use master UsersDetail(idUser) on delete cascade on delete cascade on update cascade, update cascade foreign key(idStatus) references Statuses(id) database on delete cascade on update cascade create); [KP_Saulin_INCOMSYSTEM]; create table SpecializationsEmployee(); id bigint primary key identity(1,1), create table Chats(use [KP_Saulin_INCOMSYSTEM] idEmployee bigint not null, idChat bigint primary key, idSpecialization int not null, idCustomer bigint not null, go create table Positions(foreign key(idEmployee) references idManager bigint null, id tinyint primary key identity(1,1), Employees(idUser) on delete cascade on dateCreate datetime not null, [name] nvarchar(75) not null update cascade, foreign key(idCustomer) references): foreign key(idSpecialization) references Customers(idUser), create table UsersDetail(Specializations(id) on delete cascade on foreign key(idManager) references idUser bigint primary key identity(1,1), update cascade Employees(idUser), [login] nvarchar(75) not null,); foreign key(idChat) references Orders(id) on [password] nvarchar(75) not null, delete cascade on update cascade create table Tasks(phone bigint null check((len(phone) = 11) id bigint primary key identity(1,1), OR phone IS NULL), [name] nvarchar(75) not null, create table [Messages](idPos tinyint not null, idSpecialization int not null, id bigint primary key identity(1,1), passport bigint not null check(len(passport) [description] nvarchar(max) not null, idChat bigint not null, = 10) unique, price decimal(18,2) not null, idUser bigint not null, [address] nvarchar(175) not null, discount tinyint null, [message] text null, dateStart datetime null approxCompleteTime int not null, attachment varbinary(max) null, not default(getdate()), supportPeriod int not null, fileExtension nvarchar(10) null, foreign key(idPos) references Positions(id) attachment varbinary(max) null, dateSend datetime null on delete cascade on update cascade fileExtension nvarchar(10) null, default(getdate()), foreign key(idSpecialization) references foreign key(idChat) references create table Specializations(Specializations(id) on delete cascade on Chats(idChat), id int primary key identity(1,1), update cascade foreign kev(idUser) references [name] nvarchar(75) not null UsersDetail(idUser),);); create table Statuses(create table LegalForms(insert into Positions ([name]) VALUES id tinyint primary key identity(1,1), [name] nvarchar(75) not null id int primary key identity(1,1), (N'Заказчик'); [name] nvarchar(75) not null insert into Positions ([name]) VALUES): create table Orders((N'Исполнитель');); create table Customers(id bigint primary key identity(1,1), insert into Positions ([name]) VALUES idUser bigint primary key, idCustomer bigint not null, (N'Менеджер'); [name] nvarchar(150) not null, idExecutor bigint null, insert into Statuses ([name]) VALUES idLegalForm int not null, idTask bigint not null, (N'Ожидание'); foreign key(idUser) references price decimal(18,2) not null, insert into Statuses ([name]) VALUES UsersDetail(idUser) on delete cascade on difficulty float not null default(1), (N'Обсуждение'); update cascade, dateOrder datetime not null, insert into Statuses ([name]) VALUES key(idLegalForm) planDateStart datetime null, (N'Выполнение'); foreign references LegalForms(id) on delete cascade on update factDateStart datetime null, insert into Statuses ([name]) VALUES cascade planDateComplete datetime null, (N'Выполнено'): factDateComplete datetime null, insert into Statuses ([name]) VALUES): create table Employees(attachment varbinary(max) null, (N'Отклонён'); fileExtension nvarchar(10) null, insert into LegalForms ([name]) VALUES idUser bigint primary key, [surname] nvarchar(75) not null, idStatus tinyint not null, (N'Физическое лицо'); [name] nvarchar(75) not null, foreign key(idCustomer) references insert into LegalForms ([name]) VALUES [patronymic] nvarchar(75) null, Customers(idUser), (N'Юридическое лицо'); key(idExecutor) foreign references Employees(idUser),

foreign

key(idUser)

foreign key(idTask) references Tasks(id) on

Приложение А.

Приложение В.	public AuthWindow()	<rowdefinition height="Auto"></rowdefinition>
AuthWindow.xaml	{	<rowdefinition height="Auto"></rowdefinition>
<window< td=""><td><pre>InitializeComponent();</pre></td><td><rowdefinition></rowdefinition></td></window<>	<pre>InitializeComponent();</pre>	<rowdefinition></rowdefinition>
x:Class="INCOMSYSTEM.Windows.Auth	AuthFrame = AFrame;	
Window"	AuthFrame.Navigated +=	<grid.columndefinitions></grid.columndefinitions>
xmlns="http://schemas.microsoft.com/winf	AuthFrameOnNavigated;	<columndefinition width="50"></columndefinition>
x/2006/xaml/presentation"	AuthFrame.Navigate(new AuthPage());	<columndefinition></columndefinition>
xmlns:x="http://schemas.microsoft.com/wi	}	
nfx/2006/xaml"	private void AuthFrameOnNavigated(object	<controls:toppanelcontrol< td=""></controls:toppanelcontrol<>
xmlns:d="http://schemas.microsoft.com/ex	sender, NavigationEventArgs e)	Grid.Column="0" Grid.ColumnSpan="2"/>
pression/blend/2008"	{	<grid <="" grid.column="0" grid.row="1" td=""></grid>
xmlns:mc="http://schemas.openxmlformats	var page = ((Frame)sender).Content as Page;	Grid.ColumnSpan="2">
.org/markup-compatibility/2006"	this.Title = page?.Title;	<grid.columndefinitions></grid.columndefinitions>
xmlns:controls="clr-	}	<columndefinition width="Auto"></columndefinition>
namespace:INCOMSYSTEM.Controls"	public static Frame AuthFrame { get; private	<columndefinition></columndefinition>
mc:Ignorable="d"	set; }	<columndefinition width="Auto"></columndefinition>
AllowsTransparency="True"	}	
WindowStyle="None"	}	<stackpanel <="" orientation="Horizontal" td=""></stackpanel>
ResizeMode="NoResize"	,	Margin="55 0">
WindowStartupLocation="CenterScreen"	MainWindow.xaml	<button <="" td="" x:name="BackBtn"></button>
Height="475"	<window< td=""><td>Content="<-"</td></window<>	Content="<-"
Width="600"	x:Class="INCOMSYSTEM.Windows.Main	Click="BackBtn_Click"/>
Background="DimGray"	Window"	<button <="" td="" x:name="BackChatBtn"></button>
BorderBrush="Black"	xmlns="http://schemas.microsoft.com/winf	Content="<-"
BorderThickness="2">	x/2006/xaml/presentation"	Visibility="Collapsed"
<windowchrome.windowchrome></windowchrome.windowchrome>	xmlns:x="http://schemas.microsoft.com/wi	IsEnabled="False"
<windowchrome <="" captionheight="0" td=""><td>nfx/2006/xaml"</td><td>Click="BackChatBtn_Click"/></td></windowchrome>	nfx/2006/xaml"	Click="BackChatBtn_Click"/>
GlassFrameThickness="0"	xmlns:d="http://schemas.microsoft.com/ex	
NonClientFrameEdges="None"	pression/blend/2008"	<stackpanel <="" grid.column="2" td=""></stackpanel>
ResizeBorderThickness="5"/>	xmlns:mc="http://schemas.openxmlformats	Orientation="Horizontal">
	.org/markup-compatibility/2006"	<textblock <="" td="" x:name="AuthBlock"></textblock>
<grid></grid>	xmlns:controls="clr-	VerticalAlignment="Center"
<grid.rowdefinitions></grid.rowdefinitions>	namespace:INCOMSYSTEM.Controls"	Style="{StaticResource TextBlockStyle}"/>
<rowdefinition height="Auto"></rowdefinition>	mc:Ignorable="d"	<image <="" source="/Images/user.png" td=""/>
<rowdefinition></rowdefinition>	AllowsTransparency="True"	Width="32"
	WindowStyle="None"	Height="32"
<controls:toppanelcontrol></controls:toppanelcontrol>	WindowStartupLocation="CenterScreen"	Margin="10 0 20 0"/>
<frame <="" td="" x:name="AFrame"/> <td>Height="600"</td> <td></td>	Height="600"	
Grid.Row="1"	MinHeight="600"	
Margin="5"	Width="950"	<pre><frame <="" pre="" x:name="MFrame"/></pre>
IsTabStop="False"	MinWidth="950"	Grid.Column="1"
NavigationUIVisibility="Hidden"/>	WindowState="Maximized"	Grid.Row="2"
	Background="DimGray"	Margin="5"
	BorderBrush="Black"	IsTabStop="False"
	BorderThickness="2">	NavigationUIVisibility="Hidden"/>
AuthWindow.xaml.cs	<windowchrome.windowchrome></windowchrome.windowchrome>	<pre><controls:borderex <="" grid.column="0" pre=""></controls:borderex></pre>
using System.Windows.Controls;	<windowchrome <="" captionheight="0" td=""><td>Grid.ColumnSpan="2"</td></windowchrome>	Grid.ColumnSpan="2"
using System.Windows.Navigation;	GlassFrameThickness="0"	HorizontalAlignment="Left"
using INCOMSYSTEM.Pages;	NonClientFrameEdges="None"	Grid.Row="1"
namespace INCOMSYSTEM.Windows	ResizeBorderThickness="5"/>	Grid.RowSpan="2"
{		BorderBrush="Coral"
public partial class AuthWindow	<grid></grid>	BorderThickness="0 0 1 0">
{	<grid.rowdefinitions></grid.rowdefinitions>	<pre><controls:borderex.style></controls:borderex.style></pre>
· ·		Shadish order 2 more from

<style targettype="controls:BorderEx"></th><th><Setter Property="FontSize" Value="18"/></th><th><Button Grid.Row="2"</th></tr><tr><td><pre><Setter Property="Width" Value="50"/></pre></td><td></style> <td>Style="{StaticResource SideBarButton}"</td>	Style="{StaticResource SideBarButton}"	
<setter <="" property="Panel.ZIndex" td=""><td></td><td>Click="QuitBtn_Click"</td></setter>		Click="QuitBtn_Click"
Value="1"/>	<grid margin="8 0 0 0"></grid>	Margin="0 0 0 15">
<setter <="" property="Background" td=""><td><grid.rowdefinitions></grid.rowdefinitions></td><td><button.content></button.content></td></setter>	<grid.rowdefinitions></grid.rowdefinitions>	<button.content></button.content>
Value="#F0696969"/>	<rowdefinition height="Auto"></rowdefinition>	<stackpanel orientation="Horizontal"></stackpanel>
<style.triggers></style.triggers>	<rowdefinition></rowdefinition>	<image< td=""></image<>
<trigger <="" property="IsMouseEnter" td=""><td><rowdefinition height="Auto"></rowdefinition></td><td>Source="/Images/shutdown.png"/></td></trigger>	<rowdefinition height="Auto"></rowdefinition>	Source="/Images/shutdown.png"/>
Value="True">		<textblock text="Выйти"></textblock>
<trigger.enteractions></trigger.enteractions>	<button <="" style="{StaticResource</td><td></StackPanel></td></tr><tr><td><BeginStoryboard></td><td>SideBarButton}" td=""><td></td></button>	
<storyboard></storyboard>	Click="MainMenuBtn_Click">	
<doubleanimation< td=""><td><button.content></button.content></td><td></td></doubleanimation<>	<button.content></button.content>	
Storyboard.TargetProperty="Width"	<stackpanel orientation="Horizontal"></stackpanel>	
To="200"	<pre><image source="/Images/logo.png"/></pre>	
Duration="0:0:0.8">	<textblock text="INCOMSYSTEM"></textblock>	
<doubleanimation.easingfunction></doubleanimation.easingfunction>		
<sineease easingmode="EaseInOut"></sineease>		MainWindow.xaml.cs
		using System.Windows;
	<stackpanel <="" grid.row="1" td=""><td>using System.Windows.Controls;</td></stackpanel>	using System.Windows.Controls;
	VerticalAlignment="Center">	using System.Windows.Navigation;
	<pre><button <="" style="{StaticResource</pre></td><td>using System. Windows. Media. Animation;</td></tr><tr><td><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre></td><td>SideBarButton}" td=""><td></td></button></pre>	
	,	using System;
<trigger.exitactions></trigger.exitactions>	Click="ReviewBtn_Click">	using System.Collections.Generic;
<beginstoryboard></beginstoryboard>	<button.content></button.content>	using System.Threading.Tasks;
<storyboard></storyboard>	<stackpanel orientation="Horizontal"></stackpanel>	using INCOMSYSTEM.Context;
<doubleanimation< td=""><td><image source="/Images/home.png"/></td><td>using System.Linq;</td></doubleanimation<>	<image source="/Images/home.png"/>	using System.Linq;
Storyboard.TargetProperty="Width"	<textblock text="Обзор"></textblock>	using INCOMSYSTEM.Pages;
To="50"		using INCOMSYSTEM.Pages.MainPages;
Duration="0:0:0.8">		using System.Windows.Input;
<doubleanimation.easingfunction></doubleanimation.easingfunction>		namespace INCOMSYSTEM.Windows
<sineease easingmode="EaseInOut"></sineease>	<button <="" td="" x:name="ProfileBtn"><td>{</td></button>	{
	Style="{StaticResource SideBarButton}"	public partial class MainWindow : Window
	Margin="0 5" Click="ProfileBtn_Click">	{
	<button.content></button.content>	public MainWindow()
	<stackpanel orientation="Horizontal"></stackpanel>	{
	<image< td=""><td><pre>InitializeComponent();</pre></td></image<>	<pre>InitializeComponent();</pre>
	Source="/Images/profileSettings.png"/>	ReviewFrame = new Frame();
	<textblock техt="Профиль"></textblock>	ChatFrame = new Frame();
		Navigation Commands. Browse Back. Input G
		estures.Clear();
<controls:borderex.resources></controls:borderex.resources>		NavigationCommands.BrowseForward.Inp
<style targettype="Image"></td><td><Button x:Name="ChatBtn"</td><td>utGestures.Clear();</td></tr><tr><td><Setter Property="Height" Value="32"/></td><td>Style="{StaticResource SideBarButton}"</td><td>NavigationCommands.BrowseHome.Input</td></tr><tr><td><Setter Property="Width" Value="32"/></td><td>Margin="0 5" Click="ChatBtn_Click"></td><td>Gestures.Clear();</td></tr><tr><td></skyle></td><td><Button.Content></td><td>NavigationCommands.BrowseStop.InputGe</td></tr><tr><td><Style TargetType="TextBlock"</td><td><pre><StackPanel Orientation="Horizontal"></pre></td><td>stures.Clear();</td></tr><tr><td>BasedOn="{StaticResource</td><td><pre><Image Source="/Images/chat.png"/></pre></td><td>_sideBarMenu.Add(MenuItems.Review,</td></tr><tr><td>TextBlockStyle}"></td><td>Source ::/mlages/chat.png /> <TextBlock Text="Mecceнджер"/></td><td>null);</td></tr><tr><td><pre><Setter Property="VerticalAlignment"</pre></td><td>*/StackPanel></td><td>_sideBarMenu.Add(MenuItems.Profile,</td></tr><tr><td>Value="Center"/></td><td></Button.Content></td><td></td></tr><tr><td></td><td></Button></td><td>null);</td></tr><tr><td><Setter Property="Margin" Value="10 0 0 0"/></td><td></Button> </StackPanel></td><td>_sideBarMenu.Add(MenuItems.Chat, null); IsClosed = false;</td></tr><tr><td></td><td></td><td>IsClosed = false;</td></tr></tbody></table></style>		

```
ReviewFrame.Navigated
                                                   break;
                                                                                                                      (!Equals(MFrame.Content,
                                      +=
ReviewFrameOnNavigated;
                                                   case
                                                                                          3:
                                                                                                      _sideBarMenu[MenuItems.Chat])) return;
ChatFrame.Navigated
                                                   ReviewFrame.Navigate(new
                                                                                                      var page = ((Frame)sender).Content as Page;
                                      +=
ChatFrameOnNavigated;
                                                   ManagerPage());
                                                                                                      this.Title
                                                                                                                                    page?.Title;
if
         (AuthUser
                                                   break:
                                                                                                      _sideBarMenu[MenuItems.Chat] = page;
                                    null)
                                                   default:
                                                                                                      Nav(page);
AuthBlock.Text = "Вы вошли как гость!";
                                                                                                      BackChatBtn.IsEnabled
                                                   MessageBox.Show("Такого
                                                                                                                                    !(page
                                                                                 окна
                                                                                         не
                                                                                                                                             is
ProfileBtn.Visibility = Visibility.Collapsed;
                                                                                                      ChatListPage);
                                                   существует, ты
                                                                      как
                                                                            вообше
                                                                                       сюда
ChatBtn.Visibility = Visibility.Collapsed;
                                                   попал?");
ReviewFrame.Navigate(new
                                                   this.Close();
                                                                                                      public static Frame ReviewFrame { get;
ViewTasksPage());
                                                   break;
                                                                                                      private
                                                                                                                            set;
return;
                                                                                                      public static Frame ChatFrame { get; private
                                                                                                      private void BackBtn_Click(object sender,
SetAuthUserText();
                                                   public static UsersDetail AuthUser { get; set;
SetStartPage();
                                                                                                      RoutedEventArgs
                                                   }
                                                   private
                                                                                        void
private
            void
                       SetAuthUserText()
                                                   ReviewFrameOnNavigated(object sender,
                                                                                                      ReviewFrame.GoBack();
                                                   NavigationEventArgs
{
                                                                                          e)
                                                                                                      private void QuitBtn_Click(object sender,
using
           (var
                     db
                                     new
                                                   {
INCOMSYSTEMEntities())
                                                   if
                                                                   (!Equals(MFrame.Content,
                                                                                                      RoutedEventArgs
                                                   _sideBarMenu[MenuItems.Review]))
                                                                                                      IsClosed
if
        (AuthUser.idPos
                                       1)
                                                   return;
                                                                                                                                           true:
                                                   var page = ((Frame)sender). Content as Page;
                                                                                                      new
                                                                                                                         AuthWindow().Show();
var user = db.Customers.First(s => s.idUser
                                                   this.Title
                                                                                                      this.Close();
                                                                                 page?.Title;
                        AuthUser.idUser);
                                                   _sideBarMenu[MenuItems.Review] = page;
AuthBlock.Text
                         $"{user.name}";
                                                   Nav(page);
                                                                                                      private readonly Dictionary<MenuItems,
                                                   BackBtn.IsEnabled
}
                                                                                                                  _sideBarMenu
                                                                                                                                           new
else
                                                   ReviewFrame.CanGoBack;
                                                                                                      Dictionary<MenuItems,
                                                                                                                                       Page>();
                                                                                                      public static bool IsClosed { get; private set;
var user = db.Employees.First(s => s.idUser
                                                                          Nav(Page
                                                   private
                                                           async
                                                                    void
                                                                                      page)
==
                        AuthUser.idUser);
                                                                                                      private Page ChatListPage { get; set; }
AuthBlock.Text
                                                                                                                        ReviewBtn_Click(object
                               $"{user}";
                                                                          DoubleAnimation
                                                                                                      private
                                                     var
                                                         anim =
                                                                     new
                                                                                                      sender,
                                                                                                                     RoutedEventArgs
                                                                                                                                             e)
                                                   EasingFunction
                                                                               SineEase
                                                                    =
                                                                        new
                                                                                                      this.Title
}
                                                   EasingMode = EasingMode.EaseInOut },
                           SetStartPage()
                                                   To
                                                                                                      _sideBarMenu[MenuItems.Review].Title;
private
               void
                                                   Duration = TimeSpan.FromSeconds(0.35)
                                                                                                      BackBtn.Visibility =
                                                                                                                              Visibility. Visible;
if
         (AuthUser
                                    null)
                                                                                                      BackChatBtn.Visibility
                                                                                                      Visibility.Collapsed;
ReviewFrame.Navigate(new
                                                   MFrame.BeginAnimation(OpacityProperty,
                                                                                                      MFrame.Content
ViewTasksPage());
                                                   anim):
                                                                                                      _sideBarMenu[MenuItems.Review];
return;
                                                     await
                                                                           Task.Delay(350);
                                                     MFrame.Content
                                                                                                      private void ProfileBtn_Click(object sender,
}
                                                                                       page;
switch
                                                                                                      RoutedEventArgs
                        (AuthUser.idPos)
                                                     await
                                                                             Task.Delay(25);
                                                                                                                                             e)
                                                     anim.To
{
                                                                                       1.0d:
                                                                         =
                                       1:
                                                   anim.Duration
                                                                                                      if (_sideBarMenu[MenuItems.Profile] ==
case
ReviewFrame.Navigate(new
                                                   TimeSpan.FromSeconds(0.35);
                                                                                                      null) _sideBarMenu[MenuItems.Profile] =
ViewTasksPage());
                                                   MFrame.BeginAnimation(OpacityProperty,
                                                                                                                         ProfilePage(AuthUser);
break;
                                                   anim);
                                                                                                      this.Title
                                                                                                                                    "Профиль";
                                       2:
                                                                                                      BackBtn.Visibility = Visibility.Collapsed;
                                                                                                      BackChatBtn.Visibility
ReviewFrame.Navigate(new
                                                   private void ChatFrameOnNavigated(object
ViewOrdersPage());
                                                   sender,
                                                                NavigationEventArgs
                                                                                                      Visibility.Collapsed;
ChatBtn.Visibility = Visibility.Collapsed;
                                                                                                      MFrame.Content
                                                   {
```

_sideBarMenu[MenuItems.Profile];	<page< th=""><th>Margin="0 0 5 0"></th></page<>	Margin="0 0 5 0">
}	x:Class="INCOMSYSTEM.Pages.AuthPag	<pre><controls:inputtextbox <="" pre="" x:name="LogBox"></controls:inputtextbox></pre>
private void ChatBtn_Click(object sender,	e"	Margin="0 5" PlaceHolder="example"/>
RoutedEventArgs e)	xmlns="http://schemas.microsoft.com/winf	<grid></grid>
{	x/2006/xaml/presentation"	<grid.columndefinitions></grid.columndefinitions>
if (_sideBarMenu[MenuItems.Chat] ==	xmlns:x="http://schemas.microsoft.com/wi	<columndefinition></columndefinition>
null)	nfx/2006/xaml"	<columndefinition width="Auto"></columndefinition>
{	xmlns:mc="http://schemas.openxmlformats	
ChatListPage = new ChatListPage();	.org/markup-compatibility/2006"	<controls:inputtextbox< td=""></controls:inputtextbox<>
_sideBarMenu[MenuItems.Chat] =	xmlns:d="http://schemas.microsoft.com/ex	x:Name="PassBox" Margin="0 5"
ChatListPage;	pression/blend/2008"	PlaceHolder="password"
}	xmlns;controls="clr-	IsPassword="True" Width="155"/>
this.Title =	namespace:INCOMSYSTEM.Controls"	<checkbox <="" grid.column="1" td=""></checkbox>
_sideBarMenu[MenuItems.Chat].Title;	mc:Ignorable="d"	IsChecked="{Binding
BackBtn.Visibility = Visibility.Collapsed;	Title="Окно авторизации">	ElementName=PassBox,
BackChatBtn. Visibility = Visibility. Visible;	<page.resources></page.resources>	Path=IsShowed}">
MFrame.Content =	<style <="" targettype="TextBlock" td=""><td><CheckBox.Style></td></tr><tr><td>_sideBarMenu[MenuItems.Chat];</td><td>BasedOn="{StaticResource</td><td><Style TargetType="CheckBox"></td></tr><tr><td>_sideBanvienu[ivienuneins.char], }</td><td>TextBlockStyle}"/></td><td><pre><Style TargetType= CheckBox > </pre> <pre><Setter Property="Cursor" Value="Hand"/></pre></td></tr><tr><td>public void GoChat(Chats chat)</td><td><pre><Style TargetType="Label"</pre></td><td><pre><Setter Property="IsTabStop"</pre></td></tr><tr><td>public volu Goenat(Chats chat)</td><td>BasedOn="{StaticResource LabelStyle}"/></td><td>Value="False"/></td></tr><tr><td>ChatBtn_Click(null, new</td><td><pre></pre></pre><pre></pre><pre></pre><pre></pre></pre><pre></pre><pre></pre></pre><pre></pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pre></pre><pr</td><td><pre><Setter Property="Template"></pre></td></tr><tr><td></td><td><Grid></td><td><Setter. Value></td></tr><tr><td>RoutedEventArgs()); var chatPage = ChatListPage as</td><td><Grid.ColumnDefinitions></td><td><ControlTemplate</td></tr><tr><td></td><td><pre><ColumnDefinition Width="Auto"/></pre></td><td>TargetType="CheckBox"></td></tr><tr><td>ChatListPage; chatPage.Chat = chat;</td><td><ColumnDefinition/></td><td><pre><Image x:Name="eyeImage" Width="24"</pre></td></tr><tr><td>chatPage.Chat = chat; chatPage.ChatEnter_LeftBtnUp(null, null);</td><td><ColumnDefinition Width="250"/></td><td>Height="24"/></td></tr><tr><td></td><td><ColumnDefinition Width="Auto"/></td><td><pre><ControlTemplate.Triggers></pre></td></tr><tr><td>}</td><td></ri></td><td></td></tr><tr><td>private void MainMenuBtn_Click(object sender, RoutedEventArgs e)</td><td><Grid Grid.Column="0"</td><td><pre><Trigger Property="IsChecked" Value="False"></pre></td></tr><tr><td>(KouledEventAigs</td><td>Grid.ColumnSpan="2"</td><td><pre><Setter TargetName="eyeImage"</pre></td></tr><tr><td>ReviewBtn_Click(sender, e);</td><td>Margin="15 0"</td><td>Property="Source"</td></tr><tr><td>ReviewBtn_Click(sender, e); ReviewFrame = new Frame();</td><td>HorizontalAlignment="Center"</td><td>Value="/Images/eyeOpen.png"/></td></tr><tr><td>ReviewFrame.Navigated +=</td><td>VerticalAlignment="Center"></td><td></Trigger></td></tr><tr><td>ReviewFrameOnNavigated;</td><td><pre><Grid.RowDefinitions></pre></td><td></td></tr><tr><td>· ·</td><td><RowDefinition/></td><td><Trigger Property="IsChecked" Value="True"></td></tr><tr><td>SetStartPage();</td><td></td><td></td></tr><tr><td>mivoto void PoolsChotPtm Click(skipst</td><td>· ·</td><td></td></tr><tr><td>private void BackChatBtn_Click(object</td><td>· ·</td><td>Property="Source" Valve=" / (propers/cyc/Cless pro="/c</td></tr><tr><td>sender, RoutedEventArgs e)</td><td><RowDefinition Height="Auto"/> </Grid.RowDefinitions></td><td>Value="/Images/eyeClose.png"/></td></tr><tr><td>{ ChatFrame.Navigate(ChatListPage);</td><td><Grid.ColumnDefinitions></td><td></Trigger> </ControlTemplate.Triggers></td></tr><tr><td></td><td><ColumnDefinition/></td><td></ControlTemplate></td></tr><tr><td>}</td><td><ColumnDefinition Width="150"/></td><td></Setter.Value></td></tr><tr><td>} internal enum MenuItems</td><td></ri></td><td></setter></td></tr><tr><td>internal enum MenuItems</td><td><StackPanel Grid.Row="0"</td><td></style>	
Review = 0 ,	Grid.Column="0">	·
Review = 0, Profile = 1,	Grid.Column= 0 > <textblock <="" td="" text="Введите логин:"><td> </td></textblock>	
Chat = 2	· ·	
}	<textblock <="" td="" text="Введите пароль:"><td></td></textblock>	
}	Margin="0 15 0 0"/>	<stackpanel <="" grid.row="1" td=""></stackpanel>
AuthDaga yaml		Grid Column="0"
AuthPage.xaml	<stackpanel <="" grid.row="0" td=""><td>Grid.ColumnSpan="2"</td></stackpanel>	Grid.ColumnSpan="2"
	Grid.Column="1"	Margin="0 20 0 5"

Orientation="Horizontal"	"ИНКОМСИСТЕМ".	return;
HorizontalAlignment="Center">	<paragraph>Научно-Инженерный Центр</paragraph>	}
<button <="" td="" x:name="AuthBtn"><td>«ИНКОМСИСТЕМ» является</td><td>var login = LogBox.Value;</td></button>	«ИНКОМСИСТЕМ» является	var login = LogBox.Value;
Content="Войти"	крупнейшим системным интегратором	var password = PassBox.Value;
IsEnabled="False"	Российской Федерации и стратегическим	using (var db = new
Margin="5 0"/>	партнёром нефтегазовых и	INCOMSYSTEMEntities())
<button <="" td="" x:name="RegBtn"><td>нефтехимических</td><td>{</td></button>	нефтехимических	{
Content="Зарегистрироваться"	компаний.	var user = db.UsersDetail.Include(s =>
Margin="5 0"/>		s.Positions)
		.Include(s => s.Employees)
<button <="" grid.row="2" td=""><td></td><td>.Include(s => s.Customers)</td></button>		.Include(s => s.Customers)
Grid.Column="0"		.Include(s => s.Customers.LegalForms)
Grid.ColumnSpan="2"		.FirstOrDefault(s => s.login == login &&
Style="{StaticResource LabelBtn}"		s.password == password);
Click="GuestBtn_Click"		if (user != null)
Content="Войти как гость"	AuthPage.xaml.cs	{
HorizontalAlignment="Center"/>	using System.Ling;	MainWindow.AuthUser = user;
<border <="" grid.row="3" td=""><td>using System. Windows;</td><td>var window = new MainWindow();</td></border>	using System. Windows;	var window = new MainWindow();
Grid.Column="0"	using System. Windows,	window.Show();
Grid.ColumnSpan="2"	using System. Windows. Controls,	Application.Current.Windows.OfType <aut< td=""></aut<>
x:Name="ErrorBorder"	using INCOMSYSTEM.Context;	hWindow>().First().Close();
Background="#560b0b"	using INCOMSYSTEM.Windows;	return;
BorderBrush="Red"	using System.Data.Entity;	}
BorderThickness="1"	namespace INCOMSYSTEM.Pages	; ShowError("Неверный логин или
Visibility="Collapsed"	{	пароль");
HorizontalAlignment="Center"	public partial class AuthPage : Page	
Padding="10 5"	f	1
I adding 10	1	}
· ·	public AuthPage()	private bool CheckEmpty()
Margin="0 10 0 15">	public AuthPage()	private bool CheckEmpty()
Margin="0 10 0 15"> <textblock <="" td="" x:name="ErrorBlock"><td>{</td><td>{</td></textblock>	{	{
Margin="0 10 0 15"> <textblock <="" foreground="White" td="" x:name="ErrorBlock"><td>{ InitializeComponent();</td><td>{ if (LogBox.IsWhiteSpace) return</td></textblock>	{ InitializeComponent();	{ if (LogBox.IsWhiteSpace) return
Margin="0 10 0 15"> <textblock <="" foreground="White" td="" verticalalignment="Center" x:name="ErrorBlock"><td>{ InitializeComponent(); LogBox.TextChanged += (s, e) =></td><td>{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть</td></textblock>	{ InitializeComponent(); LogBox.TextChanged += (s, e) =>	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть
Margin="0 10 0 15"> <textblock <="" foreground="White" td="" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"><td>{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty();</td><td>{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым");</td></textblock>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty();	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым");
Margin="0 10 0 15"> <textblock <="" foreground="White" horizontalalignment="Center" td="" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"><td>{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) =></td><td>{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return</td></textblock>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) =>	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty();	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key ==	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым");
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled)	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true;
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border <="" grid.column="2" td=""><td>{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); };</td><td>{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty;</td></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); };	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty;
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border <="" borderbrush="Coral" grid.column="2" td=""><td>{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick;</td><td>{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility =</td></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick;	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility =
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border <="" borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" td=""><td>{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick;</td><td>{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed;</td></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick;	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed;
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; }	if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed; return false;
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender,	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed; return false; }
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions></grid.rowdefinitions></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; }	if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed; return false;
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions> <rowdefinition></rowdefinition></grid.rowdefinitions></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) {	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed; return false; } private bool ShowError(string error) {
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions> <rowdefinition></rowdefinition> <rowdefinition></rowdefinition> </grid.rowdefinitions></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBorder.Visibility = triorBorder.Visibility = Visibility.Collapsed; return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; }
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions> <rowdefinition></rowdefinition> <rowdefinition></rowdefinition> </grid.rowdefinitions></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) {	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed; return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; }
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions> <rowdefinition></rowdefinition> </grid.rowdefinitions> <td>{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); }</td><td>{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed; return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBlock.Text = error;</td></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); }	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed; return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBlock.Text = error;
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions> <rowdefinition></rowdefinition> </grid.rowdefinitions> <image margin="5" source="/Images/authImage.png"/></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); } private void AuthBtnClick(object sender,	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBorder.Visibility = string.Empty; ErrorBorder.Visibility = return { AuthBtn.IsEnabled = false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBlock.Text = error; return true;
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions> <rowdefinition></rowdefinition> <rowdefinition></rowdefinition> </grid.rowdefinitions> <image margin="5" source="/Images/authImage.png"/> <richtextbox <="" isreadonly="True" td=""><td>{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); }</td><td>{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBorder.Visibility = string.Empty; ErrorBorder.Visibility = return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible;</td></richtextbox></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); }	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBorder.Visibility = string.Empty; ErrorBorder.Visibility = return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible;
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions> <rowdefinition></rowdefinition> </grid.rowdefinitions> <richtextbox <="" grid.row="1" isreadonly="True" td=""><td>{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); } private void AuthBtnClick(object sender, RoutedEventArgs e) {</td><td>{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed; return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Click(object sender,</td></richtextbox></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); } private void AuthBtnClick(object sender, RoutedEventArgs e) {	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBlock.Text = string.Empty; ErrorBorder.Visibility = Visibility.Collapsed; return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Click(object sender,
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions> <rowdefinition></rowdefinition> </grid.rowdefinitions> <image margin="5" source="/Images/authImage.png"/> <richtextbox grid.row="1" horizontalalignment="Center" isreadonly="True"></richtextbox></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); } private void AuthBtnClick(object sender, RoutedEventArgs e) { if(CheckEmpty())	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBorder.Visibility = string.Empty; ErrorBorder.Visibility = visibility.Collapsed; return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBlock.Text = error; return true; } private void GuestBtn_Click(object sender, RoutedEventArgs e)
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions> <rowdefinition></rowdefinition> <rowdefinition></rowdefinition> </grid.rowdefinitions> <image margin="5" source="/Images/authImage.png"/> <richtextbox grid.row="1" horizontalalignment="Center" isreadonly="True"> <flowdocument< td=""><td>{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled)} AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); } private void AuthBtnClick(object sender, RoutedEventArgs e) { if(CheckEmpty()) {</td><td>{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBorder.Visibility = string.Empty; ErrorBorder.Visibility = return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible;</td></flowdocument<></richtextbox></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled)} AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); } private void AuthBtnClick(object sender, RoutedEventArgs e) { if(CheckEmpty()) {	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBorder.Visibility = string.Empty; ErrorBorder.Visibility = return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible;
Margin="0 10 0 15"> <textblock fontsize="14" foreground="White" horizontalalignment="Center" textalignment="Center" verticalalignment="Center" x:name="ErrorBlock"></textblock> <border borderbrush="Coral" borderthickness="3 0 0 0" grid.column="2" margin="5 30"> <grid verticalalignment="Center"> <grid.rowdefinitions> <rowdefinition></rowdefinition> </grid.rowdefinitions> <image margin="5" source="/Images/authImage.png"/> <richtextbox grid.row="1" horizontalalignment="Center" isreadonly="True"></richtextbox></grid></border>	{ InitializeComponent(); LogBox.TextChanged += (s, e) => CheckEmpty(); PassBox.TextChanged += (s, e) => CheckEmpty(); this.KeyDown += (s, e) => { if (e.Key == Key.Enter && AuthBtn.IsEnabled) AuthBtnClick(s, e); }; AuthBtn.Click += AuthBtnClick; RegBtn.Click += RegBtnClick; } private void RegBtnClick(object sender, RoutedEventArgs e) { AuthWindow.AuthFrame.Navigate(new RegPage()); } private void AuthBtnClick(object sender, RoutedEventArgs e) { if(CheckEmpty())	{ if (LogBox.IsWhiteSpace) return ShowError("Логин не может быть пустым"); if (PassBox.IsWhiteSpace) return ShowError("Пароль не может быть пустым"); AuthBtn.IsEnabled = true; ErrorBorder.Visibility = string.Empty; ErrorBorder.Visibility = visibility.Collapsed; return false; } private bool ShowError(string error) { AuthBtn.IsEnabled = false; ErrorBorder.Visibility = Visibility.Visible; ErrorBorder.Visibility = Visibility.Visible; ErrorBlock.Text = error; return true; } private void GuestBtn_Click(object sender, RoutedEventArgs e)

Application.Current.Windows.OfType <aut< th=""><th><columndefinition width="200"></columndefinition></th><th></th></aut<>	<columndefinition width="200"></columndefinition>	
hWindow>().First().Close();	<columndefinition width="40"></columndefinition>	<pre><stackpanel <="" orientation="Horizontal" pre=""></stackpanel></pre>
}		Margin="5 0 0 0">
}	<grid.rowdefinitions></grid.rowdefinitions>	<textblock fontsize="16" text="No: "></textblock>
}	<rowdefinition></rowdefinition>	<pre><controls:inputtextbox <="" maxlength="6" pre=""></controls:inputtextbox></pre>
	<rowdefinition height="Auto"></rowdefinition>	Width="65" PlaceHolder="Номер"
RegPage.xaml	<rowdefinition height="Auto"></rowdefinition>	x:Name="InpPassportNumber"/>
<page< td=""><td></td><td></td></page<>		
x:Class="INCOMSYSTEM.Pages.RegPage	<stackpanel <="" grid.column="0" td=""><td></td></stackpanel>	
"	Grid.Row="0">	<pre><controls:inputtextbox< pre=""></controls:inputtextbox<></pre>
xmlns="http://schemas.microsoft.com/winf	<textblock <="" td="" x:name="NameBlock"><td>x:Name="InpAddress" MaxLength="175"</td></textblock>	x:Name="InpAddress" MaxLength="175"
x/2006/xaml/presentation"	Text="Название: " Margin="0"/>	PlaceHolder="Адрес"/>
xmlns:x="http://schemas.microsoft.com/wi	<textblock <="" td="" text="Правовая форма: "><td><pre><controls:inputtextbox< pre=""></controls:inputtextbox<></pre></td></textblock>	<pre><controls:inputtextbox< pre=""></controls:inputtextbox<></pre>
nfx/2006/xaml"	Height="40" Width="110"/>	x:Name="InpLogin" MaxLength="30"
xmlns:mc="http://schemas.openxmlformats	<textblock text="Телефон: "></textblock>	PlaceHolder="Логин"/>
.org/markup-compatibility/2006"	<textblock <="" td="" text="Паспортные данные: "><td><grid></grid></td></textblock>	<grid></grid>
xmlns:d="http://schemas.microsoft.com/ex	Height="40" Width="110"/>	<grid.columndefinitions></grid.columndefinitions>
pression/blend/2008"	<textblock text="Адрес: "></textblock>	<columndefinition></columndefinition>
xmlns:controls="clr-	<textblock text="Логин: "></textblock>	<pre><columndefinition width="Auto"></columndefinition></pre>
namespace:INCOMSYSTEM.Controls"	<textblock text="Пароль: "></textblock>	
mc:Ignorable="d"		<pre><controls:inputtextbox< pre=""></controls:inputtextbox<></pre>
Title="Регистрация">	<stackpanel <="" grid.column="1" td=""><td>x:Name="InpPassword"</td></stackpanel>	x:Name="InpPassword"
<page.resources></page.resources>	Grid.Row="0"	IsPassword="True" MaxLength="50"
<pre><style <="" pre="" targettype="TextBlock"></td><td>HorizontalAlignment="Left"></td><td>Width="155" PlaceHolder="Пароль"/></td></tr><tr><td>BasedOn="{StaticResource</td><td><pre><controls:InputTextBox</pre></td><td><pre></pre> <pre> </pre> <pre> <pre> <pre> CheckBox</td></tr><tr><td>TextBlockStyle}"></td><td>x:Name="InpName"</td><td>ElementName=InpPassword,</td></tr><tr><td>• •</td><td>•</td><td>Path=IsShowed}"</td></tr><tr><td><pre><Setter Property="Height" Value="30"/> <Setter Property="FortSize" Value="18"/></pre></td><td>MaxLength="150"</td><td>Grid.Column="1"></td></tr><tr><td><pre><Setter Property="FontSize" Value="18"/> </pre></td><td>Margin="0" PlaceHolder="Название\ФИО"/></td><td><CheckBox.Style></td></tr><tr><td><pre><Setter Property="TextWrapping" Value="Wrap"/></pre></td><td>1</td><td>•</td></tr><tr><td>Value="Wrap"/></td><td>8</td><td><pre><Style TargetType="CheckBox"></pre></td></tr><tr><td><Setter Property="Margin" Value="0 5 0 0"/></td><td>Selection Changed — "Crab Legal On Selection</td><td><Setter Property="Cursor" Value="Hand"/> <Setter Property="IsTabStop"</td></tr><tr><td></td><td>SelectionChanged="CmbLegal_OnSelectio</td><td><Setter Property="IsTabStop" Value="False"/></td></tr><tr><td></style></pre>	nChanged">	
<style< td=""><td><combobox.itemtemplate></combobox.itemtemplate></td><td><setter property="Template"></setter></td></style<>	<combobox.itemtemplate></combobox.itemtemplate>	<setter property="Template"></setter>
TargetType="controls:InputTextBox"	<pre><datatemplate></datatemplate></pre>	<setter.value></setter.value>
BasedOn="{StaticResource {x:Type	<textblock <="" td="" text="{Binding name}"><td><controltemplate< td=""></controltemplate<></td></textblock>	<controltemplate< td=""></controltemplate<>
controls:InputTextBox}}">	Style="{StaticResource	TargetType="CheckBox">
<pre><setter property="Height" value="30"></setter></pre>	ComboBoxTextBlockStyle}"/>	<pre><image <="" pre="" width="24" x:name="eyeImage"/></pre>
<pre><setter 24"="" property="Margin" value="0 5 0</pre></td><td></br></td><td>Height="></setter></pre>		
0"/>		<controltemplate.triggers></controltemplate.triggers>
		<trigger <="" property="IsChecked" td=""></trigger>
<style 10="(G) to Part 10 = " <="" combobox"="" part="" targettype="ComboBox" td=""><td><pre><controls:InputTextBox</pre></td><td>Value="False"></td></tr><tr><td>BasedOn="{StaticResource {x:Type</td><td>x:Name="InpPhone" Margin="0 0 0 0"</td><td><Setter TargetName="eyeImage"</td></tr><tr><td>ComboBox}}"></td><td>PlaceHolder="+7(900)-000-00-00"/></td><td>Property="Source"</td></tr><tr><td><pre><Setter Property="Height" Value="30"/></pre></td><td><StackPanel Height="40"</td><td>Value="/Images/eyeOpen.png"/></td></tr><tr><td><pre><Setter Property="Margin" Value="0 5 0</pre></td><td>VerticalAlignment="Center" Margin="0 10</td><td>Trigger></td></tr><tr><td>0"/></td><td>0 0" Orientation="Horizontal"></td><td><Trigger Property="IsChecked"</td></tr><tr><td></style>	<stackpanel orientation="Horizontal"></stackpanel>	Value="True">
	<textblock <="" td="" text="Серия: "><td><setter <="" targetname="eyeImage" td=""></setter></td></textblock>	<setter <="" targetname="eyeImage" td=""></setter>
<grid <="" horizontalalignment="Center" td=""><td>FontSize="16"/></td><td>Property="Source"</td></grid>	FontSize="16"/>	Property="Source"
VerticalAlignment="Center">	<pre><controls:inputtextbox <="" maxlength="4" pre=""></controls:inputtextbox></pre>	Value="/Images/eyeClose.png"/>
<grid.columndefinitions></grid.columndefinitions>	Width="55" PlaceHolder="Серия"	
<columndefinition width="110"></columndefinition>	x:Name="InpPassportSerie"/>	

```
</ControlTemplate>
                                                 TextAlignment="Center"
                                                                                                   ShowError("Длина
                                                                                                                         пароля
                                                                                                                                   должна
</Setter.Value>
                                                 HorizontalAlignment="Center"
                                                                                                   достигать
                                                                                                               минимум
                                                                                                                           6
                                                                                                                               символов");
</Setter>
                                                 FontSize="14"/>
                                                                                                   return;
</Style>
                                                 </Border>
                                                 </Grid>
                                                                                                   if
</CheckBox.Style>
</CheckBox>
                                                 </Page>
                                                                                                   (PasswordDifficulty.CheckDifficultyPassw
</Grid>
                                                                                                   ord(InpPassword.Value)
</StackPanel>
                                                 RegPage.xaml.cs
                                                                                                   DifficultyPassword.Easy)
<Button
            VerticalAlignment="Bottom"
                                                 using
                                                                                  System;
Grid.Column="2"
                                                 using
                                                                             System.Linq;
                                                                                                   ShowError("Пароль не может
                                                                                                                                     быть
Click="GenerateNewPasswordBtn_Click"
                                                                         System.Windows;
                                                 using
                                                                                                   слабым");
Margin="5
                          0
                                                 using
                                                                System.Windows.Controls;
                                                                                                   return;
                                                 using
<Button.ToolTip>
                                                                   System.Windows.Input;
                                                          INCOMSYSTEM.BehaviorsFiles;
<ToolTip
                                                 using
                                                                                                   using
                                                                                                              (var
                                                                                                                        db
                                                                                                                                       new
ToolTipService.InitialShowDelay="1000"
                                                                 INCOMSYSTEM.Context;
                                                                                                   INCOMSYSTEMEntities())
                                                 using
Height="35">
                                                               INCOMSYSTEM.Windows;
                                                 using
<TextBlock Text="Сгенерировать новый
                                                                   INCOMSYSTEM.Pages
                                                                                                   long?
                                                                                                                                      null:
                                                 namespace
                                                                                                                phone
                        FontSize="16"/>
                                                                                                   if
пароль"
                                                                                                                   (!InpPhone.IsWhiteSpace)
</ToolTip>
                                                 public partial class RegPage : Page
</Button.ToolTip>
                                                                                                   try { phone = long.Parse(InpPhone.Value); }
<Button.Content>
                                                 public
                                                                               RegPage()
                                                                                                   catch
                                                                                                                          return;
                                                                                                                 {
<Image Source="../Images/doorKey.png"/>
</Button.Content>
                                                 InitializeComponent();
                                                                                                   var passportStr = InpPassportSerie.Value +
</Button>
                                                                      db
                                                                                                   InpPassportNumber.Value;
                                                 using
                                                            (var
                                                                                     new
<StackPanel
                          Grid.Row="1"
                                                 INCOMSYSTEMEntities())
                                                                                                   long.TryParse(passportStr,
                                                                                                                                        var
Grid.Column="0"
                                                                                                   passport);
Grid.ColumnSpan="2"
                                                 CmbLegal.ItemsSource
                                                                                                      (db.UsersDetail.FirstOrDefault(s
                                     5"
Margin="0
                            0
                                                 db.LegalForms.ToList();
                                                                                                   s.login.ToLower()
Orientation="Horizontal"
                                                                                                   InpLogin.Value.ToLower().Trim()) != null)
HorizontalAlignment="Center">
                                                 this.KeyDown += (s, e) => { if (e.Key ==
<Button
                      x:Name="RegBtn"
                                                 Key.Enter
                                                                &&
                                                                        RegBtn.IsEnabled)
                                                                                                   ShowError("Такой логин уже занят");
Content="Зарегистрироваться"
                                                 RegBtn_Click(s,
                                                                                                   return:
                                                                          e):
                                                                                       }:
IsEnabled="False"
                                                 InpName.TextChanged +=
                                                                             (s,
                                                                                  e)
                                                                                      =>
Margin="5
                                     0"
                                                 CheckEmpty();
                                                                                                   if
                                                                                                          (phone
                                                                                                                      !=
                                                                                                                              null
                                                                                                                                       &&
Click="RegBtn_Click"/>
                                                 InpPhone.TextChanged += (s, e) =>
                                                                                                   db.UsersDetail.FirstOrDefault(s => s.phone
<Button
                       Content="Назад"
                                                 CheckEmpty();
                                                                                                              phone)
                                                                                                                                      null)
Margin="5
                                     0"
                                                 InpPassportSerie.TextChanged += (s, e) =>
Click="BackBtn_Click"/>
                                                 CheckEmpty();
                                                                                                   ShowError("Номер телефона уже занят");
                                                 InpPassportNumber.TextChanged += (s, e)
</StackPanel>
                                                                                                   return;
<Border
                          Grid.Row="2"
                                                                           CheckEmpty();
Grid.Column="0"
                                                 InpAddress.TextChanged += (s, e) =>
                                                                                                       (db.UsersDetail.FirstOrDefault(s
Grid.ColumnSpan="3"
                                                 CheckEmpty();
                                                                                                   s.passport
                                                                                                                                      null)
                                                                                                                     passport)
x:Name="ErrorBorder"
                                                 InpLogin.TextChanged += (s, e) =>
Background="#560b0b"
                                                                                                   ShowError("Паспорт
                                                 CheckEmpty();
                                                                                                                           уже
                                                                                                                                   занят"):
BorderBrush="Red"
                                                 InpPassword.TextChanged += (s, e) =>
                                                                                                   return:
BorderThickness="1"
                                                 CheckEmpty();
Visibility="Collapsed"
                                                                                                         userDetail
                                                                                                   var
                                                                                                                                UsersDetail
HorizontalAlignment="Center"
                                                 private void RegBtn_Click(object sender,
Padding="10
                                     5"
                                                 RoutedEventArgs
                                                                                                                            InpLogin.Value,
                                                                                                   login
Margin="0
                 10
                          0
                                   15">
                                                 {
                                                                                                   password
                                                                                                                         InpPassword.Value,
<TextBlock
                   x:Name="ErrorBlock"
                                                 if
                                                            (CheckEmpty())
                                                                                                   phone
                                                                                                                                     phone,
                                                                                   return:
Foreground="White"
                                                 if
                                                      (InpPassword.Value.Length
                                                                                                   idPos
                                                                                                                                         1,
                                                                                       6)
VerticalAlignment="Center"
                                                 {
                                                                                                   passport
                                                                                                                                   passport,
```

```
address
                       InpAddress.Value,
                                                  out )) return ShowError("Номер паспорта
                                                                                                    nfx/2006/xaml"
dateStart
                          DateTime.Now
                                                       может
                                                                 содержать
                                                                               символы");
                                                                                                    xmlns:mc="http://schemas.openxmlformats
};
                                                  if (InpPassportNumber.Value.Length != 6)
                                                                                                    .org/markup-compatibility/2006"
db.UsersDetail.Add(userDetail);
                                                           ShowError("Номер
                                                                                                    xmlns:d="http://schemas.microsoft.com/ex
                                                  return
                                                                                 паспорта
                                                                                                    pression/blend/2008"
      customer
                 =
                      new
                              Customers
                                                             содержать
                                                                                 цифры");
var
                                                  лолжен
                                                                                                    xmlns:gl="clr-
                                                       (InpAddress.IsWhiteSpace)
                                                                                    return
{
                                                  if
                                                  ShowError("Адрес
                                                                                                    namespace:System.Globalization;assembly
name
                         InpName.Value,
                                                                      не
                                                                           может
                                                                                     быть
idLegalForm
                                                  пустым");
                                                                                                    =mscorlib'
((LegalForms)CmbLegal.SelectedItem).id,
                                                        (InpLogin.IsWhiteSpace)
                                                                                    return
                                                                                                    xmlns:wnd="clr-
                                                                                                    namespace:INCOMSYSTEM.Windows"
};
                                                  ShowError("Логин
                                                                      не
                                                                                     быть
                                                                            может
db.Customers.Add(customer);
                                                  пустым");
                                                                                                    mc:Ignorable="d">
db.SaveChanges();
                                                      (InpPassword.IsWhiteSpace)
                                                                                                    <Page.Resources>
MessageBox.Show("Вы
                               успешно
                                                  ShowError("Пароль не
                                                                                     быть
                                                                                                    <Style TargetType="{x:Type TextBlock}"
зарегистрировались");
                                                  пустым");
                                                                                                    BasedOn="{StaticResource
MainWindow.AuthUser
                                                                                                    TextBlockStyle}"
                              userDetail:
                                                  RegBtn.IsEnabled
                                                                                     true:
                                                  ErrorBlock.Text
                 MainWindow().Show();
                                                                             string.Empty;
                                                                                                    </Page.Resources>
Application.Current.Windows.OfType<Aut
                                                  ErrorBorder. Visibility
                                                                                                    <Grid
                                                                                                              HorizontalAlignment="Center"
hWindow>().First().Close();
                                                  Visibility. Collapsed;
                                                                                                    VerticalAlignment="Center">
                                                                                                    <Grid.ColumnDefinitions>
                                                  return
                                                                                    false:
                                                                                                    <ColumnDefinition
                                                                                                                         Width="Auto"
}
                                                                                                                                         />
private void BackBtn_Click(object sender,
                                                                 ShowError(string
                                                                                                    <ColumnDefinition
                                                                                                                                          />
                                                  private
                                                          bool
                                                                                    error)
RoutedEventArgs
                                                                                                    </Grid.ColumnDefinitions>
                                                  RegBtn.IsEnabled
                                                                                     false;
                                                                                                    <Border
                                                                                                                       BorderBrush="White"
AuthWindow.AuthFrame.Navigate(new
                                                  ErrorBorder.Visibility = Visibility.Visible;
                                                                                                    BorderThickness="0
                                                                                                                            0
                                                                                                                                         0"
AuthPage());
                                                  ErrorBlock.Text
                                                                                    error:
                                                                                                    Margin="5
                                                                                                                                          />
                                                  return
                                                                                     true;
                                                                                                    <StackPanel
                                                                                                                      Margin="10
                                                                                                                                        0">
private
              bool
                          CheckEmpty()
                                                                                                    <Button
                                                                                                                      Style="{StaticResource
{
                                                  private
                                                                                      void
                                                                                                    SideBarButton \}">
      (InpName.IsWhiteSpace)
                                                  GenerateNewPasswordBtn_Click(object
                                                                                                    <StackPanel
                                                                                                                  Orientation="Horizontal">
                                  return
                                                                                                                                 Width="32"
ShowError("Название/фио не может быть
                                                  sender.
                                                                RoutedEventArgs
                                                                                                    <Image
                                                                                        e)
                                                                                                    Height="32"
пустым");
if
       (!InpPhone.IsWhiteSpace
                                                  InpPassword.IsShowed
                                                                                                    Source="../Images/profile.png"
                                    &&
                                                                                     true:
long.TryParse(InpPhone.Value,
                                                  InpPassword.Text
                                                                                                    <TextBlock VerticalAlignment="Center"
                               out
                                     ))
return ShowError("Номер телефона не
                                                  PasswordGenerator.GetNewPassword();
                                                                                                    Text="Основное"
                             символы");
                                                                                                    Margin="5
может
            содержать
       (!InpPhone.IsWhiteSpace
                                                  private
                                                                                      void
                                                                                                    </StackPanel>
InpPhone.Value.Length != 11) return
                                                  CmbLegal_OnSelectionChanged(object
                                                                                                    </Button>
ShowError("Номер
                    телефона
                                                          SelectionChangedEventArgs
                                                                                                    </StackPanel>
содержать
                    11
                                цифр");
                                                                                                    <Grid
                                                                                                                          Grid.Column="1">
if (InpPassportSerie.IsWhiteSpace) return
                                                  NameBlock.Text
                                                                                                    <Grid.RowDefinitions>
ShowError("Серия паспорта не может
                                                  ((LegalForms)CmbLegal.SelectedItem).id\\
                                                                                                    <RowDefinition
                                                                                                                       Height="Auto"
                                                                "ФИО"
быть
                              пустым");
                                                                              "Название";
                                                                                                    <RowDefinition
                                                                                                                       Height="200"
                                                                                                                                         />
if (!long.TryParse(InpPassportSerie.Value,
                                                                                                    </Grid.RowDefinitions>
out )) return ShowError("Серия паспорта
                                                  }
                                                                                                    <Grid
                                                                                                             Margin="0
                                                                                                                                       15">
                                                                                                    <Grid.ColumnDefinitions>
               содержать
                             символы");
                                                  }
     может
if (InpPassportSerie.Value.Length != 4)
                                                                                                    <ColumnDefinition
                                                                                                                                          />
return ShowError("Серия паспорта должен
                                                  ProfilePage.xaml
                                                                                                    <ColumnDefinition
                                                                                                                                          />
                                                                                                    </Grid.ColumnDefinitions>
содержать
                               цифры");
if
       (InpPassportNumber.IsWhiteSpace)
                                                  x:Class="INCOMSYSTEM.Pages.ProfileP
                                                                                                    <StackPanel>
return
      ShowError("Номер паспорта не
                                                                                                    <TextBlock
                                                                                                                 x:Name="NameBlock"
               быть
                              пустым");
                                                  xmlns="http://schemas.microsoft.com/winf
                                                                                                    <TextBlock x:Name="PassportBlock"
                                                  x/2006/xaml/presentation"
                                                                                                    <TextBlock x:Name="PositionBlock"
if
(!long. Try Parse (Inp Passport Number. Value,\\
                                                  xmlns:x="http://schemas.microsoft.com/wi
                                                                                                    <TextBlock
                                                                                                                x:Name="PhoneBlock"
```

<textblock< th=""><th>Value="White" /></th><th></th></textblock<>	Value="White" />			
x:Name="CustomerLegalFormBlock" />	<setter <="" property="TextAlignment" td=""><td><border grid.column="2"></border></td></setter>	<border grid.column="2"></border>		
<textblock x:name="AddressBlock"></textblock>	Value="Center" />	<textblock></textblock>		
	<setter <="" property="VerticalAlignment" td=""><td><textblock.style></textblock.style></td></setter>	<textblock.style></textblock.style>		
<border <="" borderbrush="White" td=""><td>Value="Center" /></td><td><style <="" targettype="TextBlock" td=""></tr><tr><td>BorderThickness="0 0 5 0" /></td><td><Setter Property="Background"</td><td>BasedOn="{StaticResource {x:Type</td></tr><tr><td><StackPanel Grid.Column="1"</td><td>Value="DimGray" /></td><td>TextBlock}}"></td></tr><tr><td>Margin="5 0"></td><td><pre><Setter Property="FontWeight"</pre></td><td><Setter Property="Text"</td></tr><tr><td><TextBlock x:Name="DateStart" /></td><td>Value="Bold" /></td><td>Value="Менеджер"/></td></tr><tr><td><TextBlock</td><td></style></td><td><style.triggers></style.triggers></td></border>	Value="Center" />	<style <="" targettype="TextBlock" td=""></tr><tr><td>BorderThickness="0 0 5 0" /></td><td><Setter Property="Background"</td><td>BasedOn="{StaticResource {x:Type</td></tr><tr><td><StackPanel Grid.Column="1"</td><td>Value="DimGray" /></td><td>TextBlock}}"></td></tr><tr><td>Margin="5 0"></td><td><pre><Setter Property="FontWeight"</pre></td><td><Setter Property="Text"</td></tr><tr><td><TextBlock x:Name="DateStart" /></td><td>Value="Bold" /></td><td>Value="Менеджер"/></td></tr><tr><td><TextBlock</td><td></style>	<style.triggers></style.triggers>	
Text="{Binding ElementName=OrdersList,	<style targettype="Border"></td><td><DataTrigger</td></tr><tr><td>Path=Items.Count, StringFormat='{}Bcero</td><td><Setter Property="Background"</td><td>Binding="{Binding Source={x:Static</td></tr><tr><td>заказов: {0}'}" /></td><td>Value="DimGray" /></td><td>wnd:MainWindow.AuthUser},</td></tr><tr><td><TextBlock x:Name="PopularOrderBlock"</td><td><Setter Property="BorderBrush"</td><td>Path=idPos}"</td></tr><tr><td>/></td><td>Value="Black" /></td><td>Value="3"></td></tr><tr><td><TextBlock</td><td><Setter Property="BorderThickness"</td><td><Setter Property="Text"</td></tr><tr><td>x:Name="MostExpensiveBlock" /></td><td>Value="1" /></td><td>Value="Исполнитель" /></td></tr><tr><td></ri> <//stackPanel></td><td></sli></td><td></DataTrigger></td></tr><tr><td></Grid></td><td></Grid.Resources></td><td></br> </ri> </style.Triggers></td></tr><tr><td><ScrollViewer Grid.Row="1"</td><td><Grid.ColumnDefinitions></td><td></td></tr><tr><td>Grid.ColumnSpan="2"</td><td></td><td></style>			
VerticalScrollBarVisibility=" Disabled "	<columndefinition width="100"></columndefinition> <columndefinition width="200"></columndefinition>			
·				
HorizontalScrollBarVisibility="Auto"> <grid></grid>		<border grid.column="3"></border>		
<grid.rowdefinitions></grid.rowdefinitions>		<textblock text="Задача"></textblock>		
		STEXTIFICATE SAGARDA (Border)		
Č	<columndefinition width="100"></columndefinition> <columndefinition width="225"></columndefinition>	<border grid.column="4"></border>		
<rowdefinition></rowdefinition>	<columndefinition width="100"></columndefinition>	<textblock text="Цена"></textblock>		
		STEXT DIOCK TEXT LIERA //> //> //		
<button <="" click="RefreshOrdersBtn_Click" p=""> Horizontal Alianment = "Loft"</button>	<border></border>			
HorizontalAlignment=" Left "> <button.content></button.content>		<border grid.column="5"> <textblock text="Сложность"></textblock></border>		
	<textblock text="№Заказа"></textblock>			
<grid as<="" calcum="" definition="" td=""><td></td><td></td></grid>				
<grid.columndefinitions></grid.columndefinitions>	<border grid.column="1"></border>	<border grid.column="6"></border>		
<columndefinition width="32"></columndefinition>	<textblock></textblock>	<textblock auto"="" text="Дата формирования</td></tr><tr><td><ColumnDefinition Width="></textblock>	<textblock.style></textblock.style>	заказа" />
	<style <="" targettype="TextBlock" td=""><td></Border></td></tr><tr><td><Image Source="/Images/refresh.png" /></td><td>BasedOn="{StaticResource {x:Type</td><td><Border Grid.Column="7"></td></tr><tr><td><TextBlock Text="Обновить"</td><td>TextBlock}}"></td><td><TextBlock Text="Crarryc" /></td></tr><tr><td>Grid.Column="1" /></td><td><Setter Property="Text"</td><td></Border></td></tr><tr><td></Grid></td><td>Value="Заказчик"/></td><td></ri></td></tr><tr><td></Button.Content></td><td><Style.Triggers></td><td><pre><ItemsControl x:Name="OrdersList"</pre></td></tr><tr><td></Button></td><td><DataTrigger</td><td>Grid.Row="1"</td></tr><tr><td><Grid Grid.Row="1"></td><td>Binding="{Binding Source={x:Static</td><td>Style="{StaticResource ItemsVisible}"></td></tr><tr><td><Grid.RowDefinitions></td><td>wnd:MainWindow.AuthUser},</td><td><ItemsControl.ItemTemplate></td></tr><tr><td><RowDefinition Height="Auto" /></td><td>Path=idPos}"</td><td><DataTemplate></td></tr><tr><td><RowDefinition /></td><td>Value="1"></td><td><CheckBox x:Name="OrderGrid"</td></tr><tr><td></Grid.RowDefinitions></td><td><Setter Property="Text"</td><td>Margin="0 0 0 10"</td></tr><tr><td><Grid></td><td>Value="Исполнитель" /></td><td>Style="{StaticResource</td></tr><tr><td><Grid.Resources></td><td></DataTrigger></td><td>ClearCheckBoxTemplate}"></td></tr><tr><td><Style TargetType="TextBlock"</td><td></Style.Triggers></td><td><CheckBox.Content></td></tr><tr><td>BasedOn="{StaticResource</td><td></style>	<grid></grid>		
ComboBoxTextBlockStyle}">		<grid.resources></grid.resources>		
<setter <="" property="Foreground" td=""><td></td><td><style <="" targettype="TextBlock" td=""></tr></tbody></table></style></td></setter>		<style <="" targettype="TextBlock" td=""></tr></tbody></table></style>		

BasedOn="{StaticResource	<setter property="Text" th="" value="{Binding</th><th><TextBlock Text=" {binding<=""></setter>			
ComboBoxTextBlockStyle}">	Employees,	Statuses.name}" />		
<setter <="" property="Foreground" td=""><td>TargetNullValue='Oтсутствует'}" /></td><td></td></setter>	TargetNullValue='Oтсутствует'}" />			
Value="White" />		<uniformgrid <="" grid.row="1" td=""></uniformgrid>		
<setter <="" property="TextAlignment" td=""><td></td><td>Grid.ColumnSpan="9"</td></setter>		Grid.ColumnSpan="9"		
Value="Center" />		Grid.Column="0"		
<setter <="" property="VerticalAlignment" td=""><td></td><td>Columns="4"</td></setter>		Columns="4"		
Value="Center" />		Height="80">		
<setter <="" property="Background" td=""><td></td><td><uniformgrid.style></uniformgrid.style></td></setter>		<uniformgrid.style></uniformgrid.style>		
Value="DimGray" />	<border grid.column="2"></border>	<style targettype="UniformGrid"></td></tr><tr><td></style>	<textblock></textblock>	<setter <="" property="Visibility" td=""></setter>
<style targettype="Border"></td><td><TextBlock.Style></td><td>Value="Collapsed"/></td></tr><tr><td><Setter Property="Background"</td><td><Style TargetType="TextBlock"</td><td><Style.Triggers></td></tr><tr><td>Value="DimGray" /></td><td>BasedOn="{StaticResource {x:Type</td><td><DataTrigger Binding="{Binding</td></tr><tr><td><Setter Property="BorderBrush"</td><td>TextBlock}}"></td><td>ElementName=OrderGrid,</td></tr><tr><td>Value="Black" /></td><td><Setter Property="Text" Value="{Binding</td><td>Path=IsChecked}" Value="True"></td></tr><tr><td><Setter Property="BorderThickness"</td><td>Chats.Employees,</td><td><Setter Property="Visibility"</td></tr><tr><td>Value="1" /></td><td>TargetNullValue='Отсутствует'} "/></td><td>Value="Visible"/></td></tr><tr><td><Setter Property="Padding" Value="5" /></td><td><Style.Triggers></td><td></DataTrigger></td></tr><tr><td></style>	<datatrigger< td=""><td></td></datatrigger<>			
	Binding="{Binding Source={x:Static			
<grid.rowdefinitions></grid.rowdefinitions>	wnd:MainWindow.AuthUser},			
<rowdefinition height="40"></rowdefinition>	Path=idPos}"	<border></border>		
<rowdefinition height="Auto"></rowdefinition>	Value="3">	<textblock <="" td="" text="Плановая дата начала"></textblock>		
	<setter bold"="" property="Text" value="{Binding</td><td>FontWeight="></setter>			
<grid.columndefinitions></grid.columndefinitions>	Employees,			
<columndefinition width="100"></columndefinition>	TargetNullValue='Отсутствует'}" />	<border></border>		
<columndefinition width="200"></columndefinition>		<textblock 200"="" text="Фактическая дата</td></tr><tr><td><ColumnDefinition Width="></textblock>		начала" FontWeight="Bold" />
<columndefinition width="150"></columndefinition>				
<columndefinition width="100"></columndefinition>		<border></border>		
<columndefinition width="100"></columndefinition>		<textblock 225"="" техt="Плановая дата</td></tr><tr><td><ColumnDefinition Width="></textblock>		завершения" FontWeight="Bold" />
<columndefinition width="100"></columndefinition>	<border grid.column="3"></border>			
	<textblock <="" td="" text="{Binding Tasks.name}"><td><border></border></td></textblock>	<border></border>		
<border></border>	/>	<textblock id}"="" text="Фактическая дата</td></tr><tr><td><TextBlock Text=" {binding=""></textblock>		завершения" FontWeight="Bold" />
	<border grid.column="4"></border>			
<border grid.column="1"></border>	<textblock plandatestart,<="" td="" text="{Binding price,</td><td><Border></td></tr><tr><td><TextBlock></td><td>StringFormat='{}{0:0,0} py6.',</td><td><TextBlock</td></tr><tr><td><TextBlock.Style></td><td>ConverterCulture={x:Static</td><td>Text=" {binding=""></textblock>			
<style <="" targettype="TextBlock" td=""><td>gl:CultureInfo.CurrentCulture}}" /></td><td>StringFormat='{}{0:dd MMMM yyyy} r.',</td></tr><tr><td>BasedOn="{StaticResource {x:Type</td><td></Border></td><td>ConverterCulture={x:Static</td></tr><tr><td>TextBlock}}"></td><td><Border Grid.Column="5"></td><td>gl:CultureInfo.CurrentCulture},</td></tr><tr><td><pre><Setter Property="Text" Value="{Binding</pre></td><td><TextBlock Text="{Binding difficulty}" /></td><td>TargetNullValue='OrcyrcrByer'}" /></td></tr><tr><td>Customers.name,</td><td></br> </ra> </Border></td><td></br> </Border></td></tr><tr><td>TargetNullValue='Oтсутствует'}"/></td><td><Border Grid.Column="6"></td><td><Border></td></tr><tr><td><pre><Style.Triggers></pre></td><td><TextBlock Text="{Binding dateOrder,</td><td><pre><TextBlock Text="{Binding factDateStart,</pre></td></tr><tr><td><Style. Triggers > <DataTrigger</td><td>StringFormat='{}{0:dd MMMM yyyy} r.',</td><td>StringFormat='{} {0:dd MMMM yyyy} r.',</td></tr><tr><td></td><td></td><td></td></tr><tr><td>Binding="{Binding Source={x:Static</td><td>ConverterCulture={x:Static</td><td>ConverterCulture={x:Static</td></tr><tr><td>wnd:MainWindow.AuthUser},</td><td>gl:CultureInfo.CurrentCulture}}" /></td><td>gl:CultureInfo.CurrentCulture},</td></tr><tr><td>Path=idPos}" Nales "1"</td><td></Border></td><td>TargetNullValue='Отсутствует'}" /></td></tr></tbody></table></style>				

<Border

Grid.Column="7">

</Border>

Value="1">

```
<Border>
                                                   {usersDetail.Positions.name}";
                                                                                                      Orders
                                                                                                                                           null;
                                                                                                                    order
<TextBlock
                         Text="{Binding
                                                   PassportBlock.Text
                                                                                 $"Паспорт:
                                                                                                      var groupBy = list.GroupBy(s => s.idTask);
planDateComplete,
                                                   {usersDetail.SeriePassport}
                                                                                                      foreach (var ordersEnumerable in groupBy)
StringFormat='{}{0:dd MMMM yyyy} r.',
                                                   {usersDetail.NumberPassport}";
                                                   PhoneBlock.Text = $"Номер телефона:
ConverterCulture={x:Static
                                                                                                                     ordersEnumerable.Count();
                                                                                                      var
gl:CultureInfo.CurrentCulture},
                                                   {usersDetail.phone}";
                                                                                                      if
                                                                                                            (k
                                                                                                                           count)
                                                                                                                                      continue:
TargetNullValue='Oтсутствует'}"
                                      />
                                                   PhoneBlock.Text = "Номер телефона: " +
                                                                                                                                             k:
                                                                                                      count
</Border>
                                                   (usersDetail.phone
                                                                          !=
                                                                                 null
                                                                                           ?
                                                                                                      order
                                                                                                                       ordersEnumerable.First();
<Border>
                                                   usersDetail.phone.ToString()
                                                                                                      PopularOrderBlock.Text = "Популярный
<TextBlock
                         Text="{Binding
                                                   "Отсутствует");
                                                                                                      заказ: " + (order != null ? order.Tasks.name
factDateComplete,
                                                          (usersDetail.idPos
                                                                                          1)
StringFormat='{}{0:dd MMMM yyyy} r.',
                                                   CustomerLegalFormBlock.Text
                                                                                                                                "Отсутствует");
ConverterCulture={x:Static
                                                   $"Правовая
                                                                                                      MostExpensiveBlock.Text
                                                                                                                                       "Самый
                                                                                     форма:
                                                                                                      дорогой заказ: " + (list.Count > 0 ?
gl:CultureInfo.CurrentCulture},
                                                   \{usersDetail.Customers.LegalForms.name\}
TargetNullValue='Oтсутствует'}"
                                      />
                                                                                                      list.Max(s
                                                                                                                         s.price).ToString("#,#",
                                                                                                      CultureInfo.CurrentCulture) + " py6." :
</Border>
                                                   else CustomerLegalFormBlock.Visibility =
</UniformGrid>
                                                   Visibility. Collapsed;
                                                                                                      "Отсутствует");
</Grid>
                                                   AddressBlock.Text
                                                                                    $"Адрес
</CheckBox.Content>
                                                   проживания:
                                                                      {usersDetail.address}";
                                                                                                      private void RefreshOrdersBtn_Click(object
</CheckBox>
                                                   DateStart.Text = "Дата регистрации: " +
                                                                                                      sender,
                                                                                                                     RoutedEventArgs
                                                                                                                                             e)
</DataTemplate>
                                                   usersDetail.dateStart.ToString("dd MMMM
</ItemsControl.ItemTemplate>
                                                   уууу",
                                                                 CultureInfo.CurrentCulture);
                                                                                                      RefreshOrders();
</ItemsControl>
                                                   }
</Grid>
                                                   private readonly UsersDetail _usersDetail;
</Grid>
                                                               ObservableCollection<Orders>
</ScrollViewer>
                                                   OrdersCollection { get; set; } = new
</Grid>
                                                   ObservableCollection<Orders>();
                                                                                                      TopPanelControl.xaml
</Grid>
                                                   private
                                                             async
                                                                     void
                                                                             RefreshOrders()
                                                                                                      <UserControl
</Page>
                                                                                                      x:Class="INCOMSYSTEM.Controls.TopP
                                                                          Task.Run(()
                                                                                                      anelControl"
                                                        list
                                                   var
                                                              =
                                                                   await
ProfilePage.xaml.cs
                                                                                                      xmlns="http://schemas.microsoft.com/winf
         System.Collections.ObjectModel;
                                                   using
                                                                                                      x/2006/xaml/presentation"
using
                                                                        db
                                                              (var
                                                                                        new
using
                      System.Data.Entity;
                                                   INCOMSYSTEMEntities())
                                                                                                      xmlns:x="http://schemas.microsoft.com/wi
using
                    System.Globalization;
                                                                                                      nfx/2006/xaml"
                            System.Ling;
                                                   return
                                                                                   db.Orders
                                                                                                      xmlns:mc="http://schemas.openxmlformats
using
                 System.Threading.Tasks;
                                                                                                      .org/markup-compatibility/2006"
using
                                                   .Include(s
                                                                               s.Employees)
                        System.Windows;
                                                   .Include(s
                                                                    s.Employees.UsersDetail)
                                                                                                      xmlns:d="http://schemas.microsoft.com/ex
using
using
               System. Windows. Controls;
                                                   .Include(s
                                                                    s.Customers.UsersDetail)
                                                                                                      pression/blend/2008"
using
               INCOMSYSTEM.Context;
                                                   .Include(s
                                                                                  s.Statuses)
                                                                                                      mc:Ignorable="d">
                                                                      =>
namespace
                  INCOMSYSTEM.Pages
                                                   .Include(s
                                                                       =>
                                                                                    s.Tasks)
                                                                                                      <Border
{
                                                   .Include(s
                                                                                    s.Chats)
                                                                                                      MouseEnter="PnlControlBar_OnMouseEnt
public partial class ProfilePage: Page
                                                   .Include(s
                                                                         s.Chats.Employees)
                                                                 =>
                                                   .Where(s
                                                                        s.idCustomer
                                                                                                      MouseLeftButtonDown="PnlControlBar_O
                                                                =>
                                                                                                      nMouseLeftButtonDown"
public ProfilePage(UsersDetail usersDetail)
                                                   _usersDetail.idUser || s.idExecutor ==
                                                                                                                                  Padding="5">
                                                   _usersDetail.idUser || s.Chats.idManager ==
                                                                                                      <Grid>
_usersDetail
                             usersDetail;
                                                   _usersDetail.idUser)
                                                                                                      <Grid.RowDefinitions>
InitializeComponent();
                                                                                                      <RowDefinition
                                                   .ToList();
                                                                                                                          Height="Auto"
                                                                                                                                             />
                                                                                                      <RowDefinition
RefreshOrders():
                                                                                                                                             />
OrdersList.ItemsSource = OrdersCollection;
                                                                                                      </Grid.RowDefinitions>
                                                   });
NameBlock.Text = usersDetail.idPos == 1?
                                                   OrdersCollection
                                                                                        new
                                                                                                      <Grid
                                                                                                                                Grid.Row="0">
$"ФИО: {usersDetail.Customers.name}":
                                                   ObservableCollection<Orders>(list);
                                                                                                      <Grid.ColumnDefinitions>
$"ФИО:
                {usersDetail.Employees}";
                                                   OrdersList.ItemsSource = OrdersCollection;
                                                                                                      <ColumnDefinition
                                                                                                                                Width="Auto"/>
PositionBlock.Text
                           $"Должность:
                                                                                          0;
                                                                                                      <ColumnDefinition/>
                                                   var
                                                                count
```

<columndefinition width="Auto"></columndefinition>		WindowCurrentState.Text =
	<button <="" td="" x:name="SwitchStateBtn"><td>window.WindowState ==</td></button>	window.WindowState ==
<stackpanel <="" grid.column="0" td=""><td>Margin="3 0"</td><td>WindowState. Maximized ? "2" : "1";</td></stackpanel>	Margin="3 0"	WindowState. Maximized ? "2" : "1";
Orientation="Horizontal"	Click="SwitchStateBtn_Click"	};
VerticalAlignment="Center">	Width="40">	}
<image <="" margin="5 0" td=""/> <td><textblock< td=""><td>[DllImport("user32.dll")]</td></textblock<></td>	<textblock< td=""><td>[DllImport("user32.dll")]</td></textblock<>	[DllImport("user32.dll")]
Source="/Images/logo.png"/>	x:Name="WindowCurrentState" Text="1"	private static extern IntPtr
<grid></grid>	FontFamily="Webdings"	SendMessage(IntPtr hWnd, int wMsg, int
<grid.rowdefinitions></grid.rowdefinitions>	TextAlignment="Center"	wParam, int lParam);
<rowdefinition></rowdefinition>	VerticalAlignment="Center"	private void
<rowdefinition></rowdefinition>	Foreground="White"/>	$PnlControlBar_OnMouseLeftButtonDown (\\$
		object sender, MouseButtonEventArgs e)
<textblock <="" grid.row="0" td=""><td><button <="" click="CloseBtn_Click" td=""><td>{</td></button></td></textblock>	<button <="" click="CloseBtn_Click" td=""><td>{</td></button>	{
FontFamily="/INCOMSYSTEM;componen	Margin="3 0" Width="40">	$if \qquad \text{(e.ClickCount} \qquad == \qquad 2)$
t/Fonts/#Roboto"	<textblock <="" td="" text="r"><td>{</td></textblock>	{
FontSize="16"	FontFamily="Webdings"	SwitchState();
Foreground="White"	TextAlignment="Center"	return;
Margin="0 5 5 0"	VerticalAlignment="Center"	}
Техт="ИНКОМСИСТЕМ"/>	Foreground="White"/>	var window = (this.Parent as Grid).Parent as
<textblock <="" grid.row="1" td=""><td></td><td>Window;</td></textblock>		Window;
FontFamily="/INCOMSYSTEM;componen		var helper = new
t/Fonts/#Roboto"		WindowInteropHelper(window);
FontSize="12"	<contentpresenter grid.row="1"></contentpresenter>	SendMessage(helper.Handle, 161, 2, 0);
Foreground="White"		WindowCurrentState.Text =
Margin="0 0 5 5"		window.WindowState ==
Техt="Научный Инженерный Центр"/>		WindowState.Maximized ? "2" : "1";
		}
	TopPanelControl.xaml.cs	private void
<textblock <="" grid.column="1" td=""><td>using System;</td><td>PnlControlBar_OnMouseEnter(object</td></textblock>	using System;	PnlControlBar_OnMouseEnter(object
FontFamily="/INCOMSYSTEM;componen	using System.Runtime.InteropServices;	sender, MouseEventArgs e)
t/Fonts/#Roboto"	using System.Windows;	1
TextAlignment="Center"	using System.Windows.Controls;	var window = (this.Parent as Grid).Parent as
VerticalAlignment="Center"	using System.Windows.Input;	Window;
FontSize="18"	using System.Windows.Interop;	window.MaxHeight =
Foreground="White"	namespace INCOMSYSTEM.Controls	SystemParameters.MaximizedPrimaryScree
Margin="5,5"	{	nHeight;
Text="{Binding Title,	public partial class TopPanelControl :	}
RelativeSource={RelativeSource	UserControl	private void CloseBtn_Click(object sender,
AncestorType=Window}}">	{	RoutedEventArgs e)
<textblock.tooltip></textblock.tooltip>	public TopPanelControl()	{
*	{	var window = (this.Parent as Grid).Parent as
<binding mode='Self}"' path="Text" relativesource="{RelativeSource</td><td></td><td></td></tr><tr><td><Binding RelativeSource=" {relativesource=""></binding>	InitializeComponent():	Window:
Mode= Self }" Path="Text"/>	InitializeComponent(); Loaded += (s, e) =>	Window; window.Close();
	* "	Window; window.Close(); }
Mode=Self}" Path="Text"/>	Loaded += (s, e) => {	<pre>window.Close(); }</pre>
Mode=Self}" Path="Text"/> <stackpanel <="" grid.column="2" td=""><td>Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as</td><td><pre>window.Close(); } private void MinimizeBtn_Click(object</pre></td></stackpanel>	Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as	<pre>window.Close(); } private void MinimizeBtn_Click(object</pre>
Mode=Self}" Path="Text"/> <stackpanel grid.column="2" orientation="Horizontal"></stackpanel>	Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as Window;	window.Close(); } private void MinimizeBtn_Click(object sender, RoutedEventArgs e)
Mode=Self}" Path="Text"/> <stackpanel grid.column="2" orientation="Horizontal"> <button <="" click="MinimizeBtn_Click" td=""><td>Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as Window; window.MaxHeight =</td><td><pre>window.Close(); } private void MinimizeBtn_Click(object sender, RoutedEventArgs e) {</pre></td></button></stackpanel>	Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as Window; window.MaxHeight =	<pre>window.Close(); } private void MinimizeBtn_Click(object sender, RoutedEventArgs e) {</pre>
Mode=Self}" Path="Text"/> <stackpanel grid.column="2" orientation="Horizontal"> <button click="MinimizeBtn_Click" margin="3 0" width="40"></button></stackpanel>	Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as Window; window.MaxHeight = SystemParameters.MaximizedPrimaryScree	<pre>window.Close(); } private void MinimizeBtn_Click(object sender, RoutedEventArgs e) { var window = (this.Parent as Grid).Parent as</pre>
Mode=Self}" Path="Text"/> <stackpanel grid.column="2" orientation="Horizontal"> <button click="MinimizeBtn_Click" margin="3 0" width="40"> <textblock <="" td="" text="0"><td>Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as Window; window.MaxHeight = SystemParameters.MaximizedPrimaryScree nHeight;</td><td><pre>window.Close(); } private void MinimizeBtn_Click(object sender, RoutedEventArgs e) { var window = (this.Parent as Grid).Parent as Window;</pre></td></textblock></button></stackpanel>	Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as Window; window.MaxHeight = SystemParameters.MaximizedPrimaryScree nHeight;	<pre>window.Close(); } private void MinimizeBtn_Click(object sender, RoutedEventArgs e) { var window = (this.Parent as Grid).Parent as Window;</pre>
Mode=Self}" Path="Text"/> <stackpanel grid.column="2" orientation="Horizontal"> <button click="MinimizeBtn_Click" margin="3 0" width="40"> <textblock <="" fontfamily="Webdings" td="" text="0"><td>Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as Window; window.MaxHeight = SystemParameters.MaximizedPrimaryScree nHeight; SwitchStateBtn.Visibility =</td><td><pre>window.Close(); } private void MinimizeBtn_Click(object sender, RoutedEventArgs e) { var window = (this.Parent as Grid).Parent as Window; window.WindowState =</pre></td></textblock></button></stackpanel>	Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as Window; window.MaxHeight = SystemParameters.MaximizedPrimaryScree nHeight; SwitchStateBtn.Visibility =	<pre>window.Close(); } private void MinimizeBtn_Click(object sender, RoutedEventArgs e) { var window = (this.Parent as Grid).Parent as Window; window.WindowState =</pre>
Mode=Self}" Path="Text"/> <stackpanel grid.column="2" orientation="Horizontal"> <button click="MinimizeBtn_Click" margin="3 0" width="40"> <textblock <="" td="" text="0"><td>Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as Window; window.MaxHeight = SystemParameters.MaximizedPrimaryScree nHeight;</td><td><pre>window.Close(); } private void MinimizeBtn_Click(object sender, RoutedEventArgs e) { var window = (this.Parent as Grid).Parent as Window;</pre></td></textblock></button></stackpanel>	Loaded += (s, e) => { var window = (this.Parent as Grid).Parent as Window; window.MaxHeight = SystemParameters.MaximizedPrimaryScree nHeight;	<pre>window.Close(); } private void MinimizeBtn_Click(object sender, RoutedEventArgs e) { var window = (this.Parent as Grid).Parent as Window;</pre>

```
sender,
              Routed Event Args \\
                                      e)
{
SwitchState();
}
private
               void
                           SwitchState()
{
var window = (this.Parent as Grid).Parent as
Window;
window.WindowState
window.WindowState
                                     ==
                                      ?
WindowState. Maximized
Window State. \textbf{Normal}
WindowState.Maximized;
Window Current State. Text\\
window. Window State\\
WindowState. Maximized ? "2" : "1";
}
}
}
```

```
Приложение С.
                                                  </Border>
AdditionalWindow.xaml
                                                  </Grid>
                                                  </Window>
<Window
x{:}Class{=}"INCOMSYSTEM.Windows.Addi\\
tionalWindow"
                                                  AdditionalWindow.xaml.cs
xmlns="http://schemas.microsoft.com/winf
                                                  using
                                                                          System.Windows;
                                                                 System.Windows.Controls;
x/2006/xaml/presentation"
                                                  using
xmlns:x="http://schemas.microsoft.com/wi
                                                  using
                                                               System. Windows. Navigation;
                                                                 INCOMSYSTEM.Windows
nfx/2006/xaml"
                                                  namespace
xmlns:mc="http://schemas.openxmlformats
.org/markup-compatibility/2006"
                                                  public partial class AdditionalWindow:
xmlns:d="http://schemas.microsoft.com/ex
pression/blend/2008"
                                                                       AdditionalWindow()
xmlns:controls="clr-
                                                  public
namespace: INCOMSYSTEM. Controls"
                                                  {
mc:Ignorable="d"
                                                  InitializeComponent();
AllowsTransparency="True"
                                                  _errorStaticBorder
                                                                               ErrorBorder;
WindowStyle="None"
                                                  _errorStaticBlock
                                                                                ErrorBlock;
WindowStartupLocation="CenterScreen"
                                                  MFrame.Navigated
                                                                                        +=
WindowState="Normal"
                                                  MainFrameOnNavigated;
Background="DimGray"
ResizeMode="NoResize"
                                                  private static Border _errorStaticBorder;
Height="575"
                                                  private static TextBlock _errorStaticBlock;
Width="600"
                                                  private void MainFrameOnNavigated(object
BorderBrush="Black"
                                                  sender,
                                                               NavigationEventArgs
BorderThickness="2">
<Grid>
                                                  var page = ((Frame)sender). Content as Page;
<Grid.RowDefinitions>
                                                  this.Title
                                                                                page?.Title;
<RowDefinition
                        Height="Auto"/>
                                                  this.Width
                                                                                      50d;
                                                                    page.Width
<RowDefinition/>
                                                  this.Height
                                                                    page.Height
                                                                                      50d;
<RowDefinition
                        Height="Auto"/>
                                                  public static void ShowError(string error)
</Grid.RowDefinitions>
<controls:TopPanelControl/>
<Frame
                      x:Name="MFrame"
                                                  _errorStaticBlock.Text
                                                                                     error:
Grid.Row="1"
                                                  _errorStaticBorder.Visibility
Margin="5"
                                                  Visibility. Visible;
IsTabStop="False"
                                                  }
NavigationUIVisibility="Hidden"/>
                                                  public
                                                             static
                                                                       void
                                                                                HideError()
                           Grid.Row="2"
<Border
x:Name="ErrorBorder"
                                                  _errorStaticBlock.Text =
                                                                              string.Empty;
Background="#560b0b"
                                                  _errorStaticBorder.Visibility
BorderBrush="Red"
                                                  Visibility. Collapsed;
BorderThickness="1"
Visibility="Collapsed"
                                                  }
HorizontalAlignment="Center"
                                                  }
Padding="10
                                      5"
Margin="0
                          0
                                    15">
<TextBlock
                   x:Name="ErrorBlock"
Foreground="White"
VerticalAlignment="Center"
TextAlignment="Center"
HorizontalAlignment="Center"
FontSize="14"/>
```

Приложение D.	MainWindow.AuthUser.idUser)	else
ChatListPageViewModel.cs	.ToList();	{
using INCOMSYSTEM.Context;	var messages = db.Messages	chatMessColl.IsConnected =
using INCOMSYSTEM.Windows;	.Include(s => s.UsersDetail)	chatMess.IsConnected;
using System;	.ToList();	if (chatMessColl.Chat.Employees !=
using System.Data.Entity;	var chatColl =	chat.Employees)
using System.Collections.ObjectModel;	ChatsCollection.ToList().Where(s =>	{
using System.ComponentModel;	!chats.Select(k =>	chatMessColl.Chat.idManager =
using System.Ling;	k.idChat).Contains(s.Id));	chat.idManager;
using System.Threading.Tasks;	if (MainWindow.AuthUser.idPos == 3 &&	chatMessColl.Chat.Employees =
using System.Windows.Data;	chatColl.Any())	chat.Employees;
using INCOMSYSTEM.BehaviorsFiles;	{	chatMessColl.Recipient =
namespace INCOMSYSTEM.ViewModels	foreach (var c in chatColl)	chatMess.Recipient;
f	(vai e in chatcon)	}
public class ChatListPageViewModel :	ChatsCollection.Remove(c);	J
ObservableObject	` ` `	if (chatMassCall Chat Employees myll
ObservableObject	} continue.	if (chatMessColl.Chat.Employees == null chatMessColl.Message.id == chatMess.Id)
	continue;	,
public ChatListPageViewModel()	}	continue;
	foreach (var chat in chats)	1.34 0.110 15
ChatsCollection = new	1	chatMessColl.SendDate =
ObservableCollection <chatmess>();</chatmess>	var message = messages.LastOrDefault(s =>	chatMess.SendDate;
CollectionViewChat =	s.idChat == chat.idChat) ?? new Messages	chatMessColl.Message =
CollectionViewSource.GetDefaultView(Ch	{	chatMess.Message;
atsCollection);	idChat = chat.idChat,	}
CollectionViewChat.SortDescriptions.Inser	dateSend = chat.dateCreate,	}
t(0, new SortDescription("SendDate",	message = \$"Подробности заказа	}
ListSortDirection. Descending));	\"{chat.Orders.Tasks.name}\"",	
ChatsCollection.CollectionChanged += (s,	idUser = chat.idCustomer,	await Task.Delay(2500);
e) => OnRaiseChanged();	UsersDetail = chat.Customers.UsersDetail	}
RefreshCollection();	};	}
}	var chatMess = new ChatMess	}
public ObservableCollection <chatmess></chatmess>	{	public class ChatMess : ObservableObject
ChatsCollection { get; }	Id = chat.idChat,	{
public ICollectionView	Chat = chat,	private long _id;
CollectionViewChat { get; }	SendDate = message.dateSend,	private string _recipient;
public async void RefreshCollection()	Message = message,	private Chats _chat;
{	Order = chat.Orders,	private Orders _order;
while (!MainWindow.IsClosed)	IsConnected = chat.Employees != null	private Messages _message;
{	};	private DateTime? _sendDate;
using (var db = await Task.Run(() => new		private bool _isConnected;
INCOMSYSTEMEntities()))	if (MainWindow.AuthUser.idUser ==	public long Id
{	chat.idCustomer) chatMess.Recipient =	{
var chats = db.Chats.Include(s =>	chat.Employees == null	get => _id;
s.Customers)	? "Ожидание подключения менеджера"	set
.Include(s => s.Employees)	: \$"{chat.Employees}";	{
.Include(s => s.Employees.UsersDetail)	else chatMess.Recipient =	_id = value;
.Include(s => s.Employees.UsersDetail)	chat.Customers.name;	OnRaiseChanged();
.Include(s => s.Orders)		}
.Include(s => s.Orders.Tasks)	var chatMessColl =	}
.Include(s => s.Orders.Statuses)	ChatsCollection.FirstOrDefault(s => s.Id ==	public string Recipient
.Where(s => s.idCustomer ==	chatMess.Id);	-{
MainWindow.AuthUser.idUser	if (chatMessColl == null)	get => _recipient;
s.idManager ==	ChatsCollection.Add(chatMess);	set
-	•	

```
ChatListPage.xaml
                                                                                                     </Trigger>
                                                                                                     </Style.Triggers>
_recipient
                                  value;
                                                  <Page
OnRaiseChanged();
                                                  x:Class="INCOMSYSTEM.Pages.ChatList
                                                                                                     </Style>
                                                  Page"
                                                                                                     </Grid.Style>
                                                                                                     <Grid.Resources>
}
                                                  xmlns="http://schemas.microsoft.com/winf
public
                  Chats
                                    Chat
                                                  x/2006/xaml/presentation"
                                                                                                     <Style
                                                                                                                     TargetType="TextBlock"
                                                  xmlns:x="http://schemas.microsoft.com/wi
                                                                                                     BasedOn="{StaticResource
                                                  nfx/2006/xaml"
                                                                                                     TextBlockStyle}"/>
                                   chat:
get
                 =>
set
                                                  xmlns:mc="http://schemas.openxmlformats
                                                                                                     </Grid.Resources>
                                                                                                     <Grid.ColumnDefinitions>
                                                  .org/markup-compatibility/2006"
{
                                                  xmlns:d="http://schemas.microsoft.com/ex
                                                                                                     <ColumnDefinition
                                                                                                                               Width="225"/>
_chat
                                  value;
OnRaiseChanged();
                                                  pression/blend/2008"
                                                                                                     <ColumnDefinition/>
                                                  xmlns:vm="clr-
                                                                                                     <ColumnDefinition
                                                                                                                              Width="Auto"/>
}
                                                  namespace:INCOMSYSTEM.ViewModels
                                                                                                     </Grid.ColumnDefinitions>
                                                                                                     <Border
                                                                                                                            Grid.Column="0"
public
                 Orders
                                   Order
                                                                                                     Grid.ColumnSpan="3"
                                                  mc:Ignorable="d"
                                                                                                     BorderBrush="White"
                                                  Title="Мессенджер">
get
                                  order:
                                                                                                     BorderThickness="0
                                                                                                                                    0
                                                                                                                                           2"
                                                  <Page.DataContext>
set
                                                  <vm:ChatListPageViewModel/>
                                                                                                     x:Name="MainBorder"/>
{
                                  value;
                                                  </Page.DataContext>
                                                                                                     <Border
                                                                                                                            Grid.Column="0"
order
OnRaiseChanged();
                                                  <Grid>
                                                                                                     BorderBrush="#fff"
                                                                                                                                         0"/>
}
                                                  <ItemsControl
                                                                       BorderThickness="0"
                                                                                                     BorderThickness="0
                                                  ItemsSource="{Binding
                                                                                                     <Border
                                                                                                                            Grid.Column="1"
}
              Messages
                                Message
                                                  CollectionViewChat,
                                                                                                     BorderBrush="#fff"
public
                                                  UpdateSourceTrigger=PropertyChanged}
                                                                                                     BorderThickness="0
                                                                                                                            0
                                                                                                                                  2
                                                                                                                                         0"/>
                               _message;
                                                                                                     <Grid>
get
                                                                                                     <Grid.ColumnDefinitions>
set
                                                  BorderBrush="Transparent"
                                                                                       0">
{
                                                  Margin="75
                                                                                                     <ColumnDefinition
                                                                                                                                Width="32"/>
                                                                                                     <ColumnDefinition
                                                  <ItemsControl.ItemsPanel>
_message
                                  value;
                                                                                                     </Grid.ColumnDefinitions>
OnRaiseChanged();
                                                  <ItemsPanelTemplate>
                                                                          Cursor="Hand"/>
}
                                                  <StackPanel
                                                                                                     <Image
                                                                                                                   Source="../Images/user.png"
                                                  </ItemsPanelTemplate>
                                                                                                     Width="32"
                                                                                                                               Height="32"/>
}
public
             DateTime?
                               SendDate
                                                  </ItemsControl.ItemsPanel>
                                                                                                     <StackPanel Grid.Column="1" Margin="5
                                                  <ItemsControl.Style>
                                                                                                     0">
                                                                                                     <TextBlock
                                                                                                                              Text="{Binding
               =>
                               _sendDate;
                                                  <Style
                                                                TargetType="ItemsControl"
get
                                                  BasedOn="{StaticResource
                                                                                                     Order.Tasks.name}"
set
                                                  ItemsVisible}">
                                                                                                     TextWrapping="Wrap"/>
_sendDate
                                   value;
                                                  </Style>
                                                                                                     <Border
                                                                                                                           BorderBrush="#fff"
OnRaiseChanged();
                                                  ItemsControl.Style>
                                                                                                     BorderThickness="0
                                                                                                                            0
                                                                                                                                  0
                                                                                                                                         2"/>
                                                  <ItemsControl.ItemTemplate>
                                                                                                     <TextBlock Text="{Binding Recipient}"
                                                                                                     TextWrapping="Wrap">
}
                                                  <DataTemplate>
public
               bool
                             IsConnected
                                                  <Border>
                                                                                                     <TextBlock.Style>
                                                                                                                     TargetType="TextBlock"
                                                  <Grid
                                                                                                     <Style
{
                                                                                                     BasedOn="{StaticResource
                           _isConnected;
                                                  Margin="0
get
                                                  MouseLeftButtonUp="ChatEnter_LeftBtnU
                                                                                                     TextBlockStyle}">
set
                                                  p">
                                                                                                     <Setter
                                                                                                                       Property="Foreground"
_isConnected
                                                  <Grid.Style>
                                                                                                     Value="#ccc"/>
                                  value;
OnRaiseChanged();
                                                                       TargetType="Grid">
                                                                                                     <Style.Triggers>
                                                  <Style
                                                  <Style.Triggers>
                                                                                                     <DataTrigger
                                                                                                                           Binding="{Binding
                                                  <Trigger
                                                                   Property="IsMouseOver"
                                                                                                     IsConnected,
                                                  Value="True">
                                                                                                     UpdateSourceTrigger=PropertyChanged}
                                                                                                                               Value="True">
                                                  <Setter
                                                                    Property="Background"
                                                  Value="#9B9B9B"/>
                                                                                                     <Setter
                                                                                                                       Property="Foreground"
```

```
Value="#fff"/>
                                                                                                   pression/blend/2008" xmlns:controls="clr-
                                                     (ChatPages.ContainsKey(Chat.idChat))
                                                                                                   namespace:INCOMSYSTEM.Controls"
</DataTrigger>
</Style.Triggers>
                                                                                                   mc:Ignorable="d">
</Style>
                                                 Main Window. Chat Frame. Navigate (Chat Pa\\
                                                                                                   <Grid
                                                                                                                  Margin="30
                                                                                                                                       10">
                                                 ges[Chat.idChat]);
</TextBlock.Style>
                                                                                                   <Grid.RowDefinitions>
</TextBlock>
                                                                                                   <RowDefinition/>
                                                  }
</StackPanel>
                                                 else
                                                                                                   <RowDefinition
                                                                                                                            Height="100"/>
</Grid>
                                                                                                   <RowDefinition
                                                                                                                           Height="Auto"/>
<TextBlock
                       Grid.Column="1"
                                                 chatMess
                                                                       new
                                                                                ChatMess
                                                                                                   </Grid.RowDefinitions>
                                     0"
Margin="5
                                                                                                   <ItemsControl
                                                                                                                     x:Name="MessagesList"
                                                  {
                                                                              Chat.idChat,
                                                                                                   Style="{StaticResource
TextWrapping="Wrap"
                                                                                                                             ItemsVisible}"
VerticalAlignment="Center"
                                                  IsConnected
                                                                                                   BorderBrush="Wheat"
                                                                                                   BorderThickness="1"
Text="{Binding Message.shortMessage}"
                                                 Recipient
                                                                     Chat.Customers.name
Foreground="#ccc"/>
                                                                                                   Padding="10
                                                                                                                                       0">
<TextBlock
                       Grid.Column="2"
                                                 var chatPage = new ChatPage(chatMess);
                                                                                                   <ItemsControl.ItemTemplate>
                                     0"
Margin="5
                                                 ChatPages.Add(Chat.idChat,
                                                                                                   <DataTemplate>
                                                                                chatPage);
                                                 Main Window. Chat Frame. Navigate (chat Pag
VerticalAlignment="Top"
                                                                                                                                         3"
                                                                                                   <Border
                                                                                                                     Margin="5
                                                                                                   CornerRadius="15"
Text="{Binding
                              SendDate,
                                                 e);
                                                                                                   BorderBrush="Wheat"
StringFormat='{}Дата
                              отправки:
                                                  }
{0:dd.MM.yy
                           HH:mm}'}"/>
                                                                                                   BorderThickness="1"
                                                  }
</Grid>
                                                 else
                                                                                                   Width="{Binding
</Border>
                                                                                                   RelativeSource={RelativeSource
</DataTemplate>
                                                 chatMess = (sender as Grid).DataContext as
                                                                                                   Mode=FindAncestor,
</ItemsControl.ItemTemplate>
                                                 ChatMess;
                                                                                                   AncestorType=ItemsControl},
(ChatPages.ContainsKey(chatMess.Id))
                                                                                                   Path=ActualWidth,
</Grid>
                                                                                                   Converter={StaticResource
</Page>
                                                 MainWindow.ChatFrame.Navigate(ChatPa
                                                                                                   WidthConverter}}">
                                                  ges[chatMess.Id]);
                                                                                                   <Border.Style>
ChatListPage.xaml.cs
                                                                                                   <Style
                                                                                                                      TargetType="Border">
              System.Collections.Generic;
                                                 else
                                                                                                   <Setter
                                                                                                            Property="HorizontalAlignment"
using
using
               System.Windows.Controls;
                                                                                                   Value="Left"/>
                  System.Windows.Input;
                                                 var chatPage = new ChatPage(chatMess);
                                                                                                   <Style.Triggers>
using
               INCOMSYSTEM.Context;
                                                 ChatPages.Add(chatMess.Id,
                                                                                                   <DataTrigger
                                                                                                                         Binding="{Binding
using
                                                                               chatPage);
          INCOMSYSTEM. ViewModels;
                                                 MainWindow.ChatFrame.Navigate(chatPag
                                                                                                   ThisUser}"
                                                                                                                             Value="True">
using
             INCOMSYSTEM.Windows;
using
                                                 e);
                                                                                                   <Setter Property="HorizontalAlignment"
                   Page
                                                                                                   Value="Right"/>
using
                                                  }
System.Windows.Controls.Page;
                                                                                                   </DataTrigger>
                                                  }
namespace
                 INCOMSYSTEM.Pages
                                                 Chat
                                                                                     null;
                                                                                                   </Style.Triggers>
                                                                                                   </Style>
                                                  }
public partial class ChatListPage: Page
                                                                                                   </Border.Style>
                                                                                                   <Grid
                                                  }
                                                                                                                               Margin="5">
                                                                                                   <Grid.Resources>
public
                          ChatListPage()
                                                 ChatPage.xaml
                                                                                                   <Style
                                                                                                                    TargetType="TextBlock"
                                                                                                   BasedOn="{StaticResource
InitializeComponent();
                                                  <Page
                                                 x:Class="INCOMSYSTEM.Pages.ChatPag
                                                                                                   TextBlockStyle}"/>
private Dictionary<long, Page> ChatPages {
                                                                                                   </Grid.Resources>
                                                                                                   <Grid.RowDefinitions>
get; } = new Dictionary<long, Page>();
                                                  xmlns="http://schemas.microsoft.com/winf
                                                                                                   <RowDefinition
                                                                                                                              Height="30"/>
       Chats Chat { get; set; }
                                                  x/2006/xaml/presentation"
public
      void ChatEnter_LeftBtnUp(object
                                                  xmlns:x="http://schemas.microsoft.com/wi
                                                                                                   <RowDefinition
                                                                                                                           Height="Auto"/>
sender,
           MouseButtonEventArgs
                                                  nfx/2006/xaml"
                                                                                                   </Grid.RowDefinitions>
                                                 xmlns:mc="http://schemas.openxmlformats
                                                                                                   <Grid.ColumnDefinitions>
                                                                                                   <ColumnDefinition
                                                  .org/markup-compatibility/2006"
ChatMess
                              chatMess:
                                                                                                                                         />
if
          (Chat
                                                  xmlns:d="http://schemas.microsoft.com/ex
                                                                                                   <ColumnDefinition
                                                                                                                            Width="Auto"/>
                        !=
                                   null)
```

	Foreground="White"	private ICollectionView
<textblock text="{Binding Sender}"></textblock>	VerticalAlignment="Center"	CollectionViewMessages =>
<textblock <="" td="" text="{Binding Message}"><td>TextAlignment="Center"</td><td>Collection View Source. Get Default View (Me</td></textblock>	TextAlignment="Center"	Collection View Source. Get Default View (Me
Grid.Row="1"	HorizontalAlignment="Center"	ssagesCollection);
TextWrapping="Wrap"	FontSize="14"/>	private readonly ChatMess _chatMess;
Margin="0 0 30 0"/>		private async void RefreshMessages()
<textblock <="" td="" text="{Binding SendDate,</td><td></Grid></td><td>{</td></tr><tr><td>StringFormat='{}{0:dd.MM.yy HH:ss}'}"><td></td><td>while (!MainWindow.IsClosed)</td></textblock>		while (!MainWindow.IsClosed)
FontSize="12"		{
Margin="10 0"	ChatPage.xaml.cs	using (var db = await Task.Run(() => new
VerticalAlignment="Bottom"	using System;	INCOMSYSTEMEntities()))
Grid.Row="1"	using System.Collections.ObjectModel;	{
Grid.Column="1"/>	using System.ComponentModel;	var messages = db.Messages
	using System.Data.Entity;	.Include(s => s.UsersDetail)
	using System.Linq;	.Include(s => s.UsersDetail.Employees)
	using System.Threading.Tasks;	.Include(s => s.UsersDetail.Customers)
	using System.Windows;	.Include(s => s.Chats)
	using System.Windows.Data;	.Include(s => s.Chats.Orders)
<border <="" borderbrush="Wheat" td=""><td>using System.Windows.Input;</td><td>.Include(s => s.Chats.Orders.Tasks)</td></border>	using System.Windows.Input;	.Include(s => s.Chats.Orders.Tasks)
BorderThickness="1"	using INCOMSYSTEM.Context;	.Where(s => s.idChat == _chatMess.Id)
Grid.Row="1"	using INCOMSYSTEM.ViewModels;	.ToList();
Margin="0 10 0 0">	using INCOMSYSTEM.Windows;	foreach (var message in messages)
<grid></grid>	using Page =	{
<grid.columndefinitions></grid.columndefinitions>	System.Windows.Controls.Page;	var dialogMess = new DialogMess
<columndefinition></columndefinition>	namespace INCOMSYSTEM.Pages	{
<columndefinition width="Auto"></columndefinition>	{	Id = message.id,
	public partial class ChatPage : Page	Message = message.message,
<controls:inputtextbox< td=""><td>{</td><td>File = message.attachment,</td></controls:inputtextbox<>	{	File = message.attachment,
x:Name="InputMessageBox"	public ChatPage(ChatMess chatMess)	FileExtension = message.fileExtension,
PlaceHolder="Tekct"	{	SendDate = message.dateSend,
BorderBrush="Transparent"	_chatMess = chatMess;	FileName = message.attachment != null ?
BorderThickness="0"	InitializeComponent();	\$"{message.Chats.Orders.Tasks.name}.{me
MaxLength="255"	Title = chatMess.IsConnected ?	
TextAlignment=" Left "	\$"Переписка с {chatMess.Recipient}" :	ssage.fileExtension}" : string.Empty };
TextWrapping="Wrap"/>	chatMess.Recipient;	if (MainWindow.AuthUser.idUser ==
<pre><stackpanel< td=""><td>CollectionViewMessages.SortDescriptions.</td><td>message.idUser) dialogMess.ThisUser =</td></stackpanel<></pre>	CollectionViewMessages.SortDescriptions.	message.idUser) dialogMess.ThisUser =
VerticalAlignment="Center">	Insert(0, new SortDescription("SendDate",	
<pre><button <="" pre="" x:name="SendMessageBtn"></button></pre>	ListSortDirection. Ascending));	true; switch (message.UsersDetail.idPos)
Content="Отправить"	MessagesList.ItemsSource =	switch (message. OsersDetail.idr os)
Click="SendMessageBtn_Click"/>	•	1
	CollectionViewMessages; InputMessageBox.KeyDown += (s, e) =>	case 1: dialogMess.Sender =
	inputwiessageBox.ReyDown += (s, e) =>	· ·
	f (a Kara Kara Fratan) matrices	message.UsersDetail.Customers.name;
	if (e.Key != Key. Enter) return;	break;
<border <="" grid.row="2" td=""><td>var start = InputMessageBox.SelectionStart;</td><td>case 3:</td></border>	var start = InputMessageBox.SelectionStart;	case 3:
x:Name="ErrorBorder"	InputMessageBox.Text = InputMessageBox Text Incert(start	dialogMess.Sender =
Background="#560b0b"	InputMessageBox.Text.Insert(start++, "\n");	\$"{message.UsersDetail.Employees}";
BorderBrush="Red"	InputMessageBox.SelectionStart = start;	break;
BorderThickness="1"	};	}
Visibility="Collapsed"	Task.Run(RefreshMessages);	if (MessagesCollection.FirstOrDefault(s =>
HorizontalAlignment="Center"	}	s.Id == dialogMess.Id) == null)
Padding="10 5"	private ObservableCollection <dialogmess></dialogmess>	Application.Current.Dispatcher.Invoke(()
Margin="0 5 0 15">	MessagesCollection { get; } = new	=> MessagesCollection.Add(dialogMess));
<textblock <="" td="" x:name="ErrorBlock"><td>ObservableCollection<dialogmess>();</dialogmess></td><td>}</td></textblock>	ObservableCollection <dialogmess>();</dialogmess>	}

```
await
                       Task.Delay(2500);
                                                   using (var db = await Task.Run(() => new
                                                   INCOMSYSTEMEntities()))
}
                                                   var
                                                          message
                                                                           new
                                                                                  Messages
                  _canSend
                                                   {
                                                                                                      public
private
          bool
                                    true;
private void SendMessageBtn_Click(object
                                                   idChat
                                                                               _chatMess.Id,
                                                                                                      public
                                                                                                      public
              RoutedEventArgs
                                                   idUser = MainWindow.AuthUser.idUser,
sender.
                                                                     InputMessageBox.Text,
                                                                                                      public
{
                                                   message
if
         (InputMessageBox.IsWhiteSpace)
                                                   dateSend
                                                                             DateTime.Now
                                                   };
ErrorBorder.Visibility = Visibility.Visible;
                                                   db.Messages.Add(message);
SetError("Сообщение не может быть
                                                                    db.SaveChangesAsync();
пустым");
                                                       dialogMess
                                                                     =
                                                                         new DialogMess
return;
                                                   Id
}
                                                                                 message.id,
if (InputMessageBox.Text.Length > 255)
                                                   Message
                                                                           message.message,
                                                   SendDate
                                                                          message.dateSend,
ErrorBorder.Visibility = Visibility.Visible;
                                                   ThisUser
                                                                                       true,
SetError("Сообщение
                                 должно
                                                   };
превышать
                  255
                                                              (MainWindow.AuthUser.idPos)
                             символов"):
                                                   switch
return;
                                                   {
}
                                                   case
                                                                                          1:
if (!_canSend) { ErrorBorder.Visibility =
                                                   dialogMess.Sender
Visibility. Visible;
                         return;
                                                   MainWindow.AuthUser.Customers.name;
                                       }
Task.Run(async
                                                   break;
                                                   case
                                                                                         3:
Application.Current.Dispatcher.Invoke(()
                                                   dialogMess.Sender
=> SendMessageBtn.IsEnabled = false);
                                                   $"{MainWindow.AuthUser.Employees}";
Application.Current.Dispatcher.Invoke(()
                                                   break;
          _canSend
                                   false);
                          =
for
     (var i = 5; i >=
                                0; i--)
                                                   Application.Current.Dispatcher.Invoke(()
                                                   => MessagesCollection.Add(dialogMess));
{
string
                                timerStr;
                                                   InputMessageBox.Clear();
if (i > 1) timerStr = $"{i} секунды";
else if (i == 1) timerStr = \{i\} секунду";
                                                   if (MessagesCollection.Count >= 2) return;
       timerStr
                       $"{i}
                                                   Application.Current.Dispatcher.Invoke(()
                                секунд";
Application.Current.Dispatcher.Invoke(()
        SetError($"Перед
                              отправкой
следующего
               сообщения
                              подождите
                                                   var style = MessagesCollection.Count < 2?
                                                   MessagesList.Style
{timerStr}"));
                                                                                       null;
                       Task.Delay(1000);
await
                                                   MessagesList.Style
                                                                                       null;
                                                   MessagesList.Style
                                                                                      style;
Application.Current.Dispatcher.Invoke(()
                                                   });
=>
          canSend
                                    true):
Application.Current.Dispatcher.Invoke(()
                                                                     SetError(string
                                                   private
                                                             void
                                                                                      error)
=> SendMessageBtn.IsEnabled = true);
HideError();
                                                   ErrorBlock.Text
                                                                                      error;
});
                                                   }
HideError();
                                                                                 HideError()
                                                   private
                                                                   void
SendMessage();
                                                   ErrorBorder. Visibility
                                                   Visibility. Collapsed;
private
          async
                  void
                          SendMessage()
                                                   ErrorBlock.Text
                                                                               string.Empty;
```

```
internal
               class
                            DialogMess
        long
                Id
                         get;
                                set;
       string
               Sender
                           get;
              Message
       string
                           get; set;
        byte[]
                File
                          get;
                                set:
                     {
public string FileExtension { get; set; }
      string FileName { get; set; }
public DateTime SendDate { get; set; }
public bool ThisUser { get; set; }
```

Приложение Е.	RadiusX="50"	Content="Пользователи"/>
ManagerPage.xaml	RadiusY="50"/>	<textblock <="" grid.column="1" td=""></textblock>
<page< td=""><td><rectangle <="" grid.row="1" td=""><td>TextWrapping="Wrap"</td></rectangle></td></page<>	<rectangle <="" grid.row="1" td=""><td>TextWrapping="Wrap"</td></rectangle>	TextWrapping="Wrap"
x:Class="INCOMSYSTEM.Pages.MainPag	Grid.Column="1"	Text="Перейдя в окно с пользователями,
es.ManagerPage"	Margin="15 5"	появится список пользователей с
xmlns="http://schemas.microsoft.com/winf	Height="3"	возможность фильтрации поиска, так же
x/2006/xaml/presentation"	Fill="White"	можно просмотреть аккаунт
xmlns:x="http://schemas.microsoft.com/wi	RadiusX="50"	пользователя и перейти к чату, если
nfx/2006/xaml"	RadiusY="50"/>	существует"/>
xmlns:mc="http://schemas.openxmlformats		
.org/markup-compatibility/2006"	<grid grid.row="1"></grid>	
xmlns:d="http://schemas.microsoft.com/ex	<grid.rowdefinitions></grid.rowdefinitions>	
pression/blend/2008"	<rowdefinition></rowdefinition>	
mc:Ignorable="d"	<rowdefinition height="Auto"></rowdefinition>	ManagerPage.xaml.cs
Title="Главное окно"		using System.Windows;
Background="DimGray">	<grid.columndefinitions></grid.columndefinitions>	using System.Windows.Controls;
<page.resources></page.resources>	<columndefinition width="150"></columndefinition>	using INCOMSYSTEM.Windows;
<style <="" targettype="TextBlock" td=""><td><ColumnDefinition Width="300"/></td><td>namespace</td></tr><tr><td>BasedOn="{StaticResource</td><td></ri> /Grid.ColumnDefinitions></td><td>INCOMSYSTEM.Pages.MainPages</td></tr><tr><td>TextBlockStyle}"/></td><td><Button Margin="5"</td><td>{</td></tr><tr><td><Style TargetType="Label"</td><td>Content="Заказы"</td><td>public partial class ManagerPage : Page</td></tr><tr><td>BasedOn="{StaticResource LabelStyle}"/></td><td>Click="ViewOrderBtn_Click"/></td><td>public partial class Managerrage . Tage</td></tr><tr><td><pre>//Page.Resources></pre></td><td><TextBlock Grid.Column="1"</td><td>public ManagerPage()</td></tr><tr><td><pre><Grid HorizontalAlignment="Center"</pre></td><td>TextWrapping="Wrap"</td><td>{</td></tr><tr><td>VerticalAlignment="Center"></td><td>Техt "Перейдя в окно с заказами,</td><td>InitializeComponent();</td></tr><tr><td><Grid.RowDefinitions></td><td>появится список заказов, где можно</td><td>l</td></tr><tr><td><RowDefinition/></td><td>просмотреть, изменить, перейти к чату</td><td>private void ViewTasksBtn_Click(object</td></tr><tr><td><RowDefinition/></td><td>заказа, сформировать договор"/></td><td>sender, RoutedEventArgs e)</td></tr><tr><td><RowDefinition/></td><td><Rectangle Grid.Row="1"</td><td>{</td></tr><tr><td></ri> /Grid.RowDefinitions></td><td>Grid.Column="0"</td><td>MainWindow.ReviewFrame.Navigate(new</td></tr><tr><td><Grid></td><td>Margin="15 5"</td><td>ViewTasksPage());</td></tr><tr><td><Grid.RowDefinitions></td><td>Height="3"</td><td>lew rasksrage(j),</td></tr><tr><td><RowDefinition/></td><td>Fill="Gray"</td><td>private void ViewOrderBtn_Click(object</td></tr><tr><td><RowDefinition Height="Auto"/></td><td>RadiusX="50"</td><td>sender, RoutedEventArgs e)</td></tr><tr><td></ri> /Grid.RowDefinitions></td><td>RadiusY="50"/></td><td>schuci, RoutedEventAigs e)</td></tr><tr><td><Grid.ColumnDefinitions></td><td><Rectangle Grid.Row="1"</td><td>MainWindow.ReviewFrame.Navigate(new</td></tr><tr><td><ColumnDefinition Width="150"/></td><td>Grid.Column="1"</td><td>ViewOrdersPage());</td></tr><tr><td><ColumnDefinition Width="350"/></td><td>Margin="15 5"</td><td>\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\</td></tr><tr><td></ri> ColumnDefinitions</td><td>Height="3"</td><td>)</td></tr><tr><td><Button Margin="5"</td><td>Fill="White"</td><td>}</td></tr><tr><td>Content="Задачи"</td><td>RadiusX="50"</td><td>ſ</td></tr><tr><td>Click="ViewTasksBtn_Click"/></td><td>RadiusY="50"/></td><td>ViewOrdersPage.xaml</td></tr><tr><td><TextBlock Grid.Column="1"</td><td></Grid></td><td><Page</td></tr><tr><td>TextWrapping="Wrap"</td><td><Grid Grid.Row="2"></td><td>· ·</td></tr><tr><td>техтwтаррше wrap Техt="Перейдя в окно с задачи, появится</td><td><Grid.RowDefinitions></td><td>x:Class="INCOMSYSTEM.Pages.MainPag es.ViewOrdersPage"</td></tr><tr><td>•</td><td></td><td>· ·</td></tr><tr><td>список задач с возможность их</td><td><RowDefinition/> <RowDefinition Height="Auto"/></td><td>xmlns="http://schemas.microsoft.com/winf</td></tr><tr><td>фильтрации, так же можно их добавить,</td><td></ri> </ri> </ri> /Grid.RowDefinitions></td><td>x/2006/xaml/presentation"</td></tr><tr><td>редактировать или удалить"/></td><td></td><td>xmlns:x="http://schemas.microsoft.com/wi</td></tr><tr><td><Rectangle Grid.Row="1"</td><td><pre><Grid.ColumnDefinitions></pre></td><td>nfx/2006/xaml"</td></tr><tr><td>Grid.Column="0"</td><td><pre><ColumnDefinition Width="150"/> Width="1200"/></pre></td><td>xmlns:mc="http://schemas.openxmlformats</td></tr><tr><td>Margin="15 5"</td><td><ColumnDefinition Width="300"/></td><td>.org/markup-compatibility/2006"</td></tr><tr><td>Height="3"</td><td></ri> </Grid.ColumnDefinitions> **Marsin="5"</td><td>xmlns:d="http://schemas.microsoft.com/ex</td></tr><tr><td>Fill="Gray"</td><td><Button Margin="5"</td><td>pression/blend/2008"</td></tr></tbody></table></style>		

xmlns:gl="clr-	<setter <="" property="Background" td=""><td><textblock text="Цена"></textblock></td></setter>	<textblock text="Цена"></textblock>		
namespace:System.Globalization;assembly	Value="DimGray"/>			
=mscorlib"	<setter <="" property="BorderBrush" td=""><td><border grid.column="6"></border></td></setter>	<border grid.column="6"></border>		
xmlns:wnd="clr-	Value="Black"/>	<textblock text="Сложность"></textblock>		
namespace:INCOMSYSTEM.Windows"	<setter <="" property="BorderThickness" td=""><td></td></setter>			
mc:Ignorable="d"	Value="1"/>	<border grid.column="7"></border>		
Title="Список заказов">		<textblock text="Дата формирования</td></tr><tr><td><Page.Resources></td><td></Grid.Resources></td><td>заказа"></textblock>		
<style <="" targettype="TextBlock" td=""><td><Grid.ColumnDefinitions></td><td></Border></td></tr><tr><td>BasedOn="{StaticResource</td><td><ColumnDefinition Width="100"/></td><td><Border Grid.Column="8"></td></tr><tr><td>TextBlockStyle}"/></td><td><ColumnDefinition Width="200"/></td><td><TextBlock Text="CTaTyc"/></td></tr><tr><td><Style TargetType="Label"</td><td><ColumnDefinition Width="Auto"/></td><td></Border></td></tr><tr><td>BasedOn="{StaticResource LabelStyle}"/></td><td><ColumnDefinition Width="200"/></td><td></Grid></td></tr><tr><td></Page.Resources></td><td><ColumnDefinition Width="150"/></td><td><ItemsControl x:Name="OrdersList"</td></tr><tr><td><Grid HorizontalAlignment="Center"></td><td><ColumnDefinition Width="100"/></td><td>Grid.Row="1"</td></tr><tr><td><Grid.RowDefinitions></td><td><ColumnDefinition Width="100"/></td><td>Style="{StaticResource ItemsVisible}"></td></tr><tr><td><RowDefinition Height="Auto"/></td><td><ColumnDefinition Width="225"/></td><td><ItemsControl.ItemTemplate></td></tr><tr><td><RowDefinition/></td><td><ColumnDefinition Width="100"/></td><td><DataTemplate></td></tr><tr><td></Grid.RowDefinitions></td><td></Grid.ColumnDefinitions></td><td><CheckBox x:Name="OrderGrid"</td></tr><tr><td><StackPanel Orientation="Horizontal"</td><td><Border></td><td>Margin="0 0 0 10" Style="{StaticResource</td></tr><tr><td>HorizontalAlignment="Center" Margin="0</td><td><TextBlock Text="№Заказа"/></td><td>ClearCheckBoxTemplate}"></td></tr><tr><td>5"></td><td></Border></td><td><CheckBox.ContextMenu></td></tr><tr><td><TextBlock Text="{Binding</td><td><Border Grid.Column="1"></td><td><ContextMenu></td></tr><tr><td>ElementName=OrdersList,</td><td><TextBlock Text="Заказчик"/></td><td><MenuItem Header="Просмотреть заказ"</td></tr><tr><td>Path=Items.Count}"/></td><td></Border></td><td>Click="ViewOrderMenu_Click"</td></tr><tr><td><TextBlock Text=" из "/></td><td><Border Grid.Column="2" Width="200"></td><td>Visibility="{Binding CanViewDetail}"</td></tr><tr><td><TextBlock x:Name="AllItemsCount"/></td><td><Border.Style></td><td>CommandParameter="{Binding</td></tr><tr><td></StackPanel></td><td><Style TargetType="Border"</td><td>RelativeSource={RelativeSource</td></tr><tr><td><ScrollViewer Grid.Row="1"</td><td>BasedOn="{StaticResource {x:Type</td><td>Mode=Self},</td></tr><tr><td>VerticalScrollBarVisibility="Disabled"</td><td>Border}}"></td><td>Path=DataContext}"/></td></tr><tr><td>HorizontalScrollBarVisibility="Auto"></td><td><Setter Property="Visibility"</td><td><MenuItem Header="Перейти к чату"</td></tr><tr><td><Grid></td><td>Value="Visible"/></td><td>Click="OpenChatOrderMenu_Click"</td></tr><tr><td><Grid.RowDefinitions></td><td><Style.Triggers></td><td>Visibility="{Binding CanJoinChat}"</td></tr><tr><td></td><td><pre><DataTrigger Binding="{Binding</pre></td><td>CommandParameter="{Binding</td></tr><tr><td><RowDefinition Height="Auto"/> <RowDefinition/></td><td></td><td></td></tr><tr><td></br> </Grid.RowDefinitions></td><td>Source={x:Static wnd:MainWindow.AuthUser},</td><td>RelativeSource={RelativeSource</td></tr><tr><td><Grid></td><td>· ·</td><td>Mode=Self}, Path=DataContext}"/></td></tr><tr><td></td><td>Path=idPos}" Value="2"></td><td>•</td></tr><tr><td><Grid.Resources></td><td></td><td><MenuItem Header="Сформировать</td></tr><tr><td><Style TargetType="TextBlock"</td><td><Setter Property="Visibility"</td><td>договор"</td></tr><tr><td>BasedOn="{StaticResource</td><td>Value="Collapsed"/></td><td>Click="FormAgreementOrderMenu_Click"</td></tr><tr><td>ComboBoxTextBlockStyle}"></td><td></DataTrigger></td><td>Visibility="{Binding</td></tr><tr><td><Setter Property="Foreground"</td><td></Style.Triggers></td><td>CanFormAgreement}"</td></tr><tr><td>Value="White"/></td><td></style>	CommandParameter="{Binding			
<setter <="" property="TextAlignment" td=""><td></td><td>RelativeSource={RelativeSource</td></setter>		RelativeSource={RelativeSource		
Value="Center"/>	<textblock text="Исполнитель"></textblock>	Mode=Self},		
<setter <="" property="VerticalAlignment" td=""><td></td><td>Path=DataContext}"/></td></setter>		Path=DataContext}"/>		
Value="Center"/>	<border grid.column="3"></border>	<menuitem background"<="" header="Установить</td></tr><tr><td><Setter Property=" td=""/> <td><textblock text="Менеджер"></textblock></td> <td>фактическую дату начала"</td>	<textblock text="Менеджер"></textblock>	фактическую дату начала"
Value="DimGray"/>		Click="SetFactDateStartMenu_Click"		
<setter <="" property="FontWeight" td=""><td><border grid.column="4"></border></td><td>Visibility="{Binding</td></setter>	<border grid.column="4"></border>	Visibility="{Binding		
Value="Bold"/>	<textblock text="Задача"></textblock>	CanSetFactStartDate}"		
		CommandParameter="{Binding		
.O. 1 TE .TE !ID 1 "	D 1 G:1G1 "5"	Did G (Did G		

Grid.Column="5">

 $Relative Source = \{Relative Source$

<Style

TargetType="Border">

<Border

| Mode=Self}, | <border></border> | <border grid.column="8"></border> | | | |
|--|---|---|---|--|----------------------------|
| Path=DataContext}"/> | <textblock text="{Binding id}"></textblock> | <textblock td="" text="{Binding</td></tr><tr><td><MenuItem Header=" установить<=""><td></td><td>Statuses.name}"/></td></textblock> | | Statuses.name}"/> | |
| фактическую дату завершения" | <border grid.column="1"></border> | | | | |
| Click="SetFactDateCompleteMenu_Click" | <textblock 1"<="" td="" text="{Binding</td><td><UniformGrid Grid.Row="></textblock> | | | | |
| Visibility="{Binding | Customers.name}"/> | Grid.ColumnSpan="9" | | | |
| CanSetFactCompleteDate}" | | Grid.Column="0" | | | |
| CommandParameter="{Binding | <border grid.column="2" width="200"></border> | Columns="4" | | | |
| RelativeSource={RelativeSource | <border.style></border.style> | Height="80"> | | | |
| Mode=Self}, | <style <="" targettype="Border" td=""><td><UniformGrid.Style></td></tr><tr><td>Path=DataContext}"/></td><td>BasedOn="{StaticResource {x:Type</td><td><Style TargetType="UniformGrid"></td></tr><tr><td></ContextMenu></td><td>Border}}"></td><td><Setter Property="Visibility"</td></tr><tr><td></CheckBox.ContextMenu></td><td><Setter Property="Visibility"</td><td>Value="Collapsed"/></td></tr><tr><td><CheckBox.Content></td><td>Value="Visible"/></td><td><Style.Triggers></td></tr><tr><td><Grid></td><td><Style.Triggers></td><td><DataTrigger Binding="{Binding</td></tr><tr><td><Grid.Resources></td><td><DataTrigger Binding="{Binding</td><td>ElementName=OrderGrid,</td></tr><tr><td><Style TargetType="TextBlock"</td><td>Source={x:Static</td><td>Path=IsChecked}" Value="True"></td></tr><tr><td>BasedOn="{StaticResource</td><td>wnd:MainWindow.AuthUser},</td><td><Setter Property="Visibility"</td></tr><tr><td>ComboBoxTextBlockStyle}"></td><td>Path=idPos}"</td><td>Value="Visible"/></td></tr><tr><td><Setter Property="Foreground"</td><td>Value="2"></td><td></br></td></tr><tr><td>Value="White"/></td><td><pre><Setter Property="Visibility"</pre></td><td></style.Triggers></td></tr><tr><td><pre><Setter Property="TextAlignment"</pre></td><td>Value="Collapsed"/></td><td></style> | | | | |
| Value="Center"/> | | | | | |
| <setter <="" property="VerticalAlignment" td=""><td></td><td><border></border></td></setter> | | <border></border> | | | |
| Value="Center"/> | | <textblock <="" td="" text="Плановая дата начала"></textblock> | | | |
| <setter <="" property="Background" td=""><td></td><td>FontWeight="Bold"/></td></setter> | | FontWeight="Bold"/> | | | |
| Value="DimGray"/> | <textblock text="{Binding Employees,</td><td></Border></td></tr><tr><td></Style></td><td>TargetNullValue='Orcyrcrayer'}"></textblock> | <border></border> | | | |
| <style targettype="Border"></td><td></Border></td><td><TextBlock Text="Фактическая дата</td></tr><tr><td><Setter Property="Background"</td><td><Border Grid.Column="3"></td><td>начала" FontWeight="Bold"/></td></tr><tr><td>Value="DimGray"/></td><td><TextBlock Text="{Binding</td><td></Border></td></tr><tr><td><Setter Property="BorderBrush"</td><td>Chats.Employees,</td><td><Border></td></tr><tr><td>Value="Black"/></td><td>TargetNullValue='Отсутствует'} "/></td><td><TextBlock Техt="Плановая дата</td></tr><tr><td><Setter Property="BorderThickness"</td><td></Border></td><td>завершения" FontWeight="Bold"/></td></tr><tr><td>Value="1"/></td><td><Border Grid.Column="4"></td><td></Border></td></tr><tr><td><Setter Property="Padding" Value="5"/></td><td><TextBlock Text="{Binding</td><td><Border></td></tr><tr><td></style> | Tasks.name}"/> | <textblock fontweight="Bold" text="Фактическая дата</td></tr><tr><td></Grid.Resources></td><td></Border></td><td>завершения"></textblock> | | | |
| <grid.rowdefinitions></grid.rowdefinitions> | <border grid.column="5"></border> | | | | |
| <rowdefinition height="40"></rowdefinition> | <textblock auto"="" text="{Binding price,</td><td><Border></td></tr><tr><td><RowDefinition Height="></textblock> | StringFormat='{} {0:0,0} pyő.', | <textblock text="{Binding planDateStart,</td></tr><tr><td></Grid.RowDefinitions></td><td>ConverterCulture={x:Static</td><td>StringFormat='{}{0:dd MMMM yyyy} r.',</td></tr><tr><td><Grid.ColumnDefinitions></td><td>gl:CultureInfo.CurrentCulture}}"></textblock> | ConverterCulture={x:Static | |
| <columndefinition width="100"></columndefinition> | | gl:CultureInfo.CurrentCulture}, | | | |
| <columndefinition width="200"></columndefinition> | <border grid.column="6"></border> | TargetNullValue='Oтсутствует'}"/> | | | |
| <columndefinition width="Auto"></columndefinition> | <textblock text="{Binding difficulty}"></textblock> | | | | |
| <columndefinition width="200"></columndefinition> | | <border></border> | | | |
| <columndefinition width="150"></columndefinition> | <border grid.column="7"></border> | <textblock 100"="" text="{Binding factDateStart,</td></tr><tr><td><ColumnDefinition Width="></textblock> | <textblock 100"="" text="{Binding dateOrder,</td><td>StringFormat='{} {0:dd MMMM yyyy} r.',</td></tr><tr><td><ColumnDefinition Width="></textblock> | StringFormat='{} {0:dd MMMM yyyy} r.', | ConverterCulture={x:Static |
| <columndefinition width="225"></columndefinition> | ConverterCulture={x:Static | gl:CultureInfo.CurrentCulture}, | | | |
| <columndefinition width="100"></columndefinition> | gl:CultureInfo.CurrentCulture}}"/> | TargetNullValue='Oтсутствует'}"> | | | |
| | | <textblock.style></textblock.style> | | | |
| | | | | | |

```
<Style
                 TargetType="TextBlock"
                                                  </Border>
BasedOn="{StaticResource
                                                  </UniformGrid>
                                 {x:Type
                                                                                                     OrdersList.ItemsSource
                                                                                                                                    db.Orders
TextBlock}}">
                                                  </Grid>
                                                                                                     .Include(s
                                                                                                                      =>
                                                                                                                                 s.Customers)
<Style.Triggers>
                                                  </CheckBox.Content>
                                                                                                     .Include(s
                                                                                                                      =>
                                                                                                                                 s.Employees)
                                                  </CheckBox>
<DataTrigger
                      Binding="{Binding
                                                                                                     .Include(s
                                                                                                                     s.Employees.UsersDetail)
IsStartLateYellow}"
                          Value="True">
                                                  </DataTemplate>
                                                                                                     .Include(s
                                                                                                                      s.Customers.UsersDetail)
                                                  /ItemsControl.ItemTemplate>
<Setter
                  Property="Foreground"
                                                                                                     .Include(s
                                                                                                                                   s Statuses)
                                                                                                                       =>
Value="Yellow"/>
                                                  </ItemsControl>
                                                                                                     .Include(s
                                                                                                                                     s.Tasks)
</DataTrigger>
                                                  </Grid>
                                                                                                     .Include(s
                                                                                                                        =>
                                                                                                                                     s.Chats)
<DataTrigger
                      Binding="{Binding
                                                  </ScrollViewer>
                                                                                                     .Include(s
                                                                                                                           s.Chats.Employees)
IsStartLateRed}"
                          Value="True">
                                                  </Grid>
                                                                                                     .Where(s => MainWindow.AuthUser.idPos
<Setter
                  Property="Foreground"
                                                  </Page>
                                                                                                                          s.idExecutor
Value="Red"/>
                                                                                                     MainWindow.AuthUser.idUser)
                                                                                                     .ToList();
</DataTrigger>
                                                  ViewOrdersPage.xaml.cs
</Style.Triggers>
                                                  using
                                                                                   System;
                                                                        System.Data.Entity;
</Style>
                                                  using
</TextBlock.Style>
                                                                        System.Diagnostics;
                                                                                                     private void ViewOrderMenu_Click(object
                                                  using
</TextBlock>
                                                                      System.Globalization;
                                                                                                     sender,
                                                                                                                   RoutedEventArgs
                                                  using
                                                                                                                                           e)
</Border>
                                                  using
                                                                                System.IO;
<Border>
                                                                              System.Linq;
                                                                                                                       order
                                                  using
                                                                                                     var
<TextBlock
                         Text="{Binding
                                                                          System.Windows;
                                                                                                     ((MenuItem)sender).CommandParameter as
                                                  using
planDateComplete,
                                                  using
                                                                  System.Windows.Controls;
                                                                                                     Orders;
StringFormat='{}{0:dd MMMM yyyy} Γ.',
                                                  using
                                                           INCOMSYSTEM.BehaviorsFiles;
                                                                                                     var
                                                                                                               addWindow
                                                                                                                                         new
ConverterCulture={x:Static
                                                                  INCOMSYSTEM.Context;
                                                                                                     AdditionalWindow();
                                                  using
gl:CultureInfo.CurrentCulture},
                                                  using
                                                                                                     addWindow.MFrame.Navigate(new
TargetNullValue='Oтсутствует'} "/>
                                                  INCOMSYSTEM.Pages.MainPages.Views;
                                                                                                     ViewDetailOrderPage(order));
</Border>
                                                                INCOMSYSTEM.Windows;
                                                                                                     if (addWindow.ShowDialog() != true)
                                                  using
<Border>
                                                  using
                                                                          Microsoft.Win32;
                                                                                                     return;
<TextBlock
                         Text="{Binding
                                                                      Word
                                                                                                     ApplyFilter();
                                                  using
factDateComplete,
                                                  Microsoft.Office.Interop.Word;
StringFormat='{} {0:dd MMMM yyyy} r.',
                                                                                                                                         void
                                                  namespace
                                                                                                     private
ConverterCulture={x:Static
                                                  INCOMSYSTEM.Pages.MainPages
                                                                                                     OpenChatOrderMenu_Click(object sender,
gl:CultureInfo.CurrentCulture},
                                                                                                     RoutedEventArgs
                                                                                                                                           e)
TargetNullValue='Oтсутствует'}">
                                                  public partial class ViewOrdersPage: Page
<TextBlock.Style>
                                                                                                     var
                                                                                                                       order
<Style
                 TargetType="TextBlock"
                                                  public
                                                                          ViewOrdersPage()
                                                                                                     ((MenuItem)sender).CommandParameter as
BasedOn="{StaticResource
                                 {x:Type
                                                                                                     Orders;
TextBlock}}">
                                                  InitializeComponent();
                                                                                                     if (order.Chats.idManager == null &&
<Style.Triggers>
                                                  using
                                                             (var
                                                                       db
                                                                                       new
                                                                                                     MessageBox.Show("Перейти к чату?",
                      Binding="{Binding
                                                  INCOMSYSTEMEntities())
                                                                                                     "Подтверждение",
<DataTrigger
                          Value="True">
IsCompleteLateYellow}"
                                                                                                     MessageBoxButton.YesNo)
                                                                                                                                           !=
<Setter
                  Property="Foreground"
                                                  AllItemsCount.Text
                                                                                  db Orders
                                                                                                     MessageBoxResult.Yes)
                                                                                                                                       return:
Value="Yellow"/>
                                                  .Count(s => MainWindow.AuthUser.idPos
                                                                                                     using
                                                                                                                (var
                                                                                                                          dh
                                                                                                                                         new
                                                          2
                                                                       s.idExecutor
                                                                                                     INCOMSYSTEMEntities())
</DataTrigger>
<DataTrigger
                      Binding="{Binding
                                                  MainWindow.AuthUser.idUser)
IsCompleteLateRed}"
                          Value="True">
                                                  .ToString();
                                                                                                          chat
                                                                                                                      db.Chats.Include(s
                                                                                                     var
                   Property="Foreground"
<Setter
                                                                                                     s.Customers).First(s
                                                                                                                                s.idChat
Value="Red"/>
                                                  ApplyFilter();
                                                                                                     order.id):
</DataTrigger>
                                                                                                     var orderDb = db.Orders.First(s => s.id ==
</Style.Triggers>
                                                  private
                                                                  void
                                                                               ApplyFilter()
</Style>
                                                                                                     if(orderDb.idStatus < 2) orderDb.idStatus =
</TextBlock.Style>
                                                                       db
                                                                                                     2;
                                                  using
                                                             (var
                                                                                       new
</TextBlock>
                                                  INCOMSYSTEMEntities())
                                                                                                     chat.idManager
```

```
MainWindow.AuthUser.idUser;
                                                             employeePatronymic
                                                                                                     s}",
db.SaveChanges();
                                                  order.Employees.patronymic
                                                                                 !=
                                                                                       null
                                                                                                     ((int)order.price).GetNumberInWords().Tri
Application.Current.Windows.OfType<Mai
                                                             {order.Employees.patronymic}"
nWindow>().First().GoChat(chat);
                                                                                                     doc.ReplaceWordText("{customerPhone}",
                                                                              string.Empty;
                                                  doc.ReplaceWordText("{executorName}",
                                                                                                     order.Customers.UsersDetail.phone != null
ApplyFilter();
                                                  $"{order.Employees.surname}
                                                  {order.Employees.name}{employeePatrony
                                                                                                     order.Customers.UsersDetail.phone.Value.
                                    void
                                                                                                     ToString()
private
                                                  mic}"):
FormAgreementOrderMenu Click(object
                                                  doc.ReplaceWordText("{passportCustomer
                                                                                                                              "Отсутствует");
sender,
              RoutedEventArgs
                                                  }", order.Customers.UsersDetail.passport);
                                                                                                     doc.ReplaceWordText("{executorPhone}",
                                      e)
                                                  doc.ReplaceWordText("{passportExecutor}
                                                                                                     order.Employees.UsersDetail.phone != null
{
                  order
                                                      order.Employees.UsersDetail.passport);
((MenuItem)sender).CommandParameter as
                                                  doc.ReplaceWordText("{taskName}",
                                                                                                     order.Employees.UsersDetail.phone.Value.
Orders:
                                                  order.Tasks.name);
                                                                                                     ToString()
var saveFileDialog = new SaveFileDialog
                                                  doc.ReplaceWordText("{planDayStart}",
                                                                                                                              "Отсутствует");
                                                  order.planDateStart.Value.ToString("dd"));
                                                                                                     doc.ReplaceWordText("{customerSeriePas
                                                  doc.ReplaceWordText("{planMonthStart}",
FileName = $"Договор №{order.id}",
Filter = "Documents | *.docx, *.doc",
                                                  order.planDateStart.Value.ToString("MMM\\
                                                                                                     order.Customers.UsersDetail.SeriePassport)
                                                               CultureInfo.CurrentCulture));
DefaultExt
                                 ".docx".
                                                  M".
OverwritePrompt
                                                  doc.ReplaceWordText("{planYearStart}",
                                                                                                     doc.ReplaceWordText("{customerNumber
                                    true
                                                  order.planDateStart.Value.ToString("yy"));
                                                                                                     Passport}",
};
if (saveFileDialog.ShowDialog() != true)
                                                  doc.ReplaceWordText("{planDayComplete
                                                                                                     order.Customers.UsersDetail.NumberPassp
return;
                                                  }",
                                                  order.planDateComplete.Value.ToString("d
                                                                                                     doc.ReplaceWordText("{executorSeriePass
if
(!FormAgreement(saveFileDialog.FileNam
              order))
                                                  doc.ReplaceWordText("{planMonthCompl
                                                                                                     order.Employees.UsersDetail.SeriePassport
if (MessageBox.Show("Договор успешно
                                                  ete}",
сохранён.
              Открыть?",
                             "Успешное
                                                  order.planDateComplete.Value.ToString("
                                                                                                     doc.ReplaceWordText("{executorNumberP
сохранение", MessageBoxButton.YesNo,
                                                  MMMM".
                                                               CultureInfo.CurrentCulture));
                                                                                                     assport}",
MessageBoxImage.Question)
                                                  doc.ReplaceWordText("{planYearComplet
                                                                                                     order.Employees.UsersDetail.NumberPassp
       MessageBoxResult.Yes)
                                  return;
                                                  e}",
                                                                                                     ort);
Process.Start(saveFileDialog.FileName);
                                                  order.planDateComplete.Value.ToString("y
                                                                                                     doc.ReplaceWordText("{customerAddress}
                                                                                                         order.Customers.UsersDetail.address);
                                                  y"));
private static bool FormAgreement(string
                                                  doc.ReplaceWordText("{supportPeriod}",
                                                                                                     doc.ReplaceWordText("{executorAddress}
path,
                Orders
                                  order)
                                                  order.Tasks.supportPeriod);
                                                                                                     ", order.Employees.UsersDetail.address);
                                                  doc.ReplaceWordText("{taskPrice}",
{
                                                                                                     break;
    wApp = new Word.Application();
                                                  (int)order.Tasks.newPrice);
                 wApp.Documents.Open(
                                                  doc.ReplaceWordText("{taskPriceInWords
                                                                                                     catch
Directory.GetCurrentDirectory()
@"\Resources\AgreementTemplate.docx");
                                                  ((int)order.Tasks.newPrice).GetNumberIn
                                                                                                     var result = MessageBox.Show("Для
while
                                   (true)
                                                  Words().Trim());
                                                                                                     сохранения записи, необходимо закрыть
                                                              task Difficulty Price \\
{
                                                                                                     Word
                                                                                                                 документ",
                                                                                                                                   "Ошибка",
                                                  order.Tasks.newPrice
                                                                                                     MessageBoxButton.YesNo,
try
                                                  (decimal)order.difficulty;
                                                                                                     MessageBoxImage.Error);
doc.ReplaceWordText("{dayOrder}",
                                                  doc.ReplaceWordText("{taskDifficultyPric
                                                                                                                  != MessageBoxResult.No)
                                                                                                     if
                                                                                                        (result
order.dateOrder.ToString("dd"));
                                                                    (int)taskDifficultyPrice);
                                                                                                     continue;
                                                  e}".
doc.ReplaceWordText("{monthOrder}",
                                                  doc.ReplaceWordText("{taskDifficultyPric
                                                                                                     doc.Close();
order.dateOrder.ToString("MMMM",
                                                  eInWords}",
                                                                                                     wApp.Quit();
CultureInfo.CurrentCulture));
                                                  ((int)taskDifficultyPrice).GetNumberInWor
                                                                                                                                        false;
                                                                                                     return
doc.ReplaceWordText("{yearOrder}",
                                                  ds().Trim());
order.dateOrder.ToString("yy"));
                                                  doc.ReplaceWordText("{orderPrice}",
doc.ReplaceWordText("{customerName}",
                                                  (int)order.price);
                                                                                                     doc.SaveAs(path);
```

 $doc. Replace Word Text ("\{order Price In Word$

doc.Close();

order.Customers.name);

wApp.Quit();	ViewTasksPage.xaml	Value="Collapsed"/>	
return true;	<page< td=""><td><style.triggers></style.triggers></td></page<>	<style.triggers></style.triggers>	
}	x:Class="INCOMSYSTEM.Pages.MainPag	<datatrigger <="" binding="{Binding</td></tr><tr><td>private void</td><td>es.ViewTasksPage" td=""><td>Source={x:Static</td></datatrigger>	Source={x:Static
SetFactDateStartMenu_Click(object sender,	xmlns="http://schemas.microsoft.com/winf	windows:MainWindow.AuthUser},	
RoutedEventArgs e)	x/2006/xaml/presentation"	Path=idPos}" Value="3">	
{	xmlns:x="http://schemas.microsoft.com/wi	<setter <="" property="Visibility" td=""></setter>	
if (MessageBox.Show("Установить	nfx/2006/xaml"	Value="Visible"/>	
текущую дату начала?",	xmlns:mc="http://schemas.openxmlformats		
"Подтверждение",	.org/markup-compatibility/2006"		
MessageBoxButton.YesNo) !=	xmlns:d="http://schemas.microsoft.com/ex		
MessageBoxResult.Yes) return;	pression/blend/2008"		
var order =	xmlns:windows="clr-	<menuitem <="" header="Добавить" td=""/>	
((MenuItem)sender).CommandParameter as		Click="AddTaskMenu_Click"/>	
	namespace:INCOMSYSTEM.Windows"		
Orders;	xmlns:controls="clr-		
using (var db = new	namespace:INCOMSYSTEM.Controls"	/ItemsControl.ContextMenu>	
INCOMSYSTEMEntities())	xmlns:gl="clr-	<itemscontrol.resources></itemscontrol.resources>	
	namespace:System.Globalization;assembly	<style <="" targettype="TextBlock" td=""></tr><tr><td>var orderDb = db.Orders.First(s => s.id ==</td><td>=mscorlib"</td><td>x:Key="disc" BasedOn="{StaticResource</td></tr><tr><td>order.id);</td><td>mc:Ignorable="d"</td><td>{x:Type TextBlock}}"></td></tr><tr><td>orderDb.factDateStart = DateTime.Now;</td><td>Title="Список задач"></td><td><Setter Property="Visibility"</td></tr><tr><td>orderDb.idStatus = 3;</td><td><Page.Resources></td><td>Value="Visible"/></td></tr><tr><td>db.SaveChanges();</td><td><Style TargetType="TextBlock"</td><td><Style.Triggers></td></tr><tr><td>}</td><td>BasedOn="{StaticResource</td><td><DataTrigger Binding="{Binding</td></tr><tr><td>ApplyFilter();</td><td>TextBlockStyle}"/></td><td>discoutStyle}" Value="True"></td></tr><tr><td>}</td><td><Style TargetType="Label"</td><td><Setter Property="Foreground"</td></tr><tr><td>private void</td><td>BasedOn="{StaticResource LabelStyle}"/></td><td>Value="Yellow"/></td></tr><tr><td>SetFactDateCompleteMenu_Click(object</td><td></Page.Resources></td><td></DataTrigger></td></tr><tr><td>sender, RoutedEventArgs e)</td><td><Grid></td><td><DataTrigger Binding="{Binding</td></tr><tr><td>{</td><td><Grid.RowDefinitions></td><td>discountVisible}" Value="False"></td></tr><tr><td>if (MessageBox.Show("Установить</td><td><RowDefinition Height="Auto"/></td><td><Setter Property="Visibility"</td></tr><tr><td>текущую дату завершения?",</td><td><RowDefinition/></td><td>Value="Collapsed"/></td></tr><tr><td>"Подтверждение",</td><td></Grid.RowDefinitions></td><td></DataTrigger></td></tr><tr><td>MessageBoxButton.YesNo) !=</td><td><Grid.ColumnDefinitions></td><td></Style.Triggers></td></tr><tr><td>MessageBoxResult.Yes) return;</td><td><ColumnDefinition/></td><td></style>	
var order =	<columndefinition width="Auto"></columndefinition>	<style <="" targettype="TextBlock" td=""></tr><tr><td>((MenuItem)sender).CommandParameter as</td><td></Grid.ColumnDefinitions></td><td>x:Key="oldPrice"</td></tr><tr><td>Orders;</td><td><StackPanel Orientation="Horizontal"></td><td>BasedOn="{StaticResource {x:Type</td></tr><tr><td>using (var db = new</td><td><TextBlock Text="{Binding</td><td>TextBlock}}"></td></tr><tr><td>INCOMSYSTEMEntities())</td><td>ElementName=TasksList,</td><td><Style.Triggers></td></tr><tr><td>{</td><td>Path=Items.Count}"/></td><td><DataTrigger Binding="{Binding</td></tr><tr><td>var orderDb = db.Orders.First(s => s.id ==</td><td><TextBlock Text=" из "/></td><td>discountVisible}" Value="True"></td></tr><tr><td>order.id);</td><td><TextBlock x:Name="AllCountTask"/></td><td><Setter Property="Foreground"</td></tr><tr><td>orderDb.factDateComplete =</td><td></StackPanel></td><td>Value="#ccc"/></td></tr><tr><td>DateTime.Now;</td><td><ItemsControl x:Name="TasksList"</td><td><Setter Property="TextDecorations"</td></tr><tr><td>orderDb.idStatus = 4;</td><td>Grid.Row="1"></td><td>Value="Strikethrough"/></td></tr><tr><td>db.SaveChanges();</td><td><ItemsControl.ContextMenu></td><td></DataTrigger></td></tr><tr><td>}</td><td><ContextMenu></td><td></Style.Triggers></td></tr><tr><td>ApplyFilter();</td><td><ContextMenu.Style></td><td></style>	
}	<style <="" targettype="ContextMenu" td=""><td><Style TargetType="Button"</td></tr><tr><td>}</td><td>BasedOn="{StaticResource {x:Type</td><td>x:Key="customerBtn"</td></tr><tr><td>}</td><td>ContextMenu}}"></td><td>BasedOn="{StaticResource {x:Type</td></tr><tr><td></td><td><Setter Property="Visibility"</td><td>Button}}"></td></tr><tr><td></td><td></td><td></td></tr></tbody></table></style>		

<setter <="" property="Visibility" th=""><th></th><th>Style="{Statickesource disc}"</th></setter>		Style="{Statickesource disc}"
Value="Collapsed"/>	<textblock <="" td="" text="{Binding name}"><td>TextWrapping="Wrap"/></td></textblock>	TextWrapping="Wrap"/>
<style.triggers></style.triggers>	Grid.Row="0"	
<datatrigger binding="{Binding</td><td>Style=" td="" textblockstyle}"<="" {staticresource=""><td></td></datatrigger>		
Source={x:Static	FontSize="18"	<stackpanel <="" grid.row="3" td=""></stackpanel>
windows:MainWindow.AuthUser},	FontWeight="Bold"/>	HorizontalAlignment="Left"
Path=idPos}" Value="1">	<textblock 0="" 0"<="" 5="" td="" text="{Binding</td><td>Margin="></textblock>	
<setter <="" property="Visibility" td=""><td>shortDescription}"</td><td>Orientation="Horizontal"></td></setter>	shortDescription}"	Orientation=" Horizontal ">
Value="Visible"/>	TextWrapping=" Wrap "	<button <="" content="Подробнее" td=""></button>
	Style="{StaticResource TextBlockStyle}"	CommandParameter="{Binding
	Grid.Row="1">	RelativeSource={RelativeSource
	<textblock.tooltip></textblock.tooltip>	Mode= Self }, Path=DataContext}"
<style <="" targettype="Button" td=""><td><TextBlock Text="{Binding</td><td>Click="SelectBtn_Click"</td></tr><tr><td>x:Key="managerBtn"</td><td>description}"/></td><td>Style="{StaticResource customerBtn}"/></td></tr><tr><td>BasedOn="{StaticResource {x:Type</td><td>/TextBlock.ToolTip></td><td>Style= (Statier Source Content Plan) / Selection Content Pegaктировать"</td></tr><tr><td>Button}}"></td><td>/TextBlock></td><td>CommandParameter="{Binding</td></tr><tr><td><pre><Setter Property="Visibility"</pre></td><td><StackPanel Grid.Row="2" Margin="0 5"></td><td>RelativeSource={RelativeSource</td></tr><tr><td>* *</td><td>· ·</td><td></td></tr><tr><td>Value="Collapsed"/> <Style Triggers></td><td><TextBlock Text="{Binding</td><td>Mode=Self}, Path=DataContext}"</td></tr><tr><td><Style.Triggers></td><td>Specializations.name,</td><td>Click="EditBtn_Click"</td></tr><tr><td><DataTrigger Binding="{Binding</td><td>StringFormat='{}Область: {0}'}"</td><td>Style="{StaticResource managerBtn}"/></td></tr><tr><td>Source={x:Static</td><td>TextWrapping="Wrap"</td><td><Button Content="Удалить"</td></tr><tr><td>windows:MainWindow.AuthUser},</td><td>Style="{StaticResource TextBlockStyle}"/></td><td>Margin="5 0"</td></tr><tr><td>Path=idPos}" Value="3"></td><td><TextBlock Text="{Binding approxString,</td><td>CommandParameter="{Binding</td></tr><tr><td><Setter Property="Visibility"</td><td>StringFormat='{}Примерные дни</td><td>RelativeSource={RelativeSource</td></tr><tr><td>Value="Visible"/></td><td>выполнения: $\{0\}'\}$"</td><td>Mode=Self}, Path=DataContext}"</td></tr><tr><td></DataTrigger></td><td>Style="{StaticResource TextBlockStyle}"/></td><td>Click="DeleteBtn_Click"</td></tr><tr><td></Style.Triggers></td><td></StackPanel></td><td>Style="{StaticResource managerBtn}"/></td></tr><tr><td></style>	<border <="" grid.column="1" td=""><td></td></border>	
	Grid.Row="0"	
<itemscontrol.itemspanel></itemscontrol.itemspanel>	Grid.RowSpan="4"	
<itemspaneltemplate></itemspaneltemplate>	BorderThickness="1 0 0 0"	
<pre><wrappanel orientation="Horizontal"></wrappanel></pre>	BorderBrush="White"	/ItemsControl.ItemTemplate>
	Margin="5 0 0 0">	<itemscontrol.template></itemscontrol.template>
	<stackpanel< td=""><td><controltemplate></controltemplate></td></stackpanel<>	<controltemplate></controltemplate>
<itemscontrol.itemtemplate></itemscontrol.itemtemplate>	HorizontalAlignment="Center"	<scrollviewer< td=""></scrollviewer<>
<datatemplate></datatemplate>	VerticalAlignment="Center">	VerticalScrollBarVisibility="Auto"
<border <="" td="" width="400"><td><textblock disabled"="" text="{Binding price,</td><td>Horizontal Scroll Bar Visibility = "></textblock></td></border>	<textblock disabled"="" text="{Binding price,</td><td>Horizontal Scroll Bar Visibility = "></textblock>	
Height="160"	StringFormat='{} {0:0,0} pyб.',	<itemspresenter></itemspresenter>
BorderThickness="1"	ConverterCulture={x:Static	
BorderBrush="White"	gl:CultureInfo.CurrentCulture}}"	
CornerRadius="10"	TextWrapping="Wrap"	/ItemsControl.Template>
Padding="5"	Style="{StaticResource oldPrice}"/>	
Margin="5">	<textblock 0"<="" td="" text="{Binding discount,</td><td><Grid Grid.Column="></textblock>	
<grid></grid>	StringFormat='{}{0}%',	Grid.ColumnSpan="2"
<grid.rowdefinitions></grid.rowdefinitions>	TargetNullValue={x:Null}}"	Grid.Row="0"
<rowdefinition height="Auto"></rowdefinition>	TextWrapping="Wrap"	Grid.RowSpan="2"
<rowdefinition></rowdefinition>	Style="{StaticResource disc}"	HorizontalAlignment=" Right ">
	Margin="0 5"/>	<grid.columndefinitions></grid.columndefinitions>
<rowdefinition height="Auto"></rowdefinition>	Margin="0 5"/>	
<rowdefinition height="Auto"></rowdefinition> <rowdefinition height="Auto"></rowdefinition>	<pre><textblock auto"="" text="{Binding newPrice,</pre></td><td><ColumnDefinition Width="></textblock></pre>	
_	<textblock auto"="" text="{Binding newPrice,</td><td><ColumnDefinition Width="></textblock> <columndefinition></columndefinition>	
<rowdefinition height="Auto"></rowdefinition>	<textblock auto"="" text="{Binding newPrice,
StringFormat='{}}{0:0,0} py6.',</td><td></td></tr><tr><td><RowDefinition Height="></textblock>	

 $<\!\!/Grid. Column Definitions\!\!>$

Style="{StaticResource

disc}"

Property="Visibility"

<Setter

<button.content></button.content>	<setter <="" property="IsEnabled" th=""><th>UpdateSourceTrigger=PropertyChanged}</th></setter>	UpdateSourceTrigger= PropertyChanged }		
<pre><image source="//Images/gear.png"/></pre>	Value="False"/>	"		
		TextAlignment="Center"		
		Style="{StaticResource TextBoxStyle}"/>		
<border <="" borderbrush="Coral" td=""><td></td><td></td></border>				
BorderThickness="1 0 0 0"		<stackpanel grid.column="1"></stackpanel>		
Grid.Column="1"	<textblock <="" td="" text="r"><td><textblock <="" td="" text="До"></textblock></td></textblock>	<textblock <="" td="" text="До"></textblock>		
Width="0"	FontFamily="Webdings"/>	TextAlignment="Center"/>		
x:Name="SideBarFilter">		<slider< td=""></slider<>		
<grid></grid>		ValueChanged="Slider_ValueChanged"		
<grid.columndefinitions></grid.columndefinitions>	<combobox <="" td="" x:name="SpecBox"><td>Minimum="{Binding</td></combobox>	Minimum="{Binding		
<columndefinition width="Auto"></columndefinition>	Height="30"	ElementName=LeftSlider, Path=Value,		
<columndefinition></columndefinition>	Margin="0 5"	Mode=TwoWay,		
	SelectedIndex="0"	UpdateSourceTrigger= PropertyChanged }		
<border <="" background="#F0696969" td=""><td>SelectionChanged="FilterBox_SelectionCh</td><td>п</td></border>	SelectionChanged="FilterBox_SelectionCh	п		
Grid.RowSpan="5"	anged">	x:Name="RightSlider"/>		
Grid.ColumnSpan="5"/>	<combobox.itemtemplate></combobox.itemtemplate>	<textbox 30"="" <="" margin="0 5" td="" text="{Binding</td></tr><tr><td><StackPanel></td><td><DataTemplate></td><td>ElementName=RightSlider, Path=Value,</td></tr><tr><td><TextBlock Height="><td><textblock <="" td="" text="{Binding name}"><td>StringFormat='{}{0:0}', Mode=TwoWay,</td></textblock></td></textbox>	<textblock <="" td="" text="{Binding name}"><td>StringFormat='{}{0:0}', Mode=TwoWay,</td></textblock>	StringFormat='{}{0:0}', Mode= TwoWay ,
Text="Фильтр по названию: "/>	Style="{StaticResource	UpdateSourceTrigger= PropertyChanged }		
<textblock <="" height="30" margin="0 5" td=""><td>ComboBoxTextBlockStyle}"/></td><td>"</td></textblock>	ComboBoxTextBlockStyle}"/>	"		
Text="Сортировка по области: "/>		TextAlignment="Center"		
<textblock <="" height="30" margin="0 5" td=""><td></td><td>Style="{StaticResource TextBoxStyle}"/></td></textblock>		Style="{StaticResource TextBoxStyle}"/>		
Text="Copтировка по цене: "/>		<t< td=""></t<>		
<textblock <="" height="30" margin="0 5" td=""><td><combobox <="" td="" x:name="PriceBox"><td></td></combobox></td></textblock>	<combobox <="" td="" x:name="PriceBox"><td></td></combobox>			
Техt="Диапазон цены: "/>	Height="30"			
	Margin="0 5"			
<stackpanel 0"<="" grid.column="1" margin="5</td><td>SelectedIndex=" td=""><td></td></stackpanel>				
0">	SelectionChanged="FilterBox_SelectionCh			
<grid></grid>	anged">			
<grid.columndefinitions></grid.columndefinitions>	<textblock text="По возрастанию"></textblock>			
<columndefinition></columndefinition>	<textblock text="По убыванию"></textblock>			
<columndefinition width="Auto"></columndefinition>		ViewTasksPage.xaml.cs		
	<grid></grid>	using System;		
<controls:inputtextbox< td=""><td><grid.columndefinitions></grid.columndefinitions></td><td>using System.Collections.Generic;</td></controls:inputtextbox<>	<grid.columndefinitions></grid.columndefinitions>	using System.Collections.Generic;		
x:Name="FilterText" Height="30"	<columndefinition></columndefinition>	using System.Data.Entity;		
Width="150" PlaceHolder="Введите	<columndefinition></columndefinition>	using System.Linq;		
текст"		using System.Windows;		
TextChanged="FilterText_TextChanged"/>	<stackpanel></stackpanel>	using System.Windows.Controls;		
<button <="" td="" x:name="ClearBtn"><td><textblock <="" td="" text="Ot"><td>using System.Windows.Media.Animation;</td></textblock></td></button>	<textblock <="" td="" text="Ot"><td>using System.Windows.Media.Animation;</td></textblock>	using System.Windows.Media.Animation;		
Grid.Column="1" Width="25"	TextAlignment="Center"/>	using INCOMSYSTEM.Context;		
VerticalAlignment="Center"	<slider< td=""><td>using INCOMSYSTEM.Pages.Details;</td></slider<>	using INCOMSYSTEM.Pages.Details;		
Click="ClearBtn_Click">	ValueChanged="Slider_ValueChanged"	using		
<button.style></button.style>	Minimum="0"	INCOMSYSTEM.Pages.MainPages.Views;		
<style <="" targettype="Button" td=""><td>x:Name="LeftSlider"</td><td>using INCOMSYSTEM.Windows;</td></tr><tr><td>BasedOn="{StaticResource {x:Type</td><td>Maximum="{Binding</td><td>namespace</td></tr><tr><td>Button}}"></td><td>ElementName=RightSlider, Path=Value,</td><td>INCOMSYSTEM.Pages.MainPages</td></tr><tr><td><Setter Property="IsEnabled"</td><td>Mode=TwoWay,</td><td>{</td></tr><tr><td>Value="True"/></td><td>UpdateSourceTrigger=PropertyChanged}</td><td>public partial class ViewTasksPage</td></tr><tr><td><Style.Triggers></td><td>"/></td><td>{</td></tr><tr><td><pre><DataTrigger Binding="{Binding</pre></td><td><TextBox Text="{Binding</td><td>public ViewTasksPage()</td></tr><tr><td></td><td></td><td>riew i usasi age()</td></tr><tr><td>ElementName=FilterText,</td><td>ElementName=LeftSlider, Path=Value,</td><td>\$</td></tr></tbody></table></style>				

```
using
           (var
                     db
INCOMSYSTEMEntities())
                                                   private void ClearBtn_Click(object sender,
                                                                                                      private void EditBtn_Click(object sender,
                                                   RoutedEventArgs
                                                                                                      RoutedEventArgs
AllCountTask.Text
db.Tasks.Count().ToString();
                                                   FilterText.Clear();
                                                                                                      var
                                                                                                                          task
var max = db.Tasks.Max(s => s.price);
                                                                                                      (Tasks)((Button)sender).CommandParamet
RightSlider.Maximum
                                                                                  _isShowed;
                               (long)max;
                                                   private
                                                                   bool
LeftSlider.Value
                                       0;
                                                   private
                                                                                ApplyFilter()
                                                                                                                addWindow
                                                                   void
                                                                                                      var
                                                                                                                                            new
RightSlider.Value
                               (long)max;
                                                                                                      AdditionalWindow();
TasksList.ItemsSource = db.Tasks.Include(s
                                                   if (PriceBox == null || FilterText == null ||
                                                                                                      addWindow.MFrame.Navigate(new
                                                   TasksList == null || SpecBox == null ||
               s.Specializations).ToList();
                                                                                                      TaskDetailPage(task));
var list = new List<Specializations> { new
                                                   LeftSlider == null || RightSlider == null)
                                                                                                      if (addWindow.ShowDialog() !=
                                                                                                                                           true)
Specializations { name = "Очистить" } };
                                                   return;
                                                                                                      return;
list.AddRange(db.Specializations.ToList());
                                                   using
                                                                        db
                                                                                                      using
                                                                                                                            db
                                                              (var
                                                                                        new
                                                                                                                 (var
                                                                                                                                            new
SpecBox.ItemsSource
                                     list:
                                                   INCOMSYSTEMEntities())
                                                                                                      INCOMSYSTEMEntities())
                                                   IEnumerable<Tasks>
                                                                                                      var max = db.Tasks.Max(s => s.price);
}
                                                                               list
private void SelectBtn_Click(object sender,
                                                   db.Tasks.Include(s => s.Specializations);
                                                                                                      RightSlider.Maximum
                                                                                                                                     (long)max;
                                                   if(!FilterText.IsWhiteSpace)
RoutedEventArgs
                                       e)
                                                   list.Where(s
{
                                                                                                      ApplyFilter();
                                                                                         =>
                                                   s.name.ToLower().Trim().Contains(FilterTe
                   task
var
(Tasks)((Button)sender).CommandParamet
                                                   xt.Text.ToLower().Trim())).ToList();
                                                                                                      private void DeleteBtn_Click(object sender,
                                                         (SpecBox.SelectedIndex
                                                                                                      RoutedEventArgs
                                                   if
MainWindow.ReviewFrame.Navigate(new
                                                   list = list.Where(s => s.idSpecialization ==
ViewDetailTaskPage(task));
                                                   ((Specializations)SpecBox.SelectedItem).id
                                                                                                      if (MessageBox.Show("Вы действительно
                                                   ).ToList();
                                                                                                      хотите
                                                                                                                  удалить
}
                                                                                                                                       задачу?",
                                                                                                      "Подтверждение",
private
                                     void
                                                   switch
                                                                    (PriceBox.SelectedIndex)
FilterBox_SelectionChanged(object sender,
                                                   {
                                                                                                      MessageBoxButton.YesNo,
Selection Changed Event Args\\
                                                                                                      MessageBoxImage.Warning)
                                                   case
                                                                                                                                             !=
                                                                                                      MessageBoxResult.Yes)
                                                   list = list.OrderBy(s => s.discount != null
                                                                                                                                         return:
ApplyFilter();
                                                   && s.discount > 0 ? s.newPrice : s.price);
                                                                                                      var
                                                                                                                          task
                                                                                                      (Tasks)((Button)sender).CommandParamet
                                                   break:
private void Slider_ValueChanged(object
                                                                                          1:
                                                   case
                                                              list.OrderByDescending(s
                                                                                                                           db
sender.
                                                   list
                                                                                                      using
                                                                                                                 (var
                                                                                                                                           new
Routed Property Changed Event Args < double\\
                                                   s.discount != null && s.discount > 0 ?
                                                                                                      INCOMSYSTEMEntities())
                                                   s.newPrice
                                                                                    s.price);
                                                   break;
                                                                                                      db.Tasks.Remove(db.Tasks.First(s => s.id
ApplyFilter();
                                                                                                                                       task.id));
                                                             = (decimal)LeftSlider.Value;
                                                                                                      db.SaveChanges();
private void SettingsFilterBtn_Click(object
                                                   var max = (decimal)RightSlider.Value;
                                                                                                      var max = db.Tasks.Max(s => s.price);
sender.
              RoutedEventArgs
                                                   list = list.Where(s => (s.discount != null &&
                                                                                                      RightSlider.Maximum
                                                                                                                                     (long)max;
                                                   s.discount > 0 ? s.newPrice : s.price) >= min
                                                   && (s.discount != null && s.discount > 0?
                                                                                                      ApplyFilter();
                       DoubleAnimation
var
     anim
                 new
                                                   s newPrice
                                                                     s.price)
                                                                                      max):
                                                   TasksList.ItemsSource =
Duration = TimeSpan.FromSeconds(0.7d),
                                                                                list.ToList():
                                                                                                      private void AddTaskMenu_Click(object
               SideBarFilter.ActualWidth,
                                                                                                                     RoutedEventArgs
From
                                                                                                      sender.
                                                                                                                                             e)
EasingFunction
                 =
                     new
                           SineEase
                                                                                                                addWindow
EasingMode = EasingMode.EaseInOut },
                                                   private void FilterText_TextChanged(object
                                                                                                      var
                                                                                                                                            new
          _isShowed
                       ?
                                                               TextChangedEventArgs
                                                                                                      AdditionalWindow();
                                                   sender.
};
                                                                                                      addWindow.MFrame.Navigate(new
_isShowed
                             !_isShowed;
                                                   if (ClearBtn == null || FilterText == null)
                                                                                                      TaskDetailPage());
SideBarFilter.BeginAnimation(WidthPrope
                                                                                                      if (addWindow.ShowDialog() !=
                                                   return:
                                                                                                                                         true)
                                                   ApplyFilter();
rty,
                                  anim);
                                                                                                      return;
```

```
using (var db = new
INCOMSYSTEMEntities())
{
var max = db.Tasks.Max(s => s.price);
RightSlider.Maximum = (long)max;
}
ApplyFilter();
}
}
```

xmlns:d="http://schemas.microsoft.com/ex BorderThickness="0 0 0 1" Style="{StaticResource pression/blend/2008" Grid.ColumnSpan="3" <textblock grid.row="1" mc:ignorable="d" x:nam=""></textblock> Style="{StaticResource pression/blend/2008" Style="{StaticResource pression/blend/2008" Grid.Row="1"/> Style="{StaticResource pression/blend/2008" Sty	Width="Auto"/> Width="100"/> ns> Name="PriceTask" oldPrice}"/> ne="DiscountTask"
x:Class="INCOMSYSTEM.Pages.MainPag es.Views.ViewDetailTaskPage"	Width="Auto"/> Width="100"/> ns> Name="PriceTask" oldPrice}"/> ne="DiscountTask"
es.Views.ViewDetailTaskPage"	Width="Auto"/> Width="100"/> ns> Name="PriceTask" oldPrice}"/> ne="DiscountTask"
xmlns="http://schemas.microsoft.com/winf x/2006/xaml/presentation" TextAlignment="Center"/> <columndefinition 2006="" presentation"="" textalignment="Center" x="" xaml=""></columndefinition> <columndefinition></columndefinition> xmlns:x="http://schemas.microsoft.com/wi	Width="Auto"/> Width="100"/> ns> Name="PriceTask" oldPrice}"/> ne="DiscountTask"
x/2006/xaml/presentation" TextAlignment="Center"/> xmlns:x="http://schemas.microsoft.com/wi rfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats .org/markup-compatibility/2006" xmlns:d="http://schemas.microsoft.com/ex borderThickness="0 0 0 1" xmlns:d="http://schemas.microsoft.com/ex pression/blend/2008" Grid.ColumnSpan="3" Grid.ColumnSpan="3" TextBlock x:Name="StaticResource Title="Просмотр задачи"> TextBlock x:Name="DescriptionTask" TextBlock x:Name="Style="{StaticResource} Style="{StaticResource} TextWrapping="Wrap" Style="{StaticResource} Style=	Width="100"/> ns> Name="PriceTask" oldPrice}"/> ne="DiscountTask"
x/2006/xaml/presentation" TextAlignment="Center"/> xmlns:x="http://schemas.microsoft.com/wi nfx/2006/xaml" xmlns:mc="http://schemas.openxmlformats .org/markup-compatibility/2006" xmlns:d="http://schemas.microsoft.com/ex border BorderBrush="White" xmlns:d="http://schemas.microsoft.com/ex pression/blend/2008" Grid.ColumnSpan="3" Grid.ColumnSpan="3" TextBlock x:Name="Style="{StaticResource} Title="Просмотр задачи"> TextBlock x:Name="DescriptionTask" TextBlock x:Name="Style="{StaticResource} Style="{StaticResource} TextWrapping="Wrap" Style="{StaticResource} Style="{StaticResourc	Name="PriceTask" oldPrice}"/> ne="DiscountTask"
xmlns:x="http://schemas.microsoft.com/wi <textblock< th="">x:Name="SpecTask"/><columndefinition< th="">nfx/2006/xaml"<!--</td--><td>Name="PriceTask" oldPrice}"/> ne="DiscountTask"</td></columndefinition<></textblock<>	Name="PriceTask" oldPrice}"/> ne="DiscountTask"
nfx/2006/xaml"	Name="PriceTask" oldPrice}"/> ne="DiscountTask"
xmlns:mc="http://schemas.openxmlformats	Name="PriceTask" oldPrice}"/> ne="DiscountTask"
.org/markup-compatibility/2006"	oldPrice}"/> ne="DiscountTask"
xmlns:d="http://schemas.microsoft.com/ex BorderThickness="0 0 0 1" Style="{StaticResource pression/blend/2008" Grid.ColumnSpan="3" <textblock grid.row="1" mc:ignorable="d" x:nam=""></textblock> Style="{StaticResource x:Name="DescriptionTask" <textblock <textblock="" style="{StaticResource x:Name=" supportperiod"}<="" td="" x:name="SupportPeriod" {staticresource="" }}=""><td>oldPrice}"/> ne="DiscountTask"</td></textblock>	oldPrice}"/> ne="DiscountTask"
pression/blend/2008" Grid.ColumnSpan="3" <textblock grid.row="1" mc:ignorable="d" x:name=""></textblock> Style="{StaticResource Title="Просмотр задачи"> <textblock <textblock="" certain="" colo<="" color="" of="" td="" the="" to="" x:name=""><td>ne="DiscountTask"</td></textblock>	ne="DiscountTask"
mc:Ignorable="d" Grid.Row="1"/> Style="{StaticResource of the properties of the pro	
Title="Просмотрзадачи"> <textblock< th="">x:Name="DescriptionTask"<textblock< th="">x:Name=<page.resources>TextWrapping="Wrap"Style="{StaticResource<style< td="">TargetType="TextBlock"Margin="0"10"<textblock< td="">BasedOn="{StaticResourceGrid.Row="1"x:Name="SupportPeriod"</textblock<></style<></page.resources></textblock<></textblock<>	disc}"/>
<page.resources> TextWrapping="Wrap" Style="{StaticResource <style< td=""> TargetType="TextBlock" Margin="0 10" <textblock< td=""> BasedOn="{StaticResource Grid.Row="1" x:Name="SupportPeriod"</textblock<></style<></page.resources>	e="NewPriceTask"
<pre> <style 1"="" <="" <textblock="" basedon="{StaticResource Grid.Row=" margin="0 10" pre="" targettype="TextBlock" x:name="SupportPeriod"></td><td></td></tr><tr><td>BasedOn="{StaticResource Grid.Row="1" x:Name="SupportPeriod"</td><td>disc}"/></td></tr><tr><td></td><td>T 1 11 /</td></tr><tr><td>TextBlockStyle}"/> Text="{Binding Description}"/> </StackPanel></td><td>Task"/></td></tr><tr><td></td><td></td></tr><tr><td><Style TargetType="Label"</td><td>Grid.Column="2"</td></tr><tr><td>BasedOn="{StaticResource LabelStyle}"/></td><td>ock"</td></tr><tr><td></Page.Resources> <Style TargetType="TextBlock" VerticalAlignment="Cen</td><td>ıter"></td></tr><tr><td><Grid VerticalAlignment="Center" x:Key="disc" BasedOn="{StaticResource <TextBlock</td><td>Text="Вложение"</td></tr><tr><td>HorizontalAlignment="Center" {x:Type TextBlock}}"> TextAlignment="Center"</td><td>'"/></td></tr><tr><td>Width="400"> <Setter Property="Visibility" <Button</td><td>Content="Скачать"</td></tr><tr><td><Grid.RowDefinitions> Value="Visible"/> Margin="5</td><td>2"</td></tr><tr><td><RowDefinition Height="Auto"/> <Style.Triggers> Click="DownloadAttached"</td><td>mentBtn_Click"/></td></tr><tr><td><RowDefinition/> <DataTrigger Binding="{Binding </StackPanel></td><td></td></tr><tr><td><RowDefinition Height="Auto"/> ElementName=discountStyle, </Grid></td><td></td></tr><tr><td></Grid.RowDefinitions> Path=IsChecked}" Value="True"> <TextBlock</td><td></td></tr><tr><td><TextBlock FontSize="20" <Setter Property="Foreground" x:Name="ApproxCompleted and the complete in the complet</td><td>eteTime"</td></tr><tr><td>x:Name="TitleTask" Value="Yellow"/> Grid.Row="3"/></td><td></td></tr><tr><td>TextAlignment="Center"/> </DataTrigger> </Grid></td><td></td></tr><tr><td><Border BorderBrush="Coral" <DataTrigger Binding="{Binding} </Border></td><td></td></tr><tr><td>BorderThickness="1" ElementName=discountVisible, <StackPanel</td><td>Grid.Row="2"</td></tr><tr><td>Padding="5"</td><td>I"</td></tr><tr><td>Grid.Row="1"> <Setter Property="Visibility" HorizontalAlignment="R</td><td>Right"></td></tr><tr><td><Grid> Value="Collapsed"/> <Button C</td><td>Content="Заказать"</td></tr><tr><td><pre><Grid.RowDefinitions></pre> </pre> <pre></pre> <pre></td><td>10"</td></tr><tr><td><pre><RowDefinition Height="Auto"/> </pre> <pre></pre> <pre></pre> <pre>Click="MakeOrderBtn_C</pre></td><td>Click"/></td></tr><tr><td><RowDefinition/> </style> </pre>	
<pre><rowdefinition height="Auto"></rowdefinition> <style <="" pre="" targettype="TextBlock"></pre></td><td></td></tr><tr><td><pre><RowDefinition Height="Auto"/> x:Key="oldPrice" </pre> <pre>/Page></pre></td><td></td></tr><tr><td><RowDefinition</p> Height="Auto"/> BasedOn="{StaticResource {x:Type}</td><td></td></tr><tr><td><pre></pre> <pre></pre> <pre></pre> <pre></pre> <pre> </pre> <pre> </pre> <pre> </pre> <pre> </pre> <pre> </pre> <pre> <pre> </pre> <pre> </pre> <pre> </pre> <pre> </pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> </pre> <pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> <pre> </pre> <pre> <pre> </pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> </pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre</td><td>nl es</td></tr><tr><td>,,</td><td>ISYSTEM.Context;</td></tr><tr><td><pre><CheckBox Visibility="Collapsed" <Style.Triggers> using INCOM</pre></td><td>Microsoft.Win32;</td></tr><tr><td>v.Name="discountStyle"/></td><td></td></tr><tr><td>x:Name="discountStyle"/></td><td>System;</td></tr><tr><td><CheckBox Visibility="Collapsed" ElementName=discountStyle, using</td><td>stom Clobali:-</td></tr><tr><td><pre><CheckBox Visibility="Collapsed"</td><td>stem.Globalization;</td></tr><tr><td><pre><CheckBox Visibility="Collapsed"</td><td>System.IO;</td></tr><tr><td><CheckBox</td> Visibility="Collapsed" ElementName=discountStyle, using x:Name="discountVisible"/> Path=IsChecked}" Value="True"> using <Grid> <Setter</td> Property="Foreground" using <Grid.ColumnDefinitions> Value="#ccc"/> using</td><td></td></tr></tbody></table></style></pre>	

Value="Strikethrough"/>

IN COMSYSTEM. Behaviors Files;

using

 $<\!\!Column Definition$

Width = "Auto"/>

using INCOMSYSTEM.Windows;	FileStream(saveFileDialog.FileName,	xmlns="http://schemas.microsoft.com/winf		
namespace	FileMode.OpenOrCreate,	x/2006/xaml/presentation"		
INCOMSYSTEM.Pages.MainPages.Views	FileAccess.Write))	xmlns:x="http://schemas.microsoft.com/wi		
{	{	nfx/2006/xaml"		
public partial class ViewDetailTaskPage :	file.Write(_task.attachment, 0,	xmlns:mc="http://schemas.openxmlformats		
Page	_task.attachment.Length);	.org/markup-compatibility/2006"		
{	file.Dispose();	xmlns:d="http://schemas.microsoft.com/ex		
public ViewDetailTaskPage(Tasks task)	}	pression/blend/2008"		
{	}	xmlns:controls="clr-		
InitializeComponent();	private void MakeOrderBtn_Click(object	namespace:INCOMSYSTEM.Controls"		
discountStyle.IsChecked =	sender, RoutedEventArgs e)	xmlns:sys="clr-		
task.discoutStyle;	{	namespace:System;assembly=mscorlib"		
discountVisible.IsChecked =	if (MessageBox.Show("Вы уверены с	mc:Ignorable="d">		
task.discountVisible;	заказом?", "Подтверждение",	<grid></grid>		
TitleTask.Text = task.name;	MessageBoxButton. YesNo,	<grid.rowdefinitions></grid.rowdefinitions>		
SpecTask.Text = task.name,	· ·	<rowdefinition height="Auto"></rowdefinition>		
*		<rowdefinition></rowdefinition>		
\$"{task.Specializations.name}";	,			
DescriptionTask.Text = task.description;	var order = new Orders	<rowdefinition height="Auto"></rowdefinition>		
PriceTask.Text = \$"Цена:	{			
{task.price.ToString("#,#",	idCustomer =	<stackpanel< td=""></stackpanel<>		
CultureInfo.CurrentCulture)} pyő.";	MainWindow.AuthUser.idUser,	HorizontalAlignment="Right">		
DiscountTask.Text = \$"Скидка:	idTask = _task.id,	<stackpanel <="" orientation="Horizontal" td=""></stackpanel>		
{task.discount}%";	difficulty = 1f,	HorizontalAlignment=" Right ">		
NewPriceTask.Text = \$"Новая цена:	dateOrder = DateTime.Now,	<textblock ;<="" td="" text="Дата формирования</td></tr><tr><td>{task.newPrice:0.00}"><td>idStatus = 1</td><td>заказа: " Style="{StaticResource</td></textblock>	idStatus = 1	заказа: " Style="{StaticResource
SupportPeriodTask.Text = \$"Период	} ;	TextBlockStyle}"/>		
поддержки:	order.price = _task.price *	<textblock <="" td="" x:name="DateOrderBlock"></textblock>		
{task.supportPeriod.ConvertDay()}";	(decimal)order.difficulty;	Style="{StaticResource TextBlockStyle}"/>		
ApproxCompleteTime.Text =	var chat = new Chats			
\$"Примерное время выполнения	{	<pre><stackpanel <="" orientation="Horizontal" pre=""></stackpanel></pre>		
{task.approxCompleteTime.ConvertDay()}	idChat = order.id,	HorizontalAlignment=" Right ">		
",	idCustomer =	<textblock <="" td="" text="Статус заказа: "></textblock>		
AttachmentBlock.Visibility =	MainWindow.AuthUser.idUser,	Style="{StaticResource TextBlockStyle}"/>		
task.attachment == null ?	dateCreate = DateTime.Now	<textblock <="" td="" x:name="StatusOrderBlock"></textblock>		
Visibility. Collapsed : Visibility. Visible;	};	Style="{StaticResource TextBlockStyle}"/>		
_task = task;	using (var db = new			
}	INCOMSYSTEMEntities())			
private readonly Tasks _task;	{	<grid <="" grid.row="1" td=""></grid>		
private void	db.Orders.Add(order);	HorizontalAlignment="Center"		
DownloadAttachmentBtn_Click(object	db.Chats.Add(chat);	VerticalAlignment="Center">		
sender, RoutedEventArgs e)	db.SaveChanges();	<grid.columndefinitions></grid.columndefinitions>		
{	}	<columndefinition width="Auto"></columndefinition>		
var saveFileDialog = new SaveFileDialog	MessageBox.Show("Заказ успешно	<columndefinition></columndefinition>		
{	сформирован!");			
Title = "Скачивание файла",	MainWindow.ReviewFrame.GoBack();	<stackpanel></stackpanel>		
CreatePrompt = true,	}	<stackpanel.resources></stackpanel.resources>		
FileName =	}	<style <="" targettype="TextBlock" td=""></tr><tr><td>\$"{_task.name}.{_task.fileExtension}",</td><td>}</td><td>BasedOn="{StaticResource</td></tr><tr><td>Filter = \$"File *.{_task.fileExtension}"</td><td></td><td>TextBlockStyle}"></td></tr><tr><td>};</td><td>ViewDetailOrderPage.xaml</td><td><Setter Property="Height" Value="30"/></td></tr><tr><td>if (saveFileDialog.ShowDialog() != true)</td><td><Page</td><td><pre><Setter Property="TextWrapping"</pre></td></tr><tr><td>return;</td><td>x:Class="INCOMSYSTEM.Pages.MainPag</td><td>Value="Wrap"/></td></tr><tr><td>using(var file = new</td><td>es.Views.ViewDetailOrderPage"</td><td><Setter Property="Margin" Value="0 5 0</td></tr><tr><td></td><td>5</td><td></td></tr></tbody></table></style>		

0"/>	IsReadOnly="True"/>	x:Name="FileDownload"
	<combobox x:name="ExecutorBox"></combobox>	Click="FileDownloadBtn_Click"
	<combobox.itemtemplate></combobox.itemtemplate>	VerticalAlignment="Center"
<textblock text="Заказчик"></textblock>	<datatemplate></datatemplate>	HorizontalContentAlignment="Center"
<textblock <="" td="" text="Исполнитель"><td><textblock false"="" style="{StaticResource</td><td>IsEnabled="></textblock></td></textblock>	<textblock false"="" style="{StaticResource</td><td>IsEnabled="></textblock>	
Margin="0 5 0 0"/>	ComboBoxTextBlockStyle}">	<stackpanel grid.column="1"></stackpanel>
<textblock text="Менеджер"></textblock>	<textblock.text></textblock.text>	<button <="" content="Загрузить" td=""></button>
TextBlock Text="Lena"/>	<multibinding stringformat="{}{0} {1}</td><td>Click=" td="" uploadbtn_click"<=""></multibinding>	
<textblock text="Сложность"></textblock>	{2}">	x:Name="UploadBtn"/>
<textblock employees.surname"="" техt="Плановая дата</td><td><Binding Path="></textblock>	<button <="" content="Очистить" p=""></button>	
начала"/>	<binding path="Employees.name"></binding>	IsEnabled="False" Click="ClearBtn_Click"
	<binding path="Employees.name"></binding> <binding path="Employees.patronymic"></binding>	x:Name="ClearBtn"/>
<ТехtВlоск Техt="Плановая дата выполнения"/>		<button <="" content="Вернуть" td=""></button>
SEINOTHERINA /> <textblock <="" p="" text="Вложение"></textblock>		IsEnabled="False"
		Click="ReturnBtn_Click"
Height="85"/>		x:Name="ReturnBtn"/>
<pre><stackpanel "175"="" "5="" 0"="" <="" grid.column="1" magazin="" pre="" width=""></stackpanel></pre>		
Width="175" Margin="5 0">		
<stackpanel.resources></stackpanel.resources>	<controls:inputtextbox< td=""><td></td></controls:inputtextbox<>	
<style< td=""><td>x:Name="ExecutorBlock"</td><td></td></style<>	x:Name="ExecutorBlock"	
TargetType="controls:InputTextBox"	Visibility="Collapsed"	<stackpanel <="" grid.row="2" td=""></stackpanel>
BasedOn="{StaticResource {x:Type	IsReadOnly="True"/>	Orientation="Horizontal"
controls:InputTextBox}}">	<controls:inputtextbox< td=""><td>HorizontalAlignment="Center"</td></controls:inputtextbox<>	HorizontalAlignment="Center"
<setter property="Height" value="30"></setter>	x:Name="ManagerBlock"	Margin="0 5">
<setter property="Margin" true"="" value="0 5 0</td><td>IsReadOnly="></setter>	<button <="" td="" x:name="SaveBtn"></button>	
0"/>	<controls:inputtextbox< td=""><td>Content="Сохранить"</td></controls:inputtextbox<>	Content="Сохранить"
	x:Name="PriceBox" IsReadOnly="True"/>	Click="SaveBtn_Click"/>
<style <="" targettype="ComboBox" td=""><td><controls:InputTextBox</td><td><Button x:Name="CancelBtn"</td></tr><tr><td>BasedOn="{StaticResource {x:Type</td><td>x:Name="DifficultyBox"</td><td>Content="Отменить" Margin="5 0"</td></tr><tr><td>ComboBox}}"></td><td>TextChanged="DifficultyBox_TextChange</td><td>Click="CancelBtn_Click"/></td></tr><tr><td><Setter Property="Height" Value="30"/></td><td>d"/></td><td></StackPanel></td></tr><tr><td><Setter Property="Margin" Value="0 5 0</td><td><controls:DatePickerEx</td><td></Grid></td></tr><tr><td>0"/></td><td>x:Name="PlanDateStartBox"</td><td></Page></td></tr><tr><td></style>	DisplayDateStart="{x:Static	
<style< td=""><td>sys:DateTime.Today}"</td><td>ViewDetailOrderPage.xaml.cs</td></style<>	sys:DateTime.Today}"	ViewDetailOrderPage.xaml.cs
TargetType="controls:DatePickerEx"	$Selected Date Changed = "Date Box_Selected"$	using System.Linq;
BasedOn="{StaticResource {x:Type	DateChanged"/>	using System.Windows.Controls;
controls:DatePickerEx}}">	<controls:datepickerex< td=""><td>using INCOMSYSTEM.Context;</td></controls:datepickerex<>	using INCOMSYSTEM.Context;
<setter <="" property="BorderBrush" td=""><td>x:Name="PlanDateCompleteBox"</td><td>using System.Data.Entity;</td></setter>	x:Name="PlanDateCompleteBox"	using System.Data.Entity;
Value="White"/>	DisplayDateStart="{x:Static	using System.Globalization;
<style.triggers></style.triggers>	sys:DateTime.Today}"	using System.Windows;
<datatrigger binding="{Binding</td><td><math display=" block"="">Selected Date Changed = "Date Box_Selected"</datatrigger>	using Microsoft.Win32;	
RelativeSource={RelativeSource	DateChanged"/>	using System.IO;
Mode= Self }, Path=IsWrong}"	<grid <="" height="100" td=""><td>using System;</td></grid>	using System;
Value="True">	Drop="FileDownload_OnDrop"	using INCOMSYSTEM.Windows;
<setter <="" property="BorderBrush" td=""><td>AllowDrop="True"</td><td>namespace</td></setter>	AllowDrop="True"	namespace
Value="Red"/>	x:Name="FileDownloadGrid">	INCOMSYSTEM.Pages.MainPages.Views
	<grid.columndefinitions></grid.columndefinitions>	{
	<columndefinition></columndefinition>	public partial class ViewDetailOrderPage :
	<columndefinition width="Auto"></columndefinition>	Page
•		(
		{
<controls:inputtextbox< td=""><td> </td></controls:inputtextbox<>		

```
order;
                                                    CancelBtn.Content
                                                                                    "Закрыть";
                                                                                                         FileDownload.IsEnabled
                                                                                                                                              false;
order
                                                                                                         ReturnBtn.Visibility = Visibility.Collapsed;
InitializeComponent();
                                                    }
Title
               $"Подробности
                                                    private
                                                             void SetInputBoxValues(Orders
                                    заказа
№{order.id}";
                                                                                                         else
                                                    order)
_fileOrder
                                                    {
                         order.attachment:
_fileOrderExtension = order.fileExtension;
                                                    CustomerBlock.Text
                                                                                                         FileDownload.Content = $"Дополнение к
SetInputBoxValues(order);
                                                    $"{order.Customers.name}";
                                                                                                         договору. {order.fileExtension}";
SetFileValues(order);
                                                    PriceBox.Text
                                                                                                         FileDownload.IsEnabled
                                                                                                                                              true:
if (order.idStatus >= 4 || order.factDateStart
                                                    $"{order.price.ToString("#,#",
                                                                                          new
                                                                                                         TempFile
                                                                                                                                        fileOrder;
!= null || MainWindow.AuthUser.idPos ==
                                                    CultureInfo("ru-RU"))}
                                                                                        руб.";
                                                                                                         TempFileExtension = _fileOrderExtension;
2)
                                                    DifficultyBox.Text
                                                                                                         ClearBtn.IsEnabled
                                                                                                                                              true;
                                                    order.difficulty.ToString(CultureInfo.Invari
ChangeExecutorBox();
                                                    antCulture);
DisableBoxes();
                                                         (order.Chats.Employees
                                                                                                         private
                                                                                                                    readonly
                                                                                                                                 Orders
                                                                                                                                            _order;
return;
                                                                                                         private string _fileName = "Дополнение к
                                                    ManagerBlock.Text
                                                                                                         договору";
                                                                                                         private
                     db
                                                    $"{order.Chats.Employees.surname}
                                                                                                                                         FileName
using
           (var
                                      new
                                                                                                                          string
INCOMSYSTEMEntities())
                                                    {order.Chats.Employees.name}";
                                                    ManagerBlock.Text
                                                                                                         get
                                                                                                                                        fileName:
ExecutorBox.ItemsSource
                                                    order.Chats.Employees.patronymic != null
                                                                                                         set
db.Employees.SelectMany(s
                                                    ? $" {order.Chats.Employees.patronymic}"
                                       =>
s.SpecializationsEmployee)
                                                                                                         FileDownload.IsEnabled
.Include(s => s.Employees.UsersDetail)
                                                                                                         !string.IsNullOrWhiteSpace(value);
                                                    }
.Where(s => s.Employees.UsersDetail.idPos
                                                    if
                                                             (order.idStatus
                                                                                            3)
                                                                                                         _fileName
                                                                                                                                             value;
                   2
                                      &&
                                                                                                         FileDownload.Content
                                                                                                                                             value;
s.idSpecialization
                                                    ExecutorBlock.Text
order.Tasks.idSpecialization).ToList();
                                                    $"{order.Employees.surname}
if
       (order.idExecutor
                                     null)
                                                    {order.Employees.name}";
                                                                                                         private
                                                                                                                   readonly
                                                                                                                              byte[]
                                                                                                                                        fileOrder;
ExecutorBox.SelectedItem
                                                    ExecutorBlock.Text
                                                                                                         private readonly string _fileOrderExtension;
ExecutorBox.ItemsSource.Cast<Specializat
                                                                                                         private byte[] TempFile { get; set; }
                                                    order.Employees.patronymic
                                                                                          null
ionsEmployee>()
                                                               {order.Employees.patronymic}"
                                                                                                         private string TempFileExtension { get; set;
.First(s
         =>
               s.Employees.idUser
order.idExecutor);
                                                                                                                           UploadBtn_Click(object
                                                    }
                                                                                                         private
                                                                                                                   void
                                                    if(order.factDateStart
                                                                                          null)
                                                                                                         sender,
                                                                                                                        RoutedEventArgs
}
}
                    ChangeExecutorBox()
                                                    PlanDateStartBox.IsHitTestVisible = false;
                                                                                                              openFile
                                                                                                                       = new OpenFileDialog
private
           void
                                                    PlanDateCompleteBox.IsHitTestVisible =
ExecutorBox. Visibility
                                                    false;
                                                                                                         Title
                                                                                                                           "Загрузка
                                                                                                                                           файла",
                                                                                                         Filter = "Document | *.docx; *.doc | Portable
Visibility.Collapsed;
ExecutorBlock. Visibility
                                                    PlanDateStartBox. SelectedDate \\
                                                                                                         Document
                                                                                                                                            *.pdf",
Visibility. Visible;
                                                                                                         DefaultExt
                                                    order.planDateStart;
                                                                                                                                            ".docx"
                                                    PlanDate Complete Box. Selected Date \\
private
              void
                           DisableBoxes()
                                                    order.planDateComplete;
                                                                                                         if (openFile.ShowDialog() != true) return;
                                                    DateOrderBlock.Text
{
                                                                                                         TempFile
ExecutorBox. Visibility
                                                    order.dateOrder.ToString("dd
                                                                                                         File.ReadAllBytes(openFile.FileName);
                                                                                     MMMM
Visibility. Collapsed;
                                                                   CultureInfo.CurrentCulture);
                                                                                                         TempFileExtension
                                                    уууу",
ExecutorBlock.Visibility
                                                    StatusOrderBlock.Text
                                                                                                         openFile.SafeFileName.Split('.').Last();
Visibility. Visible;
                                                    order.Statuses.name;
                                                                                                                              $"Дополнение
                                                                                                                                                 к
DifficultyBox.IsReadOnly
                                                                                                         договору. {TempFileExtension}";
                                     true:
FileDownloadGrid.AllowDrop
                                     false;
                                                    private void SetFileValues(Orders order)
                                                                                                         ClearBtn.IsEnabled
                                                                                                                                              true;
UploadBtn.IsEnabled
                                     false;
                                                                                                         ReturnBtn.IsEnabled
                                                                                                                                              true;
                                                    if
ClearBtn.IsEnabled
                                     false:
                                                             (_fileOrder
                                                                                          null)
SaveBtn.Visibility = Visibility.Collapsed;
                                                    {
                                                                                                         private void ClearBtn_Click(object sender,
```

```
RoutedEventArgs
                                                                                                       byte?
                                                                                                                                            null;
                                       e)
                                                                                                                    status
{
                                                    AdditionalWindow.ShowError("Не верный
                                                                                                       if (_order.factDateStart
                                                                                                                                  !=
                                                                                                                                      null
                                                                                                                                             &&
TempFile
                                     null:
                                                   формат
                                                                                     файла");
                                                                                                       _order.factDateComplete != null) status = 4;
TempFileExtension
                                                   return;
                                                                                                       else if ((_order.factDateStart != null &&
                                    null;
FileName
                                                                                                                                            null)
                            string.Empty;
                                                                                                       _order.factDateComplete
                                                                                                       || (PlanDateStartBox.SelectedDate != null
ClearBtn.IsEnabled
                                    false;
                                                   TempFile
                                                                      File.ReadAllBytes(file);
                                                                =
ReturnBtn.IsEnabled
                                                   TempFileExtension
                                                                                fileExtension;
                                                                                                       && PlanDateCompleteBox.SelectedDate !=
                                    true:
                                                    ClearBtn.IsEnabled
                                                                                                       null))
                                                                                                                     status
                                                                                                                                              3;
                                                                                        true:
private void ReturnBtn_Click(object sender,
                                                    ReturnBtn.IsEnabled
                                                                                        true;
                                                                                                       using
                                                                                                                  (var
                                                                                                                            db
                                                                                                                                            new
                                                                                                       INCOMSYSTEMEntities())
RoutedEventArgs
                                                    FileName
                                                                        $"Дополнение
                                                   договору. {TempFileExtension}";
TempFile
                               _fileOrder;
                                                    AdditionalWindow.HideError();
                                                                                                       var order = db.Orders.First(s => s.id ==
TempFileExtension = _fileOrderExtension;
                                                                                                       _order.id);
FileName
                    $"Дополнение
                                                    private
                                                                                         void
                                                                                                       order.idExecutor
договору. {TempFileExtension}";
                                                   DifficultyBox_TextChanged(object sender,
                                                                                                       ((SpecializationsEmployee)ExecutorBox.Se
ReturnBtn.IsEnabled
                                                   TextChangedEventArgs
                                                                                                       lectedItem)?.idEmployee;
                                    false;
ClearBtn.IsEnabled
                                    true:
                                                                                                       order.price
                                                   if
}
                                                                                                       Math.Round(_order.Tasks.newPrice
private void FileDownloadBtn_Click(object
                                                   (!decimal.TryParse(DifficultyBox.Text.Rep
                                                                                                       difficulty);
                                                                                                                               (double)difficulty;
               RoutedEventArgs
                                                   lace(',',
                                                                                                       order.difficulty
sender.
                                                                                          '.').
                                                   NumberStyles. Allow Decimal Point,
                                                                                                       order.planDateStart
if
     (TempFile
                          null)
                                  return;
                                                   CultureInfo.InvariantCulture,
                                                                                                       PlanDateStartBox.SelectedDate;
                                                                                          var
var saveFileDialog = new SaveFileDialog
                                                   difficulty))
                                                                                                       order.planDateComplete
                                                                                                       PlanDateCompleteBox.SelectedDate;
{
Title
                "Скачивание
                                  файла",
                                                    AdditionalWindow.ShowError("He
                                                                                                       if (status != null) order.idStatus
FileName
                          $"{FileName}",
                                                   удалось
                                                             преобразовать
                                                                               коэффициент
                                                                                                       status. Value;
Filter = $"File | * {TempFileExtension}",
                                                   сложности
                                                                                    строку");
                                                                                                       order.attachment
                                                                                                                                       TempFile;
DefaultExt = $".{TempFileExtension}"
                                                   return;
                                                                                                       order.fileExtension = TempFileExtension;
                                                                                                       db.SaveChanges();
                                                                   _order.Tasks.newPrice *
if (saveFileDialog.ShowDialog() != true)
                                                    var price =
return:
                                                   difficulty:
                    file
                                                   PriceBox.Text
                                                                                                       AdditionalWindow.HideError():
using
           (var
FileStream(saveFileDialog.FileName,
                                                   Math.Round(price).ToString("#,#",
                                                                                                                       addWindow
                                                                                         new
FileMode.OpenOrCreate,
                                                    CultureInfo("ru-RU"))
                                                                                       руб.";
                                                                                                       Application.Current.Windows.OfType<Ad
FileAccess.Write))
                                                    AdditionalWindow.HideError();
                                                                                                       ditionalWindow>().First();
                                                                                                       addWindow.DialogResult
                                                                                                                                            true:
file.Write(TempFile, 0, TempFile.Length);
                                                   private void SaveBtn_Click(object sender,
                                                                                                       addWindow.Close();
                                                   RoutedEventArgs
                                                    {
                                                                                                       private void CancelBtn_Click(object sender,
                                                   if
private void FileDownload_OnDrop(object
                                                                                                       RoutedEventArgs
sender.
               DragEventArgs
                                                   (!decimal.TryParse(DifficultyBox.Text.Rep
                                                                                          '.'),
                                                                                                       if
{
                                                   lace(',',
                                                                                                                                     (!IsEquals()
if
                                                   NumberStyles. Allow Decimal Point,
                                                                                                       &&
                                                                                                                   MessageBox.Show("Отменить
(!e.Data.GetDataPresent(DataFormats.File
                                                   CultureInfo.InvariantCulture.
                                                                                                       внесённые
                                                                                                                                   изменения?".
                                                                                          var
                                                   difficulty))
Drop))
                                                                                                       "Подтверждение",
                                   return:
                    file
                                                                                                       MessageBoxButton.YesNo,
var
((string[])e.Data.GetData(DataFormats.File
                                                    AdditionalWindow.ShowError("He
                                                                                                       MessageBoxImage.Warning)
                                                                                                       MessageBoxResult.No)
Drop))?[0];
                                                   удалось преобразовать
                                                                               коэффициент
                                                                                                                                          return:
if
       (file
                                                                                                                       addWindow
                         null)
                                                    сложности
                                                                                    строку");
     fileExtension = file.Split('.').Last();
                                                   return;
                                                                                                       Application.Current.Windows.OfType<Ad
     (fileExtension
                      !=
                           "docx"
                                                                                                       ditionalWindow>().First();
fileExtension != "doc" && fileExtension !=
                                                   if
                                                                                 (!IsEquals())
                                                                                                       addWindow.DialogResult
                                                                                                                                           false:
"pdf")
                                                    {
                                                                                                       addWindow.Close();
```

}	<page.resources></page.resources>	<setter property="Height" value="30"></setter>		
private bool IsEquals()	<style <="" targettype="TextBlock" td=""><td><pre><Setter Property="Margin" Value="0 5 0</pre></td></tr><tr><td>{</td><td>BasedOn="{StaticResource</td><td>0"/></td></tr><tr><td>if (_order.idStatus >= 3) return true;</td><td>TextBlockStyle}"/></td><td></style>			
return	<style <="" targettype="Label" td=""><td></StackPanel.Resources></td></tr><tr><td>((SpecializationsEmployee)ExecutorBox.Se</td><td>BasedOn="{StaticResource LabelStyle}"/></td><td><pre><controls:InputTextBox</pre></td></tr><tr><td>lectedItem)?.idEmployee ==</td><td></Page.Resources></td><td>PlaceHolder="Название"</td></tr><tr><td>_order.idExecutor</td><td><Grid HorizontalAlignment="Center"</td><td>x:Name="NameBox"/></td></tr><tr><td>&& DifficultyBox.Text ==</td><td>VerticalAlignment="Center"></td><td><ComboBox x:Name="SpecBox"</td></tr><tr><td>_order.difficulty.ToString(CultureInfo.Inva</td><td><Grid.RowDefinitions></td><td>Height="30" Margin="0 10 0 0"</td></tr><tr><td>riantCulture)</td><td><RowDefinition/></td><td>SelectedIndex="0"></td></tr><tr><td>&& PlanDateStartBox.SelectedDate ==</td><td><RowDefinition Height="Auto"/></td><td><ComboBox.ItemTemplate></td></tr><tr><td>_order.planDateStart</td><td></Grid.RowDefinitions></td><td><DataTemplate></td></tr><tr><td>&& PlanDateCompleteBox.SelectedDate</td><td><Grid.ColumnDefinitions></td><td><TextBlock Style="{StaticResource</td></tr><tr><td>== _order.planDateComplete</td><td><ColumnDefinition Width="150"/></td><td>ComboBoxTextBlockStyle}"</td></tr><tr><td>&& (fileOrder == TempFile &&</td><td><ColumnDefinition Width="200"/></td><td>Text="{Binding name}"/></td></tr><tr><td>_fileOrderExtension ==</td><td><ColumnDefinition Width="30"/></td><td></DataTemplate></td></tr><tr><td>TempFileExtension);</td><td></Grid.ColumnDefinitions></td><td></ComboBox.ItemTemplate></td></tr><tr><td>}</td><td><StackPanel></td><td></ComboBox></td></tr><tr><td>private void</td><td><StackPanel.Resources></td><td><controls:InputTextBox</td></tr><tr><td>DateBox_SelectedDateChanged(object</td><td><Style TargetType="TextBlock"</td><td>PlaceHolder="Описание"</td></tr><tr><td>sender, SelectionChangedEventArgs e)</td><td>BasedOn="{StaticResource</td><td>x:Name="DescriptionBox"/></td></tr><tr><td>{</td><td>TextBlockStyle}"></td><td><pre><controls:InputTextBox</pre></td></tr><tr><td>PlanDateStartBox.IsWrong =</td><td><pre><Setter Property="Height" Value="30"/></pre></td><td>PlaceHolder="Цена" x:Name="PriceBox"</td></tr><tr><td>PlanDateStartBox.SelectedDate != null &&</td><td><pre><Setter Property="TextWrapping"</pre></td><td>TextChanged="PriceBox_OnTextChanged"</td></tr><tr><td>PlanDateCompleteBox.SelectedDate != null</td><td>Value="Wrap"/></td><td>/></td></tr><tr><td>&& PlanDateStartBox.SelectedDate.Value</td><td></style>	<pre><controls:inputtextbox< pre=""></controls:inputtextbox<></pre>		
>		PlaceHolder="Скилка"		
PlanDateCompleteBox.SelectedDate.Value;	<textblock text="Введите название"></textblock>	x:Name="DiscountBox"/>		
if(PlanDateStartBox.IsWrong)	<textblock td="" плановая<="" техt="Выберите</td><td><controls:InputTextBox</td></tr><tr><td>AdditionalWindow.ShowError("><td>специализацию" Height="45"/></td><td>PlaceHolder="Период поддержки"</td></textblock>	специализацию" Height="45"/>	PlaceHolder="Период поддержки"	
дата начала не может быть больше	<textblock text="Введите описание"></textblock>	x:Name="SupportPeriod" Margin="0 10 0		
плановой даты завершения");	<textblock text="Введите цену"></textblock>	0"/>		
else AdditionalWindow.HideError();	<textblock text="Введите скидку"></textblock>	<controls:inputtextbox< td=""></controls:inputtextbox<>		
}	<textblock td="" text="Введите преиод</td><td>PlaceHolder=" время<="" примерное=""></textblock>			
}	поддержки (дни)" Height="40"/>	выполнения" x:Name="ApproxTimeBox"		
}	<textblock 0="" 15="" 15"="" text="Введите примерное</td><td>Margin="></textblock>			
	время выполнения (дни)" Height="70"/>	<grid <="" height="100" td=""></grid>		
TaskDetailPage.xaml	<textblock <="" td="" text="Добавьте вложение"><td>Drop="FileDownload_OnDrop"</td></textblock>	Drop="FileDownload_OnDrop"		
<page< td=""><td>Height="100"/></td><td>AllowDrop="True"></td></page<>	Height="100"/>	AllowDrop="True">		
x:Class="INCOMSYSTEM.Pages.Details.		<grid.columndefinitions></grid.columndefinitions>		
TaskDetailPage"	<stackpanel grid.column="1" http:="" margin="5</td><td><ColumnDefinition/></td></tr><tr><td>xmlns=" schemas.microsoft.com="" td="" winf<=""><td>0"></td><td><columndefinition width="Auto"></columndefinition></td></stackpanel>	0">	<columndefinition width="Auto"></columndefinition>	
x/2006/xaml/presentation"	<stackpanel.resources></stackpanel.resources>			
xmlns:x="http://schemas.microsoft.com/wi	<style< td=""><td><button <="" style="{StaticResource</td></tr><tr><td>nfx/2006/xaml" td=""><td>TargetType="controls:InputTextBox"</td><td>LabelBtn}"</td></button></td></style<>	<button <="" style="{StaticResource</td></tr><tr><td>nfx/2006/xaml" td=""><td>TargetType="controls:InputTextBox"</td><td>LabelBtn}"</td></button>	TargetType="controls:InputTextBox"	LabelBtn}"
xmlns:mc="http://schemas.openxmlformats	BasedOn="{StaticResource {x:Type	x:Name="FileDownload"		
.org/markup-compatibility/2006"	controls:InputTextBox}}">	Click="FileDownloadBtn_Click"		
xmlns:d="http://schemas.microsoft.com/ex	<setter property="Height" value="30"></setter>	VerticalAlignment="Center"		
pression/blend/2008"		HorizontalContentAlignment="Center"		
xmlns:controls="clr-	<style <="" targettype="ComboBox" td=""><td>IsEnabled="False"/></td></tr><tr><td>namespace:INCOMSYSTEM.Controls"</td><td>BasedOn="{StaticResource {x:Type</td><td><StackPanel Grid.Column="1"></td></tr><tr><td>mc:Ignorable="d"></td><td>ComboBox}}"></td><td><Button Content="Загрузить"</td></tr><tr><td></td><td></td><td></td></tr></tbody></table></style>			

```
Click="UploadBtn_Click"/>
                                                   using
                                                                 INCOMSYSTEM. Windows;
                                                                                                      $"{task.name}.{task.fileExtension}";
<Button
                                                                                                      FileDownload.IsEnabled
                     Content="Очистить"
                                                   using
                                                                           Microsoft.Win32;
                                                                                                                                           true;
IsEnabled="False" Click="ClearBtn_Click"
                                                   namespace INCOMSYSTEM.Pages.Details
                                                                                                      TempFile
                                                                                                                                      _fileTask;
x:Name="ClearBtn"/>
                                                                                                      TempFileExtension = _fileTaskExtension;
<Rutton
                                                                                                      ClearBtn.IsEnabled
                      Content="Вернуть"
                                                   public partial class TaskDetailPage: Page
                                                                                                                                           true:
IsEnabled="False"
Click="ReturnBtn_Click"
                                                   public
                                                              TaskDetailPage(Tasks
                                                                                        task)
x:Name="ReturnBtn"/>
                                                                                                      public
                                                                                                                               TaskDetailPage()
</StackPanel>
                                                   InitializeComponent();
</Grid>
                                                   Title
                                                                "Редактирование
                                                                                                      InitializeComponent();
                                                                                    задачи":
</StackPanel>
                                                                                                      Title
                                                   _isEdit
                                                                                                                      "Добавление
                                                                                                                                       задачи";
                                                                                        true;
<StackPanel Grid.Column="2" Margin="5
                                                   SaveBtn.Content
                                                                               "Сохранить";
                                                                                                      isEdit
                                                                                                                                           false:
0">
                                                   _task
                                                                                        task;
                                                                                                      SaveBtn.Content
                                                                                                                                    "Добавить";
<StackPanel.Resources>
                                                   _fileTask
                                                                             task.attachment;
                                                                                                      FileDownload.IsEnabled
                                                                                                                                           false;
<Style
                                                                                                      ReturnBtn.Visibility = Visibility.Collapsed;
                 TargetType="TextBlock"
                                                   _fileTaskExtension = task.fileExtension;
BasedOn = "\{StaticResource
                                                   SetFileValues(task);
                                                                                                                            db
                                 {x:Type
                                                                                                      using
                                                                                                                  (var
                                                                                                                                            new
                                                                                                      INCOMSYSTEMEntities())
TextBlock}}">
                                                   _oldName
                                                                                  task name:
<Setter Property="Height" Value="30"/>
                                                   SetInputBoxValues(task);
<Setter
                Property="TextWrapping"
                                                   using
                                                              (var
                                                                                        new
                                                                                                      SpecBox.ItemsSource
Value="Wrap"/>
                                                   INCOMSYSTEMEntities())
                                                                                                      db.Specializations.ToList();
</Style>
</StackPanel.Resources>
                                                   SpecBox.ItemsSource
<TextBlock
                              Text="*"/>
                                                   db.Specializations.ToList();
                                                                                                      private
                                                                                                                  readonly
                                                                                                                               Tasks
                                                                                                                                          _task;
<TextBlock Text="*" Margin="0 15 0 5"/>
                                                   SpecBox.SelectedItem
                                                                                                                                     _fileName;
                                                                                                      private
                                                                                                                      string
<TextBlock
                              Text="*"/>
                                                   SpecBox.ItemsSource.Cast<Specializations
                                                                                                      private
                                                                                                                       string
                                                                                                                                      FileName
                              Text="*"/>
<TextBlock
<TextBlock
                               Text=""/>
                                                   .First(s => s.id == task.idSpecialization);
                                                                                                      get
                                                                                                                                      _fileName;
<TextBlock Text="*" Margin="0 10 0 5"/>
                                                   }
                                                                                                      set
<TextBlock Text="*" Margin="0 15 0 0"/>
<TextBlock
                               Text=""/>
                                                   private void SetInputBoxValues(Tasks task)
                                                                                                      FileDownload.IsEnabled
</StackPanel>
                                                                                                      !string.IsNullOrWhiteSpace(value);
<StackPanel
                           Grid.Row="1"
                                                   NameBox.Text
                                                                                                      fileName
                                                                                  task.name:
                                                                                                                                          value:
Grid.Column="0"
                                                   DescriptionBox.Text =
                                                                            task.description;
                                                                                                      FileDownload.Content
                                                                                                                                          value:
Grid.ColumnSpan="3"
                                                   PriceBox.Text
Orientation="Horizontal"
                                                   $"{task.price.ToString("#,#",
HorizontalAlignment="Center"
                                                   CultureInfo.CurrentCulture)}";
                                                                                                                                         isEdit;
                                                                                                      private
                                                                                                                  readonly
                                                                                                                               bool
Margin="0
                                      5">
                                                   if (task.discount != null) DiscountBox.Text
                                                                                                                readonly
                                                                                                                            string
                                                                                                                                     _oldName;
                                                                                                      private
<Button
                    Content="Сохранить"
                                                                          $"{task.discount}";
                                                                                                      private
                                                                                                                 readonly
                                                                                                                            byte[]
                                                                                                                                      _fileTask;
Click="SaveBtn_Click"
                                                   ApproxTimeBox.Text
                                                                                                      private readonly string _fileTaskExtension;
x:Name="SaveBtn"/>
                                                                                                      private byte[] TempFile { get; set; }
                                                   $"{task.approxCompleteTime}";
<Button Content="Отменить" Margin="5
                                                   SupportPeriod.Text
                                                                                                      private string TempFileExtension { get; set;
              Click="CancelBtn_Click"/>
                                                   task.supportPeriod.ToString();
</StackPanel>
                                                                                                      private
                                                                                                                         UploadBtn_Click(object
</Grid>
                                                   private void SetFileValues(Tasks task)
                                                                                                      sender.
                                                                                                                     RoutedEventArgs
                                                                                                                                              e)
</Page>
                                                   {
                                                   if
                                                            (_fileTask
                                                                                        null)
                                                                                                          openFile
                                                                                                                          new OpenFileDialog
                                                                                                      var
                                                                                                                     =
TaskDetailPage.xaml.cs
                    System.Globalization;
                                                   FileDownload.IsEnabled
using
                                                                                       false:
                                                                                                      Title
                                                                                                                        "Загрузка
                                                                                                                                        файла".
                              System.IO;
                                                   ReturnBtn.Visibility = Visibility.Collapsed;
                                                                                                      Filter = "Document | *.docx; *.doc | Portable
using
using
                            System.Linq;
                                                   }
                                                                                                      Document
                                                                                                                                         *.pdf",
using
                        System.Windows;
                                                   else
                                                                                                      DefaultExt
                                                                                                                                         ".docx"
               System.Windows.Controls;
using
                                                                                                      };
               INCOMSYSTEM.Context;
                                                   FileDownload.Content
                                                                                                      if (openFile.ShowDialog() != true) return;
using
```

```
TempFile
                                                    Drop))
                                                                                                         task.name
                                                                                        return;
                                                                                                                                            _name;
                                                                        file
File.ReadAllBytes(openFile.FileName);
                                                    var
                                                                                             =
                                                                                                         task.description
                                                                                                                                             _desc;
TempFileExtension
                                                    ((string[])e.Data.GetData(DataFormats.File
                                                                                                         task.price
                                                                                                                                            _price;
openFile.SafeFileName.Split('.').Last();\\
                                                    Drop))?[0];
                                                                                                         if (_disc > 0) task.discount =
                                                                                                                                            _disc;
FileName
                                                    if
                                                            (file
                  openFile.SafeFileName;
                                                                              null)
                                                                                                         else
                                                                                                                   task.discount
                                                                                                                                              null;
                                                                                        return:
ClearBtn.IsEnabled
                                     true:
                                                         fileExtension =
                                                                            file.Split('.').Last();
                                                                                                         task.approxCompleteTime
                                                                                                                                          _approx;
ReturnBtn.IsEnabled
                                                          (fileExtension
                                                                          !=
                                                                                 "docx"
                                                                                                         task.idSpecialization
                                     frue:
                                                                                                                                          _spec.id;
                                                    fileExtension != "doc" && fileExtension !=
                                                                                                         task.supportPeriod
                                                                                                                                          _support;
private void ClearBtn_Click(object sender,
                                                     "pdf")
                                                                                                         task.attachment
                                                                                                                                         TempFile;
RoutedEventArgs
                                                    {
                                                                                                         task.fileExtension = TempFileExtension;
                                                    AdditionalWindow.ShowError("Не верный
                                                                                                         db.SaveChanges();
TempFile
                                     null:
                                                                                      файла");
TempFileExtension
                                     null;
                                                    return;
                                                                                                         return
                                                                                                                                              true;
FileName
                             string.Empty;
ClearBtn.IsEnabled
                                     false;
                                                    TempFile
                                                                       File.ReadAllBytes(file);
                                                                                                         private
                                                                                                                          bool
                                                                                                                                         AddTask()
ReturnBtn.IsEnabled
                                                    TempFileExtension
                                                                                 fileExtension;
                                     true;
                                                    ClearBtn.IsEnabled
                                                                                                         if
                                                                                                                            (IsNullOrWhiteSpace())
                                                                                          true;
private void ReturnBtn_Click(object sender,
                                                    ReturnBtn.IsEnabled
                                                                                =
                                                                                          true;
RoutedEventArgs
                                                    FileName
                                                                                                         AdditionalWindow.ShowError("Поля
                                        e)
                                                                           file.Split('\\').Last();
                                                    AdditionalWindow.HideError();
                                                                                                                        быть
                                                                                                                                       пустыми");
                                                                                                         могут
TempFile
                                fileTask;
                                                                                                         return
                                                                                                                                              false;
TempFileExtension = _fileTaskExtension;
                                                    private void SaveBtn_Click(object sender,
FileName
                                                    RoutedEventArgs
                                                                                                         if
                                                                                                                (!GetValues())
                                                                                                                                   return
                                                                                                                                              false;
$"{_oldName}.{_fileTaskExtension}";
                                                                                                                              db
                                                                                                         using
                                                                                                                    (var
                                                                                                                                               new
ReturnBtn.IsEnabled
                                                    switch
                                                                                      (_isEdit)
                                                                                                         INCOMSYSTEMEntities())
ClearBtn.IsEnabled
                                     true;
                                                    {
                                                    case
                                                              true
                                                                       when
                                                                                  !SaveTask():
                                                                                                         var
                                                                                                                  task
                                                                                                                                   new
                                                                                                                                             Tasks
private void FileDownloadBtn_Click(object
                                                    case
                                                              false
                                                                        when
                                                                                   !AddTask():
sender,
               RoutedEventArgs
                                                    return:
                                                                                                         name
                                                                                                                                            _name,
                                                                                                         description
                                                                                                                                             _desc,
if
                                                    AdditionalWindow.HideError();
     (TempFile
                           null)
                                   return;
                                                                                                         price
                                                                                                                                            _price,
var saveFileDialog = new SaveFileDialog
                                                                    addWindow
                                                                                                         approxCompleteTime
                                                                                                                                           _approx,
                                                    Application.Current.Windows.OfType<Ad
                                                                                                         idSpecialization
{
                                                                                                                                          spec.id.
Title
                "Скачивание
                                  файла",
                                                    ditionalWindow>().First();
                                                                                                         supportPeriod
                                                                                                                                          _support,
FileName
                    $"{NameBox.Value}",
                                                    addWindow.DialogResult
                                                                                          true;
                                                                                                         attachment
                                                                                                                                         TempFile,
Filter = $"File | * {TempFileExtension}",
                                                    addWindow.Close();
                                                                                                         fileExtension
                                                                                                                                TempFileExtension
DefaultExt = $".{TempFileExtension}"
                                                    }
                                                    private
                                                                                    SaveTask()
                                                                                                         if (_disc > 0) task.discount
                                                                                                                   task.discount
if (saveFileDialog.ShowDialog() != true)
                                                                                                         else
                                                                                                                                              null;
                                                    {
                                                    if
                                                                       (IsNullOrWhiteSpace())
                                                                                                         db.Tasks.Add(task);
return:
                     file
                                                                                                         db.SaveChanges();
using
           (var
                                      new
FileStream(saveFileDialog.FileName,
                                                    AdditionalWindow.ShowError("Поля
FileMode. OpenOrCreate,
                                                    отмеченные звёздочкой не могут быть
                                                                                                         return
                                                                                                                                              true;
FileAccess.Write))
                                                    пустыми");
                                                                                         false;
                                                    return
                                                                                                         private
                                                                                                                           string
                                                                                                                                            name;
file.Write(TempFile, 0, TempFile.Length);
                                                    }
                                                                                                                       Specializations
                                                                                                         private
                                                                                                                                             _spec;
                                                    if
                                                           (!GetValues())
}
                                                                                         false:
                                                                                                         private
                                                                                                                           string
                                                                                                                                             _desc;
                                                    if
                                                          (IsEqualsOrNull())
                                                                                                                          decimal
                                                                                return
                                                                                          true;
                                                                                                         private
                                                                                                                                            _price;
private void FileDownload_OnDrop(object
                                                    using
                                                                          db
                                                                (var
                                                                                          new
                                                                                                         public
                                                                                                                           int
                                                                                                                                          _support;
sender,
                DragEventArgs
                                                    INCOMSYSTEMEntities())
                                                                                                         private
                                                                                                                            byte
                                                                                                                                             _disc;
                                                                                                         private
                                                                                                                            int
                                                                                                                                          _approx;
if
                                                    var task = db.Tasks.First(s => s.id ==
                                                                                                         private
                                                                                                                         bool
                                                                                                                                       GetValues()
(!e.Data.GetDataPresent(DataFormats.File
                                                    _task.id);
```

```
if
                  (IsNullOrWhiteSpace())
                                                   MessageBoxImage.Warning)
                                                   MessageBoxResult. No) \\
                                                                                      return;
AdditionalWindow.ShowError("Поля
                                                                  addWindow
               быть
                             пустыми");
                                                   Application.Current.Windows.OfType<Ad
могут
return
                                    false;
                                                   ditionalWindow>().First();
                                                   addWindow.DialogResult
}
                                                                                       false:
                                                   addWindow.Close();
name
                         NameBox. Value:
              SpecBox.SelectedItem
_spec
Specializations:
                                                   private
                                                                bool
                                                                            IsEqualsOrNull()
_desc
                   DescriptionBox.Value;
                                                   {
if (!decimal.TryParse(PriceBox.Value, out
                                                   if (_isEdit) return NameBox.Value ==
_price))
                                                   _task.name
                                                   &&
{
AdditionalWindow.ShowError("He
                                                   ((Specializations) SpecBox. Selected Item). id\\
удалось преобразовать строку в число");
                                                                        _task.idSpecialization
                                                             DescriptionBox.Value
return
                                    false;
                                                   &&
                                                   _task.description
}
if(!int.TryParse(SupportPeriod.Value,
                                                   &&
                                                                PriceBox.Value
                                                   _task.price.ToString("#,#",
_support))
                                                   CultureInfo.CurrentCulture)
{
AdditionalWindow.ShowError("He
                                                   &&
                                                              SupportPeriod.Value
удалось
             преобразовать
                                  период
                                                   _task.supportPeriod.ToString()
                                 число");
                                                   &&
                                                               DiscountBox.Value
поддержки
                     В
return
                                    false;
                                                   _task.discount.ToString()
                                                             ApproxTimeBox.Value
if
      (!DiscountBox.IsWhiteSpace
                                     &&
                                                   _task.approxCompleteTime.ToString()
                                                                     == TempFile
!byte.TryParse(DiscountBox.Value,
                                     out
                                                         (_fileTask
                                                                                        &&
_disc))
                                                   _fileTaskExtension
                                                   TempFileExtension);
AdditionalWindow.ShowError("He
                                                   return
                                                                     NameBox.IsWhiteSpace
                                                   &&
удалось преобразовать скидку в число");
                                                                Description Box. Is White Space \\
return
                                                   &&
                                                                      PriceBox.IsWhiteSpace
                                    false:
                                                   &&
                                                                  DiscountBox.IsWhiteSpace
}
if (int.TryParse(ApproxTimeBox.Value, out
                                                   &&
                                                                 SupportPeriod.IsWhiteSpace
                    return
                                                   &&
                                                               ApproxTimeBox.IsWhiteSpace
approx))
                                    true;
                                                   && string.IsNullOrWhiteSpace(FileName);
return
                                    false:
}
                                                   }
private
           bool
                    IsNullOrWhiteSpace()
                                                   private
                                                                                        void
                                                   PriceBox_OnTextChanged(object
                                                                                     sender,
                                                   TextChangedEventArgs
return
           NameBox.IsWhiteSpace
                                        DescriptionBox.IsWhiteSpace
                                        \parallel
                                                   {
PriceBox.IsWhiteSpace
                                        \|
                                                         (PriceBox.IsWhiteSpace)
                                                                                     return:
ApproxTimeBox.IsWhiteSpace
                                        \parallel
                                                   if (!decimal.TryParse(PriceBox.Value, out
SupportPeriod.IsWhiteSpace;
                                                   _price))
private void CancelBtn_Click(object sender,
                                                   AdditionalWindow.ShowError("He
RoutedEventArgs
                                       e)
                                                   удалось преобразовать строку в число");
{
                                                   return;
if
                       (!IsEqualsOrNull()
&&
            MessageBox.Show("Отменить
                                                   AdditionalWindow.HideError();
внесённые
                            изменения?",
                                                        selStart = PriceBox.SelectionStart;
"Подтверждение",
                                                   var selLength = PriceBox.SelectionLength;
MessageBoxButton.YesNo,
                                                                      PriceBox.Text.Length;
                                                         length
                                                                 =
```

```
PriceBox.Text = _price.ToString("#,#",
CultureInfo.CurrentCulture);
if (length == PriceBox.Text.Length - 1)
selStart++;
PriceBox.SelectionStart = selStart;
PriceBox.SelectionLength = selLength;
}
}
```

```
Приложение G.
                                                            DependencyProperty.UnsetValue;
                                                   return
                                                                                                       //Наименования
                                                                                                                                          сотен
ActualWidthConverter.cs
                                                                                                       private static readonly string[] Hunds =
                                                   }
using
                                 System;
                                                                                                       "", "сто ", "двести ", "триста ", "четыреста
                    System.Globalization;
using
                                                   }
using
                        System.Windows;
using
                   System.Windows.Data;
                                                                                                       "пятьсот ", "шестьсот ", "семьсот ",
                                                   Extension.cs
                                                                       Word
                                                                                                       "восемьсот
namespace
                                                   using
                                                                                           =
                                                                                                                              "девятьсот
INCOMSYSTEM.BehaviorsFiles
                                                   Microsoft.Office.Interop.Word;
                                                                                                       };
{
                                                   namespace
                                                                                                       //Наименования
                                                                                                                                      десятков
                                                   INCOMSYSTEM.BehaviorsFiles
public
        class
               ActualWidthConverter
                                                                                                       private static readonly string[] Tens =
IValueConverter
                                                   public
                                                              static
                                                                         class
                                                                                   Extension
                                                                                                       "", "десять ", "двадцать ", "тридцать ",
public object Convert(object value, Type
                                                                                                                            "пятьдесят
                                                                                                       "шестьдесят", "семьдесят", "восемьдесят
targetType, object parameter, CultureInfo
                                                   public static bool ReplaceWordText(this
culture)
                                                   Word.Document document, string findText,
                                                                                                                      "девяносто
                                                   object replaceText, bool replaceAll = true)
                                                                                                       };
                                    125d:
return
           (double)value
                                                                                                       ///
                                                                                                                                    <summarv>
}
                                                   var
                                                           range
                                                                          document.Content;
                                                                                                       /// Перевод в строку числа с учётом
public object ConvertBack(object value,
                                                   range.Find.ClearFormatting();
                                                                                                       падежного окончания относящегося к
                     object
                                                                range.Find.Execute(FindText:
Type
        targetType,
                              parameter.
                                                   return
                                                                                                       числу
                                                                                                                             существительного
CultureInfo
                                 culture)
                                                   findText,
                                                                ReplaceWith:
                                                                                 replaceText,
                                                                                                      ///
                                                                                                                                   </summary>
{
                                                   Forward:
                                                                                        true.
                                                                                                       /// <param name="val">Число</param>
return
         DependencyProperty.UnsetValue;
                                                   Replace:
                                                                      replaceAll
                                                                                                       ///
                                                                                                                             name="male">Pod
                                                                                                               <param</pre>
                                                   Word.WdReplace.wdReplaceAll
}
                                                                                                       существительного, которое относится
                                                   Word.WdReplace.wdReplaceOne);
                                                                                                                                числу</param>
                                                                                                       ///
                                                                                                                           пате="опе">Форма
                                                                                                              <param</pre>
                                                   public static string ConvertDay(this int
                                                                                                       существительного
                                                                                                                                 единственном
AnyItemsToVisibilityConverter.cs
                                                   value)
                                                                                                       числе</param>
using
                                 System;
                                                                                                              <param
                                                                                                                           name="two">Форма
                      System.Collections;
                                                           valStr
using
                                                                            value.ToString();
                                                   var
                                                                                                       существительного
                                                                                                                              om
                                                                                                                                     двух
using
                    System.Globalization;
                                                   string
                                                                                         str;
                                                                                                       четырёх</param>
                                                                                                              <param</pre>
                            System.Ling;
                                                              newVal
                                                                                      value:
                                                                                                                           name="five">Форма
using
                                                   var
                                                       (valStr.Length
                        System.Windows;
using
                                                   if
                                                                       >
                                                                            1)
                                                                                 newVal
                                                                                          =
                                                                                                       существительного
                                                                                                                              om
                                                                                                                                     пяти
                   System.Windows.Data;
                                                   int.Parse(valStr[valStr.Length
using
                                                                                                       больше</param>
                                                   1].ToString());
                                                                                                                           <returns></returns>
namespace
INCOMSYSTEM.BehaviorsFiles
                                                   if (value == 14) str = $"{value} дней";
                                                                                                       private static string Str(int val, bool male,
                                                   else if (newVal == 1) str = $"{value} день";
                                                                                                                    string
                                                                                                       string
                                                                                                              one.
                                                                                                                            two,
                                                                                                                                   string
public class AnyItemsToVisibilityConverter
                                                   else if (\text{newVal} > 0 \&\& \text{newVal} < 5) \text{ str} =
                         IValueConverter
                                                   $"{value}
                                                                                       дня";
                                                                                                       string[]
                                                                                                                           frac20
                                                   else
                                                           str
                                                                        $"{value}
                                                                                      дней";
                                                                                                       "", "один ", "два ", "три ", "четыре ", "пять
public object Convert(object value, Type
                                                   return
                                                                                         str;
targetType, object parameter, CultureInfo
                                                   }
                                                                                                                        "шесть
culture)
                                                                                                       "семь ", "восемь ", "девять ", "десять ",
                                                   }
{
                                                   }
                                                                                                       "одиннадцать
if (!(value is IEnumerable collection))
                                                                                                       "двенадцать
                                                                                                                              "тринадцать
                     Visibility. Collapsed;
                                                   NumberInWords.cs
                                                                                                       "четырналцать
                                                                                                                              "пятналцать
return collection.OfType<object>().Any() ?
                                                   using
                                                                                     System;
                                                                                                       "шестнадцать
                                                                                                                              "семнадцать
Visibility. Collapsed : Visibility. Visible;
                                                   using
                                                                                System.Text;
                                                                                                       "восемнадцать
                                                                                                                              "девятнадцать
}
                                                   namespace
                                                                                                       };
public object ConvertBack(object value,
                                                   INCOMSYSTEM.BehaviorsFiles
                                                                                                       var
                                                                                                              num
                                                                                                                                           1000;
Type
       targetType,
                     object
                               parameter,
                                                                                                       if(0)
                                                                                                                        num)
                                                                                                                                  return
CultureInfo
                                                                            NumberInWords
                                 culture)
                                                   public
                                                            static
                                                                    class
                                                                                                       if(num
                                                                                                                         0)
                                                                                                                                 throw
                                                                                                                                            new
                                                                                                       ArgumentOutOfRangeException("val",
{
                                                   {
```

```
"Параметр
                                   быть
                                                  ///
                                                                                                     PropertyChanged?.Invoke(this,
                        может
                                                                                <summary>
                                                                                                                                         new
                                                                                                     PropertyChangedEventArgs(prop));
отрицательным");
                                                  /// Перевод целого числа в строку
if(!male)
                                                                               </summary>
                                                     <param name="val">Число</param>
                                       ";
                                                       <returns>Возвращает
frac20[1]
                                                  ///
                                                                                строковую
                          "одна
frac20[2]
                          "лве
                                       ۳;
                                                                           числа</returns>
                                                  запись
                                                  public static string GetNumberInWords(this
                                                                                                     PasswordGenerator.cs
var r = new StringBuilder(Hunds[num /
                                                                     val)
                                                                                                                                      System;
                                                  int
                                                                                                     using
100]);
                                                  var
                                                              minus
                                                                                      false;
                                                                                                     namespace
                                                                                                     INCOMSYSTEM.BehaviorsFiles
                                                  if (val < 0) { val = -val; minus = true; }
                    100
if(num
                                     20)
                                                                                   (int)val;
                                                  var
                                                              n
                                                  var
                                                                    new
                                                                            StringBuilder();
                                                                                                     public
                                                                                                             static
                                                                                                                    class
                                                                                                                           PasswordGenerator
r.Append(frac20[num
                                   100]);
                                                  if
                                                       (0)
                                                                   n)
                                                                         r.Append("0
}
                                                  if
                                                                %
                                                                        1000
                                                                                  !=
                                                                                         0)
                                                                                                     public static
                                                                                                                   string
                                                                                                                           GetNewPassword()
                                                                                      ""));
else
                                                  r.Append(Str(n,
                                                                    true,
                                                                   /=
                                                                                     1000;
                                                                                                     const
                                                                                                                  string
                                                                                                                                text
                                                  r.Insert(0, Str(n, false, "тысяча", "тысячи",
r.Append(Tens[num
                          100
                                    10]);
                                                                                                     "abcdefghijklmnopqrstuvwxyz"
r.Append(frac20[num
                                                  "тысяч"));
                                                                                                     "ABCDEFGHIJKLMNOPQRSTUVWXY
                                    10]);
                                                                                     1000;
                                                                   /=
r.Append(Case(num, one, two, five));
                                                  r.Insert(0,
                                                               Str(n,
                                                                                "миллион",
                                                                                                     "0123456789"
                                                                        true.
if(r.Length
                  0)
                        r.Append("
                                                  "миллиона",
                                                                            "миллионов"));
                                                                                                     "!@#$%&";
return
                            r.ToString();
                                                                   /=
                                                                                                     var
                                                                                                           random
                                                                                                                           new
                                                                                                                                   Random():
}
                                                  r.Insert(0,
                                                              Str(n,
                                                                       true,
                                                                               "миллиард",
                                                                                                     var
                                                                                                               newPassword
///
                                                  "миллиарда",
                                                                           "миллиардов"));
                                                                                                     while(string.IsNullOrWhiteSpace(newPass
                             <summary>
              правильного
///
     Выбор
                             падежного
                                                                   /=
                                                                                     1000;
                       сущесвительного
                                                  r.Insert(0,
                                                              Str(n,
                                                                       true,
                                                                               "триллион",
окончания
                                                                                                     PasswordDifficulty.CheckDifficultyPasswo
///
                            </summary>
                                                  "триллиона",
                                                                           "триллионов"));
/// <param name="val">Число</param>
                                                  n
                                                                                     1000;
                                                                                                     rd(newPassword)
                    пате="опе">Форма
                                                                              "триллиард",
                                                                                                     DifficultyPassword. Easy)
       <param
                                                  r.Insert(0,
                                                              Str(n,
                                                                      true,
                                                                                                     for (var i = 0; i < random.Next(5, 15); i++)
существительного
                          единственном
                                                  "триллиарда",
                                                                          "триллиардов"));
числе</param>
                                                       (minus)
                                                                 r.Insert(0,
                                                                              "минус
                                                                                                     newPassword
                                                                                                                   +=
                                                                                                                          text[random.Next(0,
                    name="two">Форма
                                                                                                     text.Length)];
       <param
                                                  //Делаем
                                                                                 заглавной
                                                              первую
                                                                        букву
                                                  r[0]
                                                                        char.ToUpper(r[0]);
                                                                                                     return
                                                                                                                                newPassword;
существительного
                                      do
                      om
                             двух
четырёх</param>
                                                  return
                                                                               r.ToString();
                    name="five">Форма
                                                  }
       <param
существительного
                              пяти
                                                  }
                       om
больше</param>
                   <returns>Возвращает
                                                                                                     PasswordDifficulty.cs
существительное
                             падежным
                                                  ObservableObject.cs
                                                                                                     namespace
                                                                                                     INCOMSYSTEM.BehaviorsFiles
окончанием,
              которое
                         соответсвует
                                                  using
                                                                  System.ComponentModel;
                                                  using System.Runtime.CompilerServices;
числу</returns>
private static string Case(int val, string one,
                                                  namespace
                                                                                                     public
                                                                                                            static
                                                                                                                   class
                                                                                                                           PasswordDifficulty
string
                                                  INCOMSYSTEM.BehaviorsFiles
            two,
                       string
                                    five)
                                                                                                     public
                                                                                                                           DifficultyPassword
{
                                                                                                                 static
                                                                     ObservableObject
var t=(val % 100 > 20) ? val % 10 : val %
                                                                                                     CheckDifficultyPassword(string password)
                                                  public
                                                            class
20;
                                                  INotifyPropertyChanged
switch
                                                                                                                     DifficultyPassword.Hard;
                                      (t)
                                                                                                     return
                                                  public event PropertyChangedEventHandler
{
                                                  PropertyChanged;
            1:
                                    one:
case 2:
         case 3: case 4: return
                                    two;
                                                  protected
                                                                                       void
                                                                                                     public
                                                                                                                 enum
                                                                                                                           DifficultyPassword
default:
                   return
                                    five;
                                                  OnRaiseChanged([CallerMemberName]
                                                                                                                                           0,
}
                                                  string
                                                                prop
                                                                                                     Easy
                                                                                                     Medium
                                                                                                                                           1,
}
                                                  {
```

```
Hard
                                        2
                                                   timer.Start();
                                                                                                      var calendar = popup.Child as Calendar;
}
                                                                                                      if
                                                                                                            (calendar
                                                                                                                                null)
                                                                                                                                         return:
                                                   private void OnMouseLeave(object sender,
                                                                                                      calendar. Set Resource Reference (Calendar. St\\
                                                   MouseEventArgs
                                                                                                      yleProperty,
                                                                                                      "DatePickerEx_CustomPopup");
BorderEx.cs
                                 System;
                                                   IsMouseEnter
                                                                                       false:
using
                        System.Windows;
                                                   timer.Stop();
using
               System.Windows.Controls;
using
                                                   }
using
                   System.Windows.Data;
                                                   }
                  System. Windows. Input;
                                                                                                      InputTextBox.cs
using
using
              System. Windows. Threading;
                                                                                                      using
                                                                                                                                  System.Ling;
namespace
                INCOMSYSTEM.Controls
                                                   DatePicker.cs
                                                                                                      using
                                                                                                                              System.Windows;
                                                   using
                                                                           System.Windows;
                                                                                                      using
                                                                                                                     System. Windows. Controls;
{
public
         class
                 BorderEx
                                  Border
                                                   using
                                                                   System.Windows.Controls;
                                                                                                      using
                                                                                                                         System.Windows.Data;
{
                                                   using System.Windows.Controls.Primitives;
                                                                                                      using
                                                                                                                         System.Windows.Input;
                                                                                                                     INCOMSYSTEM.Controls
                                                                      System.Windows.Data;
public
                              BorderEx()
                                                   using
                                                                                                      namespace
                                                                   INCOMSYSTEM.Controls
                                                   namespace
MouseEnter
                          OnMouseEnter:
                                                                                                               InputTextBox
                                                                                                                                       TextBox
                                                                                                      class
                         OnMouseLeave:
MouseLeave
                                                   public class DatePickerEx : DatePicker
timer.Interval
                                                                                                      public
                                                                                                                                         IsNull
                                                                                                                        bool
TimeSpan.FromSeconds(0.5d);
                                                   static
                                                                             DatePickerEx()
timer.Tick
                 +=
                           TimerOnTick;
                                                                                                                (bool)GetValue(IsNullProperty);
                                                                                                      get
}
                                                   DefaultStyleKeyProperty.OverrideMetadata
                                                                                                      set => SetValue(IsNullProperty, value);
private void TimerOnTick(object sender,
                                                   (typeof(DatePickerEx),
EventArgs
                                                   FrameworkPropertyMetadata(typeof(DatePi
                                                                                                      public static readonly DependencyProperty
                                                   ckerEx)));
                                                                                                      IsNullProperty
IsMouseEnter
                           _isMouseOver;
                                                                                                      DependencyProperty.Register(nameof(IsNu
timer.Stop();
                                                   public override void OnApplyTemplate()
                                                                                                      ll), typeof(bool), typeof(InputTextBox),
                                                                                                                        PropertyMetadata(true));
                                                                                                      new
                                                                                                      public
                                                                                                                                  IsWhiteSpace
private DispatcherTimer timer
                                                   base.OnApplyTemplate();
                                                                                                                     bool
                                     new
DispatcherTimer();
                                                   var
                                                                     popup
private
                                                   GetTemplateChild("PART_Popup")
         bool
               isMouseOver
                                   false:
                              =
                                                                                          as
                                                                                                      get
public
                            IsMouseEnter
                                                                                                      (bool)GetValue(IsWhiteSpaceProperty);
              bool
                                                   Popup;
                                                                                       null)
                                                                                                                SetValue(IsWhiteSpaceProperty,
                                                             (popup
                                                                                                      set
                                      =>
                                                   ApplyCustomTemplate(popup);
                                                                                                      value);
get
(bool)GetValue(IsMouseEnterProperty);
          SetValue(IsMouseEnterProperty,
                                                   public
                                                                                    IsWrong
                                                                                                      public static readonly DependencyProperty
                                                                    bool
value);
                                                                                                      IsWhiteSpaceProperty
                                                   get => (bool)GetValue(IsWrongProperty);
                                                                                                      DependencyProperty.Register(nameof(IsW
                                                   set => SetValue(IsWrongProperty, value);
public static readonly DependencyProperty
                                                                                                      hiteSpace),
                                                                                                                                   typeof(bool),
IsMouseEnterProperty
                                                   }
                                                                                                      typeof(InputTextBox),
Dependency Property. Register (name of (IsM\\
                                                   public static readonly DependencyProperty
                                                                                                               FrameworkPropertyMetadata(true,
ouse Enter),\ type of (bool),\ type of (Border Ex),
                                                                                                      FrameworkPropertyMetadataOptions.Binds
                                                   IsWrongProperty
                                                                                                      TwoWayByDefault,
new
                                                   DependencyProperty.Register(nameof(IsWr
                                                                                typeof(bool),
FrameworkPropertyMetadata(default(bool),
                                                   ong),
                                                                                                      null.
                                                                                                                        null,
                                                                                                                                          false.
Framework Property Metadata Options. \textbf{Binds}
                                                   typeof(DatePickerEx),
                                                                                                      UpdateSourceTrigger.PropertyChanged));
                                                                                        new
TwoWayByDefault,
                                                   FrameworkPropertyMetadata(default(bool),
                                                                                                      public
                                                                                                                        string
                                                                                                                                          Value
                                                   FrameworkPropertyMetadataOptions.Binds
                  null.
UpdateSourceTrigger.PropertyChanged));
                                                   TwoWayByDefault,
                                                                                                               (string)GetValue(ValueProperty);
private void OnMouseEnter(object sender,
                                                                                       false,
                                                                                                      set
                                                                                                          => SetValue(ValueProperty, value);
MouseEventArgs
                                                   UpdateSourceTrigger.PropertyChanged));
                                                   void ApplyCustomTemplate(Popup popup)
                                                                                                      public static readonly DependencyProperty
isMouseOver
                                    true;
                                                   {
                                                                                                      ValueProperty
```

```
DependencyProperty.Register(nameof(Valu
                                                                                                         inputTextBox.IsPlaceHolder
                                                    ject
                                                                                                         !string. Is Null Or White Space ((string) e. New \\
e), typeof(string), typeof(InputTextBox),
                                                    DependencyPropertyChangedEventArgs e)
new
                   PropertyMetadata(""));
                                                                                                         Value):
                            IsPlaceHolder
                                                    if (!(d is InputTextBox inputTextBox))
                                                                                                         inputTextBox.Text = (string)e.NewValue;
public
               bool
{
                                                    return:
                                                    if
                                                        (inputTextBox.IsPlaceHolder)
                                                                                                         public
                                                                                                                                    InputTextBox()
get
                                       =>
                                                                                        return:
(bool)GetValue(IsPlaceHolderProperty);
                                                    inputTextBox._isChanged
                                                                                         false:
set => SetValue(IsPlaceHolderProperty,
                                                    if
                                                                                                         GotFocus
                                                                                                                                    GotFocusText;
                                                                           ((bool)e.NewValue)
value);
                                                                                                         LostFocus
                                                                                                                          +=
                                                                                                                                    LostFocusText;
                                                    var start = inputTextBox.SelectionStart;
                                                                                                         TextChanged
                                                                                                                                  OnTextChanged;
}
                                                    inputTextBox.Text = inputTextBox.Value;
public static readonly DependencyProperty
                                                                                                         PreviewKeyDown
                                                                                                                                                +=
IsPlaceHolderProperty
                                                    input Text Box. Selection Start\\
                                                                                                         OnPreviewKeyDown;
DependencyProperty.Register(nameof(IsPla
ceHolder),
                             typeof(bool),
                                                    else
                                                                                                         private void OnPreviewKeyDown(object
typeof(InputTextBox),
                                                    {
                                                                                                         sender,
                                                                                                                         KeyEventArgs
         FrameworkPropertyMetadata(true,
                                                    if
Framework Property Metadata Options. \textbf{Binds}
                                                    (string. Is Null Or White Space (input Text Box.\\
                                                                                                         switch
                                                                                                                                           (e.Key)
TwoWayByDefault,
                                                    Value))
                                                                                        return:
                                                                                                         {
                                                                                                                                       Key. Delete:
null,
                  null,
                                                    var start = inputTextBox.SelectionStart;
                                                                                                         case
UpdateSourceTrigger.PropertyChanged));
                                                                           inputTextBox.Text;
                                                                                                                                        Key.Back:
                                                    var
                                                            text
                                                                                                        case
public
                               IsPassword
                                                    foreach
                                                                                                         if
                                                                                                                 (SelectionLength
                bool
                                                                 (var
                                                                                  in
                                                                                          text)
                                                                                                                                                 0)
{
                                                    {
                                                                      text.Replace(c,
                                                                                          '•');
                                                                                                         _isRemove
                                                                                                                                              true;
get
                                                    text
(bool)GetValue(IsPasswordProperty);
                                                                                                         RemoveFromSecureString(SelectionStart,
            SetValue(IsPasswordProperty,
                                                    inputTextBox.Text
                                                                                          text:
                                                                                                         SelectionLength);
value);
                                                    inputTextBox.SelectionStart
                                                                                          start;
}
                                                                                                         else
                                                                                                                         switch
                                                                                                                                            (e.Key)
public static readonly DependencyProperty
                                                    inputTextBox._isChanged
                                                                                          true;
\\Is Password Property
                                                                                                         case Key. Delete when SelectionStart <
Dependency Property. Register (name of (Is Pa\\
                                                    public
                                                                    string
                                                                                  PlaceHolder
                                                                                                         Text.Length:
ssword),
                             typeof(bool),
                                                    {
                                                                                                         isRemove
                                                                                                                                              true:
                                                                                                         RemoveFromSecureString(SelectionStart,
typeof(InputTextBox),
        FrameworkPropertyMetadata(false,
                                                    (string)GetValue(PlaceHolderProperty);
                                                                                                         1):
Framework Property Metadata Options. \textbf{Binds}
                                                                SetValue(PlaceHolderProperty,
                                                    set
                                                                                                         break:
TwoWayByDefault,
                                                    value);
                                                                                                        case Key.Back when SelectionStart > 0:
                                     false,
null.
                                                    }
UpdateSourceTrigger.PropertyChanged));
                                                    public static readonly DependencyProperty
                                                                                                                caretIndex
                                                                                                                                    SelectionStart;
public
                 bool
                                IsShowed
                                                    PlaceHolderProperty
                                                                                                         if (SelectionStart > 0 && SelectionStart <
                                                    DependencyProperty.Register(nameof(Plac
                                                                                                         Text.Length)
{
get => (bool)GetValue(IsShowedProperty);
                                                    eHolder),
                                                                                typeof(string),
                                                                                                         caretIndex
                                                                                                                                                 1;
set => SetValue(IsShowedProperty, value);
                                                    typeof(InputTextBox),
                                                                                                         RemoveFromSecureString(SelectionStart -
                                                               FrameworkPropertyMetadata("",
                                                                                                                                                1);
public static readonly DependencyProperty
                                                    FrameworkPropertyMetadataOptions.Binds
                                                                                                         SelectionStart
                                                                                                                                        caretIndex;
                                                    TwoWayByDefault,
                                                                                                         break:
IsShowedProperty
                                                                                                        default:
                                                    PlaceHolderPropertyChanged, null, false,
Dependency Property. Register (name of (IsSh \\
owed), typeof(bool), typeof(InputTextBox),
                                                    UpdateSourceTrigger.PropertyChanged));
                                                                                                         _isRemove
                                                                                                                                             false;
        FrameworkPropertyMetadata(false,
                                                    private
                                                                        static
                                                                                          void
                                                                                                         break:
FrameworkPropertyMetadataOptions.Binds
                                                    PlaceHolderPropertyChanged(Dependency
TwoWayByDefault,
                                                                                                         e.Handled
                                                                                                                                              true:
IsShowedPropertyChanged, null,
                                                    DependencyPropertyChangedEventArgs e)
                                                                                                         break;
UpdateSourceTrigger.PropertyChanged));
                                                                                                         default:
                                                    if (!(d is InputTextBox inputTextBox))
                                                                                                         _isChanged
private
                   static
                                     void
                                                                                                                                              true:
Is Showed Property Changed (Dependency Ob\\
                                                                                                         _isRemove
                                                    return;
                                                                                                                                             false;
```

```
break;
}
                 _isRemove
                                                     private void RemoveFromSecureString(int
private
         bool
                                     false;
private
                                                                                   trimLength)
         bool
                 _isChanged =
                                     false;
                                                     startIndex,
                                                                        int
private readonly object _lock = new object();
                                                     {
private void OnTextChanged(object sender,
                                                            (string. Is Null Or White Space (Value)) \\
TextChangedEventArgs
                                                     return;
{
                                                     _isChanged
                                                                                          false;
if (Text != PlaceHolder || !IsPlaceHolder)
                                                     _isRemove
                                                                                          true;
                                                     var
                                                            caretIndex
                                                                                 SelectionStart;
IsPlaceHolder
                                     false;
                                                     Value
                                                                      Value.Remove(startIndex,
IsNull
                                     false;
                                                     trimLength);
}
                                                     Text
                                                                       Text.Remove(startIndex,
else
                                                     trimLength);
                                                     SelectionStart
                                                                                    caretIndex;
IsPlaceHolder
                                     true;
IsNull
                                                     private void LostFocusText(object sender,
                                     true;
}
                                                     RoutedEventArgs
lock
                                   (Value)
                                                     {
                                                             (string.IsNullOrWhiteSpace(Text))
IsWhiteSpace
                       IsPlaceHolder
                                                     {
string.IsNullOrWhiteSpace(Text);
                                                     _isChanged
                                                                                          false;
if
           (IsPlaceHolder)
                                    return;
                                                     Value
                                                                                  string.Empty;
if
                               (IsShowed)
                                                     IsPlaceHolder
                                                                                          true;
                                                     Text
                                                                                  PlaceHolder;
                                                     IsNull
Value
                                     Text;
                                                                                           true;
return;
                                                     }
                                                     else
                                                                 IsNull
                                                                                         false;
                           (IsWhiteSpace)
if
                                                     private void GotFocusText(object sender,
{
Value
                             string.Empty;
                                                     RoutedEventArgs
return:
                                                     if
                                                                                (IsPlaceHolder)
     (!IsPassword)
if
                      Value
                                     Text;
                                                     {
else
                                                     _isChanged
                                                                                         false;
{
                                                     IsPlaceHolder
                                                                                         false;
if (_isRemove || !_isChanged) return;
                                                     Text
                                                                                  string.Empty;
_isChanged
                                     false;
                                                     IsNull
                            SelectionStart;
                                                     string.IsNullOrWhiteSpace(base.Text);
var
                   Text[start
Value = !Text.Contains("•") ? Text :
                                                     }
Value.Insert(start - 1, r.ToString());
                                                     public
                                                                             void
                                                                                        Clear()
                                                                  new
if
                     (!Text.Contains("•"))
                                                     {
                                                     base.Text
{
                                                                                  string.Empty;
var str = Text.Where(s => s !=
                                      '•');
                                                     LostFocusText(new
                                                                             object(),
                                                                                           new
foreach
                                                     RoutedEventArgs());
             (var
                              in
                                       str)
{
                                                     }
Text
                 Text.Replace(c,
                                      '•');
                                                     }
}
                                                     }
      Text
                    Text.Replace(r,
                                      '•');
SelectionStart
                                     start;
```

Приложение Н.	<setter <="" property="Background" th=""><th><columndefinition></columndefinition></th></setter>	<columndefinition></columndefinition>
ButtonStyle.xaml	Value="#4C29FEFF" />	<columndefinition width="20"></columndefinition>
<resourcedictionary< td=""><td></td><td></td></resourcedictionary<>		
xmlns="http://schemas.microsoft.com/winf	<trigger <="" property="IsPressed" td=""><td><border< td=""></border<></td></trigger>	<border< td=""></border<>
x/2006/xaml/presentation"	Value="True">	x:Name="Border"
xmlns:x="http://schemas.microsoft.com/wi	<setter <="" property="Background" td=""><td>Grid.ColumnSpan="2"</td></setter>	Grid.ColumnSpan="2"
nfx/2006/xaml">	Value="#4C03090B" />	CornerRadius="0"
<style <="" targettype="{x:Type Button}" td=""><td></Trigger></td><td>Background="#FF3F3F3F"</td></tr><tr><td>BasedOn="{StaticResource MainStyle}"></td><td><Trigger Property="IsEnabled"</td><td>BorderBrush="#FF97A0A5"</td></tr><tr><td><Setter Property="Background"</td><td>Value="False"></td><td>BorderThickness="1" /></td></tr><tr><td>Value="#4C37B1D4"/></td><td><Setter Property="Background"</td><td><Path</td></tr><tr><td><Setter Property="Height" Value="30" /></td><td>Value="#4D4D4D"/></td><td>x:Name="Arrow"</td></tr><tr><td><Setter Property="FontSize" Value="18" /></td><td></Trigger></td><td>Grid.Column="1"</td></tr><tr><td><Setter Property="Margin" Value="3 0" /></td><td></Style.Triggers></td><td>Fill="White"</td></tr><tr><td><Setter Property="Padding" Value="5 2" /></td><td></style>	HorizontalAlignment="Center"	
<setter <="" property="Foreground" td=""><td><style <="" targettype="Button" td=""><td>VerticalAlignment="Center"></td></tr><tr><td>Value="Black" /></td><td>x:Key="LabelBtn"</td><td><Path.Style></td></tr><tr><td><Setter Property="Cursor" Value="Hand"</td><td>BasedOn="{StaticResource MainStyle}"></td><td><Style TargetType="Path"></td></tr><tr><td>/></td><td><pre><Setter Property="Cursor" Value="Hand"</pre></td><td><Style.Triggers></td></tr><tr><td><Setter</td><td>/></td><td><pre><DataTrigger Binding="{Binding</pre></td></tr><tr><td>Property="HorizontalContentAlignment"</td><td><Setter Property="Template"></td><td>RelativeSource={RelativeSource</td></tr><tr><td>Value="Center" /></td><td><Setter Value></td><td>AncestorType=ToggleButton},</td></tr><tr><td><Setter</td><td><pre><ControlTemplate TargetType="Button"></pre></td><td>••</td></tr><tr><td></td><td>Control rempiate Target Type= Button > <Border BorderBrush="White"</p></td><td>,</td></tr><tr><td>Property="VerticalContentAlignment" Value="Center" /></td><td></td><td><pre><Setter Property="Data" Value="M0,8 L0,6</pre></td></tr><tr><td></td><td>BorderThickness="0 0 0 2"</td><td>L4,2 L8,6 L8,8 L4,4 z"/></td></tr><tr><td><Setter Property="Template"></td><td>VerticalAlignment="{TemplateBinding</td><td></DataTrigger></td></tr><tr><td><Setter.Value></td><td>VerticalAlignment}"</td><td></Style.Triggers></td></tr><tr><td><ControlTemplate TargetType="{x:Type</td><td>HorizontalAlignment="{TemplateBinding</td><td><Style.Setters></td></tr><tr><td>Button}"></td><td>HorizontalAlignment}"></td><td><Setter Property="Data" Value="M0,0 L0,2</td></tr><tr><td><Border Background="{TemplateBinding</td><td><TextBlock Text="{TemplateBinding</td><td>L4,6 L8,2 L8,0 L4,4 z"/></td></tr><tr><td>Background}"</td><td>Content}"</td><td></Style.Setters></td></tr><tr><td>CornerRadius="5"</td><td>Foreground="White"</td><td></style></td></setter>	<style <="" targettype="Button" td=""><td>VerticalAlignment="Center"></td></tr><tr><td>Value="Black" /></td><td>x:Key="LabelBtn"</td><td><Path.Style></td></tr><tr><td><Setter Property="Cursor" Value="Hand"</td><td>BasedOn="{StaticResource MainStyle}"></td><td><Style TargetType="Path"></td></tr><tr><td>/></td><td><pre><Setter Property="Cursor" Value="Hand"</pre></td><td><Style.Triggers></td></tr><tr><td><Setter</td><td>/></td><td><pre><DataTrigger Binding="{Binding</pre></td></tr><tr><td>Property="HorizontalContentAlignment"</td><td><Setter Property="Template"></td><td>RelativeSource={RelativeSource</td></tr><tr><td>Value="Center" /></td><td><Setter Value></td><td>AncestorType=ToggleButton},</td></tr><tr><td><Setter</td><td><pre><ControlTemplate TargetType="Button"></pre></td><td>••</td></tr><tr><td></td><td>Control rempiate Target Type= Button > <Border BorderBrush="White"</p></td><td>,</td></tr><tr><td>Property="VerticalContentAlignment" Value="Center" /></td><td></td><td><pre><Setter Property="Data" Value="M0,8 L0,6</pre></td></tr><tr><td></td><td>BorderThickness="0 0 0 2"</td><td>L4,2 L8,6 L8,8 L4,4 z"/></td></tr><tr><td><Setter Property="Template"></td><td>VerticalAlignment="{TemplateBinding</td><td></DataTrigger></td></tr><tr><td><Setter.Value></td><td>VerticalAlignment}"</td><td></Style.Triggers></td></tr><tr><td><ControlTemplate TargetType="{x:Type</td><td>HorizontalAlignment="{TemplateBinding</td><td><Style.Setters></td></tr><tr><td>Button}"></td><td>HorizontalAlignment}"></td><td><Setter Property="Data" Value="M0,0 L0,2</td></tr><tr><td><Border Background="{TemplateBinding</td><td><TextBlock Text="{TemplateBinding</td><td>L4,6 L8,2 L8,0 L4,4 z"/></td></tr><tr><td>Background}"</td><td>Content}"</td><td></Style.Setters></td></tr><tr><td>CornerRadius="5"</td><td>Foreground="White"</td><td></style>	
VerticalAlignment="{TemplateBinding	TextWrapping="Wrap"	
VerticalAlignment}"	TextAlignment="{TemplateBinding	
HorizontalAlignment="{TemplateBinding	HorizontalContentAlignment}"/>	
HorizontalAlignment}">		
<label <="" content="{TemplateBinding</td><td></ControlTemplate></td><td><ControlTemplate</td></tr><tr><td>Content}" td=""><td></td><td>x:Key="ComboBoxTextBox"</td></label>		x:Key="ComboBoxTextBox"
Foreground="White"		TargetType="{x:Type TextBox}">
Style="{StaticResource LabelStyle}"		<border <="" td="" x:name="PART_ContentHost"></border>
FontFamily="{TemplateBinding		Focusable="False"
FontFamily}"		CornerRadius="5"
VerticalAlignment="{TemplateBinding	ComboBoxStyle.xaml	BorderBrush="#0AD2CF"
VerticalContentAlignment}"	<resourcedictionary< td=""><td>BorderThickness="1"</td></resourcedictionary<>	BorderThickness="1"
HorizontalAlignment="{TemplateBinding	xmlns="http://schemas.microsoft.com/winf	Background="{TemplateBinding
HorizontalContentAlignment}"/>	x/2006/xaml/presentation"	Background}" />
	xmlns:x="http://schemas.microsoft.com/wi	
	nfx/2006/xaml">	<style targettype="{x:Type</td></tr><tr><td></Setter.Value></td><td><ControlTemplate</td><td>ComboBox}"></td></tr><tr><td></Setter></td><td>x:Key="ComboBoxToggleButton"</td><td><Setter</td></tr><tr><td><Style.Triggers></td><td>TargetType="{x:Type ToggleButton}"></td><td>Property="HorizontalContentAlignment"</td></tr><tr><td><Trigger Property="IsMouseOver"</td><td><Grid></td><td>Value="Left"/></td></tr><tr><td>1 ,</td><td></td><td></td></tr></tbody></table></style>

 $<\!\!Grid.ColumnDefinitions\!\!>$

 $<\!\!Setter \quad Property \!\!=\! "Snaps To Device Pixels"$

Value="True">

Value="True"/> IsReadOnly}"/> Property="Popup.AllowsTransparency" Property="MinHeight" Value="true"> <Setter <Popup Value="30"/> x:Name="Popup" <Setter TargetName="DropDownBorder" <Setter Property="Foreground" Placement="Bottom" Property="CornerRadius" Value="0"/> Value="White"/> HorizontalAlignment="Center" <Setter TargetName="DropDownBorder" <Setter Property="OverridesDefaultStyle" IsOpen="{TemplateBinding Property="Margin" Value="0,2,0,0"/> Value="False"/> IsDropDownOpen}" </Trigger> <Setter Property="Padding" Value="5 5 20 AllowsTransparency="True" <Trigger Property="IsEditable" 0"/> Focusable="False" Value="true"> Property="Template"> Property="IsTabStop" <Setter PopupAnimation="Slide"> <Setter <Setter.Value> <Grid x:Name="DropDown" Value="false"/> <ControlTemplate SnapsToDevicePixels="True" <Setter TargetType="ComboBox"> MinWidth="{TemplateBinding TargetName="PART_EditableTextBox" Value="Visible"/> <Grid> ActualWidth}" Property="Visibility" <Setter <ToggleButton MaxHeight="{TemplateBinding TargetName="ContentSite" Property="Visibility" Value="**Hidden**"/> Name="ToggleButton" MaxDropDownHeight}"> Template="{StaticResource <Border </Trigger> ComboBoxToggleButton}" </ControlTemplate.Triggers> x:Name="DropDownBorder" Grid.Column="2" Background="#FF3F3F3F" </ControlTemplate> Focusable="false" BorderThickness="1" </Setter.Value> IsChecked="{Binding BorderBrush="#888888"/> </Setter> Path=IsDropDownOpen,Mode=TwoWay,R <ScrollViewer Margin="4,6,4,6" </Style> <Style x:Key="ComboBoxTextBlockStyle" elativeSource={RelativeSource Width="{Binding TemplatedParent}}" RelativeSource={RelativeSource TargetType="TextBlock"> ClickMode="Press"> AncestorType=ComboBox}, <Setter Property="FontSize" Value="15"/> Path=ActualWidth}" </ToggleButton> <Setter Property="ToolTip"> Horizontal Scroll Bar Visibility = "Disabled"<ContentPresenter <Setter. Value> SnapsToDevicePixels="True"> x:Name="ContentSite" <Binding RelativeSource="{RelativeSource IsHitTestVisible="False" Path="Text"/> <StackPanel IsItemsHost="True" Mode=Self}" Content="{TemplateBinding KeyboardNavigation.DirectionalNavigation </Setter.Value> SelectionBoxItem}" ="Contained" /> </Setter> ContentTemplate="{TemplateBinding </ScrollViewer> </Style> SelectionBoxItemTemplate}" </Grid> <Style TargetType="ToolTip"> ContentTemplateSelector="{TemplateBindi </Popup> <Setter Property="Background" ItemTemplateSelector}" </Grid> Value="Gray"/> VerticalAlignment="{TemplateBinding <ControlTemplate.Triggers> <Setter Property="Foreground" VerticalContentAlignment}" <Trigger Property="HasItems" Value="White"/> HorizontalAlignment="{TemplateBinding Value="false"> </Style> <Setter TargetName="DropDownBorder" HorizontalContentAlignment}" </ResourceDictionary> Margin="{TemplateBinding Padding}"/> Property="MinHeight" Value="30"/> <TextBox </Trigger> DatePickerExStyle.xaml x:Name="PART_EditableTextBox" <Trigger Property="IsEnabled" < Resource Dictionary Style="{x:Null}" Value="false"> xmlns="http://schemas.microsoft.com/winf Template="{StaticResource <Setter x/2006/xaml/presentation" Property="Foreground" ComboBoxTextBox}" Value="#888888"/> xmlns:x="http://schemas.microsoft.com/wi HorizontalAlignment="Left" </Trigger> nfx/2006/xaml" VerticalAlignment="Center" Property="IsGrouping" <Trigger xmlns:local="clr-Margin="3,3,23,3" Value="true"> namespace:INCOMSYSTEM.Controls"> Focusable="True" <Setter <ControlTemplate Background="#FF3F3F3F" Property="ScrollViewer.CanContentScroll" x:Key="CalendarItemTemplate" Foreground="Green" Value="false"/> TargetType="{x:Type CalendarItem}"> Visibility="Hidden" </Trigger> <ControlTemplate.Resources>

<Trigger

SourceName="Popup"

<DataTemplate

x:Key="{x:Static

IsReadOnly="{TemplateBinding

Controller	CalendarItem.DayTitleTemplateResourceK	<visualstate x:name="MouseOver"></visualstate>	
Foreground="#FFFFF" Storyboard_TargetNome="path" Storyboard_TargetNome="path" To- FortWeight="Blod" Storyboard_TargetNome="path" Storyboard_TargetNome="path" To- FortWeight="Blodget" Storyboard_TargetNome="path" Storyboard_TargetNome="path" Storyboard_TargetNome="path" To- FortSize="9.5" Storyboard_TargetNome="path" Storyb	ey}">	<storyboard></storyboard>	<visualstate x:name="Disabled"></visualstate>
Found smally="Trebusher	<textblock< td=""><td><coloranimation <="" duration="0" td=""><td><storyboard></storyboard></td></coloranimation></td></textblock<>	<coloranimation <="" duration="0" td=""><td><storyboard></storyboard></td></coloranimation>	<storyboard></storyboard>
FornWeight="Bold"	Foreground="#FFFFFF"	Storyboard.TargetName="path"	<doubleanimation <="" duration="0" td=""></doubleanimation>
Femilian Alignmenta Center	FontFamily="Trebuchet MS"	To="#FF73A9D8"	Storyboard.TargetName="path" To=".5"
Margin="Oco.106" Cylsius State	FontWeight="Bold"	Storyboard.TargetProperty="(Shape.Fill).(S	Storyboard.TargetProperty="(Shape.Fill).(B
Murgin="0.0.06"	FontSize="9.5"	olidColorBrush.Color)"/>	rush.Opacity)"/>
Margin="0.60.6"	HorizontalAlignment="Center"		
VerticalAlignment="Center"	Margin="0,6,0,6"		
Verical Alignment="Center"	Text="{Binding}"	<visualstate x:name="Disabled"></visualstate>	
ControlTemplate		<storyboard></storyboard>	
<pcontroltemplate.resources> Storyboard.TargetProperty="(Shape.Fill)(B Obata="M12-96875" d. 28125" b.11.52 Cil.1.52 Grid x.Name="PART_Root"> Storyboard.TargetProperty="(Shape.Fill)(B Data="M12-96875" d. 28125" b.11.52 Local Storyboard. 26.28125" b.11.52 Local Storyboard. Horizontal Alignment="Center" ControlTemplate. Local Storyboard. Local Storyboard. Horizontal Alignment="Center" ControlTemplate. ControlTemplate. Local Storyboard. Storyboard. Local Storyboard. Width="6"/"></pcontroltemplate.resources>	•	•	
Storyboard.TargetProperty="(Shape.Fill).(B	•		
Colord Resources			•
\$\times \times			
Color="#A5FFFFFF"/>		• •	
15.28125Z* 15.28125Z* Fill="#FFFFFF" 4 Fill="#FFFFFFF" Fill="#FFFFFFF" Fill="#FFFFFFF" Height="10" <ind> x.Name="Posmal"/>x.Name="Posmal"/>y. Path x.Name="path" Height="10" <ind> x.Name="Disabled">x.Name="Disabled">x.Name="Disabled">x.Name="Disabled">x.Name="Disabled">x.Name="Disabled">x.Name="Disabled">x.Name="Disabled">x.Name="Disabled">x.Name="Disabled">x.Name="Disabled">x.Name="Disabled 15.28125 L19.03125 27.71875 L20.46875 Width="6">Width="6">Width="6">X.Grid>X.</ind></ind>	•	•	
<visualstatemanager, td="" visualstategroups<=""> Fill="#FFFFFF" <visualstategroup< td=""> Grid> HorizontalAlignment="Center" x:Name="CommonStates"> -Path x:Name="path" Heights="10" x:Name="CommonStates"> -Path x:Name="path" Heights="10" x:VisualState x:Name="Disabled"> 15:28125 17:34375 16 L8.03125 VicticalAlignment="Center" <visualstate< td=""> x:Name="Disabled"> 16:71875 L10.03125 27:71875 L20.46875 Width="6"> <notation="part disabled"=""> 57:718752 16 L20.46875 Width="6"> Grid> Storyboard.TargetName="PART_Disabled 57:18752* -VisualCarget Property="Opacity"> To="1" Fill="#FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</notation="part></visualstate<></visualstategroup<></visualstatemanager,>			
<visualstategroup< td=""> <grid> K:Name="path" HorizontalAlignment="Center" x:Name="CommonStates"> <path< td=""> x:Name="path" Height="10" <visualstate< td=""> x:Name="Disabled"> 15.28125 L7.34375 16 18.03125 VerticalAlignment="Center" <visualstate< td=""> x:Name="Disabled"> 16.71875 L19.03125 27.71875 L20.46875 Width="6"/> <cobach-imation< td=""> Duration="0" 26.28125 L10.1875 16 L20.46875 Width="6"/> Storyboard TargetName="PART_Disabled 5.718752" Visual To="1" Fill="#FFFFFFF" VisualTargetProperty="Opacity"/> HorizontalAlignment="Center" </cobach-imation<></visualstate<></visualstate<></path<></grid></visualstategroup<>			
x:Name="CommonStates"> < Path x:Name="path" Height="10" <visualstate< th=""> x:Name="Normal"/> Data="M19.03125" 4.28125" L8.03125 Stretch="Uniform" <visualstate< th=""> x:Name="Disabled"> 15.28125" L7.34375" 16 L8.03125 VerticalAlignment="Center" <volume< th=""> CobubleAnimation Duration="0" 26.28125" L10.1875 16 L20.46875 Width="6"/> Grid Storyboard.TargetName="PART_Disabled 5.71875Z 16 L20.46875 Width="6"/> ControlTemplate VisualState To="1" Fill="#FFFFFFF" ControlTemplate ControlTemplate VisualState To="1" Fill="#FFFFFFFF" ControlTemplate VisualStateGroup VisualStateGroup VisualStateGroup VisualStateGroup VisualStateGroup VisualStateGroup VisualStateGroup VisualStateGroup VisualStateGroup ControlTemplate VisualState x:Name="Normal of the</volume<></visualstate<></visualstate<>			
VisualState x:Name="Normal"/>	_		· ·
VisualState		•	· ·
Storyboard			
Storyboard.TargetName="PART_Disabled 5.71875Z" Clouders ControlTemplate Storyboard.TargetName="PART_Disabled 5.71875Z" ControlTemplate Cont			•
Storyboard.TargetName="PART_Disabled Visual" To="1" Fill="#FFFFFF"	·		
Visual" To="1" Fill="#FFFFFF" 《ControlTemplate> Storyboard.TargetProperty="Opacity"/> HorizontalAlignment="Center" ControlTemplate **Storyboard.** Height="8" x:Key="HeaderButtonTemplate" **VisualState> Stretch="Uniform" TargetType="{x:Type_Button_Template"} **VisualStateGroup> VerticalAlignment="Center" Grid **Cursor="Hard **VisualStateGroup> Width="6"/>> **VisualStateGroup **VisualStateGroup **Border Background="{TemplateBinding **Grid> **VisualStateGroup ***VisualStateGroup **Border Brush="{TemplateBinding **ControlTemplate> **VisualState ***Name="Normal BorderBrush="{TemplateBinding} ***ControlTemplate ***VisualState ***Name="Normal BorderBrush="{TemplateBinding} ***Xey="NextButtonTemplate" ***Storyboard ***Storyboard **Border Thickness="{TemplateBinding} ***Xey="NextButtonTemplate" ***Storyboard ***Storyboard ****Storyboard ************************************			
Storyboard.TargetProperty="Opacity"/> HorizontalAlignment="Center"			
Height="8" x:Key="HeaderButtonTemplate" TargetType="{x:Type Button Cursor="Hard" </td <td></td> <td></td> <td>•</td>			•
VisualState> Stretch="Uniform" TargetType="{x:Type Buttor VisualStateGroup> VerticalAlignment="Center" <fri>doi:10 Cursor="Har VisualStateManager.VisualStateGroups> Width="6"/> Buttor</fri>		· ·	•
VerticalAlignment="Center" Grid Cursor="Hard" VisualStateManager.VisualStateGroups Width="6"/> VisualStateManager.VisualStateGroup Border Background="{TemplateBinding	•		
<td< td=""><td></td><td></td><td></td></td<>			
Border Background="{TemplateBinding <visualstategroup< td=""> Background}" x:Name="CommonStates"> BorderBrush="{TemplateBinding <visualstate< td=""> x:Name="Normal AvisualState BorderBrush}" <controltemplate< td=""> <visualstate< td=""> x:Name="MouseOver BorderThickness="{TemplateBinding x:Key="NextButtonTemplate" <storyboard> BorderThickness="{TemplateBinding x:Key="NextButtonTemplate" <coloranimation< td=""> CornerRadius="15"> <border td="" templatebinding<=""> x:Key="NextButtonTemplate" <coloranimation< td=""> CornerRadius="15"> <border button}"="" template=""> <coloranimation< td=""> CornerRadius="15"> <border <="" cornerradius="20" td=""> Duration="0" Border Bush="#FFFFFFFFF" Cursor="Hand" Storyboard.TargetName="buttonContent Background="#FD413C"> Corid. Resources <visualstatemanager.visualstategroups> Storyboard.TargetProperty="(TextElems Color)"/2 CornerTemplate x:Name="CommonStates"> x:Key="PreviousButtonTemplate" <visualstate x:name="Normal"></visualstate> <visualstate< td=""> x:Name="CommonStates"> <visualstate< td=""> x:Name="Distallate"> x:Key="PreviousButtonTemplate" <visuals< td=""><td>*</td><td>Ç</td><td></td></visuals<></visualstate<></visualstate<></visualstatemanager.visualstategroups></border></coloranimation<></border></coloranimation<></border></coloranimation<></storyboard></visualstate<></controltemplate<></visualstate<></visualstategroup<>	*	Ç	
Background}"	•		
BorderBrush="{TemplateBinding			
BorderThickness="{TemplateBinding x:Key="NextButtonTemplate"	-		
BorderThickness="{TemplateBinding x:Key="NextButtonTemplate"		*	
BorderThickness}" TargetType="{x:Type Button}"> <coloranimation cornerradius="15"> <border 0"="" <border="" borderbrush="#FFFFFFF" cornerradius="1" cursor="Hand" duration="0" storyboard.targetname="buttonContent BorderThickness="> Background="#FD413C"> To="#606467" <grid> Corrid> <visualstatemanager.visualstategroup> Storyboard.TargetProperty="(TextElemt Corid.Resources> <visualstategroup 2="" <controltemplate="" foreground).(solidcolorbrush.color)"="" x:name="CommonStates"> <xisualstate x:name="Normal"></xisualstate> <xisualstate> x:Name="MouseOver"> <visualstate disable="" x:name="BouseOver"> <visualstate <="" disable="" td="" x:name="Disabl</td><td>BorderBrush}"><td><controltemplate< td=""><td><visualstate x:name="MouseOver"></visualstate></td></controltemplate<></td></visualstate></visualstate></xisualstate></visualstategroup></visualstatemanager.visualstategroup></grid></border></coloranimation>	<controltemplate< td=""><td><visualstate x:name="MouseOver"></visualstate></td></controltemplate<>	<visualstate x:name="MouseOver"></visualstate>	
CornerRadius="15"> <border <="" cornerradius="20" duration="0" td=""><td>BorderThickness="{TemplateBinding</td><td>x:Key="NextButtonTemplate"</td><td><storyboard></storyboard></td></border>	BorderThickness="{TemplateBinding	x:Key="NextButtonTemplate"	<storyboard></storyboard>
<border< th="">BorderBrush="#FFFFFF"Cursor="Hand"Storyboard.TargetName="buttonContentBorderThickness="0"CornerRadius="1">Background="#FD413C">To="#606467"<grid><visualstatemanager.visualstategroups>Storyboard.TargetProperty="(TextElem<grid.resources><visualstategroup< td="">Foreground).(SolidColorBrush.Color)"/<controltemplate< td="">x:Name="CommonStates">x:Key="PreviousButtonTemplate"<visualstate< td="">x:Name="Normal"/>TargetType="{x:TypeButton}">Button}">Button}"><visualstate< td="">x:Name="MouseOver"><visualstate< td="">x:Name="Disable of the color of th</visualstate<></visualstate<></visualstate<></controltemplate<></visualstategroup<></grid.resources></visualstatemanager.visualstategroups></grid></border<>	,		
BorderThickness="0" CornerRadius="1"> Background="#FD413C"> To="#606467" <grid> <visualstatemanager.visualstategroups> Storyboard.TargetProperty="(TextElement of ControlTemplate visualStateGroup visualStateGroup visualStateGroup visualStateGroup visualStateGroup visualStateGroup visualState v</visualstatemanager.visualstategroups></grid>	CornerRadius="15">	<border <="" cornerradius="20" td=""><td>Duration="0"</td></border>	Duration="0"
<grid> < VisualStateManager.VisualStateGroups> Storyboard.TargetProperty="(TextElement Corid.Resources)</grid>	<border <="" borderbrush="#FFFFFFF" td=""><td>Cursor="Hand"</td><td>Storyboard.TargetName="buttonContent"</td></border>	Cursor="Hand"	Storyboard.TargetName="buttonContent"
<pre><grid.resources></grid.resources></pre>	BorderThickness="0" CornerRadius="1">	Background="#FD413C">	To="#606467"
<controltemplate< td=""> x:Name="CommonStates"> /Storyboard> x:Key="PreviousButtonTemplate" <visualstate< td=""> x:Name="Normal"/> /VisualState> TargetType="{x:Type Button}"> <visualstate< td=""> x:Name="MouseOver"> <visualstate< td=""> x:Name="Disable values"> <border< td=""> CornerRadius="20" <storyboard> <storyboard> Cursor="Hand" <coloranimation< td=""> Duration="0" <doubleanimation< td=""> Background="#FD413C"> Storyboard.TargetName="path" Duration="0"</doubleanimation<></coloranimation<></storyboard></storyboard></border<></visualstate<></visualstate<></visualstate<></controltemplate<>	<grid></grid>	<visualstatemanager.visualstategroups></visualstatemanager.visualstategroups>	Story board. Target Property = "(Text Element.
x:Key="PreviousButtonTemplate"	<grid.resources></grid.resources>	<visualstategroup< td=""><td>$Foreground). (Solid Color Brush. Color) "/\!\!>$</td></visualstategroup<>	$Foreground). (Solid Color Brush. Color) "/\!\!>$
TargetType="{x:Type Button}"> <visualstate x:name="MouseOver"> <visualstate commonstates"="" disable="" x:name="Disable x:</td><td><ControlTemplate</td><td>x:Name="></visualstate></visualstate>			
<pre> <border< td=""><td>$x{:}Key{=}"PreviousButtonTemplate"\\$</td><td><visualstate x:name="Normal"></visualstate></td><td></td></border<></pre>	$x{:}Key{=}"PreviousButtonTemplate"\\$	<visualstate x:name="Normal"></visualstate>	
Cursor="Hand"	TargetType="{x:Type Button}">	<visualstate x:name="MouseOver"></visualstate>	<visualstate x:name="Disabled"></visualstate>
Background="#FD413C"> Storyboard.TargetName="path" Duration="0"	<border <="" cornerradius="20" td=""><td><storyboard></storyboard></td><td><storyboard></storyboard></td></border>	<storyboard></storyboard>	<storyboard></storyboard>
	Cursor="Hand"	<coloranimation <="" duration="0" td=""><td><doubleanimation< td=""></doubleanimation<></td></coloranimation>	<doubleanimation< td=""></doubleanimation<>
<visualstatemanager visualstategroups=""> To="#FF73A9D8" Storyboard TargetName="buttonContent</visualstatemanager>	Background="#FD413C">	Storyboard.TargetName="path"	Duration="0"
the state of the s	<visualstatemanager.visualstategroups></visualstatemanager.visualstategroups>	To="#FF73A9D8"	Storyboard.TargetName="buttonContent"
<visualstategroup .5"<="" storyboard.targetproperty="(Shape.Fill).(S To=" td=""><td><visualstategroup< td=""><td>Storyboard.TargetProperty="(Shape.Fill).(S</td><td></td></visualstategroup<></td></visualstategroup>	<visualstategroup< td=""><td>Storyboard.TargetProperty="(Shape.Fill).(S</td><td></td></visualstategroup<>	Storyboard.TargetProperty="(Shape.Fill).(S	
x:Name="CommonStates"> olidColorBrush.Color)"/> Storyboard.TargetProperty="Opacity"/>	x:Name="CommonStates">	olidColorBrush.Color)"/>	Storyboard.TargetProperty="Opacity"/>
<pre><visualstate x:name="Normal"></visualstate> </pre>	<visualstate x:name="Normal"></visualstate>		

VisualSancGroup Vis		NextButtonTemplate}"	Stretch="Fill" StrokeThickness="1"			
CytinalStateMumer=VisualStateGroups		• •	Visibility="Collapsed"/>			
ContentPersenter	•		• •			
Second Content Complete Template Binding	•					
Content Template Template Binding		_				
ContentTemplate		1	1 ,			
Content="(TemplateBinding		-				
Text Element Foreground="#FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	• •	•	11.0			
Morgin="8" 4 1 9" ColumnDefinitions			_			
Horizontal Alignment="Template Binding Column Definition Width="30"> Column Definition	•	•				
NotizontalComenAlignment						
VerticalAlignment="(Templateblinding						
VerticalContentAlignment ' '>	· ·		* *			
ColumnDefinition			·			
ColumnDefinition	-					

	•					
ColumnDefinition			_			
MinWidth="79">						
ColumnDefinition Width="avi" KnowDefinition Height="Auto" Value="Visible" KnowDefinition Height="Auto" AbataTrigger Binding="[Binding Grid.ColumnDefinitions KnowDefinition Height="Auto" ObataTrigger Binding="[Binding Grid.RowDefinition Height="Auto" ObataTrigger Binding="[Binding Grid.RowDefinition Height="Auto" DisplayMode Calendar Plant			• • •			
ColumnDefinition		· ·	· ·			
		· ·				
		· ·				
RelativeSource Rela		8				
		ε	• •			
	· ·	· ·	•			
Selution x:Name="PART_HeaderButton"			, , , , , , , , , , , , , , , , , , , ,			
Focusable="False" Grid x:Name="PART_YearView" TargetName="PART_MonthView" Focusable="False" Grid ColumnSpan="3" Value="Hidden" Value="Hidden" Value="Hidden" Value="Hidden" FontSize="10.5" Margin="6" Grid.Row="1" TargetName="PART_YearView" Value="Visible" Value="Visible			,,,			
Focusable="False" Grid.ColumnSpan="3" Value="Hidden">			1 ,			
FontWeight="Bold" Margin="Center" Setter Property="Visibility" FontSize="10.5" Margin="6" Grid.Row="1" TargetName="PART_YearView" Margin="6" Grid.Row="1" TargetName="PART_YearView" Value="Visible" Value="CalendarStyle" Value="CalendarStyle" Value="CalendarStyle" Value="CalendarStyle" Value="CalendarStyle" Value="Grid.Column="I" Value="Involve" Value="Froreground"			· ·			
Font Size="10.5"						
HorizontalAlignment="Center" Visibility="Hidden" Value="Visible" Value="Center" Cfrid.Row="0" Cfrid.ColumnDefinition Width="Auto" C/ControlTemplate-Triggers ColumnDefinition Width="Auto" C/ControlTemplate-Triggers ColumnDefinition Width="Auto" C/ControlTemplate-Triggers ColumnDefinition Width="Auto" C/ControlTemplate-Triggers CalendarStyle" C/ControlTemplate VerticalAlignment="Center" C/ColumnDefinition Width="Auto" C/ControlTemplate X:Key="CalendarStyle" C/ColumnDefinition Width="Auto" C/ControlTemplate X:Key="CalendarStyle" C/ColumnDefinition Width="Auto" C/ControlTemplate X:Key="CalendarStyle" C/ColumnDefinition Width="Auto" C/ControlTemplate X:Key="CalendarStyle" C/ColumnDefinition C/ColumnDefini	•	· ·	1 ,			
Clid.Row="0" ColumnDefinition Width="Auto"/> ControlTemplate. Triggers			_			
ColumnDefinition Width="Auto"/> Columto Template-Triggers> ColumnDefinition Width="Auto"/> Columto Template-Triggers> ColumnDefinition Width="Auto"/> Columto Template-Triggers> ColumnDefinition Width="Auto"/> Columto Template-Triggers> ColumnDefinition Width="Auto"/> ColumnDefinition ColumnDefinition Width="Auto"/> ColumnDefinition Co		·				
VerticalAlignment="Center"/>			••			
VerticalAlignment="Center">			1 66			
StackPanel Orientation="Horizontal" ColumnDefinition Width="Auto"/> TargetType="{x:Type Calendar}"> Calendar}"> Grid.Column="1" Cyfrid.ColumnDefinitions Setter Property="Foreground"	Ī,		-			
	-		, , , , , , , , , , , , , , , , , , , ,			
Margin="0 13 10 5"			<setter <="" property="Foreground" td=""></setter>			
HorizontalAlignment="Right"> <rowdefinition height="Auto"></rowdefinition> Value="#282F33"/> <pre></pre>						
<pre> <button <="" td="" x:name="PART_PreviousButton"><td>e</td><td>,</td><td></td></button></pre>	e	,				
Focusable="False"	•	ε				
Height="17"		ε	• •			
Margin="0"/"> Value="0"/> Template="{StaticResource Setter Property="Template"> PreviousButtonTemplate}" Setter. Value> Width="17"/> ControlTemplate TargetType="{x:Type <button <="" td="" x:name="PART_NextButton"> x:Name="PART_DisabledVisual" Calendar}"> Calendar}"> Focusable="False" Fill="{StaticResource DisabledColor}" <stackpanel <="" td="" x:name="PART_Root"> Height="17" Opacity="0" RadiusX="2" RadiusY="2" RadiusY="2" HorizontalAlignment="Center"></stackpanel></button>	Focusable="False"		Value="Transparent"/>			
Template="{StaticResource	Height="17"		• •			
PreviousButtonTemplate}" <setter.value> Width="17"/> <rectangle <controltemplate="" calendar}"="" part_nextbutton"="" targettype="{x:Type} <Button x:Name=" x:name="PART_DisabledVisual"> Calendar}"> Focusable="False" Fill="{StaticResource DisabledColor}" <stackpanel height="17" horizontalalignment="Center" opacity="0" radiusx="2" radiusy="2" x:name="PART_Root"> HorizontalAlignment="Center"> HorizontalAlignment="Center"> HorizontalAlignment="Center"> Calendar}"> Calendar}" Calendar}" Calendar}" Calendar}" Calendar}" Calendar}" Calendar</stackpanel></rectangle></setter.value>						
Width="17"/>			<setter property="Template"></setter>			
<pre><button calendar}"="" x:name="PART_DisabledVisual"> Focusable="False" Fill="{StaticResource DisabledColor}" <stackpanel height="17" horizontalalignment="Center" opacity="0" radiusx="2" radiusy="2" x:name="PART_Root"></stackpanel></button></pre>						
Focusable="False" Fill="{StaticResource DisabledColor}" <stackpanel height="17" horizontalalignment="Center" opacity="0" radiusx="2" radiusy="2" x:name="PART_Root"></stackpanel>		•	<controltemplate 17"="" horizontalalignment="Center" opacity="0" radiusx="2" radiusy="2" targettype="{x:Type</td></tr><tr><td>Height="></controltemplate>	<button <="" td="" x:name="PART_NextButton"><td>x:Name="PART_DisabledVisual"</td><td>Calendar}"></td></button>	x:Name="PART_DisabledVisual"	Calendar}">
	Focusable="False"		<stackpanel <="" td="" x:name="PART_Root"></stackpanel>			
Template="{StaticResource Stroke="{StaticResource DisabledColor}" <calendaritem< td=""><td>Height="17"</td><td>Opacity="0" RadiusX="2" RadiusY="2"</td><td>· ·</td></calendaritem<>	Height="17"	Opacity="0" RadiusX="2" RadiusY="2"	· ·			
	Template="{StaticResource	Stroke="{StaticResource DisabledColor}"	<calendaritem< td=""></calendaritem<>			

Template="{DynamicResource	<visualstate x:name="Pressed"></visualstate>	<visibility>Visible</visibility>
CalendarItemTemplate}"	<storyboard></storyboard>	
x:Name="PART_CalendarItem"	<doubleanimation <="" duration="0" td=""><td></td></doubleanimation>	
Background="{TemplateBinding	Storyboard.TargetName="HighlightBackgr	
Background}"	ound" To="0.5"	
BorderBrush="{TemplateBinding	Storyboard.TargetProperty="Opacity"/>	
BorderBrush}"		<visualstate< td=""></visualstate<>
BorderThickness="{TemplateBinding		x:Name="CalendarButtonUnfocused">
BorderThickness}"	<visualstate x:name="Disabled"></visualstate>	<storyboard></storyboard>
Style="{TemplateBinding	<storyboard></storyboard>	<objectanimationusingkeyframes< td=""></objectanimationusingkeyframes<>
CalendarItemStyle}"/>	<doubleanimation <="" duration="0" p=""></doubleanimation>	Duration="0"
	Storyboard.TargetName="HighlightBackgr	Storyboard.TargetName="DayButtonFocus
	ound" To="0"	Visual"
	Storyboard.TargetProperty="Opacity"/>	Storyboard.TargetProperty="Visibility">
	<doubleanimation <="" duration="0" td=""><td><pre><discreteobjectkeyframe keytime="0"></discreteobjectkeyframe></pre></td></doubleanimation>	<pre><discreteobjectkeyframe keytime="0"></discreteobjectkeyframe></pre>
	Storyboard.TargetName="NormalText"	<pre><discreteobjectkeyframe.value></discreteobjectkeyframe.value></pre>
<style< td=""><td>To=".35"</td><td><visibility>Collapsed</visibility></td></style<>	To=".35"	<visibility>Collapsed</visibility>
x:Key="CalendarCalendarDayButtonStyle"	Storyboard.TargetProperty="Opacity"/>	
TargetType="{x:Type		
CalendarDayButton}">		
<setter property="Width" value="25"></setter>		
<setter property="Height" value="25"></setter>	<visualstategroup< td=""><td></td></visualstategroup<>	
<setter property="FontSize" value="10"></setter>	x:Name="SelectionStates">	
<setter property="Cursor" value="Hand"></setter>	<visualstategroup.transitions></visualstategroup.transitions>	<visualstategroup< td=""></visualstategroup<>
<setter <="" property="Background" td=""><td><visualtransition< td=""><td>x:Name="ActiveStates"></td></visualtransition<></td></setter>	<visualtransition< td=""><td>x:Name="ActiveStates"></td></visualtransition<>	x:Name="ActiveStates">
Value="Transparent"/>	GeneratedDuration="0"/>	<visualstategroup.transitions></visualstategroup.transitions>
<setter< td=""><td></td><td><visualtransition< td=""></visualtransition<></td></setter<>		<visualtransition< td=""></visualtransition<>
Property="HorizontalContentAlignment"	<visualstate x:name="Unselected"></visualstate>	GeneratedDuration="0"/>
Value="Center"/>	<visualstate x:name="Selected"></visualstate>	
<setter< td=""><td><storyboard></storyboard></td><td><visualstate x:name="Active"></visualstate></td></setter<>	<storyboard></storyboard>	<visualstate x:name="Active"></visualstate>
Property="VerticalContentAlignment"	<doubleanimation <="" duration="0" td=""><td><storyboard></storyboard></td></doubleanimation>	<storyboard></storyboard>
Value="Center"/>	Story board. Target Name = "Selected Backgro"	<coloranimation< td=""></coloranimation<>
<setter property="Template"></setter>	und" To=".75"	Duration="0"
<setter.value></setter.value>	Storyboard.TargetProperty="Opacity"/>	Storyboard.TargetName="NormalText"
<controltemplate #ffffff"<="" targettype="{x:Type</td><td></Storyboard></td><td>To=" td=""></controltemplate>		
CalendarDayButton}">		Story board. Target Property = "(Text Element.
<grid></grid>		Foreground).(SolidColorBrush.Color)"/>
<visualstatemanager.visualstategroups></visualstatemanager.visualstategroups>	<visualstategroup< td=""><td></td></visualstategroup<>	
<visualstategroup< td=""><td>x:Name="CalendarButtonFocusStates"></td><td></td></visualstategroup<>	x:Name="CalendarButtonFocusStates">	
x:Name="CommonStates">	<visualstategroup.transitions></visualstategroup.transitions>	<visualstate x:name="Inactive"></visualstate>
<visualstategroup.transitions></visualstategroup.transitions>	<visualtransition< td=""><td><storyboard></storyboard></td></visualtransition<>	<storyboard></storyboard>
<visualtransition< td=""><td>GeneratedDuration="0"/></td><td><coloranimation< td=""></coloranimation<></td></visualtransition<>	GeneratedDuration="0"/>	<coloranimation< td=""></coloranimation<>
GeneratedDuration="0:0:0.1"/>		Duration="0"
	<visualstate< td=""><td>Storyboard.TargetName="NormalText"</td></visualstate<>	Storyboard.TargetName="NormalText"
<visualstate x:name="Normal"></visualstate>	x:Name="CalendarButtonFocused">	To="#606467"
<visualstate x:name="MouseOver"></visualstate>	<storyboard></storyboard>	Storyboard.TargetProperty="(TextElement.
<storyboard></storyboard>	<objectanimationusingkeyframes< td=""><td>Foreground).(SolidColorBrush.Color)"/></td></objectanimationusingkeyframes<>	Foreground).(SolidColorBrush.Color)"/>
<doubleanimation <="" duration="0" td=""><td>Duration="0"</td><td></td></doubleanimation>	Duration="0"	
Storyboard.TargetName="HighlightBackgr	Storyboard.TargetName="DayButtonFocus	
ound" To="0.5"	Visual"	
Storyboard.TargetProperty="Opacity"/>	Storyboard.TargetProperty="Visibility">	<pre><visualstategroup x:name="DayStates"></visualstategroup></pre>
	<pre><discreteobjectkeyframe keytime="0"></discreteobjectkeyframe></pre>	<visualstategroup.transitions></visualstategroup.transitions>
	<pre><discreteobjectkeyframe.value></discreteobjectkeyframe.value></pre>	<visualtransition< td=""></visualtransition<>
v risualistatos	Discionogenicy Fame. value/	V ISUALLIANSIUUII

GeneratedDuration="0"/>	Opacity="0"	<setter property="FontSize" value="10"></setter>		
	RadiusX="20"	<setter< td=""></setter<>		
<visualstate x:name="RegularDay"></visualstate>	RadiusY="20"/>	Property="HorizontalContentAlignment"		
<visualstate x:name="Today"></visualstate>	<contentpresenter <="" td="" x:name="NormalText"><td>Value="Center"/></td></contentpresenter>	Value="Center"/>		
<storyboard></storyboard>	TextElement.Foreground="#FF333333"	<setter< td=""></setter<>		
<doubleanimation <="" duration="0" td=""><td>HorizontalAlignment="{TemplateBinding</td><td>Property="VerticalContentAlignment"</td></doubleanimation>	HorizontalAlignment="{TemplateBinding	Property="VerticalContentAlignment"		
Storyboard.TargetName="TodayBackgroun	HorizontalContentAlignment}"	Value="Center"/>		
d" To="1"	Margin="5,1,5,1"	<setter property="Template"></setter>		
Storyboard.TargetProperty="Opacity"/>	VerticalAlignment="{TemplateBinding	<setter.value></setter.value>		
<coloranimation <="" duration="0" td=""><td>VerticalContentAlignment}"/></td><td><controltemplate normaltext"<="" targettype="{x:Type</td></tr><tr><td>Storyboard.TargetName=" td=""><td><path <="" td="" x:name="Blackout"><td>CalendarButton}"></td></path></td></controltemplate></td></coloranimation>	VerticalContentAlignment}"/>	<controltemplate normaltext"<="" targettype="{x:Type</td></tr><tr><td>Storyboard.TargetName=" td=""><td><path <="" td="" x:name="Blackout"><td>CalendarButton}"></td></path></td></controltemplate>	<path <="" td="" x:name="Blackout"><td>CalendarButton}"></td></path>	CalendarButton}">
To="#FFFFFFF"	Data="M8.1772461,11.029181	<grid></grid>		
Storyboard.TargetProperty="(TextElement.	L10.433105,11.029181	<visualstatemanager.visualstategroups></visualstatemanager.visualstategroups>		
Foreground).(SolidColorBrush.Color)"/>	L11.700684,12.801641	<visualstategroup< td=""></visualstategroup<>		
	L12.973633,11.029181	x:Name="CommonStates">		
	L15.191895,11.029181	<visualstategroup.transitions></visualstategroup.transitions>		
	L12.844727,13.999395	<visualtransition< td=""></visualtransition<>		
<visualstategroup< td=""><td>L15.21875,17.060919</td><td>GeneratedDuration="0:0:0.1"/></td></visualstategroup<>	L15.21875,17.060919	GeneratedDuration="0:0:0.1"/>		
x:Name="BlackoutDayStates">	L12.962891,17.060919	VisualStateGroup.Transitions>		
<visualstategroup.transitions></visualstategroup.transitions>	L11.673828,15.256231	<visualstate x:name="Normal"></visualstate>		
<visualtransition< td=""><td>L10.352539,17.060919</td><td><visualstate x:name="MouseOver"></visualstate></td></visualtransition<>	L10.352539,17.060919	<visualstate x:name="MouseOver"></visualstate>		
GeneratedDuration="0"/>	L8.1396484,17.060919	<storyboard></storyboard>		
VisualStateGroup.Transitions>	L10.519043,14.042364 z"	<doubleanimation <="" duration="0" td=""></doubleanimation>		
<pre><visualstate x:name="NormalDay"></visualstate></pre>	Fill="#FFFFFF"	Storyboard.TargetName="Background"		
<pre><visualstate x:name="BlackoutDay"></visualstate></pre>	HorizontalAlignment="Stretch"	To=".5"		
<storyboard></storyboard>	Margin="3"	Storyboard.TargetProperty="Opacity"/>		
<doubleanimation <="" duration="0" p=""></doubleanimation>	Opacity="0"	/>		
	RenderTransformOrigin="0.5,0.5"	VisualState>		
Storyboard.TargetName="Blackout" To=".2"	Stretch="Fill"	<pre></pre> <pre><visualstate x:name="Pressed"></visualstate></pre>		
Storyboard. TargetProperty="Opacity"/>	VerticalAlignment="Stretch"/>	<storyboard> <doubleanimation <="" duration="0" td=""></doubleanimation></storyboard>		
	<rectangle< td=""><td></td></rectangle<>			
	x:Name="DayButtonFocusVisual"	Storyboard.TargetName="Background"		
<td>IsHitTestVisible="false"</td> <td>To=".5"</td>	IsHitTestVisible="false"	To=".5"		
VisualStateManager.VisualStateGroups>	RadiusX="20"	Storyboard.TargetProperty="Opacity"/>		
<rectangle <="" td="" x:name="TodayBackground"><td>RadiusY="20"</td><td></td></rectangle>	RadiusY="20"			
Fill="#FD413C"	Stroke="Transparent"	All the last of the control of		
Opacity="0"	Visibility="Collapsed"/>			
RadiusX="20"		<visualstategroup< td=""></visualstategroup<>		
RadiusY="20"/>		x:Name="SelectionStates">		
<rectangle <="" td="" x:name="SelectedBackground"><td></td><td><visualstategroup.transitions></visualstategroup.transitions></td></rectangle>		<visualstategroup.transitions></visualstategroup.transitions>		
Fill="#FD413C"		<visualtransition< td=""></visualtransition<>		
Opacity="0"		GeneratedDuration="0"/>		
RadiusX="20"	<style< td=""><td></td></style<>			
RadiusY="20"/>	x:Key="CalendarCalendarButtonStyle"	<visualstate x:name="Unselected"></visualstate>		
<border background="{TemplateBinding</td><td>TargetType=" calendarbutton}"="" {x:type=""></border>	<visualstate x:name="Selected"></visualstate>			
Background}"	<setter <="" property="Background" td=""><td><storyboard></storyboard></td></setter>	<storyboard></storyboard>		
BorderBrush="{TemplateBinding	Value="#FD413C"/>	<doubleanimation <="" duration="0" td=""></doubleanimation>		
BorderBrush}"	<setter <="" property="Foreground" td=""><td>Storyboard. TargetName = "SelectedBackgro"</td></setter>	Storyboard. TargetName = "SelectedBackgro"		
BorderThickness="{TemplateBinding	Value="#FFFFFF"/>	und" To=".75"		
BorderThickness}"/>	<setter <="" property="MinWidth" td=""><td>Storyboard.TargetProperty="Opacity"/></td></setter>	Storyboard.TargetProperty="Opacity"/>		
<rectangle< td=""><td></td><td></td></rectangle<>				
	Value="40"/>			
x:Name="HighlightBackground"	Value="40"/> <setter <="" property="MinHeight" td=""><td> </td></setter>			

Value="42"/>

 $<\!\!/Visual State Group\!\!>$

Fill="#FD413C"

<visualstategroup< th=""><th></th><th><setter <="" property="Foreground" th=""></setter></th></visualstategroup<>		<setter <="" property="Foreground" th=""></setter>	
x:Name="ActiveStates">		Value="#FFFFFF"/>	
$<\!$		<setter <="" property="Background" td=""></setter>	
<visualtransition< td=""><td></td><td>Value="#282F33"/></td></visualtransition<>		Value="#282F33"/>	
GeneratedDuration="0"/>	<pre><rectangle <="" pre="" x:name="SelectedBackground"></rectangle></pre>	<setter< td=""></setter<>	
	Fill="{TemplateBinding Background}"	Property="VerticalContentAlignment"	
<visualstate x:name="Active"></visualstate>	Opacity="0" RadiusX="20"	Value="Center"/>	
<visualstate x:name="Inactive"></visualstate>	RadiusY="20"/>	<setter <="" property="BorderBrush" td=""></setter>	
<storyboard></storyboard>	<rectangle <="" td="" x:name="Background"><td>Value="Transparent"/></td></rectangle>	Value="Transparent"/>	
<coloranimation <="" duration="0" td=""><td>Fill="{TemplateBinding Background}"</td><td><setter <="" property="BorderThickness" td=""></setter></td></coloranimation>	Fill="{TemplateBinding Background}"	<setter <="" property="BorderThickness" td=""></setter>	
Storyboard.TargetName="NormalText"	Opacity="0" RadiusX="20"	Value="0"/>	
To="#FF777777"	RadiusY="20"/>	<setter property="Template"></setter>	
Storyboard.TargetProperty="(TextElement.	<contentpresenter <="" td="" x:name="NormalText"><td><setter.value></setter.value></td></contentpresenter>	<setter.value></setter.value>	
Foreground).(SolidColorBrush.Color)"/>	TextElement.Foreground="#FFFFFF"	<controltemplate targettype="{x:Type</td></tr><tr><td></Storyboard></td><td>HorizontalAlignment=" td="" {templatebinding<=""><td>Calendar}"></td></controltemplate>	Calendar}">
	HorizontalContentAlignment}"	<stackpanel <="" td="" x:name="PART_Root"></stackpanel>	
	Margin="1,0,1,1"	HorizontalAlignment=" Center ">	
<visualstategroup< td=""><td>VerticalAlignment="{TemplateBinding</td><td><calendaritem< td=""></calendaritem<></td></visualstategroup<>	VerticalAlignment="{TemplateBinding	<calendaritem< td=""></calendaritem<>	
x:Name="CalendarButtonFocusStates">	VerticalContentAlignment}"/>	Template="{DynamicResource	
<visualstategroup.transitions></visualstategroup.transitions>	<rectangle< td=""><td>CalendarItemTemplate }"</td></rectangle<>	CalendarItemTemplate }"	
<visualtransition< td=""><td>x:Name="CalendarButtonFocusVisual"</td><td>x:Name="PART_CalendarItem"</td></visualtransition<>	x:Name="CalendarButtonFocusVisual"	x:Name="PART_CalendarItem"	
GeneratedDuration="0"/>	IsHitTestVisible="false" RadiusX="20"	Background="{TemplateBinding	
	RadiusY="20" Stroke="#FD413C"	Background}"	
<visualstate< td=""><td>Visibility="Collapsed"/></td><td>BorderBrush="{TemplateBinding</td></visualstate<>	Visibility="Collapsed"/>	BorderBrush="{TemplateBinding	
x:Name="CalendarButtonFocused">		BorderBrush}"	
<storyboard></storyboard>	<controltemplate.triggers></controltemplate.triggers>	BorderThickness="{TemplateBinding	
<objectanimationusingkeyframes< p=""></objectanimationusingkeyframes<>	<trigger <="" property="IsFocused" td=""><td>BorderThickness}"</td></trigger>	BorderThickness}"	
Duration="0"	Value="True">	Style="{TemplateBinding	
Storyboard.TargetName="CalendarButtonF	<pre><setter <="" pre="" property="Visibility"></setter></pre>	CalendarItemStyle}"/>	
ocusVisual"	TargetName="CalendarButtonFocusVisual"		
Storyboard.TargetProperty="Visibility">	Value="Visible"/>		
<pre><discreteobjectkeyframe keytime="0"></discreteobjectkeyframe></pre>			
<pre><discreteobjectkeyframe.value></discreteobjectkeyframe.value></pre>	/ControlTemplate.Triggers>		
,	1 66		
<visibility>Visible</visibility>			
<pre></pre>			
	<style <="" targettype="{x:Type</td><td>L C IN IT</td></tr><tr><td></VisualState></td><td>local:DatePickerEx}" td=""><td>ItemsControlNulTarget.xaml</td></tr><tr><td><VisualState</td><td>BasedOn="{StaticResource</td><td><ResourceDictionary</td></tr><tr><td>x:Name="CalendarButtonUnfocused"></td><td>DatePickerStyle}"></td><td>xmlns="http://schemas.microsoft.com/winf</td></tr><tr><td><Storyboard></td><td><Style.Resources></td><td>x/2006/xaml/presentation"</td></tr><tr><td><ObjectAnimationUsingKeyFrames</td><td><Style TargetType="{x:Type Calendar}"</td><td>xmlns:x="http://schemas.microsoft.com/wi</td></tr><tr><td>Duration="0"</td><td>x:Key="DatePickerEx_CustomPopup"></td><td>nfx/2006/xaml"</td></tr><tr><td>Storyboard.TargetName="CalendarButtonF</td><td><Setter Property="CalendarButtonStyle"</td><td>xmlns:bf="clr-</td></tr><tr><td>ocusVisual"</td><td>Value="{DynamicResource</td><td>namespace:INCOMSYSTEM.BehaviorsFil</td></tr><tr><td>Storyboard.TargetProperty="Visibility"></td><td>CalendarCalendarButtonStyle}"/></td><td>es"></td></tr><tr><td><DiscreteObjectKeyFrame KeyTime="0"></td><td><Setter</td><td> bf:AnyItemsToVisibilityConverter</td></tr><tr><td><DiscreteObjectKeyFrame.Value></td><td>Property="CalendarDayButtonStyle"</td><td>x:Key="AnyItemsToVisibilityConverter"/></td></tr><tr><td><Visibility>Collapsed</Visibility></td><td>Value="{DynamicResource</td><td><Style TargetType="{x:Type</td></tr><tr><td></DiscreteObjectKeyFrame.Value></td><td>CalendarCalendarDayButtonStyle}"/></td><td>ItemsControl}" x:Key="ItemsVisible"></td></tr><tr><td></DiscreteObjectKeyFrame></td><td><Setter Property="FontFamily"</td><td><Setter Property="Template"></td></tr><tr><td></ObjectAnimationUsingKeyFrames></td><td>Value="Trebuchet MS"/></td><td><Setter.Value></td></tr></tbody></table></style>		

<controltemplate 2"="" 5="" targettype="{x:Type</th><th>Value="></controltemplate>	Value="White"/>			
ItemsControl}">	<setter <="" property="TextBox.IsTabStop" td=""><td><setter <="" property="Background" td=""></setter></td></setter>	<setter <="" property="Background" td=""></setter>		
<scrollviewer< td=""><td>Value="True"/></td><td>Value="Transparent"/></td></scrollviewer<>	Value="True"/>	Value="Transparent"/>		
VerticalScrollBarVisibility="Auto"	<setter <="" property="TextBox.Margin" td=""><td><setter <="" property="Cursor" td=""></setter></td></setter>	<setter <="" property="Cursor" td=""></setter>		
HorizontalScrollBarVisibility=" Disabled ">	Value="0"/>	Value="IBeam"/>		
<grid></grid>	<setter <="" property="TextBox.CaretBrush" td=""><td><setter property="Template"></setter></td></setter>	<setter property="Template"></setter>		
<textblock< td=""><td>Value="White"/></td><td><setter.value></setter.value></td></textblock<>	Value="White"/>	<setter.value></setter.value>		
Margin="30 5 0 0"	<setter <="" property="TextBox.Background" td=""><td><controltemplate< td=""></controltemplate<></td></setter>	<controltemplate< td=""></controltemplate<>		
Text="Список пуст"	Value="Transparent"/>	TargetType="controls:InputTextBox">		
VerticalAlignment="Top"	<setter <="" property="TextBox.Cursor" td=""><td><grid background="{TemplateBinding</td></tr><tr><td>Style=" td="" textblockstyle}"<="" {staticresource=""><td>Value="IBeam"/></td><td>Background}"</td></grid></td></setter>	<grid background="{TemplateBinding</td></tr><tr><td>Style=" td="" textblockstyle}"<="" {staticresource=""><td>Value="IBeam"/></td><td>Background}"</td></grid>	Value="IBeam"/>	Background}"
Visibility="{Binding Path=Items,	<setter property="TextBox.Template"></setter>	Width="{TemplateBinding Width}"		
RelativeSource={RelativeSource	<setter.value></setter.value>	x:Name="PART_Grid">		
TemplatedParent},	<controltemplate></controltemplate>	<border borderbrush="{TemplateBinding</td></tr><tr><td>UpdateSourceTrigger=PropertyChanged,</td><td><Grid Background=" td="" {templatebinding<=""><td>BorderBrush}"</td></border>	BorderBrush}"	
Converter={StaticResource	TextBox.Background}"	BorderThickness="{TemplateBinding		
AnyItemsToVisibilityConverter}}"/>	Width="{TemplateBinding	BorderThickness}"		
<itemspresenter></itemspresenter>	TextBox.Width}"	CornerRadius="5"		
	x:Name="PART_Grid">	Width="{TemplateBinding ActualWidth}"		
	<border <="" borderbrush="White" td=""><td>HorizontalAlignment="Left"</td></border>	HorizontalAlignment=" Left "		
	BorderThickness="0 0 0 1"	VerticalAlignment=" Bottom "		
	Margin="5 0"	x:Name="PART_Border"/>		
	x:Name="PART_Border"/>	<scrollviewer margin="{TemplateBinding</td></tr><tr><td></td><td><pre><ScrollViewer Margin=" pre="" {templatebinding<=""></scrollviewer>	Padding}"	
	TextBox.Padding}"	Foreground="{TemplateBinding		
M : 0, 1	Foreground="{TemplateBinding	Foreground}"		
MainStyle.xaml	TextBox.Foreground}"	x:Name="PART_ContentHost"/>		
<resourcedictionary< td=""><td>x:Name="PART_ContentHost"/></td><td></td></resourcedictionary<>	x:Name="PART_ContentHost"/>			
xmlns="http://schemas.microsoft.com/winf		<controltemplate.triggers></controltemplate.triggers>		
x/2006/xaml/presentation"		<trigger <="" property="IsPlaceHolder" td=""></trigger>		
xmlns:x="http://schemas.microsoft.com/wi		Value="True">		
nfx/2006/xaml"		<pre><setter <="" pre="" targetname="PART_ContentHost"></setter></pre>		
xmlns:controls="clr-		Property="Opacity" Value="0.7"/>		
namespace:INCOMSYSTEM.Controls">	<style <="" td="" x:key="TextBlockStyle"><td></Trigger></td></tr><tr><td><Style x:Key="MainStyle"></td><td>TargetType="TextBlock"</td><td></ControlTemplate.Triggers></td></tr><tr><td><Setter Property="TextBlock.FontFamily"</td><td>BasedOn="{StaticResource</td><td></ControlTemplate></td></tr><tr><td>Value="/INCOMSYSTEM;component/Fon</td><td>MainStyle}"></style>			
ts/#Roboto"/>	<style <="" td="" x:key="TextBoxStyle"><td></Setter></td></tr><tr><td><Setter Property="TextBlock.FontSize"</td><td>TargetType="TextBox"</td><td></style>			
Value="16"/>	BasedOn="{StaticResource	<style <="" targettype="PasswordBox" td=""></tr><tr><td><Setter Property="TextBlock.Foreground"</td><td>TextPassBoxStyle}"></style>	BasedOn="{StaticResource	
Value="White"/>	<style< td=""><td>TextPassBoxStyle}"></td></style<>	TextPassBoxStyle}">		
<setter <="" property="TextBox.IsTabStop" td=""><td>TargetType="controls:InputTextBox"</td><td><style <="" td="" x:key="LabelStyle"></tr><tr><td>Value="False"/></td><td>BasedOn="{StaticResource MainStyle}"></td><td>TargetType="Label"</td></tr><tr><td></style></td><td><setter <="" property="BorderBrush" td=""><td>BasedOn="{StaticResource</td></setter></td></setter>	TargetType="controls:InputTextBox"	<style <="" td="" x:key="LabelStyle"></tr><tr><td>Value="False"/></td><td>BasedOn="{StaticResource MainStyle}"></td><td>TargetType="Label"</td></tr><tr><td></style>	<setter <="" property="BorderBrush" td=""><td>BasedOn="{StaticResource</td></setter>	BasedOn="{StaticResource
<style <="" td="" x:key="TextPassBoxStyle"><td>Value="White"/></td><td>MainStyle}"></style>				
BasedOn="{StaticResource MainStyle}">	<setter <="" property="BorderThickness" td=""><td><style <="" targettype="RichTextBox" td=""></tr><tr><td><Setter Property="TextBox.BorderBrush"</td><td>Value="0 0 0 1"/></td><td>BasedOn="{StaticResource MainStyle}"></td></tr><tr><td>Value="White"/></td><td><Setter Property="Padding" Value="3 2"/></td><td><Setter Property="Background"</td></tr><tr><td><Setter</td><td><Setter Property="IsTabStop"</td><td>Value="Transparent"/></td></tr><tr><td>Property="TextBox.BorderThickness"</td><td>Value="True"/></td><td><pre><Setter Property="BorderBrush"</pre></td></tr><tr><td></td><td></td><td>Troperty Dorder Diam</td></tr><tr><td>Value="0 0 0 1"/></td><td><Setter Property="Margin" Value="0"/></td><td>Value="Transparent"/></td></tr></tbody></table></style></td></setter>	<style <="" targettype="RichTextBox" td=""></tr><tr><td><Setter Property="TextBox.BorderBrush"</td><td>Value="0 0 0 1"/></td><td>BasedOn="{StaticResource MainStyle}"></td></tr><tr><td>Value="White"/></td><td><Setter Property="Padding" Value="3 2"/></td><td><Setter Property="Background"</td></tr><tr><td><Setter</td><td><Setter Property="IsTabStop"</td><td>Value="Transparent"/></td></tr><tr><td>Property="TextBox.BorderThickness"</td><td>Value="True"/></td><td><pre><Setter Property="BorderBrush"</pre></td></tr><tr><td></td><td></td><td>Troperty Dorder Diam</td></tr><tr><td>Value="0 0 0 1"/></td><td><Setter Property="Margin" Value="0"/></td><td>Value="Transparent"/></td></tr></tbody></table></style>		

Value="0"/>		Height="30"	
<setter <="" property="IsReadOnly" td=""><td><trigger property="Width" value="Auto"></trigger></td><td>HorizontalAlignment="Center"</td></setter>	<trigger property="Width" value="Auto"></trigger>	HorizontalAlignment="Center"	
Value="True"/>	<setter <="" property="MinWidth" td=""><td>Margin="4,3,4,3"</td></setter>	Margin="4,3,4,3"	
	Value="100"/>	Stretch="Fill"	
<style targettype="ContextMenu"></td><td></Trigger></td><td>Stroke="Wheat"</td></tr><tr><td><Setter Property="Background"</td><td><Trigger Property="Height"</td><td>Fill="#FFFFFF"</td></tr><tr><td>Value="Gray"/></td><td>Value="Auto"></td><td>StrokeThickness="2"</td></tr><tr><td><Setter Property="Foreground"</td><td><Setter Property="MinHeight"</td><td>VerticalAlignment="Center"</td></tr><tr><td>Value="White"/></td><td>Value="20"/></td><td>Width="20"></td></tr><tr><td><Setter Property="Template"></td><td></Trigger></td><td><Path.Data></td></tr><tr><td><Setter.Value></td><td></ControlTemplate.Triggers></td><td>M5,5 h50 c0,0 40,30 80,0 c0,0 -40,-30 -80,0</td></tr><tr><td><ControlTemplate</td><td></ControlTemplate></td><td>m80,0 h75 c0,0 40,30 80,0 c0,0 -40,-30 -</td></tr><tr><td>TargetType="ContextMenu"></td><td></Setter.Value></td><td>80,0</td></tr><tr><td><Border x:Name="Border"</td><td></Setter></td><td>m80,0 h75 c0,0 40,30 80,0 c0,0 -40,-30 -</td></tr><tr><td>Background="{TemplateBinding</td><td></style>	80,0		
Background}" Padding="3">	<style <="" targettype="{x:Type Button}" td=""><td>m80,0 h75 c0,0 40,30 80,0 c0,0 -40,-30 -</td></tr><tr><td><StackPanel IsItemsHost="True"</td><td>x:Key="BtnDatePickerStyle"></td><td>80,0</td></tr><tr><td>KeyboardNavigation.DirectionalNavigation</td><td><Setter Property="Cursor" Value="Hand"</td><td>m80,0 h50 v50 H5 V5 h1</td></tr><tr><td>="Cycle"/></td><td>/></td><td>M5,55 v130 h645 V55 z</td></tr><tr><td></Border></td><td><Setter Property="Template"></td><td>M5,195 h640 c0,0 2,5 0,1 h-640 c0,0 -2,-5</td></tr><tr><td></ControlTemplate></td><td><Setter.Value></td><td>0,-1</td></tr><tr><td></Setter.Value></td><td><ControlTemplate TargetType="{x:Type</td><td>M5,205 h640 c0,0 2,5 0,1 h-640 c0,0 -2,-5</td></tr><tr><td></Setter></td><td>Button}"></td><td>0,-1</td></tr><tr><td></style>	<contentpresenter></contentpresenter>	M40,70 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-5
<style <="" td="" x:key="DatePickerTextBoxStyle"><td></ControlTemplate></td><td>M160,70 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-</td></tr><tr><td>TargetType="{x:Type</td><td></setter.Value></td><td>5</td></tr><tr><td>DatePickerTextBox}"</td><td></setter></td><td>M280,70 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-</td></tr><tr><td>BasedOn="{StaticResource MainStyle}"></td><td></style>	5		
<setter <="" property="CaretBrush" td=""><td><style <="" td="" x:key="DatePickerStyle"><td>M420,70 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-</td></tr><tr><td>Value="White"/></td><td>TargetType="{x:Type DatePicker}"></td><td>5</td></tr><tr><td><Setter Property="BorderBrush"</td><td><pre><Setter Property="Template"></pre></td><td>M560,70 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-</td></tr><tr><td>Value="White"/></td><td><Setter Value></td><td>5</td></tr><tr><td><pre><Setter Property="Margin" Value="0 4"/></pre></td><td><pre><ControlTemplate TargetType="{x:Type</pre></td><td>M40,100 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-</td></tr><tr><td></td><td>DatePicker}"></td><td>5</td></tr><tr><td><Setter Property="Template"></td><td></td><td></td></tr><tr><td><Setter.Value> <ControlTemplate TargetType="{x:Type</td><td><Grid Column Definitions</td><td>M160,100 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td>1 0 11 . 11</td><td><Grid.ColumnDefinitions> <ColumnDefinition /></td><td>0,-5</td></tr><tr><td>DatePickerTextBox}"></td><td></td><td>M280,100 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td><Border CornerRadius="5"</td><td><ColumnDefinition Width="Auto" /></td><td>0,-5</td></tr><tr><td>BorderBrush="{TemplateBinding</td><td></Grid.ColumnDefinitions></td><td>M420,100 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td>BorderBrush}"</td><td><DatePickerTextBox</td><td>0,-5</td></tr><tr><td>BorderThickness="0 0 0 1"></td><td>x:Name="PART_TextBox"</td><td>M560,100 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td><ScrollViewer</td><td>IsReadOnly="True"</td><td>0,-5</td></tr><tr><td>x:Name="PART_ContentHost"/></td><td>BorderBrush="{TemplateBinding</td><td>M40,130 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-</td></tr><tr><td></Border></td><td>BorderBrush}"</td><td>5</td></tr><tr><td><ControlTemplate.Triggers></td><td>Style="{StaticResource</td><td>M160,130 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td><Trigger Property="IsEnabled"</td><td>DatePickerTextBoxStyle}"/></td><td>0,-5</td></tr><tr><td>Value="False"></td><td><Button</td><td>M280,130 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td><Setter Property="Background"</td><td>Background="Transparent"</td><td>0,-5</td></tr><tr><td>Value="{DynamicResource {x:Static</td><td>Grid.Column="1"</td><td>M420,130 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td>SystemColors.ControlBrushKey}}"/></td><td>Style="{StaticResource</td><td>0,-5</td></tr><tr><td><Setter Property="Foreground"</td><td>BtnDatePickerStyle}"</td><td>M560,130 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td>Value-"(Dynamic Decourses (v. Statio</td><td>x:Name="PART_Button"></td><td>0,-5</td></tr><tr><td>Value="{DynamicResource {x:Static</td><td>A.Tume= Truct_Button ></td><td>0, 5</td></tr></tbody></table></style></td></setter>	<style <="" td="" x:key="DatePickerStyle"><td>M420,70 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-</td></tr><tr><td>Value="White"/></td><td>TargetType="{x:Type DatePicker}"></td><td>5</td></tr><tr><td><Setter Property="BorderBrush"</td><td><pre><Setter Property="Template"></pre></td><td>M560,70 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-</td></tr><tr><td>Value="White"/></td><td><Setter Value></td><td>5</td></tr><tr><td><pre><Setter Property="Margin" Value="0 4"/></pre></td><td><pre><ControlTemplate TargetType="{x:Type</pre></td><td>M40,100 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-</td></tr><tr><td></td><td>DatePicker}"></td><td>5</td></tr><tr><td><Setter Property="Template"></td><td></td><td></td></tr><tr><td><Setter.Value> <ControlTemplate TargetType="{x:Type</td><td><Grid Column Definitions</td><td>M160,100 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td>1 0 11 . 11</td><td><Grid.ColumnDefinitions> <ColumnDefinition /></td><td>0,-5</td></tr><tr><td>DatePickerTextBox}"></td><td></td><td>M280,100 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td><Border CornerRadius="5"</td><td><ColumnDefinition Width="Auto" /></td><td>0,-5</td></tr><tr><td>BorderBrush="{TemplateBinding</td><td></Grid.ColumnDefinitions></td><td>M420,100 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td>BorderBrush}"</td><td><DatePickerTextBox</td><td>0,-5</td></tr><tr><td>BorderThickness="0 0 0 1"></td><td>x:Name="PART_TextBox"</td><td>M560,100 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td><ScrollViewer</td><td>IsReadOnly="True"</td><td>0,-5</td></tr><tr><td>x:Name="PART_ContentHost"/></td><td>BorderBrush="{TemplateBinding</td><td>M40,130 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5 0,-</td></tr><tr><td></Border></td><td>BorderBrush}"</td><td>5</td></tr><tr><td><ControlTemplate.Triggers></td><td>Style="{StaticResource</td><td>M160,130 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td><Trigger Property="IsEnabled"</td><td>DatePickerTextBoxStyle}"/></td><td>0,-5</td></tr><tr><td>Value="False"></td><td><Button</td><td>M280,130 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td><Setter Property="Background"</td><td>Background="Transparent"</td><td>0,-5</td></tr><tr><td>Value="{DynamicResource {x:Static</td><td>Grid.Column="1"</td><td>M420,130 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td>SystemColors.ControlBrushKey}}"/></td><td>Style="{StaticResource</td><td>0,-5</td></tr><tr><td><Setter Property="Foreground"</td><td>BtnDatePickerStyle}"</td><td>M560,130 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5</td></tr><tr><td>Value-"(Dynamic Decourses (v. Statio</td><td>x:Name="PART_Button"></td><td>0,-5</td></tr><tr><td>Value="{DynamicResource {x:Static</td><td>A.Tume= Truct_Button ></td><td>0, 5</td></tr></tbody></table></style>		

-	C. 1. "(N. 11)"	/0.1.	
5 MICO 100 150 -0.0 5 5 0 5 1 50 -0.0 5 5	Style="{x:Null}"		
M160,160 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5	Width="20"/>		
0,-5		Wr. J. G. J.	
M280,160 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5	<textblock <="" grid.column="1" td="" text="{TemplateBinding</td><td>WindowStyle.xaml</td></tr><tr><td>0,-5</td><td>Header}"><td><resourcedictionary< td=""></resourcedictionary<></td></textblock>	<resourcedictionary< td=""></resourcedictionary<>	
M420,160 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5	FontSize="15">	xmlns="http://schemas.microsoft.com/winf	
0,-5	<textblock.tooltip></textblock.tooltip>	x/2006/xaml/presentation"	
M560,160 h50 c0,0 5,5 0,5 h-50 c0,0 -5,-5	<binding http:="" relativesource="{RelativeSource</td><td>xmlns:x=" schemas.microsoft.com="" td="" wi<=""></binding>		
0,-5	Mode= Self }" Path="Text"/>	nfx/2006/xaml">	
		<style <="" targettype="{x:Type Button}" td=""></tr><tr><td></Path></td><td></TextBlock></td><td>x:Key="MainButton"</td></tr><tr><td></Button></td><td></Grid></td><td>BasedOn="{StaticResource {x:Type</td></tr><tr><td><Grid</td><td></Border></td><td>Button}}"></td></tr><tr><td>Grid.Column="0"</td><td><ControlTemplate.Triggers></td><td><Setter Property="Background"</td></tr><tr><td>Grid.ColumnSpan="2"</td><td><DataTrigger Binding="{Binding</td><td>Value="#2FAA87"/></td></tr><tr><td>IsHitTestVisible="False"</td><td>ElementName=IconMenu, Path=Content}"</td><td><Setter Property="Foreground"</td></tr><tr><td>Opacity="0"</td><td>Value="{x:Null}"></td><td>Value="White"/></td></tr><tr><td>x:Name="PART_DisabledVisual"></td><td><Setter TargetName="BorderIcon"</td><td><Setter Property="BorderBrush"</td></tr><tr><td><Popup</td><td>Property="Visibility" Value="Collapsed"/></td><td>Value="Coral"/></td></tr><tr><td>AllowsTransparency="True"</td><td></DataTrigger></td><td><Setter Property="BorderThickness"</td></tr><tr><td>Placement="Bottom"</td><td><Trigger Property="IsMouseOver"</td><td>Value="2"/></td></tr><tr><td>PlacementTarget="{Binding</td><td>Value="True"></td><td><Setter Property="Padding" Value="15 5"/></td></tr><tr><td>ElementName=PART_TextBox}"</td><td><Setter Property="Background"</td><td><Setter Property="Template"></td></tr><tr><td>StaysOpen="False"</td><td>Value="#5d514b"/></td><td><Setter.Value></td></tr><tr><td>x:Name="PART_Popup" /></td><td></Trigger></td><td><math display="block"><\!\!ControlTemplate TargetType="\{x:Type</math></td></tr><tr><td></Grid></td><td></ControlTemplate.Triggers></td><td>Button}"></td></tr><tr><td></Grid></td><td></ControlTemplate></td><td><Border BorderBrush="{TemplateBinding</td></tr><tr><td></ControlTemplate></td><td></Setter.Value></td><td>BorderBrush}"</td></tr><tr><td></Setter.Value></td><td></Setter></td><td>BorderThickness="{TemplateBinding</td></tr><tr><td></Setter></td><td></style>	BorderThickness}"
	<style <="" targettype="CheckBox" td=""><td>Padding="{TemplateBinding Padding}"</td></tr><tr><td><Style TargetType="MenuItem"></td><td>x:Key="ClearCheckBoxTemplate"></td><td>Background="{TemplateBinding</td></tr><tr><td><Setter Property="Template"></td><td><Setter Property="Cursor" Value="Hand"/></td><td>Background}"</td></tr><tr><td><Setter.Value></td><td><Setter Property="Template"></td><td>Cursor="{TemplateBinding Cursor}"</td></tr><tr><td><ControlTemplate</td><td><Setter.Value></td><td>CornerRadius="5"></td></tr><tr><td>TargetType="MenuItem"></td><td><ControlTemplate</td><td><TextBlock Text="{TemplateBinding</td></tr><tr><td><Border MaxWidth="175" MinWidth="30"</td><td>TargetType="CheckBox"></td><td>Content}" Foreground="{TemplateBinding</td></tr><tr><td>Background="{TemplateBinding</td><td><Label Content="{TemplateBinding</td><td>Foreground}" Style="{StaticResource</td></tr><tr><td>Background}" Padding="5"></td><td>Content}" Padding="0" Style="{x:Null}"/></td><td>TextBlockStyle}" /></td></tr><tr><td><Grid></td><td></ControlTemplate></td><td></Border></td></tr><tr><td><Grid.ColumnDefinitions></td><td></Setter.Value></td><td></ControlTemplate></td></tr><tr><td><ColumnDefinition Width="Auto"/></td><td></Setter></td><td></Setter.Value></td></tr><tr><td><ColumnDefinition Width="Auto"/></td><td></style>		
	<style <="" td="" x:key="SideBarButton"><td><Style.Triggers></td></tr><tr><td><Border x:Name="BorderIcon"</td><td>TargetType="Button"></td><td><Trigger Property="IsEnabled"</td></tr><tr><td>BorderBrush="White"</td><td><pre><Setter Property="Cursor" Value="Hand"/></pre></td><td>Value="False"></td></tr><tr><td>BorderThickness="0 0 1 0"</td><td><pre><Setter Property="Template"></pre></td><td><Setter Property="Background"</td></tr><tr><td>Padding="0 0 5 0"</td><td><Setter.Value></td><td>Value="Gray"/></td></tr><tr><td>Margin="0 0 5 0"></td><td><pre><ControlTemplate TargetType="Button"></pre></td><td><Setter Property="Foreground"</td></tr><tr><td><Label x:Name="IconMenu"</td><td><ContentPresenter/></td><td>Value="#FFAE5F"/></td></tr><tr><td>Content="{TemplateBinding Icon}"</td><td></ControlTemplate></td><td></Trigger></td></tr><tr><td>Padding="0"</td><td></setter.Value></td><td><pre><Trigger Property="IsMouseOver"</pre></td></tr><tr><td>Manaia 101</td><td>/C-44-2</td><td>Value "Topotty" islitiouseover</td></tr></tbody></table></style>		

</Setter>

Value="True">

Margin="0"

```
Property="Background"
                                                                 INCOMSYSTEM. Windows;
                                                                                                      public Visibility CanSetFactCompleteDate
<Setter
                                                   using
Value="#3946AA"/>
                                                   public
                                                              partial
                                                                           class
                                                                                     Orders
                                                                                                      =>
</Trigger>
                                                                                                      MainWindow.AuthUser?.idPos == 2 &&
                     Property="IsPressed"
<Trigger
                                                   public
                                                            long
                                                                   id
                                                                        {
                                                                             get;
                                                                                    set;
                                                                                                      factDateStart != null && factDateComplete
Value="True">
                                                   public long idCustomer { get; set; }
                                                                                                                                           null
<Setter
                  Property="Background"
                                                   public Nullable<long> idExecutor { get; set;
                                                                                                                              Visibility.Visible
Value="#171717"/>
                                                                                                                           Visibility. Collapsed;
                                                                                                                       IsStartLateYellow
</Trigger>
                                                   public
                                                                 idTask
                                                                                                      public
                                                                                                               bool
                                                           long
                                                                              get;
                                                                                    set:
                                                                          {
</Style.Triggers>
                                                   public
                                                           decimal price {
                                                                                                      factDateStart > planDateStart?.AddDays(5);
                                                                               get;
                                                                                    set;
                                                          double difficulty { get; set; }
</Style>
                                                                                                      public bool IsStartLateRed => factDateStart
                                                   public
</ResourceDictionary>
                                                   public System.DateTime dateOrder { get;
                                                                                                                             planDateComplete;
                                                                                                      public bool IsCompleteLateYellow
App.xaml
                                                   public
                                                                Nullable<System.DateTime>
                                                                                                      factDateComplete
<Application
                                                   planDateStart
                                                                                                      planDateComplete?.AddDays(5);
x:Class="INCOMSYSTEM.App"
                                                   public
                                                                Nullable<System.DateTime>
                                                                                                              bool
                                                                                                                    IsCompleteLateRed
                                                                                                      factDateComplete > planDateComplete;
xmlns="http://schemas.microsoft.com/winf
                                                   factDateStart
                                                                          get;
                                                                                  set;
x/2006/xaml/presentation"
                                                                Nullable<System.DateTime>
                                                   public
xmlns:x="http://schemas.microsoft.com/wi
                                                   planDateComplete
                                                                            get;
                                                                                   set;
nfx/2006/xam1"
                                                   public
                                                                Nullable<System.DateTime>
xmlns:bf="clr-
                                                   factDateComplete
                                                                                                      Tasks.cs
                                                                       {
                                                                            get;
                                                                                   set;
namespace:INCOMSYSTEM.BehaviorsFil
                                                                                                               INCOMSYSTEM.BehaviorsFiles;
                                                   public byte[] attachment {
                                                                                                      using
                                                                                get; set; }
es"
                                                   public string fileExtension { get; set; }
                                                                                                      namespace
                                                                                                                      INCOMSYSTEM.Context
StartupUri="Windows/AuthWindow.xaml"
                                                   public
                                                          byte
                                                                 idStatus
                                                                           {
                                                                              get;
                                                   public virtual Chats Chats { get; set; }
                                                                                                      using
                                                                                                                                       System;
<Application.Resources>
                                                   public virtual Customers Customers { get;
                                                                                                      using
                                                                                                                    System.Collections.Generic;
<ResourceDictionary>
                                                                                                      public
                                                                                                                  partial
                                                                                                                              class
                                                                                                                                         Tasks
<br/>bf:ActualWidthConverter
                                                   public virtual Employees Employees { get;
x:Key="WidthConverter"/>
                                                                                                      [System.Diagnostics.CodeAnalysis.Suppres
<ResourceDictionary.MergedDictionaries>
                                                   public virtual Statuses Statuses { get; set; }
                                                                                                      sMessage("Microsoft.Usage",
                                                                                                      "CA2214:DoNotCallOverridableMethodsIn
< Resource Dictionary
                                                   public virtual Tasks Tasks { get; set; }
Source="Resources/MainStyle.xaml"/>
                                                   public
                                                           Visibility
                                                                       CanViewDetail
                                                                                                      Constructors")]
                                                                                                      public
                                                                                                                                       Tasks()
< Resource Dictionary
                                                   Chats.idManager
                                                                            !=
                                                                                        null
Source="Resources/ButtonStyle.xaml"/>
                                                                           Visibility. Visible
<ResourceDictionary
                                                                        Visibility. Collapsed;
                                                                                                      this.Orders =
                                                                                                                     new
                                                                                                                           HashSet<Orders>():
Source="Resources/ComboBoxStyle.xaml"
                                                   public
                                                            Visibility
                                                                         CanJoinChat
/>
                                                   (Chats.idManager
                                                                                 null
                                                                                          public
                                                                                                               long
                                                                                                                      id
                                                                                                                                       set:
                                                                                                                                get;
<ResourceDictionary
                                                   MainWindow.AuthUser?.idUser
                                                                                                      public
                                                                                                              string
                                                                                                                     name
                                                                                                                             {
                                                                                                                                 get;
Source="Resources/DatePickerExStyle.xam"
                                                   Chats.idManager)
                                                                                        &&
                                                                                                      public int idSpecialization {
                                                                                                                                   get; set; }
1"/>
                                                   MainWindow.AuthUser?.idPos
                                                                                    ==
                                                                                          3
                                                                                                      public string description { get; set; }
< Resource Dictionary
                                                                            Visibility. Visible
                                                                                                      public decimal price { get; set;
Source="Resources/ItemsControlNullTarge
                                                                        Visibility. Collapsed;
                                                                                                      public Nullable<byte> discount { get; set; }
t.xaml"/>
                                                   public Visibility CanFormAgreement =>
                                                                                                      public int approxCompleteTime { get; set; }
</ResourceDictionary.MergedDictionaries>
                                                   (idExecutor != null && planDateStart !=
                                                                                                      public int supportPeriod { get; set; }
</ResourceDictionary>
                                                   null && planDateComplete != null) &&
                                                                                                      public byte[] attachment { get; set; }
                                                   MainWindow.AuthUser?.idPos
</Application.Resources>
                                                                                                      public string fileExtension { get; set; }
</Application>
                                                                           Visibility. Visible
                                                                                                      [System.Diagnostics.CodeAnalysis.Suppres
                                                                        Visibility. Collapsed;
                                                                                                      sMessage("Microsoft.Usage",
                                                   public Visibility CanSetFactStartDate =>
                                                                                                      "CA2227:CollectionPropertiesShouldBeRe
Orders.cs
                INCOMSYSTEM.Context
                                                   MainWindow.AuthUser?.idPos == 2 &&
namespace
                                                                                                      adOnly")]
{
                                                   factDateStart == null && planDateStart !=
                                                                                                      public virtual ICollection<Orders> Orders {
using
                                 System;
                                                         && planDateComplete !=
                                                                                                                          set;
                                                   9
                                                                            Visibility.Visible
using
                       System.Windows;
                                                                                                      public
                                                                                                                   virtual
                                                                                                                                Specializations
                                                                        Visibility. Collapsed;
using
              System.Collections.Generic;
                                                                                                      Specializations
                                                                                                                              get;
                                                                                                                                      set;
```

```
public bool discoutStyle => discount != null
                                                    "CA2227:CollectionPropertiesShouldBeRe
                                      10;
&&
            discount
                            >
                                                   adOnly")]
public decimal newPrice => discount != null
                                                   public virtual ICollection<Orders> Orders {
? (decimal)(price - (price * (discount /
                                                                        set;
100m)))
                                                    [System.Diagnostics.CodeAnalysis.Suppres
                                    price;
public bool discountVisible => discount !=
                                                    sMessage("Microsoft.Usage",
null
                                       0:
                                                    "CA2227:CollectionPropertiesShouldBeRe
         &&
                  discount
public
         string
                   shortDescription
                                                   adOnly")]
                                      =>
description.Length
                        >
                                        ?
                                                   public
description.Substring(0, 60)
                                                   ICollection<SpecializationsEmployee>
                              +
description;
                                                   SpecializationsEmployee { get; set; }
public
           string
                     approxString
approxCompleteTime.ConvertDay();
                                                   UsersDetail.cs
                                                    namespace
                                                                    INCOMSYSTEM.Context
Employee.cs
                INCOMSYSTEM.Context
namespace
                                                   using
                                                   using
{
                                                                  System.Collections.Generic;
                                                              partial
                                                                                  UsersDetail
using
                                 System:
                                                   public
                                                                        class
              System.Collections.Generic;
using
                               Employees
public
          partial
                     class
                                                   [System.Diagnostics.CodeAnalysis.Suppres
                                                    sMessage("Microsoft.Usage",
{
[System.Diagnostics.CodeAnalysis.Suppres
                                                    "CA2214:DoNotCallOverridableMethodsIn
sMessage("Microsoft.Usage",
                                                   Constructors")]
"CA2214:DoNotCallOverridableMethodsIn
                                                   public
                                                                                UsersDetail()
Constructors")]
public
                             Employees()
                                                   this.Messages = new HashSet<Messages>();
this.Chats
                       HashSet<Chats>():
                                                   public
                                                                  idUser {
            =
                new
                                                            long
                                                                               get;
this.Orders = new
                      HashSet<Orders>();
                                                   public
                                                            string
                                                                    login
                                                                               get;
this.SpecializationsEmployee
                                                                                get; set; }
                                                   public
                                                           string password {
HashSet<SpecializationsEmployee>();
                                                   public Nullable<long> phone { get; set; }
                                                    public
                                                            byte
                                                                   idPos
                                                                               get:
public
          override
                      string
                               ToString()
                                                    public
                                                           long
                                                                  passport { get;
{
                                                           string
                                                                  address
                                                                            { get; set;
             $"{surname}
                                                   public System.DateTime dateStart { get; set;
return
                                  {name}
{patronymic}".Trim();
                                                   public virtual Customers Customers { get;
public
        long
               idUser
public
       string
               surname
                                                   public virtual Employees Employees { get;
public
        string
                name
                        {
                            get;
                                  set;
                                                   [System. Diagnostics. Code Analysis. Suppres\\
public string patronymic { get; set; }
[System.Diagnostics.CodeAnalysis.Suppres
                                                    sMessage("Microsoft.Usage",
                                                    "CA2227:CollectionPropertiesShouldBeRe
sMessage("Microsoft.Usage",
"CA2227:CollectionPropertiesShouldBeRe
                                                   adOnly")]
adOnly")]
                                                    public
                                                             virtual
                                                                      ICollection<Messages>
public virtual ICollection<Chats> Chats {
                                                    Messages
                                                                         get;
                                                                                   set;
                                                   public virtual Positions Positions { get; set;
get;
                    set:
public virtual UsersDetail UsersDetail { get;
                                                   public
                                                              string
                                                                        SeriePassport
[System.Diagnostics.CodeAnalysis.Suppres
                                                   passport.ToString().Substring(0,
```

public

string

sMessage("Microsoft.Usage",

```
passport.ToString().Substring(4,
                                       6);
Messages.cs
namespace
                INCOMSYSTEM.Context
using
                                  System:
using
              System.Collections.Generic;
public
           partial
                      class
                                Messages
public
         long
                 id
                           get;
                                  set;
public
        long
               idChat
                            get;
public
        long
               idUser
                            get;
public
       string
               message
                             get;
public byte[] attachment { get; set;
public string fileExtension { get; set; }
public System.DateTime dateSend { get; set;
public virtual Chats Chats { get; set; }
public virtual UsersDetail UsersDetail { get;
set:
public
          string
                     shortMessage
                                       =>
                              125
                                         ?
message.Length
message.Substring(0, 125) + "...": message;
```

virtual

System;

set; }

set;

set:

4);

NumberPassport