
    BtnLoginEmployer.IsEnabled = true;
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message.ToString(),
                   "Системная ошибка",
                   MessageBoxButton.OK,
                   MessageBoxIcon.Error);
}
```

Рисунок 3.8 – Реализация логики в файле «MainEmployerPage.xaml.cs»

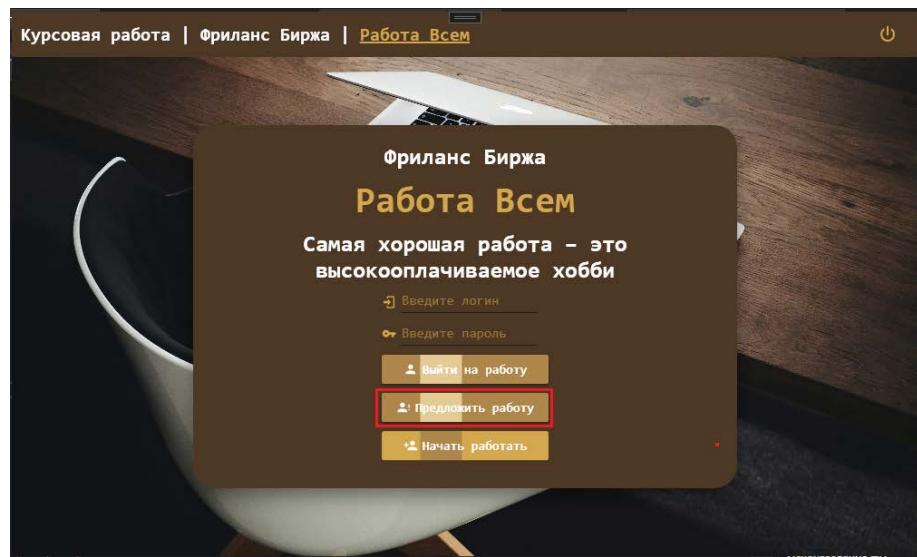


Рисунок 3.9 – Переход на страницу «MainEmployerPage»

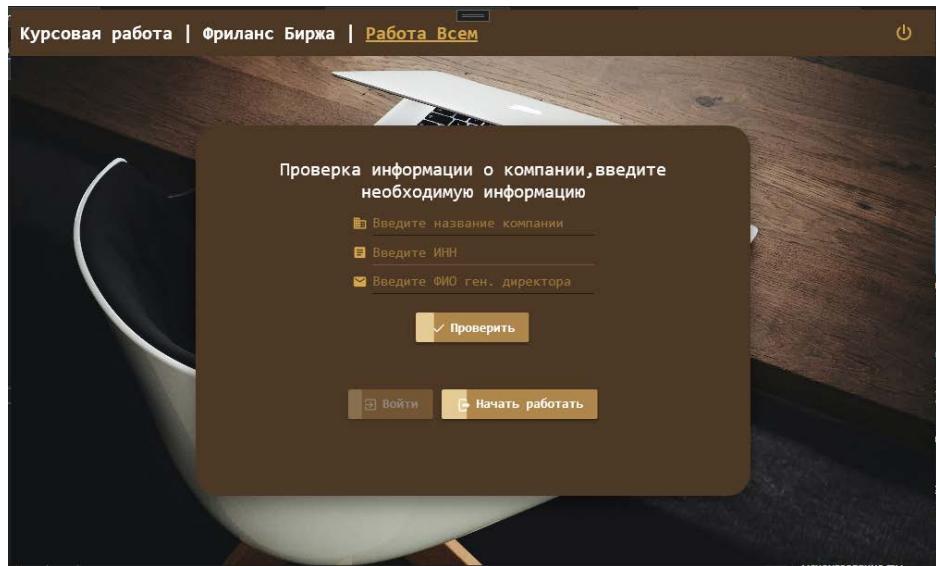


Рисунок 3.10 – Страница «MainEmployerPage»

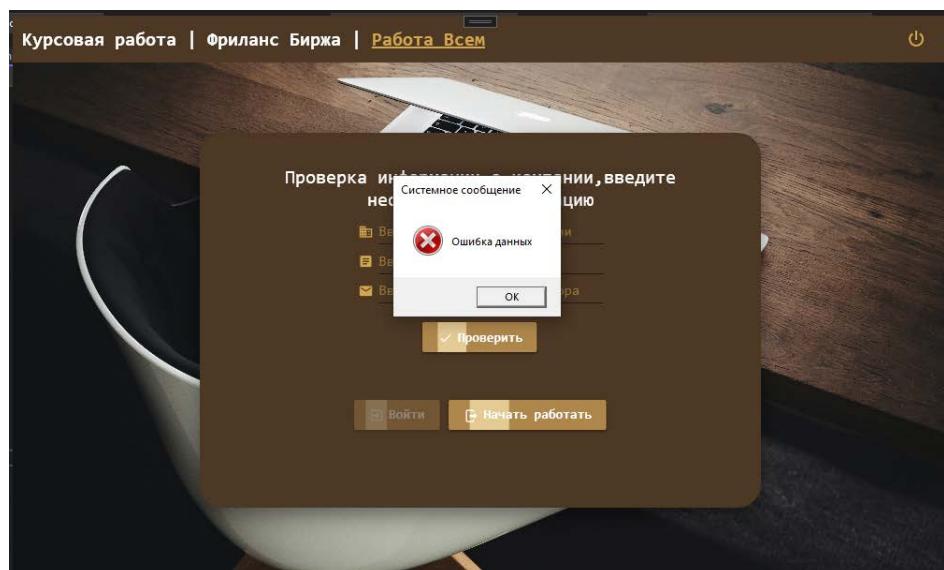


Рисунок 3.11 – Проверка на отсутствие данных

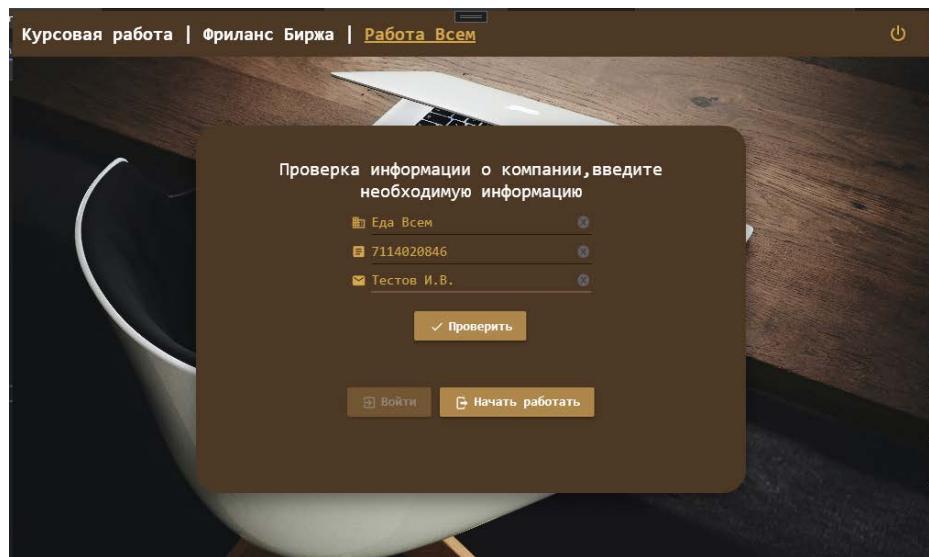


Рисунок 3.12 – Заполнение полей информацией

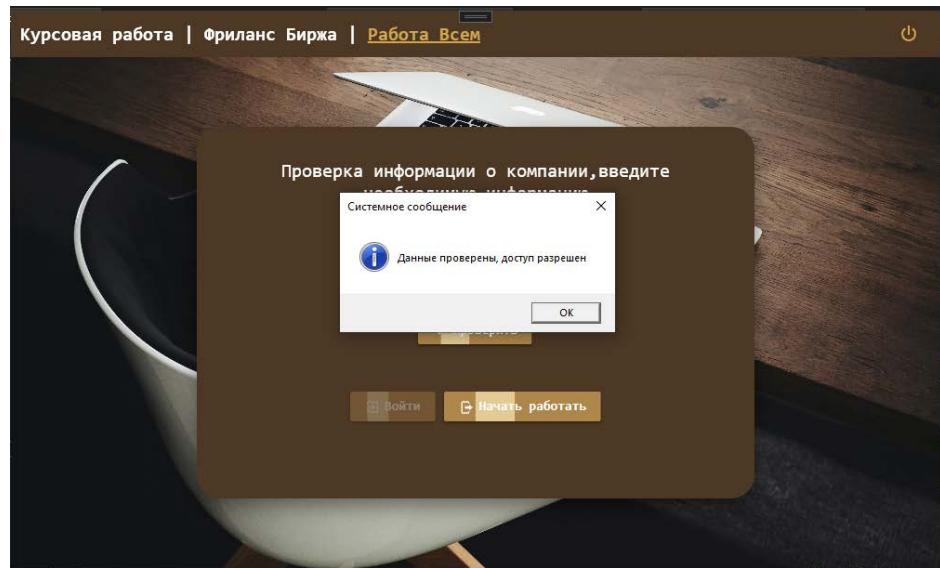


Рисунок 3.13 – Успешная проверка информации о компании

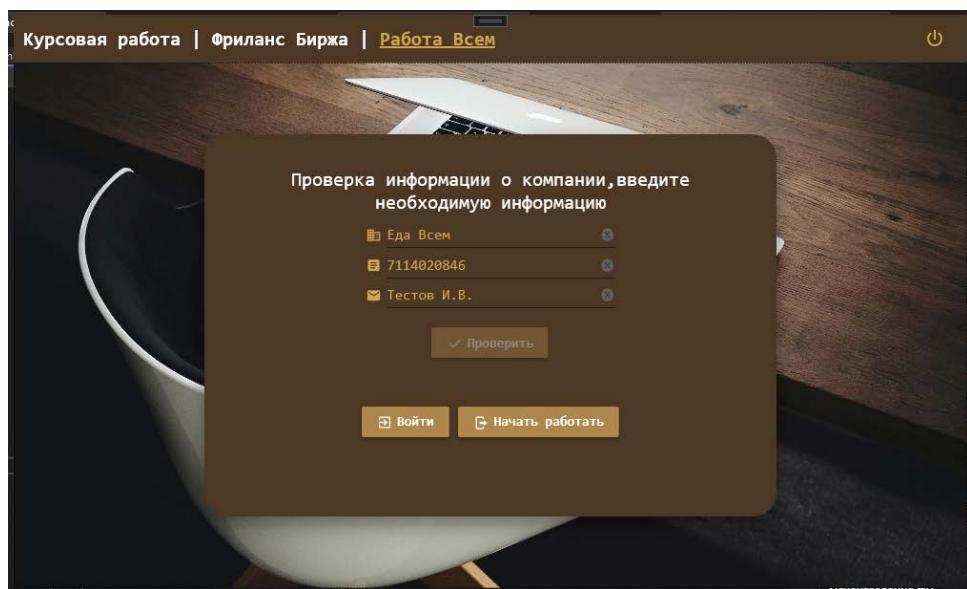


Рисунок 3.14 – Доступен переход на страницу «LoginEmployerPage»

Далее выбираем «LoginEmployerPage». На данной странице будет происходить авторизация работодателя (реализация разметки и логики представлена) на рисунках 3.15 – 3.23).

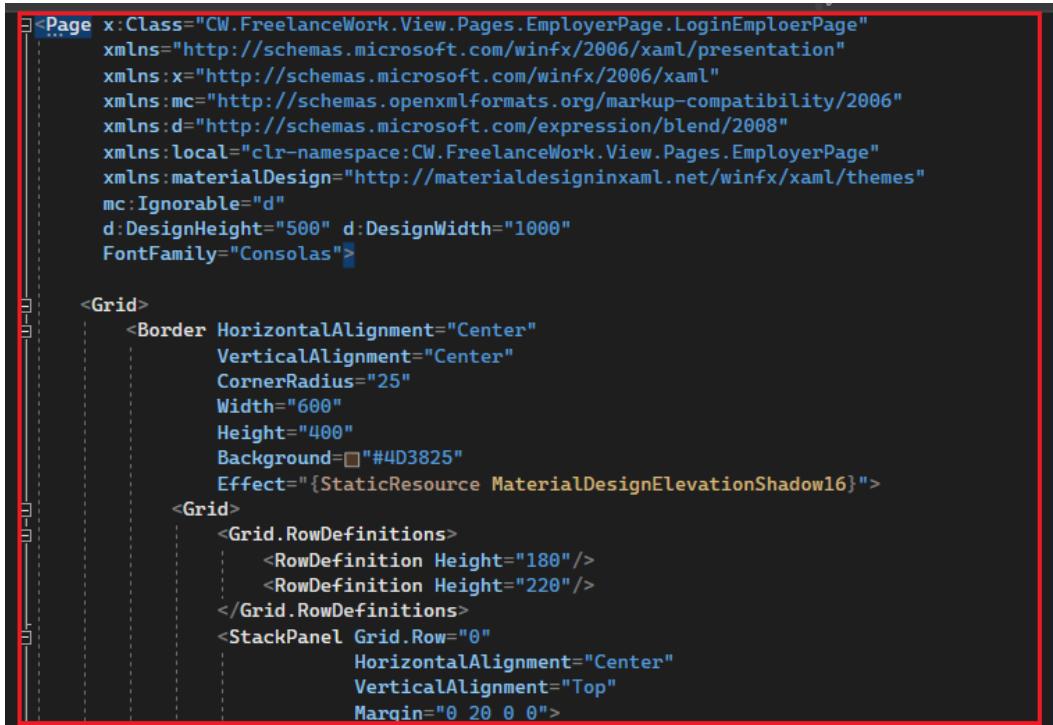


Рисунок 3.15 – Разметка «LoginEmployerPage»

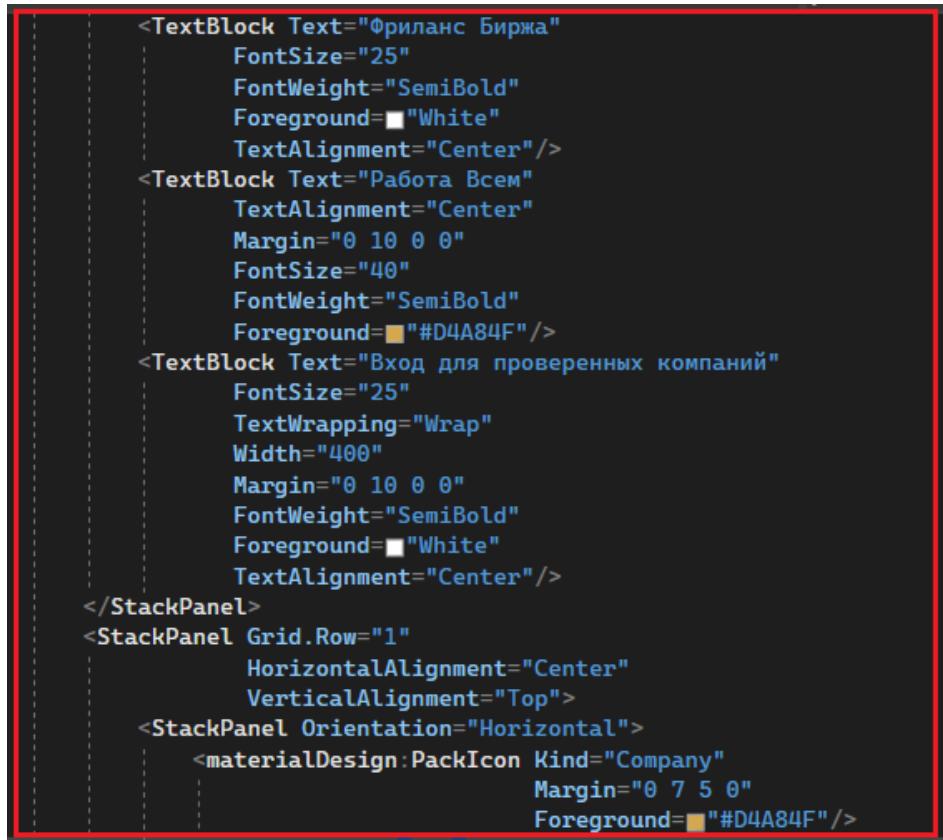


Рисунок 3.16 – Разметка «LoginEmployerPage»

```
<StackPanel>
    <TextBox Width="150"
        x:Name="TbLogin"
        FontSize="15"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Введите логин"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>

<StackPanel Orientation="Horizontal"
    Margin="0 10 0 0">
    <materialDesign:PackIcon Kind="Protected"
        Margin="0 5 5 0"
        Foreground="#D4A84F"/>
    <PasswordBox Width="150"
        x:Name="PsbPassword"
        FontSize="15"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Введите пароль"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>

<StackPanel Margin="0 10 0 0">
    <Button Background="#D4A84F"
        Foreground="White"
        x:Name="BtnLogin"
        Click="BtnLogin_Click"
        BorderThickness="0"
        Style="{StaticResource MaterialDesignRaisedButton}"
        materialDesign:ButtonProgressAssist.Value="-1"
        materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
        materialDesign:ButtonProgressAssist.IsIndeterminate="True"
        IsEnabled="{Binding DataContext.ControlsEnabled,
            RelativeSource={RelativeSource FindAncestor,
            AncestorType=Window}}">

```

Рисунок 3.17 – Разметка «LoginEmployerPage»

```
<StackPanel Orientation="Horizontal">
    <materialDesign:PackIcon Kind="Cpu32Bit"
        Margin="0 2 5 0"/>
    <TextBlock Text="Предложить работу"
        FontWeight="SemiBold"/>
</StackPanel>
</Button>

<Button Background="#D4A84F"
    Foreground="White"
    x:Name="BtnBack"
    Click="BtnBack_Click"
    BorderThickness="0"
    Margin="0 10 0 0"
    Style="{StaticResource MaterialDesignRaisedButton}"
    materialDesign:ButtonProgressAssist.Value="-1"
    materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
    materialDesign:ButtonProgressAssist.IsIndeterminate="True"
    IsEnabled="{Binding DataContext.ControlsEnabled,
        RelativeSource={RelativeSource FindAncestor,
        AncestorType=Window}}">
    <StackPanel Orientation="Horizontal">
        <materialDesign:PackIcon Kind="Logout"
            Margin="0 2 5 0"/>
        <TextBlock Text="Выйти"
            FontWeight="SemiBold"/>
    </StackPanel>
</Button>
</StackPanel>
</StackPanel>
</Grid>
</Border>
</Grid>
</Page>
```

Рисунок 3.18 – Разметка «LoginEmployerPage»

```
Ссылок: 1
private void BtnLogin_Click(object sender, RoutedEventArgs e)
{
    try
    {
        Employer userModel = FrameNavigate.DB.Employers.FirstOrDefault(u =>
            u.EmployerMail == TbLogin.Text && u.EmployerPhone == PsbPassword.Password);

        if (userModel == null)
        {
            MessageBox.Show("Ошибка данных",
                "Системное сообщение",
                MessageBoxButton.OK,
                MessageBoxImage.Error);
        }
        else
        {
            FrameNavigate.FrameObject.Navigate(new DetailEmployerPage());
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(),
            "Системная ошибка",
            MessageBoxButton.OK,
            MessageBoxImage.Error);
    }
}
```

Рисунок 3.19 – Реализация логики в файле «LoginEmployerPage.xaml.cs»

```
Ссылок: 1
private void BtnBack_Click(object sender, RoutedEventArgs e)
{
    FrameNavigate.FrameObject.Navigate(new MainWindowLoginPage());
}
```

Рисунок 3.20 – Реализация логики в файле «LoginEmployerPage.xaml.cs»

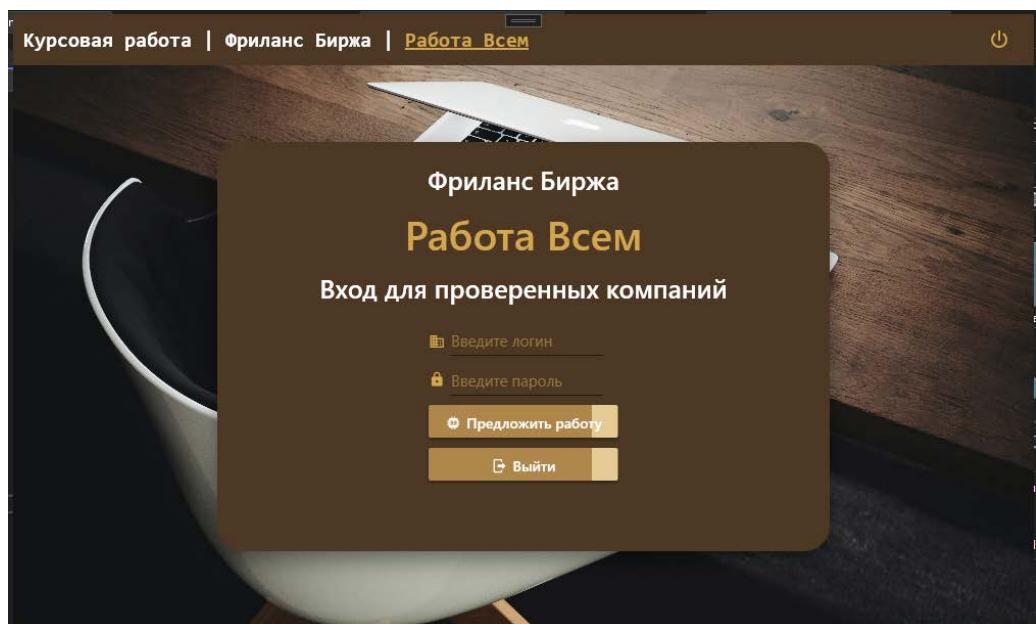


Рисунок 3.21 – Результат разметки «LoginEmployerPage»

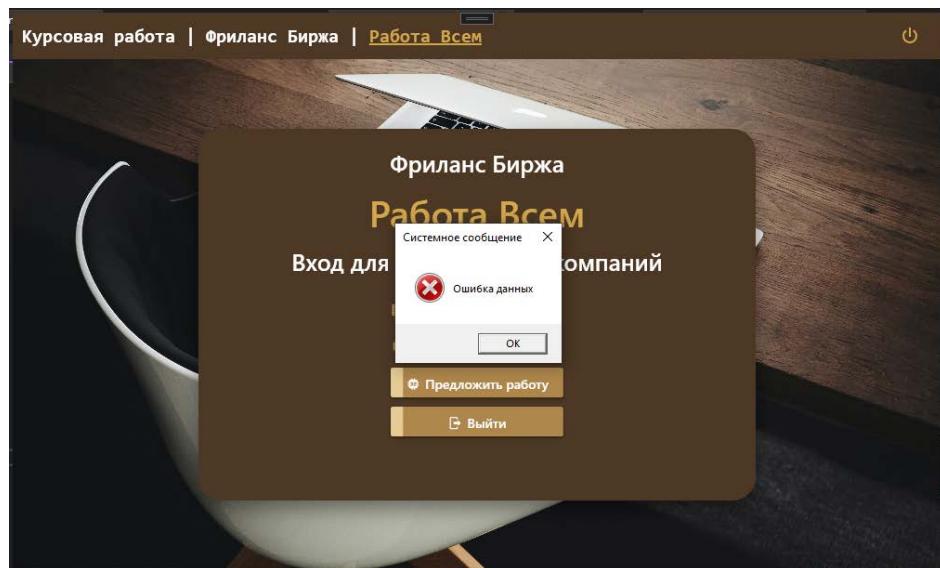


Рисунок 3.22 – Ошибка заполнения полей

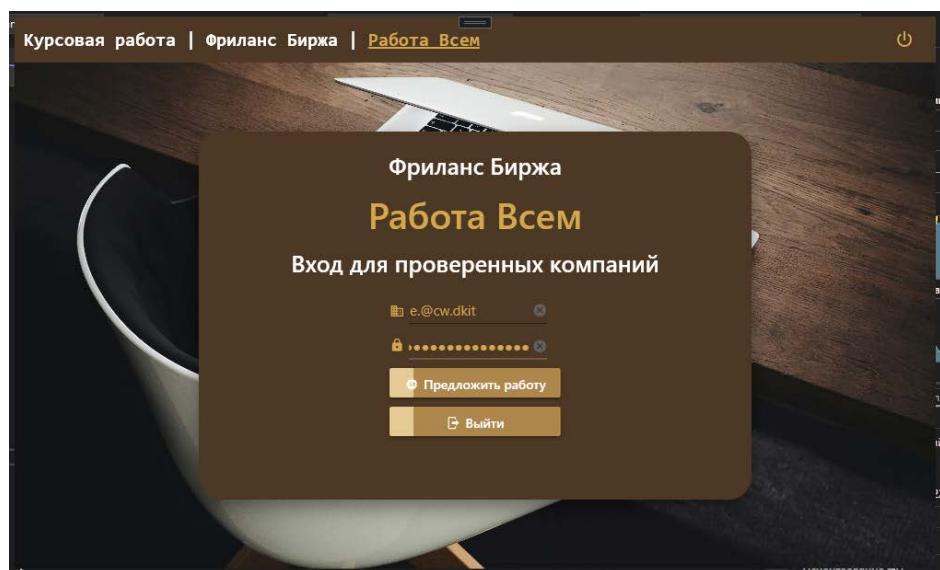


Рисунок 3.22 – Поля заполнены данными



Рисунок 3.23 – Успешная авторизация

Далее выбираем «DetailEmployerPage». На данной странице будет происходить заполнение формы по размещению заказа (реализация разметки и логики представлена) на рисунках 3.24 – 3.33).

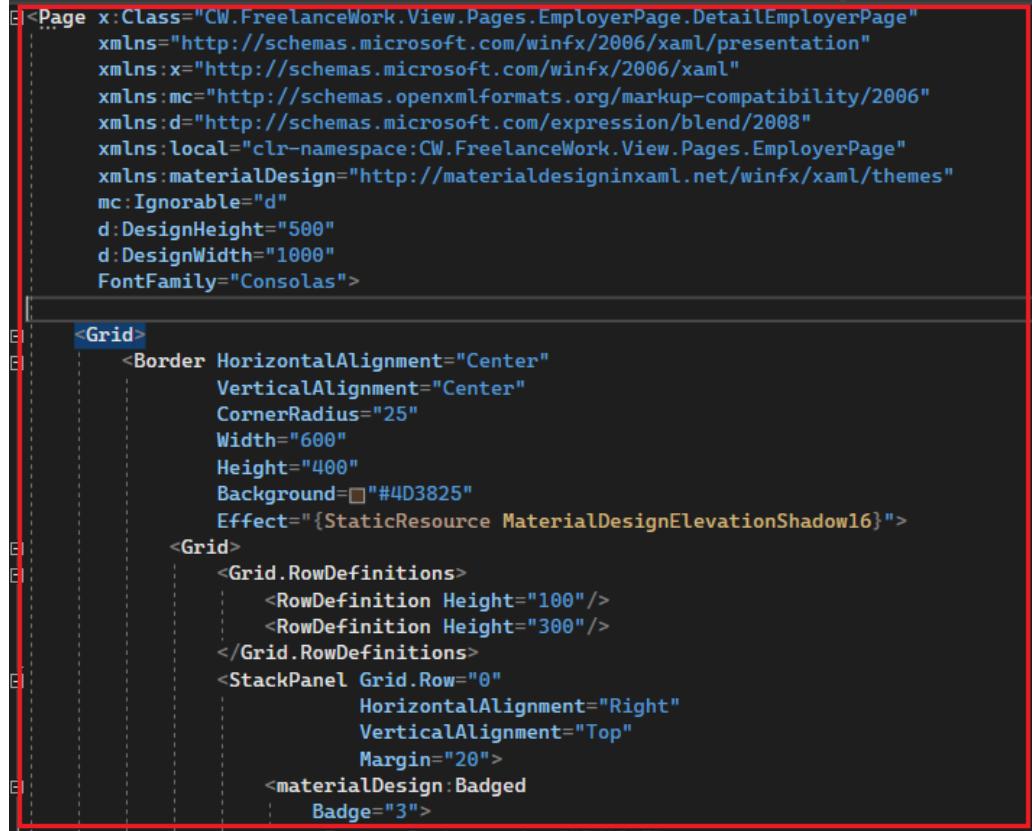


Рисунок 3.24 – Разметка «DetailEmployerPage»

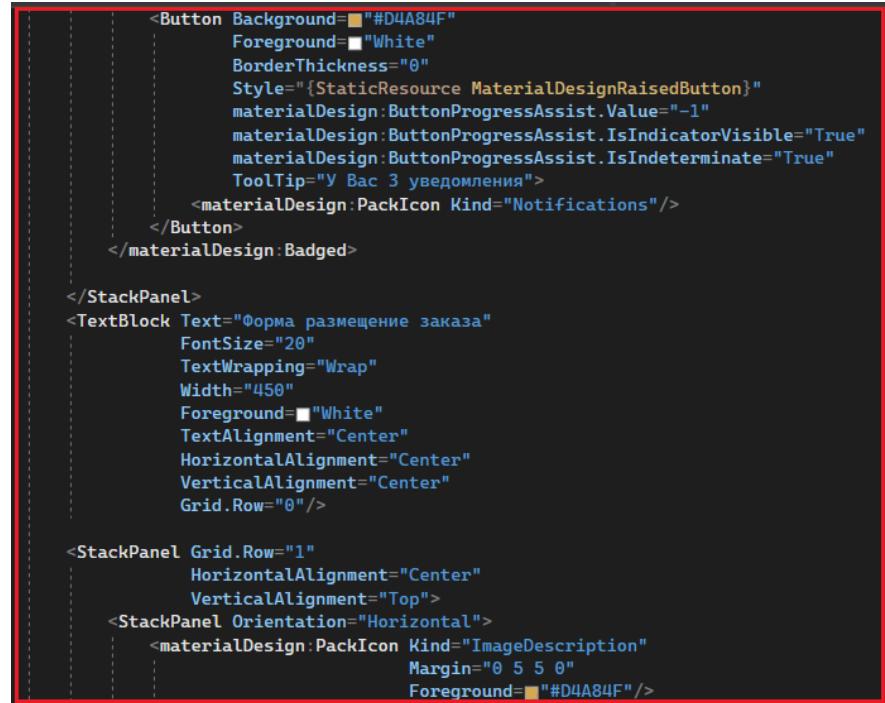


Рисунок 3.25 – Разметка «DetailEmployerPage»

```

<TextBox Width="240"
        FontSize="15"
        x:Name="TbTitle"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Введите заголовок заказа"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>

<StackPanel Orientation="Horizontal"
            Margin="0 5 0 0">
    <materialDesign:PackIcon Kind="Timer"
        Margin="0 5 5 0"
        Foreground="#D4A84F"/>
    <TextBox Width="240"
        FontSize="15"
        x:Name="TbTime"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Введите сроки исполнения"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>

<StackPanel Orientation="Horizontal"
            Margin="0 5 0 0">
    <materialDesign:PackIcon Kind="Paper"
        Margin="0 5 5 0"
        Foreground="#D4A84F"/>
    <TextBox Width="240"
        FontSize="15"
        Foreground="#D4A84F"
        IsEnabled="False"
        materialDesign:HintAssist.Hint="Добавьте файлы к заказу"
        materialDesign:TextFieldAssist.HasClearButton="True"
        ToolTip="Доступно после проверки администратором"/>
</StackPanel>

```

Рисунок 3.26 – Разметка «DetailEmployerPage»

```

<StackPanel Orientation="Horizontal"
            Margin="0 5 0 0">
    <materialDesign:PackIcon Kind="OrderAlphabeticalAscending"
        Margin="0 5 5 0"
        Foreground="#D4A84F"/>
    <TextBox Width="240"
        Height="100"
        FontSize="15"
        TextWrapping="Wrap"
        x:Name="TbOrder"
        Foreground="#D4A84F"
        materialDesign:HintAssist.Hint="Добавьте описание заказа"
        materialDesign:TextFieldAssist.HasClearButton="True"/>
</StackPanel>
</StackPanel>
<StackPanel HorizontalAlignment="Center"
            VerticalAlignment="Center"
            Margin="0 180 0 0"
            Grid.Row="2"
            Orientation="Horizontal">
    <Button Background="#D4A84F"
            Foreground="White"
            x:Name="BtnCheck"
            Click="BtnCheck_Click"
            BorderThickness="0"
            Style="{StaticResource MaterialDesignRaisedButton}"
            materialDesign:ButtonProgressAssist.Value="-1"
            materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
            materialDesign:ButtonProgressAssist.IsIndeterminate="True">
        <StackPanel Orientation="Horizontal"
                    HorizontalAlignment="Center"
                    VerticalAlignment="Center">

```

Рисунок 3.27 – Разметка «DetailEmployerPage»

```

<materialDesign:PackIcon Kind="Administrator"
    Margin="0 15 0"/>
<TextBlock Text="Отправить на согласование"
    FontWeight="SemiBold"/>
</StackPanel>
</Button>

<Button Background="#D4A84F"
    Foreground="White"
    x:Name="BtnBack"
    Click="BtnBack_Click"
    BorderThickness="0"
    Margin="10 0 0 0"
    Width="100"
    Style="{StaticResource MaterialDesignRaisedButton}"
    materialDesign:ButtonProgressAssist.Value="-1"
    materialDesign:ButtonProgressAssist.IsIndicatorVisible="True"
    materialDesign:ButtonProgressAssist.IsIndeterminate="True"
    IsEnabled="{Binding DataContext.ControlsEnabled,
        RelativeSource={RelativeSource FindAncestor,
        AncestorType=Window}}}>
<StackPanel Orientation="Horizontal">
    <materialDesign:PackIcon Kind="Logout"
        Margin="0 15 0"/>
    <TextBlock Text="Выйти"
        FontWeight="SemiBold"/>
</StackPanel>
</Button>
</StackPanel>
</Grid>
</Border>
</Grid>

```

Рисунок 3.28 – Разметка «DetailEmployerPage»

```

private void BtnCheck_Click(object sender, RoutedEventArgs e)
{
    if (string.IsNullOrEmpty(TbTitle.Text) ||
        string.IsNullOrEmpty(TbTime.Text) ||
        string.IsNullOrEmpty(TbOrder.Text))
    {
        MessageBox.Show("Все поля должны быть заполнены!",
            "Системное сообщение",
            MessageBoxButton.OK,
            MessageBoxIcon.Error);
    }
    else
    {
        MessageBox.Show("Заказ № 15 отправлен на модерирование",
            "Системное сообщение",
            MessageBoxButton.OK,
            MessageBoxIcon.Information);
        ClearTextBox();
    }
}

```

Рисунок 3.29 – Реализации логики в файле «DetailEmployerPage.xaml.cs»

```

Ссылка: 1
private void BtnBack_Click(object sender, RoutedEventArgs e)
{
    FrameNavigate.FrameObject.Navigate(new LoginEmployerPage());
}

Ссылка: 1
private void ClearTextBox()
{
    TbTitle.Text = string.Empty;
    TbTime.Text = string.Empty;
    TbOrder.Text = string.Empty;
}

```

Рисунок 3.30 – Реализации логики в файле «DetailEmployerPage.xaml.cs»

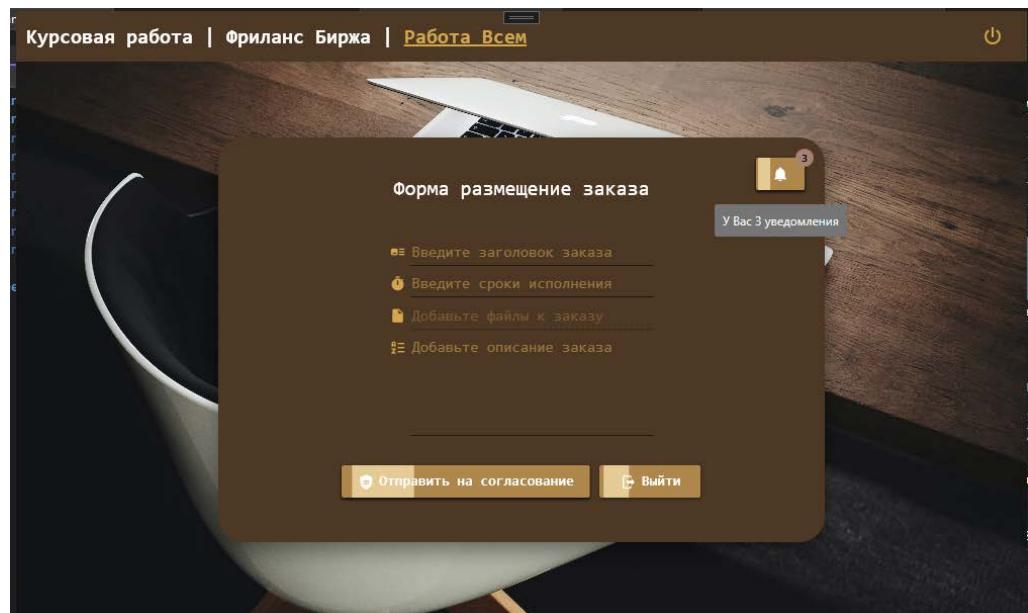


Рисунок 3.31 – Результат разметки «DetailEmployerPage»

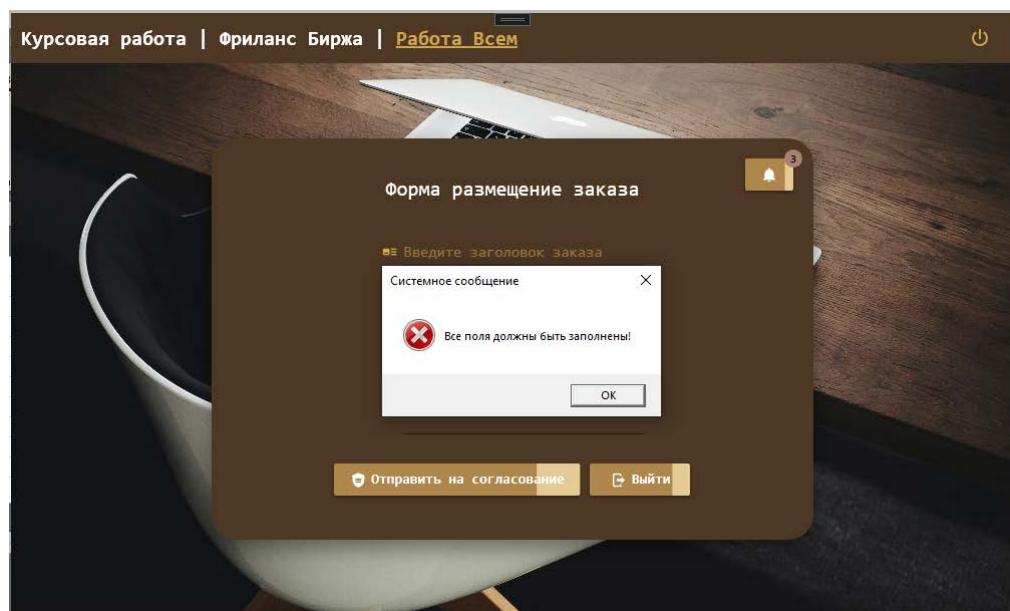


Рисунок 3.32 – Ошибка заполнения полей

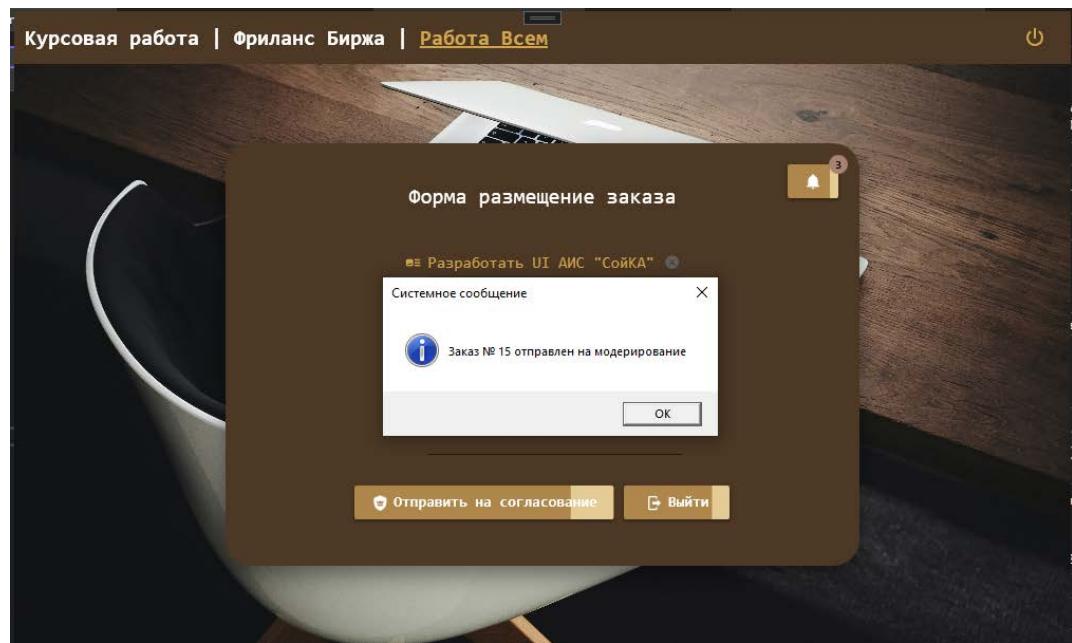
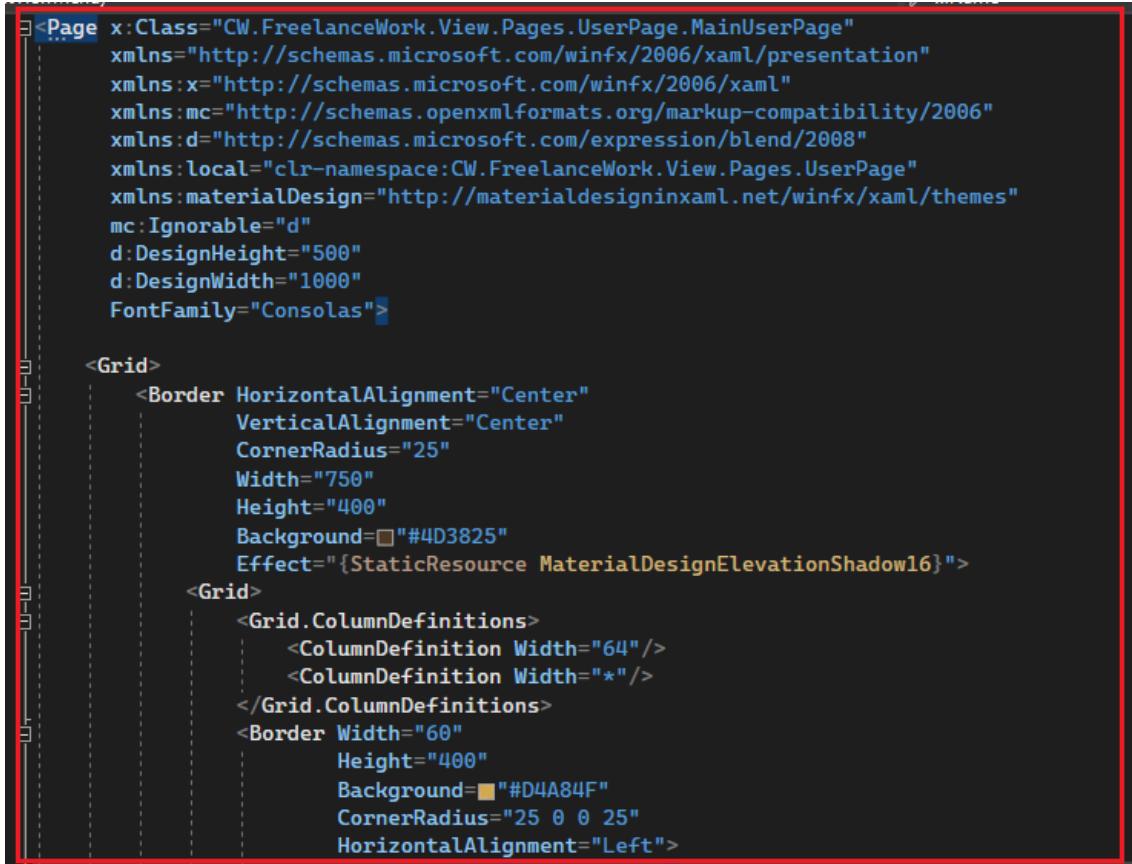


Рисунок 3.33 – Заполненная форма заказа отправлена на модерацию

4 Реализация MainUserPage

Переходим в директорию «UserPage» и создаём страницу куда будет попадать пользователь после авторизации «MainUserPage». Логика «MainUserPage» будет следующая: пользователь будет видеть заказы (которые пройдут модерацию) и сможет отправить заявку на выполнение, в свою очередь работодателю будет приходить уведомление и после одобрение пользователь может приступить к выполнению (разметка и логика «MainUserPage» представлена на рисунках 4.1 – 4.8).



```
<Page x:Class="CW.FreelanceWork.View.Pages.UserPage.MainUserPage"
      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:CW.FreelanceWork.View.Pages.UserPage"
      xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
      mc:Ignorable="d"
      d:DesignHeight="500"
      d:DesignWidth="1000"
      FontFamily="Consolas">

    <Grid>
        <Border HorizontalAlignment="Center"
                VerticalAlignment="Center"
                CornerRadius="25"
                Width="750"
                Height="400"
                Background="#4D3825"
                Effect="{StaticResource MaterialDesignElevationShadow16}">
            <Grid>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="64"/>
                    <ColumnDefinition Width="*"/>
                </Grid.ColumnDefinitions>
                <Border Width="60"
                        Height="400"
                        Background="#D4A84F"
                        CornerRadius="25 0 0 25"
                        HorizontalAlignment="Left">
```

Рисунок 4.1 – Разметка «MainUserPage»

```
<ListView x:Name="ListViewMenu"
    Margin="0 80"
    FontSize="10"
    SelectedIndex="0"
    Foreground="#White">
    <ListViewItem Cursor="Hand"
        ToolTip="Уведомления"
        IsEnabled="False">
        <StackPanel>
            <materialDesign:PackIcon Kind="Notifications"
                VerticalAlignment="Center"
                Margin="5"
                Foreground="#4D3825"/>
        </StackPanel>
    </ListViewItem>

    <ListViewItem Cursor="Hand"
        ToolTip="Настройки"
        IsEnabled="False">
        <StackPanel>
            <materialDesign:PackIcon Kind="Settings"
                VerticalAlignment="Center"
                Margin="5"
                Foreground="#4D3825"/>
        </StackPanel>
    </ListViewItem>

    <ListViewItem Cursor="Hand"
        ToolTip="Архив"
        IsEnabled="False">
    </ListViewItem>
```

Рисунок 4.2 – Разметка «MainUserPage»

```
<StackPanel>
    <materialDesign:PackIcon Kind="Archive"
        VerticalAlignment="Center"
        Margin="5"
        Foreground="#4D3825"/>
</StackPanel>
</ListViewItem>

<ListViewItem Cursor="Hand"
    ToolTip="Баланс"
    IsEnabled="False">
    <StackPanel>
        <materialDesign:PackIcon Kind="Bitcoin"
            VerticalAlignment="Center"
            Margin="5"
            Foreground="#4D3825"/>
    </StackPanel>
</ListViewItem>

<ListViewItem Cursor="Hand"
    ToolTip="Доска объявлений">
    <StackPanel>
        <materialDesign:PackIcon Kind="OrderBoolAscending"
            VerticalAlignment="Center"
            Margin="5"/>
    </StackPanel>
</ListViewItem>

<ListViewItem Cursor="Hand"
    ToolTip="Выйти"
    MouseDoubleClick="ListViewItem_MouseDoubleClick">
```

Рисунок 4.3 – Разметка «MainUserPage»

```

<StackPanel>
    <materialDesign:PackIcon Kind="Logout"
        VerticalAlignment="Center"
        Margin="5"/>
</StackPanel>
</ListViewItem>
</ListView>
</Border>
<StackPanel HorizontalAlignment="Center"
    VerticalAlignment="Center"
    Grid.Column="1"
    Margin="5">
<DataGrid x:Name="DataOrderInfo"
    FontSize="15"
    Height="380"
    Width="665"
    AutoGenerateColumns="False"
    IsReadOnly="True"
    CanUserSortColumns="False"
    Background="#D4A84F"
    RowBackground="White"
    AlternatingRowBackground="#FFF0F0"
    GridLinesVisibility="Horizontal"
    ColumnHeaderHeight="50"
    CanUserAddRows="False"
    CanUserDeleteRows="False"
    CanUserResizeRows="False"
    Effect="{StaticResource MaterialDesignElevationShadow16}">

```

Рисунок 4.4 – Разметка «MainUserPage»

```

<DataGrid.Columns>
    <DataGridTextColumn Width="Auto"
        Header="ФИРМА"
        Binding="{Binding Employer.FIO}"
        FontSize="15"/>
    <DataGridTextColumn Width="Auto"
        Header="ЗАКАЗ"
        Binding="{Binding Order}"
        FontSize="15"/>
    <DataGridTemplateColumn>
        <DataGridTemplateColumn.CellTemplate>
            <DataTemplate>
                <Button Background="#D4A84F"
                    Foreground="#4D3825"
                    x:Name="BtnOk"
                    Click="BtnOk_Click"
                    ToolTip="Оставить заявку на выполнение"
                    BorderBrush="#D4A84F"
                    Style="{StaticResource MaterialDesignFlatLightButton}">
                    <StackPanel Orientation="Horizontal">
                        <materialDesign:PackIcon Kind="Check"/>
                    </StackPanel>
                </Button>
            </DataTemplate>
        </DataGridTemplateColumn.CellTemplate>
    </DataGridTemplateColumn>
</DataGrid.Columns>
</DataGrid>
</StackPanel>
</Grid>
</Border>
</Grid>
</Page>

```

Рисунок 4.5 – Разметка «MainUserPage»

```

Ссылка 1
public MainUserPage()
{
    InitializeComponent();

    DataOrderInfo.ItemsSource = FrameNavigate.DB.OrderBoards.OrderBy(f => f.Order).ToList();
}

Ссылка 1
private void ListViewItem_MouseDoubleClick(object sender, MouseButtonEventArgs e)
{
    FrameNavigate.FrameObject.Navigate(new MainWindowLoginPage());
}

Ссылка 1
private void BtnOk_Click(object sender, RoutedEventArgs e)
{
    MessageBox.Show("Заявка отправлена на рассмотрение модерации",
                    "Системное сообщени",
                    MessageBoxButton.OK,
                    MessageBoxImage.Information);
}

```

Рисунок 4.6 – Реализация логики файла «MainUserPage.xaml.cs»

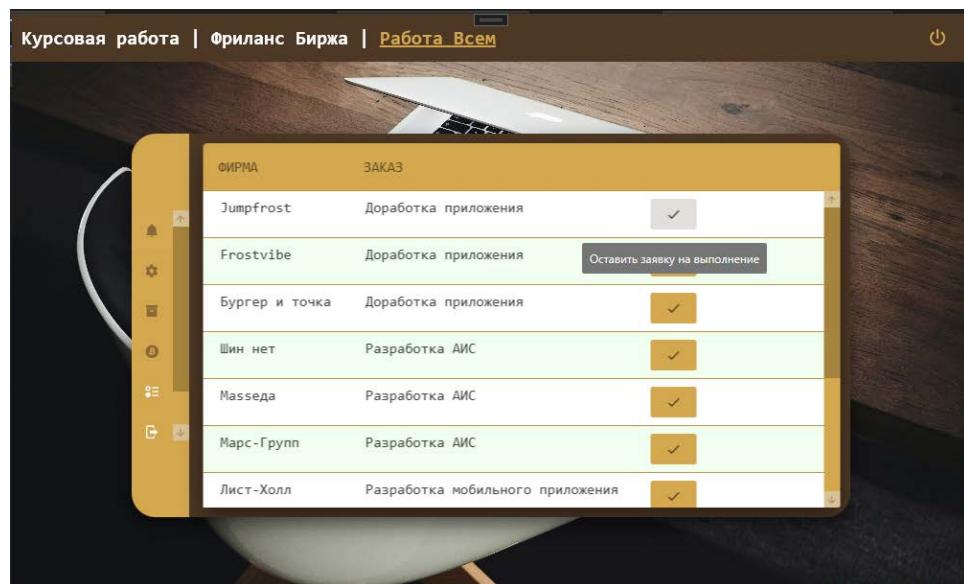


Рисунок 4.7 – Результат разметки «MainUserPage»

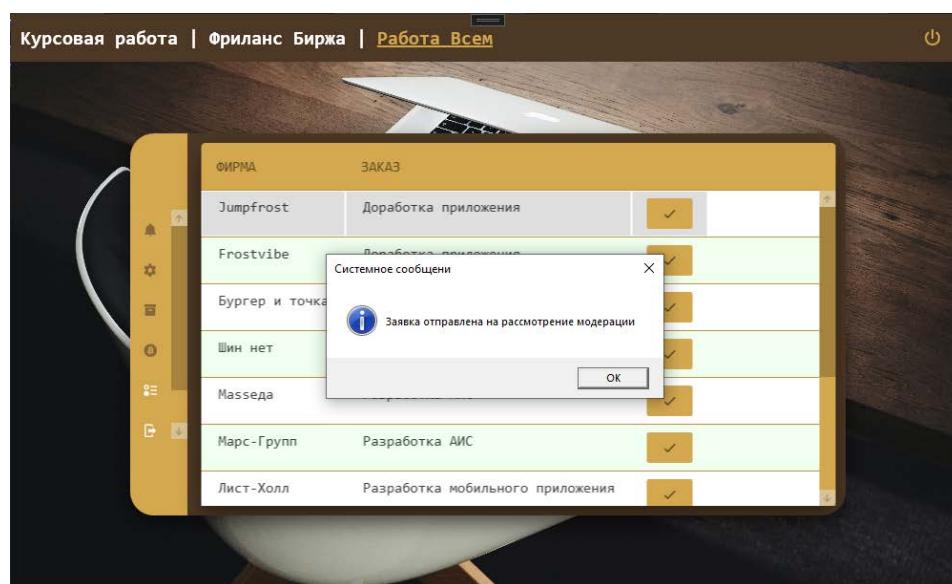


Рисунок 4.8 – Логика работы «MainUserPage»

5 Реализация MainAdministratorPage

Переходим в директорию «AdministratorPage». Для реализации логики создадим «MainAdministratorPage», директорию «AdminUserControls» и три пользовательских элемента «AdminInfoEmployerControl, AdminInfoOrderControl, AdminInfoUserControl». «MainAdministratorPage» будет содержать меню в зависимости от выбора будет подгружаться пользовательский элемент со своей логикой работы. Администратор в данном приложении будет иметь возможность удалять пользователей, заказы и работодателей (реализация разметки и разметки представлены на рисунках 5.1 – 5.).

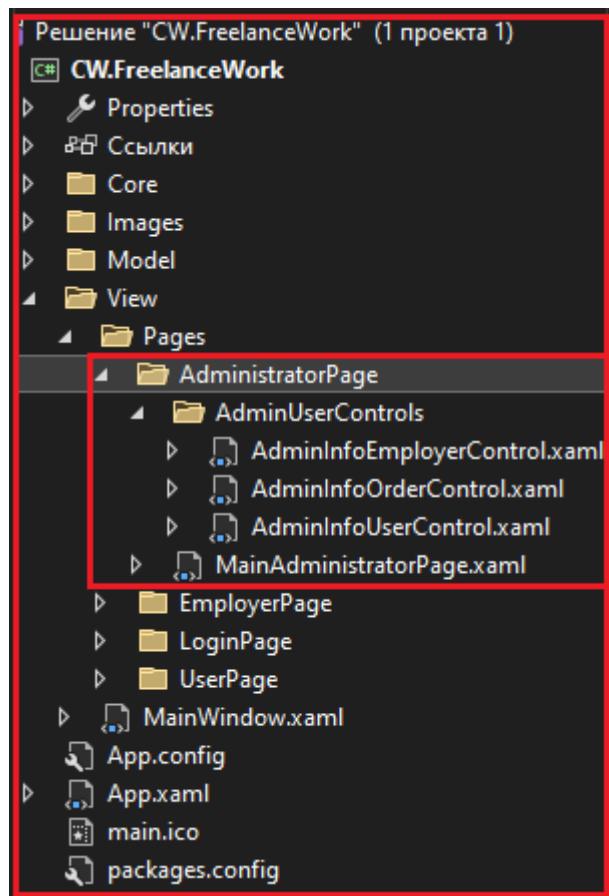


Рисунок 5.1 – Создание страницы и пользовательских элементов

```
<Page x:Class="CW.FreelanceWork.View.Pages.AdministratorPage.MainAdministratorPage"
    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:CW.FreelanceWork.View.Pages.AdministratorPage"
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    mc:Ignorable="d"
    d:DesignHeight="500"
    d:DesignWidth="1000"
    FontFamily="Consolas">

    <Grid>
        <Border HorizontalAlignment="Center"
            VerticalAlignment="Center"
            Width="980"
            Height="480"
            Background="#4D3825"
            Effect="{StaticResource MaterialDesignElevationShadow16}">
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="50"/>
                    <RowDefinition Height="*"/>
                </Grid.RowDefinitions>
                <Menu Grid.Row="0"
                    IsMainMenu="True"
                    Style="{DynamicResource MaterialDesignMenu}"
                    Background="#4D3825"
                    Foreground="LightGray">
                    <Menu.ItemsPanel>
                        <ItemsPanelTemplate>
                            <DockPanel HorizontalAlignment="Stretch" />
                        </ItemsPanelTemplate>
                    </Menu.ItemsPanel>
                
```

Рисунок 5.2 – Разметка «MainAdministratorPage»

```
<MenuItem Header="Информация о пользователях"
        Click="MenuItemUser_Click"
        HorizontalAlignment="Center"
        Cursor="Hand"
        Effect="{StaticResource MaterialDesignElevationShadow16}">
        <MenuItem.Icon>
            <materialDesign:PackIcon Kind="People"
                Foreground="LightGray"
                Width="20"
                Height="20" />
        </MenuItem.Icon>
    </MenuItem>
    <MenuItem Header="Информация о работодателях"
        Click="MenuItemEmployer_Click"
        HorizontalAlignment="Center"
        Cursor="Hand"
        Effect="{StaticResource MaterialDesignElevationShadow16}">
        <MenuItem.Icon>
            <materialDesign:PackIcon Kind="PeopleGroup"
                Foreground="LightGray"
                Width="20"
                Height="20" />
        </MenuItem.Icon>
    </MenuItem>
    <MenuItem Header="Информация о заказах"
        Click="MenuItemOrder_Click"
        HorizontalAlignment="Center"
        Cursor="Hand"
        Effect="{StaticResource MaterialDesignElevationShadow16}">
        <MenuItem.Icon>
```

Рисунок 5.3 – Разметка «MainAdministratorPage»

```
<materialDesign:PackIcon Kind="OrderBoolAscending"
    Foreground="LightGray"
    Width="20"
    Height="20"/>
</MenuItem.Icon>
</MenuItem>
</Menu>
<Grid Grid.Row="1"
    x:Name="GridContentLoad"/>
</Grid>
</Border>
</Grid>
</Page>
```

Рисунок 5.4 – Разметка «MainAdministratorPage»

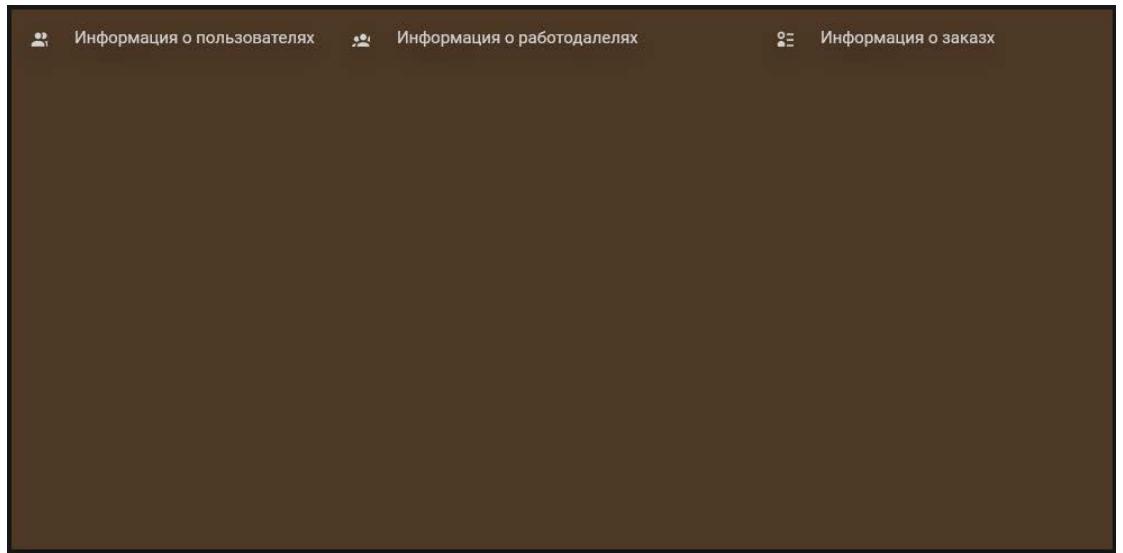


Рисунок 5.5 – Результат разметки «MainAdministratorPage»

```
Ссылок: 1
private void MenuItemUser_Click(object sender, RoutedEventArgs e)
{
    GridContentLoad.Children.Clear();
    GridContentLoad.Children.Add(new AdminInfoUserControl());
}

Ссылок: 1
private void MenuEmployer_Click(object sender, RoutedEventArgs e)
{
    GridContentLoad.Children.Clear();
    GridContentLoad.Children.Add(new AdminInfoEmployerControl());
}

Ссылок: 1
private void MenuItemOrder_Click(object sender, RoutedEventArgs e)
{
    GridContentLoad.Children.Clear();
    GridContentLoad.Children.Add(new AdminInfoOrderControl());
}
```

Рисунок 5.5 – Реализация логики файла «MainAdministratorPage.xaml.cs»

```

<UserControl x:Class="CW.FreelanceWork.View.Pages.AdministratorPage.AdminUserControls.AdminInfoEmployerControl"
    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:CW.FreelanceWork.View.Pages.AdministratorPage.AdminUserControls"
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    mc:Ignorable="d"
    d:DesignHeight="500"
    d:DesignWidth="1000"
    FontFamily="Consolas">

    <Grid>
        <StackPanel HorizontalAlignment="Center"
                    VerticalAlignment="Center"
                    Margin="5">
            <DataGrid x:Name="DataEmployerInfo"
                      FontSize="15"
                      Height="400"
                      Width="950"
                      AutoGenerateColumns="False"
                      IsReadOnly="True"
                      CanUserSortColumns="False"
                      Background="#D4A84F"
                      RowBackground="White"
                      AlternatingRowBackground="#F0FFF0"
                      GridLinesVisibility="Horizontal"
                      ColumnHeaderHeight="50"
                      CanUserAddRows="False"
                      CanUserDeleteRows="False"
                      CanUserResizeRows="False"
                      Effect="{StaticResource MaterialDesignElevationShadow16}">

```

Рисунок 5.5 – Разметка «AdminInfoEmployerControl»

```

        <DataGrid.Columns>
            <DataGridTextColumn Width="Auto" Header="НАЗВАНИЕ КОМПАНИИ" Binding="{Binding FIO}" FontSize="15"/>
            <DataGridTextColumn Width="Auto" Header="МОБ. НОМЕР" Binding="{Binding EmployerPhone}" FontSize="15"/>
            <DataGridTextColumn Width="Auto" Header="ЭЛЕКТРОННАЯ ПОЧТА" Binding="{Binding EmployerMail}" FontSize="15"/>
            <DataGridTemplateColumn>
                <DataGridTemplateColumn.CellTemplate>
                    <DataTemplate>
                        <Button Background="#D4A84F"
                                Foreground="#4D3825"
                                x:Name="BtnDelete"
                                Click="BtnDelete_Click"
                                ToolTip="Удалить компанию"
                                BorderBrush="#D4A84F"
                                Style="{StaticResource MaterialDesignFlatLightButton}">
                            <StackPanel Orientation="Horizontal">
                                <materialDesign:PackIcon Kind="Remove"/>
                            </StackPanel>
                        </Button>
                    </DataTemplate>
                </DataGridTemplateColumn.CellTemplate>
            </DataGridTemplateColumn>
        </DataGrid.Columns>
    </DataGrid>
</StackPanel>
</Grid>
</UserControl>

```

Рисунок 5.6 – Разметка «AdminInfoEmployerControl»

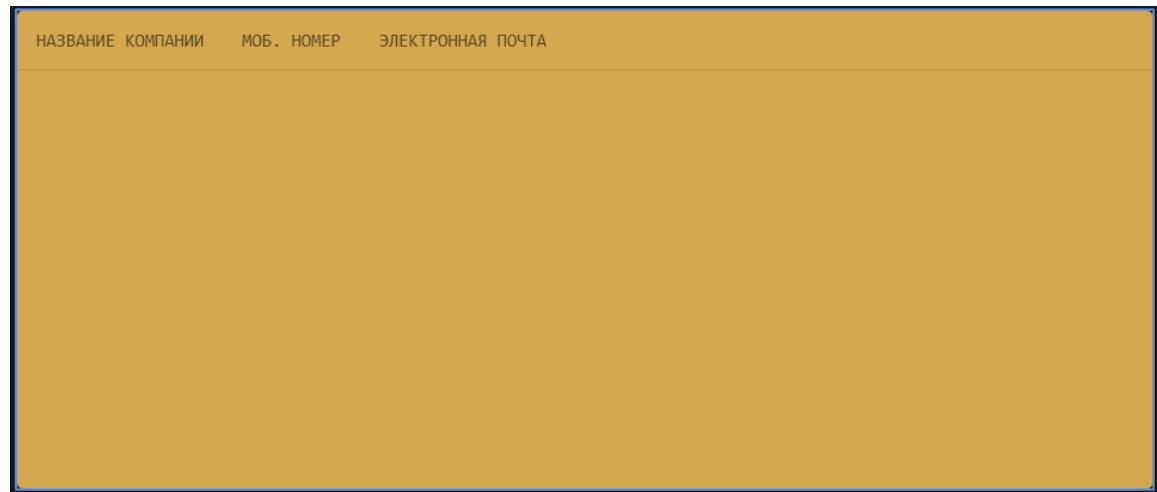


Рисунок 5.7 – Результат разметки «AdminInfoEmployerControl»

```

Ссылка 1
public AdminInfoEmployerControl()
{
    InitializeComponent();
    DataEmployerInfo.ItemsSource = FrameNavigate.DB.Employers.OrderBy(w => w.FIO).ToList();
}

Ссылка 1
private void BtnDelete_Click(object sender, RoutedEventArgs e)
{
    int idEmployer = (DataEmployerInfo.SelectedItem as Employer).EmployerID;
    var result = MessageBox.Show("Хотите удалить компанию?", "Системное сообщение",
        MessageBoxButton.YesNo,
        MessageBoxImage.Question);

    if (result == MessageBoxResult.Yes)
    {
        Employer employer = (from w in FrameNavigate.DB.Employers where w.EmployerID == idEmployer select w).SingleOrDefault();
        FrameNavigate.DB.Employers.Remove(employer);
        FrameNavigate.DB.SaveChanges();
        DataEmployerInfo.ItemsSource = FrameNavigate.DB.Employers.OrderBy(w => w.FIO).ToList();
    }
}

```

Рисунок 5.8 – Реализация логики файла «AdminInfoEmployerControl.xaml.cs»

```

<UserControl x:Class="CW.FreelanceWork.View.Pages.AdministratorPage.AdminUserControls.AdminInfoUserControl"
    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:CW.FreelanceWork.View.Pages.AdministratorPage.AdminUserControls"
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    mc:Ignorable="d"
    d:DesignHeight="500"
    d:DesignWidth="1000"
    FontFamily="Consolas">

    <Grid>
        <StackPanel HorizontalAlignment="Center"
                    VerticalAlignment="Center"
                    Margin="5">
            <DataGrid x:Name="DataUserInfo"
                      FontSize="15"
                      Height="400"
                      Width="950"
                      AutoGenerateColumns="False"
                      IsReadOnly="True"
                      CanUserSortColumns="False"
                      Background="#D4A84F"
                      RowBackground="White"
                      AlternatingRowBackground="#FFF"
                      GridLinesVisibility="Horizontal"
                      ColumnHeaderHeight="50"
                      CanUserAddRows="False"
                      CanUserDeleteRows="False"
                      CanUserResizeRows="False"
                      Effect="{StaticResource MaterialDesignElevationShadow16}">

```

Рисунок 5.9 – Разметка «AdminInfoUserControl»

```

<DataGrid.Columns>
    <DataGridTextColumn Width="Auto" Header="ФИО" Binding="{Binding FIO}" FontSize="15"/>
    <DataGridTextColumn Width="Auto" Header="МОБ. НОМЕР" Binding="{Binding UserPhone}" FontSize="15"/>
    <DataGridTextColumn Width="Auto" Header="ЭЛЕКТРОННАЯ ПОЧТА" Binding="{Binding UserMail}" FontSize="15"/>
    <DataGridTextColumn Width="Auto" Header="НАВЫКИ" Binding="{Binding UserSkills}" FontSize="15"/>
    <DataGridTemplateColumn>
        <DataGridTemplateColumn.CellTemplate>
            <DataTemplate>
                <Button Background="#D4A84F"
                        Foreground="#D3825"
                        x:Name="BtnDelete"
                        Click="BtnDelete_Click"
                        ToolTip="Удалить пользователя"
                        BorderBrush="#D4A84F"
                        Style="{StaticResource MaterialDesignFlatLightButton}">
                    <StackPanel Orientation="Horizontal">
                        <materialDesign:PackIcon Kind="Remove"/>
                    </StackPanel>
                </Button>
            </DataTemplate>
        </DataGridTemplateColumn.CellTemplate>
    </DataGridTemplateColumn>
</DataGrid.Columns>
</DataGrid>
</StackPanel>
</Grid>
</UserControl>

```

Рисунок 5.10 – Разметка «AdminInfoUserControl»



Рисунок 5.11 – Результат разметки «AdminInfoUserControl»

```
Ссылка: 1
public AdminInfoUserControl()
{
    InitializeComponent();
    DataUserInfo.ItemsSource = FrameNavigate.DB.Users.OrderBy(u => u.FIO).ToList();
}

Ссылка: 1
private void BtnDelete_Click(object sender, RoutedEventArgs e)
{
    int idUser = (DataUserInfo.SelectedItem as User).UserID;
    var result = MessageBox.Show("Хотите удалить пользователя?",
        "Системное сообщение",
        MessageBoxButton.YesNo,
        MessageBoxImage.Question);

    if (result == MessageBoxResult.Yes)
    {
        User user = (from u in FrameNavigate.DB.Users where u.UserID == idUser select u).SingleOrDefault();
        FrameNavigate.DB.Users.Remove(user);
        FrameNavigate.DB.SaveChanges();
        DataUserInfo.ItemsSource = FrameNavigate.DB.Users.OrderBy(u => u.FIO).ToList();
    }
}
```

Рисунок 5.12 – Реализация логики файла «AdminInfoUserControl.xaml.cs»

```
<UserControl x:Class="CW.FreelanceWork.View.Pages.AdministratorPage.AdminUserControls.AdminInfoOrderControl"
    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/expressionblend/2008"
    xmlns:local="clr-namespace:CW.FreelanceWork.View.Pages.AdministratorPage.AdminUserControls"
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    mc:Ignorable="d"
    d:DesignHeight="500"
    d:DesignWidth="1000"
    FontFamily="Consolas">

    <Grid>
        <StackPanel HorizontalAlignment="Center"
                    VerticalAlignment="Center"
                    Margin="5">
            <DataGrid x:Name="DataOrderInfo"
                      FontSize="15"
                      Height="400"
                      Width="950"
                      AutoGenerateColumns="False"
                      IsReadOnly="True"
                      CanUserSortColumns="False"
                      Background="#D4A84F"
                      RowBackground="White"
                      AlternatingRowBackground="#FFF"
                      GridLinesVisibility="Horizontal"
                      ColumnHeaderHeight="50"
                      CanUserAddRows="False"
                      CanUserDeleteRows="False"
                      CanUserResizeRows="False"
                      Effect="{StaticResource MaterialDesignElevationShadow16}">
                </DataGrid>
        </StackPanel>
    </Grid>

```

Рисунок 5.12 – Разметка «AdminInfoOrderControl»

```

<DataGrid>
    <DataGrid.Columns>
        <DataGridTextColumn Width="Auto" Header="ЗАКАЗ" Binding="{Binding Order}" FontSize="15"/>
        <DataGridTextColumn Width="Auto" Header="НАЗВАНИЕ КОМПАНИИ" Binding="{Binding Employer.FIO}" FontSize="15"/>
        <DataGridTextColumn Width="Auto" Header="ФРИЛАНСЕР" Binding="{Binding User.FIO}" FontSize="15"/>
        <DataGridTemplateColumn>
            <DataGridTemplateColumn.CellTemplate>
                <DataTemplate>
                    <Button Background="#D4A84F"
                            Foreground="#4D3825"
                            x:Name="BtnDelete"
                            Click="BtnDelete_Click"
                            ToolTip="Удалить заказ"
                            BorderBrush="#D4A84F"
                            Style="{StaticResource MaterialDesignFlatLightButton}">
                        <StackPanel Orientation="Horizontal">
                            <materialDesign:PackIcon Kind="Remove"/>
                        </StackPanel>
                    </Button>
                </DataTemplate>
            </DataGridTemplateColumn.CellTemplate>
        </DataGridTemplateColumn>
    </DataGrid.Columns>
</DataGrid>
</StackPanel>
</Grid>
</UserControl>

```

Рисунок 5.13 – Разметка «AdminInfoOrderControl»



Рисунок 5.14 – Результат разметки «AdminInfoOrderControl»

```

Ссылка: 1
public AdminInfoOrderControl()
{
    InitializeComponent();
    DataOrderInfo.ItemsSource = FrameNavigate.DB.OrderBoards.OrderBy(w => w.Order).ToList();
}

Ссылка: 1
private void BtnDelete_Click(object sender, RoutedEventArgs e)
{
    int idOrder = (DataOrderInfo.SelectedItem as OrderBoard).OrderBoardID;
    var result = MessageBox.Show("Хотите удалить заказ?", 
                                "Системное сообщение",
                                MessageBoxButton.YesNo,
                                MessageBoxImage.Question);

    if (result == MessageBoxResult.Yes)
    {
        OrderBoard orderBoard = (from b in FrameNavigate.DB.OrderBoards where b.OrderBoardID == idOrder select b).SingleOrDefault();
        FrameNavigate.DB.OrderBoards.Remove(orderBoard);
        FrameNavigate.DB.SaveChanges();
        DataOrderInfo.ItemsSource = FrameNavigate.DB.OrderBoards.OrderBy(w => w.Order).ToList();
    }
}

```

Рисунок 5.15 – Реализация логики файла «AdminInfoOrderControl.xaml.cs»

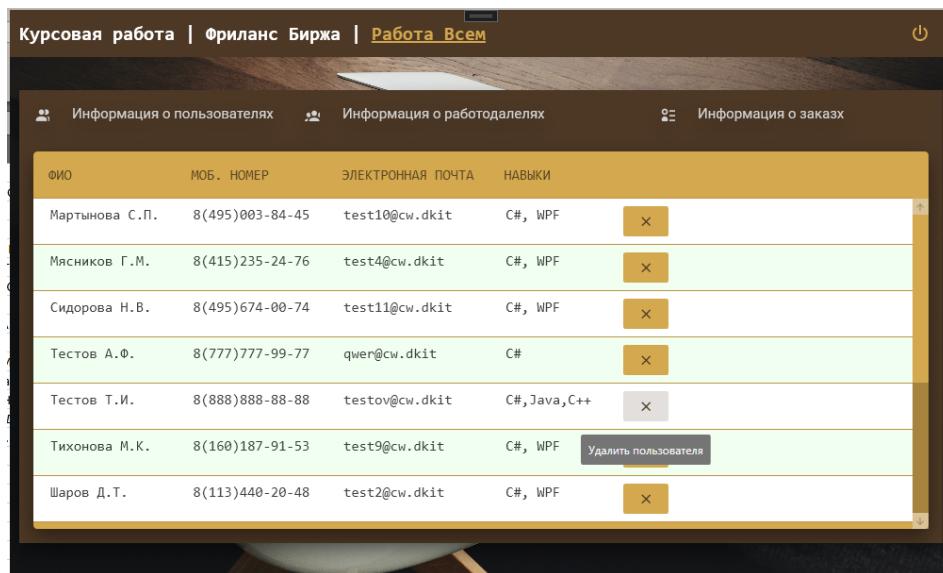


Рисунок 5.16 – Результат работы «AdminUserControls»

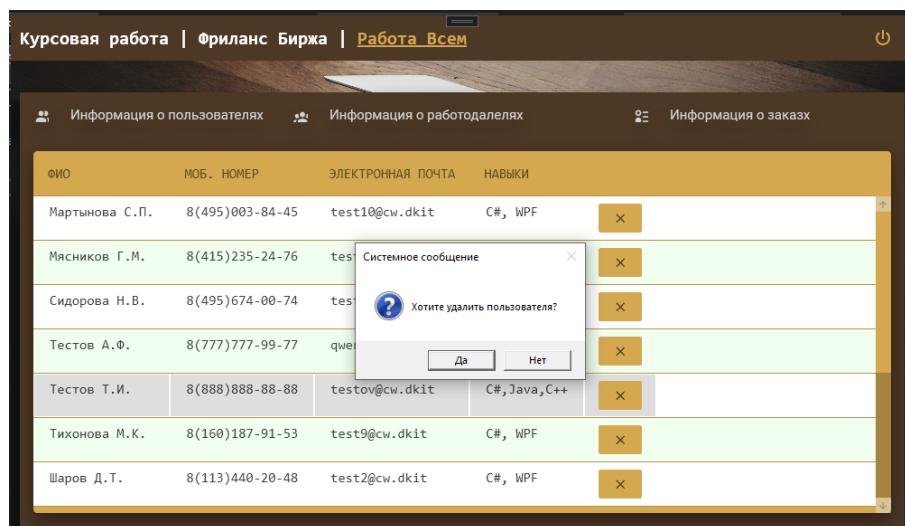


Рисунок 5.17 – Результат работы «AdminUserControls»

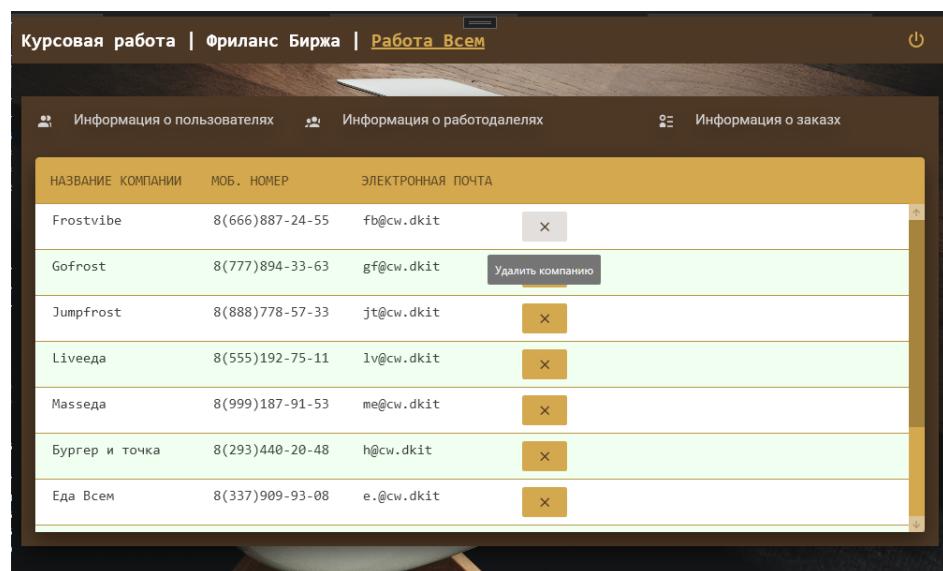


Рисунок 5.18 – Результат работы «AdminInfoEmployerControl»

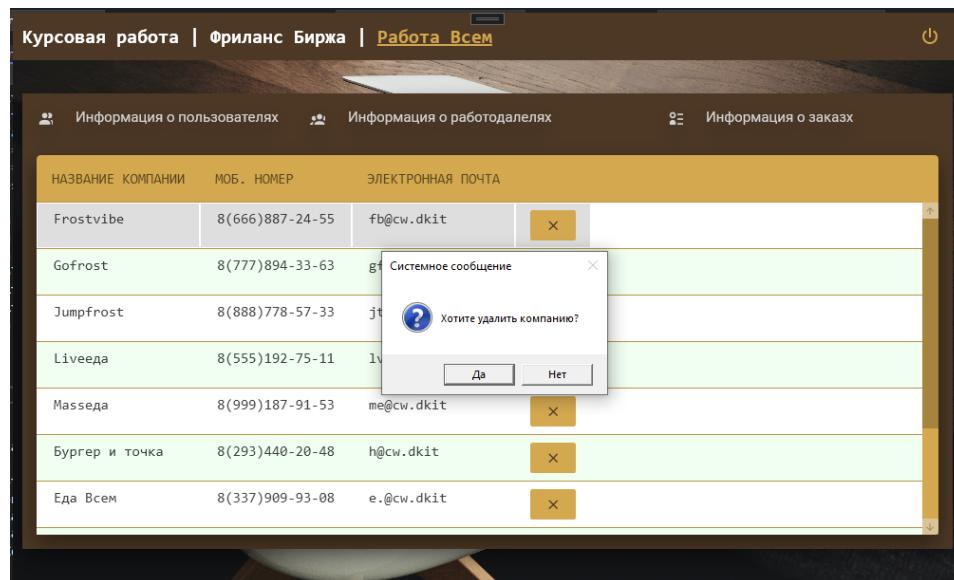


Рисунок 5.19 – Результат работы «AdminInfoEmployerControl»

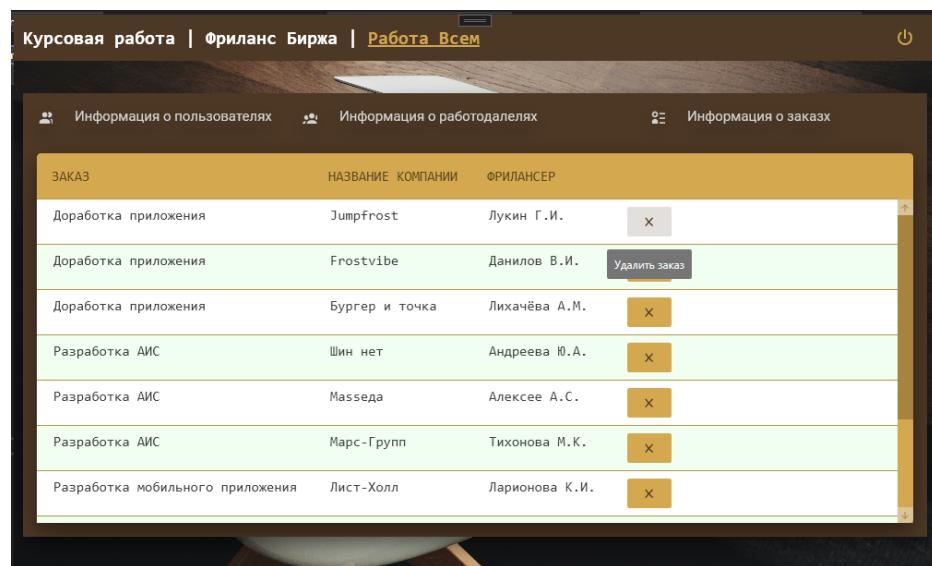


Рисунок 5.20 – Результат работы «AdminInfoOrderControl»

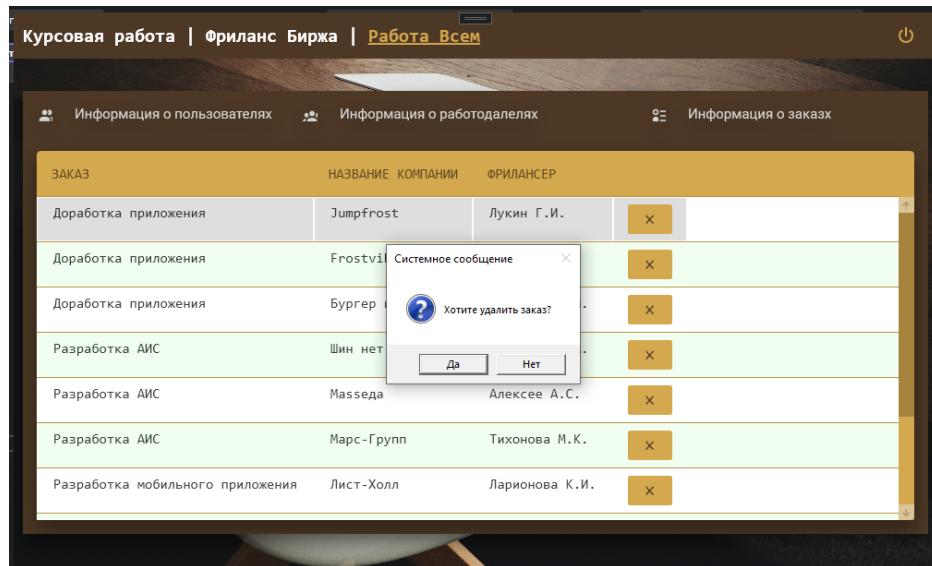


Рисунок 5.21 – Результат работы «AdminInfoOrderControl»