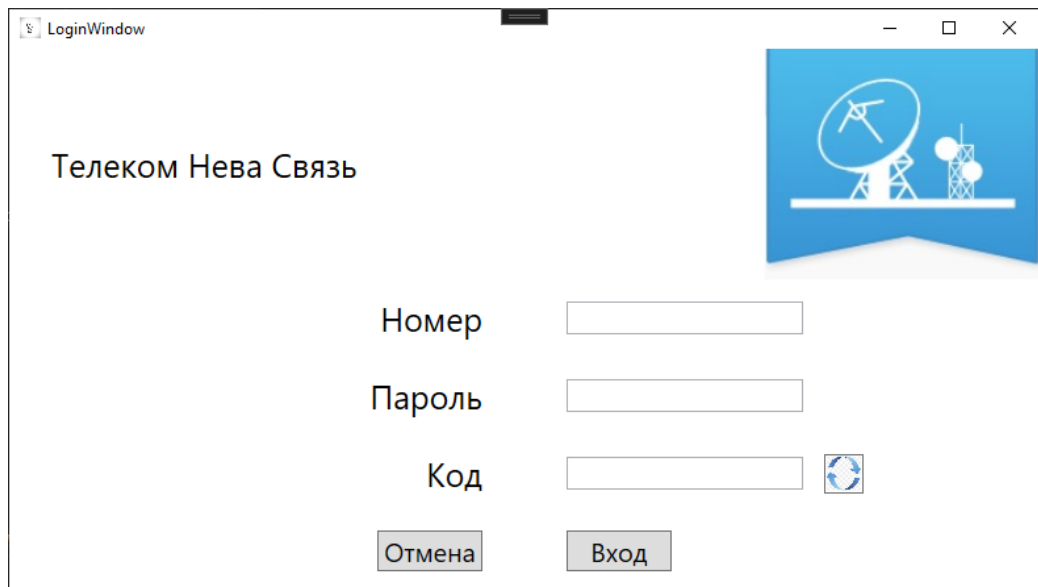


## LoginWindow.xaml



```
<Window x:Class="TelecomN.LoginWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:TelecomN"
    mc:Ignorable="d"
    Title="LoginWindow" Height="450" Width="800">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="175"/>
            <RowDefinition/>
            <RowDefinition/>
            <RowDefinition/>
            <RowDefinition/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>
        <TextBlock Text="Код" VerticalAlignment="Center" HorizontalAlignment="Right" FontSize="25" Grid.Row="3"
            Margin="32 0"/>
        <StackPanel Orientation="Horizontal" Grid.Row="3" Grid.Column="1">
            <TextBox x:Name="CodeTb" VerticalAlignment="Center" HorizontalAlignment="Left" Width="180"
                Margin="32 0 16 0" Height="25"/>
            <Button x:Name="UpdBtn" VerticalAlignment="Center" Background="Transparent" Width="30" Height="30">
                <Image VerticalAlignment="Center" HorizontalAlignment="Center" Source="\Resources\Ha
авторизацию_Обновление кода.jpg" Width="30" Height="30"/>
            </Button>
        </StackPanel>
        <Image Source="\Resources\Jloro THC.jpg" Grid.Column="1" VerticalAlignment="Center"
            HorizontalAlignment="Right"/>
        <TextBlock Text="Телеком Нева Связь" VerticalAlignment="Center" HorizontalAlignment="Left" Margin="32 0"
            FontSize="25"/>
        <TextBlock Text="Номер" VerticalAlignment="Center" HorizontalAlignment="Right" FontSize="25"
            Grid.Row="1" Margin="32 0"/>
        <TextBlock Text="Пароль" VerticalAlignment="Center" HorizontalAlignment="Right" FontSize="25"
            Grid.Row="2" Margin="32 0"/>
        <Button x:Name="CancelBtn" VerticalAlignment="Center" HorizontalAlignment="Right" Grid.Row="4"
            Margin="32 0" Content="Отмена" Width="80" FontSize="20" Click="CancelBtn_Click"/>
    </Grid>
</Window>
```

```

        <Button x:Name="AuthBtn" VerticalAlignment="Center" HorizontalAlignment="Left" Grid.Row="4" Margin="32 0" Click="AuthBtn_Click" Content="Вход" FontSize="20" Width="80" Grid.Column="1"/>
        <TextBox x:Name="LoginTb" VerticalAlignment="Center" HorizontalAlignment="left" Margin="32 0" Grid.Row="1" Grid.Column="1" Width="180" Text="{Binding Login}" Height="25" />
        <TextBox x:Name="PassTb" VerticalAlignment="Center" HorizontalAlignment="Left" Margin="32 0" Grid.Row="2" Grid.Column="1" Text="{Binding Password}" Width="180" Height="25"/>
    </Grid>
</Window>

```

## LoginWindow.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using TelecomN.Modules;
using System.Timers;

namespace TelecomN
{
    /// <summary>
    /// Логика взаимодействия для LoginWindow.xaml
    /// </summary>
    public partial class LoginWindow : Window
    {
        public LoginWindow()
        {
            InitializeComponent();
            DataContext = this;
        }

        /*кнопка отмены*/
        private void CancelBtn_Click(object sender, RoutedEventArgs e)
        {
            MessageBoxResult messageBoxResult = MessageBox.Show("Вы действительно хотите закрыть?", "Закрыть",
            MessageBoxButton.OKCancel, MessageBoxImage.Question);

            if (messageBoxResult == MessageBoxResult.OK)
            {
                this.Close();
            }
        }

        /*кнопка авторизироваться*/
        private void AuthBtn_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                User user = TelecomNevaEntities.GetContext().Users.FirstOrDefault(p => p.PhoneNumb == LoginTb.Text &&
                p.Password == PassTb.Text);
                if (user != null)
                {
                    if (CodeTb.Text == "")
                    {

```

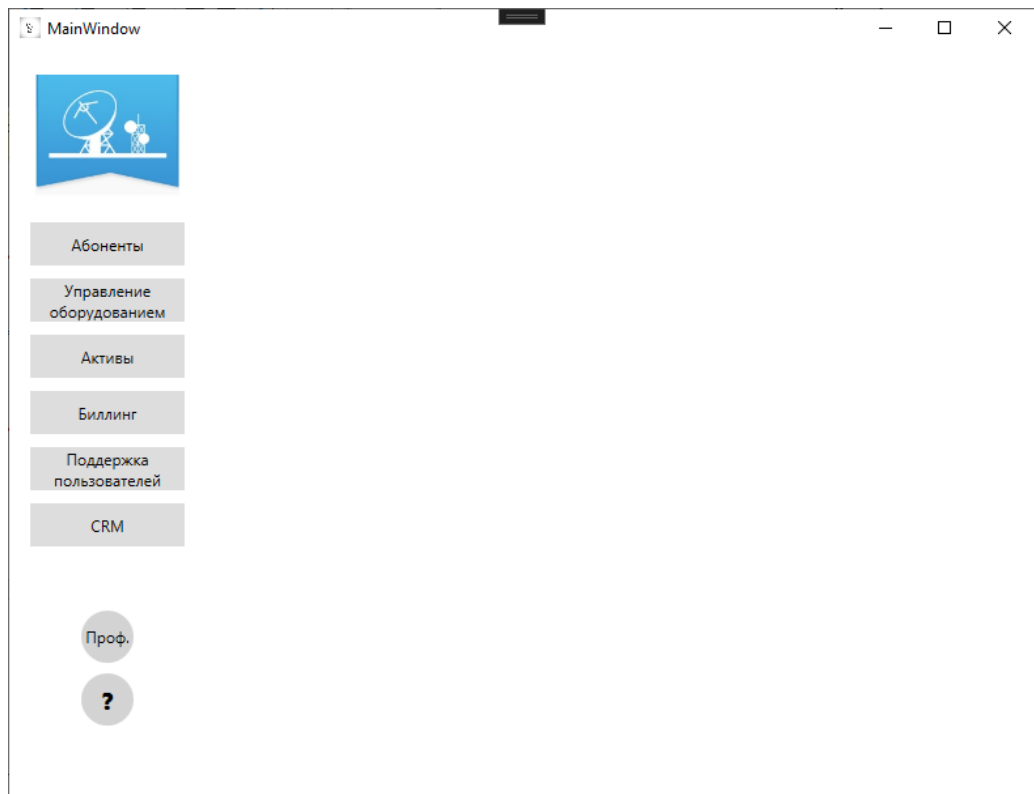
```

        CodeWindow code = new CodeWindow();
        code.Show();
    }
    if (CodeTb.Text == CodeClass.Code)
    {
        AuthStorage.Role = user.RoleID;
        AuthStorage.Name = user.PhoneNumb;
        MainWindow mainWindow = new MainWindow();
        mainWindow.Show();
        this.Close();
    }
}
else
{
    MessageBox.Show("Неверные данные, пожалуйста, попробуйте еще раз");
}
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message.ToString());
}
}

        /*кнопка для вывода кода*/
private void UpdBtn_Click(object sender, RoutedEventArgs e)
{
    CodeWindow code = new CodeWindow();
    code.Show();
}
}
}

```

## MainWindow.xaml



```
<Window x:Class="TelecomN.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:TelecomN"
    mc:Ignorable="d"
    Title="MainWindow" Height="550" Width="800">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="150"/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>
        <StackPanel Orientation="Vertical" >
            <Image Source="Resources\Лоро THC.jpg" Margin="20"/>
            <Button x:Name="AbonentBtn" BorderBrush="Transparent" Width="120" Height="35"
                VerticalAlignment="Center" HorizontalAlignment="Center" Content="Абоненты" Margin="0 0 0 8"
                Click="AbonentBtn_Click"/>
            <Button x:Name="OborBtn" Width="120" BorderBrush="Transparent" Height="35" VerticalAlignment="Center"
                HorizontalAlignment="Center" Margin="0 0 0 8" Click="OborBtn_Click">
                <TextBlock TextAlignment="Center">Управление
                <LineBreak/>оборудованием</TextBlock>
            </Button>
            <Button BorderBrush="Transparent" x:Name="ActiveBtn" Width="120" Height="35"
                VerticalAlignment="Center" HorizontalAlignment="Center" Content="Активы" Margin="0 0 0 8"
                Click="ActiveBtn_Click"/>
            <Button BorderBrush="Transparent" x:Name="BillingBtn" Width="120" Height="35"
                VerticalAlignment="Center" HorizontalAlignment="Center" Content="Биллинг" Margin="0 0 0 8"
                Click="BillingBtn_Click"/>
            <Button x:Name="SupportBtn" Width="120" Height="35" BorderBrush="Transparent"
                VerticalAlignment="Center" HorizontalAlignment="Center" Margin="0 0 0 8" Click="SupportBtn_Click">
                <TextBlock TextAlignment="Center">Поддержка
```

```

        <LineBreak/>пользователей</TextBlock>
    </Button>
    <Button BorderBrush="Transparent" x:Name="CRMBtn" Width="120" Height="35" VerticalAlignment="Center"
HorizontalAlignment="Center" Content="CRM" Margin="0 0 0 8" Click="CRMBtn_Click"/>
    <StackPanel Orientation="Vertical" VerticalAlignment="Bottom" Margin="0 40 0 0">
        <Border CornerRadius="30" Width="40" Height="40" Background="LightGray">
            <Button x:Name="ProfBtn" Background="Transparent" BorderBrush="Transparent" Content="Проф."
Click="ProfBtn_Click"/>
        </Border>
        <Border CornerRadius="30" Width="40" Height="40" Background="LightGray" Margin="0 8 0 0">
            <Button x:Name="QuestionBtn" Background="Transparent" BorderBrush="Transparent" Content="?"
FontSize="18" FontWeight="Heavy" Click="ProfBtn_Click"/>
        </Border>
    </StackPanel>
</StackPanel>
<Frame x:Name="MainFrame" ContentRendered="MainFrame_ContentRendered" NavigationUIVisibility="Hidden"
Grid.Column="1"/>
</Grid>
</Window>

```

## Mainwindow.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace TelecomN
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        /*переходит на станицу абонентов*/
        private void AbonentBtn_Click(object sender, RoutedEventArgs e)
        {
            MainFrame.Navigate(new Pages.AbonentPage());
        }

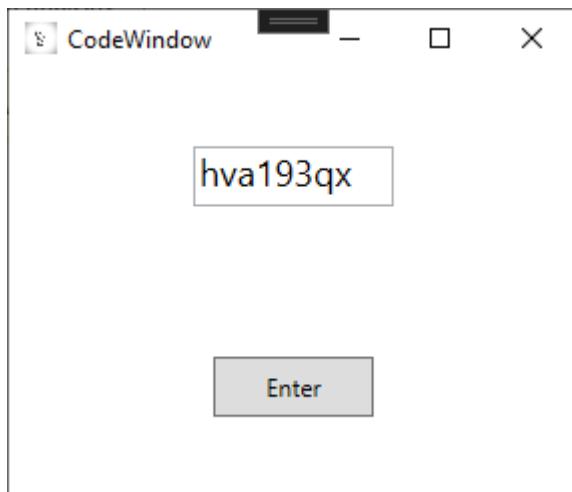
        private void OborBtn_Click(object sender, RoutedEventArgs e)
        {
        }

        private void ActiveBtn_Click(object sender, RoutedEventArgs e)

```

```
{  
  
}  
  
private void BillingBtn_Click(object sender, RoutedEventArgs e)  
{  
  
}  
  
private void SupportBtn_Click(object sender, RoutedEventArgs e)  
{  
  
}  
    /*переход на страницу CRM*/  
private void CRMBtn_Click(object sender, RoutedEventArgs e)  
{  
    MainFrame.Navigate(new Pages.CrmPage());  
}  
  
private void ProfBtn_Click(object sender, RoutedEventArgs e)  
{  
  
}  
  
private void MainFrame_ContentRendered(object sender, EventArgs e)  
{  
  
}  
}  
}
```

## CodeWindow.xaml



```
<Window x:Class="TelecomN.CodeWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:TelecomN"
        mc:Ignorable="d"
        Title="CodeWindow" Height="250" Width="300">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition/>
            <RowDefinition/>
        </Grid.RowDefinitions>
        <TextBox x:Name="CodeTb" Grid.Row="0" Width="100" Height="30" FontSize="18"
            HorizontalAlignment="Center" VerticalAlignment="Center" />
        <Button x:Name="EnterBtn" Grid.Row="1" Width="80" Height="30" Content="Enter" VerticalAlignment="Center"
            HorizontalAlignment="Center" Click="EnterBtn_Click"/>
    </Grid>
</Window>
```

## CodeWindow.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using System.IO;
using TelecomN.Modules;
using System.Timers;

namespace TelecomN
{
```

```
/// <summary>
/// Логика взаимодействия для CodeWindow.xaml
/// </summary>
public partial class CodeWindow : Window
{
    string _code { get; set; }
    public bool _isCorrect { get; set; }
    public CodeWindow()
    {
        InitializeComponent();
        _code = CodeClass.CreateCode();
        CodeTb.Text = _code;
        CodeClass.Code = _code;
    }

    /*кнопка входа*/
    private void EnterBtn_Click(object sender, RoutedEventArgs e)
    {
        this.Close();
    }
}
```



## AuthStorage.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace TelecomN.Modules
{
    internal class AuthStorage
    {
        static public int Role { get; set; }
        static public string Name { get; set; }
    }
}
```

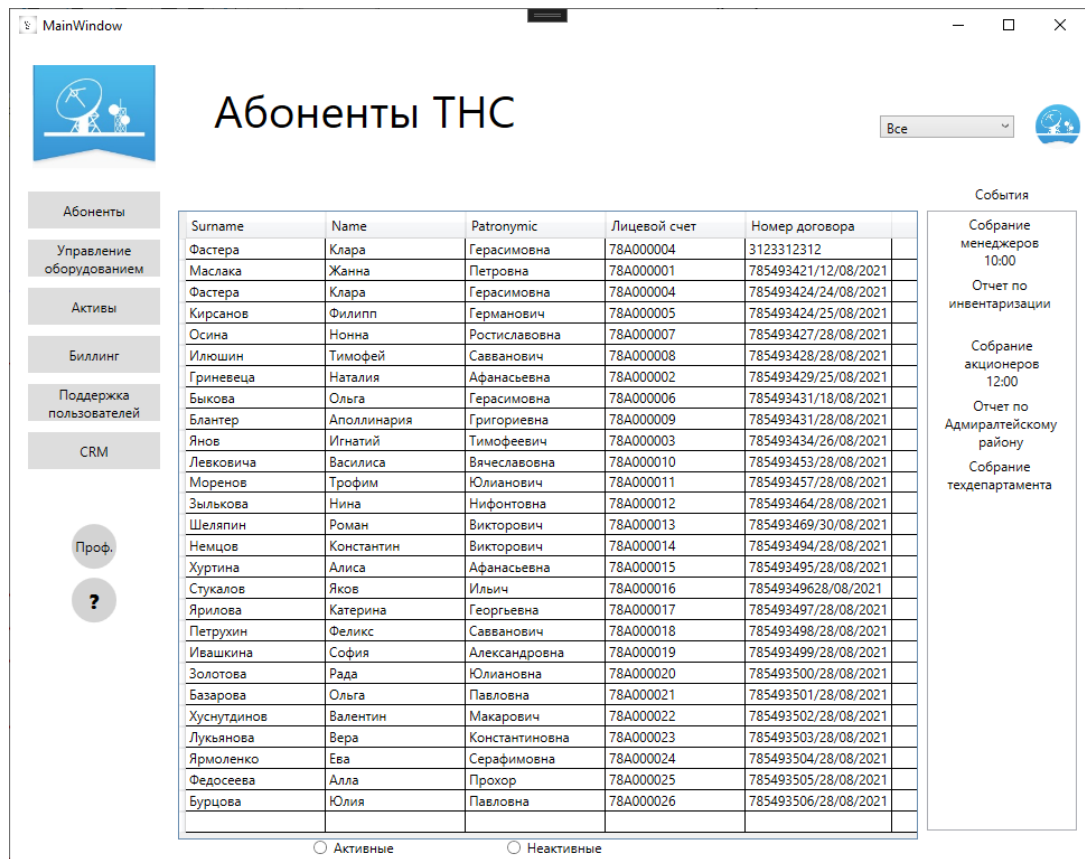
## CodeClass.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Timers;

namespace TelecomN.Modules
{
    internal class CodeClass
    {
        static public string Code;

        static public string CreateCode()
        {
            Random rnd = new Random();
            string code = "";
            string codeElement = "qwertyuiopasdfghjklzxcvbnm1234567890" ;
            for (int i = 0; i < 8; i++)
            {
                code = code + codeElement[rnd.Next(0, codeElement.Length)];
            }
            return code;
        }
    }
}
```

## AbonentPage.xaml



```
<Page x:Class="TelecomN.Pages.AbonentPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:TelecomN.Pages"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="650"
    Title="AbonentPage">

    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="150"/>
            <RowDefinition/>
            <RowDefinition Height="20"/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition Width="150"/>
        </Grid.ColumnDefinitions>
        <TextBlock Text="Абоненты THC" VerticalAlignment="Center" HorizontalAlignment="Left" FontSize="40"
            Margin="32 -32 0 0"/>
        <ComboBox x:Name="UserCb" Width="120" Height="20" SelectionChanged="UserCb_SelectionChanged"
            VerticalAlignment="Center" HorizontalAlignment="Right" Margin="64 0" Grid.ColumnSpan="2"/>
        <Ellipse Width="40" Height="40" Grid.Column="1" Margin="100 0 0 0">
            <Ellipse.Fill>
                <ImageBrush ImageSource="\Resources\Jlora THC.jpg" Stretch="UniformToFill"/>
            </Ellipse.Fill>
        </Ellipse>

        <DataGrid x:Name="AbonentData" Grid.Row="1" MinColumnWidth="125" ItemsSource="{Binding EjectApp}"
            AutoGenerateColumns="False">
```

```

        <DataGrid.Columns>
            <DataGridTextColumn Header="Surname" Binding="{Binding Abonent.LastName}"/>
            <DataGridTextColumn Header="Name" Binding="{Binding Abonent.FirstName}"/>
            <DataGridTextColumn Header="Patronymic" Binding="{Binding Abonent.Patronymic}"/>
            <DataGridTextColumn Header="Лицевой счет" Binding="{Binding Abonent.AbonentID}"/>
            <DataGridTextColumn Header="Номер договора" Binding="{Binding ApplicNumb}"/>
        </DataGrid.Columns>
    </DataGrid>
    <Border Grid.Column="1" Grid.Row="0" Width="134" Background="White" Height="30"
VerticalAlignment="Bottom">
        <TextBlock Text="События" VerticalAlignment="Center" HorizontalAlignment="Center" />
    </Border>
    <ListBox Grid.Row="1" Grid.Column="1" Margin="8 0 8 8">
        <ListBoxItem>
            <TextBlock Text="Собрание менеджеров 10:00" TextAlignment="Center" Height="50"
TextWrapping="Wrap" Margin="8 0" Width="100" VerticalAlignment="Center" HorizontalAlignment="Center"/>
        </ListBoxItem>
        <ListBoxItem>
            <TextBlock Text="Отчет по инвентаризации" TextAlignment="Center" TextWrapping="Wrap" Height="50"
Margin="8 0" Width="100" VerticalAlignment="Center" HorizontalAlignment="Center"/>
        </ListBoxItem>
        <ListBoxItem>
            <TextBlock Text="Собрание акционеров 12:00" TextAlignment="Center" TextWrapping="Wrap"
Height="50" Margin="20 0" Width="80" VerticalAlignment="Center" HorizontalAlignment="Center"/>
        </ListBoxItem>
        <ListBoxItem>
            <TextBlock Text="Отчет по Адмиралтейскому району" TextAlignment="Center" TextWrapping="Wrap"
Height="50" Margin="8 0" Width="102" VerticalAlignment="Center" HorizontalAlignment="Center"/>
        </ListBoxItem>
        <ListBoxItem>
            <TextBlock Text="Собрание техдепартамента" TextAlignment="Center" TextWrapping="Wrap"
Height="50" Margin="8 0" Width="100" VerticalAlignment="Center" HorizontalAlignment="Center"/>
        </ListBoxItem>
    </ListBox>
    <StackPanel Orientation="Horizontal" Grid.Row="2" Margin="120 0">
        <RadioButton x:Name="ActiveRb" Content="Активные" Checked="ActiveRb_Checked"/>
        <RadioButton x:Name="NotActiveRb" Content="Неактивные" Margin="100 0"
Checked="NotActiveRb_Checked"/>
    </StackPanel>
</Grid>
</Page>

```

## AbonentPage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using TelecomN.Modules;

namespace TelecomN.Pages
{

```

```

/// <summary>
/// Логика взаимодействия для AbonentPage.xaml
/// </summary>
public partial class AbonentPage : Page
{
    public List<EjectApp> Abonent { get; set; }
    public AbonentPage()
    {
        InitializeComponent();
        AbonentData.ItemsSource = null;
        Abonent = TelecomNevaEntities.GetContext().EjectApps.ToList();
        DataContext = this;
        AbonentData.ItemsSource = Abonent;
        List<Employee> employees = TelecomNevaEntities.GetContext().Employees.ToList();
        employees.Insert(0, new Employee
        {
            Surname = "Bce"
        });
        UserCb.ItemsSource = employees;
        UserCb.DisplayMemberPath = "Surname";
        UserCb.SelectedIndex = 0;
    }

    /*активные заказы*/
    private void ActiveRb_Checked(object sender, RoutedEventArgs e)
    {
        List<EjectApp> abonents = TelecomNevaEntities.GetContext().EjectApps.Where(p => p.StatusID == 1 ||
p.StatusID == 2).ToList();
        AbonentData.ItemsSource = abonents;
    }

    /*не активные заказы*/
    private void NotActiveRb_Checked(object sender, RoutedEventArgs e)
    {
        List<EjectApp> ejects = TelecomNevaEntities.GetContext().EjectApps.Where(p => p.StatusID == 3).ToList();
        AbonentData.ItemsSource = ejects;
    }

    /*метод для обновления данных*/
    private void Update()
    {
        List<EjectApp> ejects = TelecomNevaEntities.GetContext().EjectApps.ToList();
        if (UserCb.SelectedIndex > 0)
        {
            ejects = ejects.Where(p=> p.EmployeeID == (UserCb.SelectedItem as Employee).EmployeeID).ToList();
        }
        AbonentData.ItemsSource = ejects;
    }
    private void UserCb_SelectionChanged(object sender, SelectionChangedEventArgs e)
    {
        Update();
    }
}
}

```

# CrmPage.xaml

Номер заявки	Дата подключения	Лицевой счёт	Услуги	Вид услуги	Тип услуги	Статус	Описание проблемы	Тип проблемы	Дата закрытия
785493431	12/08/2021	12/08/2021 12:00:00 AM	ТВ	Подключение	Подключение услуг на существующей инфраструктуре	Новая	Подключить услуги, развода до двери есть, порт 12	ТО	8/23/2021 12:00:00 AM
785493432	04/08/2021	8/04/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Пиковая скорость соединения	Требуется выезда		ТО	8/23/2021 12:00:00 AM
785493434	15/08/2021	8/25/2021 12:00:00 AM	ТВ	Подключение	Подключение услуг на новой инфраструктуре	Визитка		ТО	8/23/2021 12:00:00 AM
785493437	08/08/2021	8/08/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Пиковая скорость соединения	Требуется выезда		ТО	
785493438	28/08/2021	8/28/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493439	13/08/2021	8/25/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493437	18/08/2021	8/18/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Низкая скорость соединения	Требуется выезда		ТО	
785493431	08/08/2021	8/08/2021 12:00:00 AM	Интернет	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493434	26/08/2021	8/26/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493431	08/08/2021	8/08/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493437	08/08/2021	8/08/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493464	28/08/2021	8/28/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493489	20/08/2021	8/20/2021 12:00:00 AM	ТВ	Подключение	Подключение услуг на новой инфраструктуре	Требуется выезда	От распределительного шкафа в подвале	ТО	
785493484	28/08/2021	8/28/2021 12:00:00 AM	ТВ	Подключение	Подключение услуг на новой инфраструктуре	Требуется выезда	От распределительного шкафа в подвале	ТО	
785493495	28/08/2021	8/28/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493498	28/08/2021	8/28/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493497	28/08/2021	8/28/2021 12:00:00 AM	Интернет	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493486	28/08/2021	8/28/2021 12:00:00 AM	Интернет	Подключение	Подключение услуг на новой инфраструктуре	Требуется выезда	От распределительного шкафа в подвале	ТО	
785493496	28/08/2021	8/28/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493500	08/08/2021	8/08/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493501	08/08/2021	8/08/2021 12:00:00 AM	Интернет	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493502	28/08/2021	8/28/2021 12:00:00 AM	Интернет	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493503	28/08/2021	8/28/2021 12:00:00 AM	ТВ	Подключение	Подключение услуг на новой инфраструктуре	Требуется выезда	От распределительного шкафа в подвале	ТО	
785493504	28/08/2021	8/28/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	
785493505	28/08/2021	8/28/2021 12:00:00 AM	ТВ	Подключение	Подключение услуг на новой инфраструктуре	Требуется выезда	От распределительного шкафа в подвале	ТО	
785493506	28/08/2021	8/28/2021 12:00:00 AM	ТВ	Диагностика и настройка оборудования/подключения	Нет доступа к услуге	Требуется выезда		ТО	

```
<Page x:Class="TelecomN.Pages.CrmPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:TelecomN.Pages"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="CrmPage">

    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="80"/>
            <RowDefinition Height="50"/>
        </Grid.RowDefinitions>
        <DataGrid x:Name="CRMDData" AutoGenerateColumns="False" ItemsSource="{Binding Application}"
            HorizontalAlignment="Center" Grid.Row="1">
            <DataGrid.Columns>
                <DataGridTextColumn Header="Номер заявки" Binding="{Binding ApplicNum}" />
                <DataGridTextColumn Header="Дата подключения" Binding="{Binding DateOfStart}" />
                <DataGridTextColumn Header="Лицевой счёт" Binding="{Binding AbonentID}" />
                <DataGridTextColumn Header="Услуги" Binding="{Binding Service.Name}" />
                <DataGridTextColumn Header="Вид услуги" Binding="{Binding ServiceView.Name}" />
                <DataGridTextColumn Header="Тип услуги" Binding="{Binding ServiceType.Name}" />
                <DataGridTextColumn Header="Статус" Binding="{Binding Status.Name}" />
                <DataGridTextColumn Header="Описание проблемы" Binding="{Binding Opisanie}" />
                <DataGridTextColumn Header="Тип проблемы" Binding="{Binding Problem.Name}" />
                <DataGridTextColumn Header="Дата закрытия" Binding="{Binding DateOfRealiz}" />
            </DataGrid.Columns>
        </DataGrid>
        <Button x:Name="AddBtn" Content="Оформить заявку" Grid.Row="2" VerticalAlignment="Center"
            HorizontalAlignment="Right" Margin="16 0" Width="120" Height="30" Click="AddBtn_Click" />
    </Grid>
</Page>
```

## CrmPage.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using TelecomN.Modules;

namespace TelecomN.Pages
{
    /// <summary>
    /// Логика взаимодействия для CrmPage.xaml
    /// </summary>
    public partial class CrmPage : Page
    {
        public List<EjectApp> Applications { get; set; }
        public CrmPage()
        {
            InitializeComponent();
            CRMDData.ItemsSource = null;
            Applications = TelecomNevaEntities.GetContext().EjectApps.ToList();
            DataContext = this;
            CRMDData.ItemsSource = Applications;
        }

        /*кнопка для доавбления новой заявки на страницу CRM*/
        private void AddBtn_Click(object sender, RoutedEventArgs e)
        {
            NavigationService.Navigate(new Pages.AddPage());
        }
    }
}
```

## AddPage.xaml

[illegible]

```
<Page x:Class="TelecomN.Pages.AddPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:TelecomN.Pages"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="AddPage">
```

[illegible]

```

        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>
        <TextBlock Text="Идентификатор" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="0"
FontSize="20"/>
        <TextBlock Text="Дата открытия заявки" VerticalAlignment="Center" HorizontalAlignment="Center"
Grid.Row="1" FontSize="20"/>
        <TextBlock Text="Абонент" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="2"
FontSize="20"/>
        <TextBlock Text="Услуга" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="3"
FontSize="20"/>
        <TextBlock Text="Тип заявки" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="4"
FontSize="20"/>
        <TextBlock Text="Тип услуги" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="5"
FontSize="20"/>
        <TextBlock Text="Статус" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="6"
FontSize="20"/>
        <TextBlock Text="Описание" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="7"
FontSize="20"/>
        <TextBlock Text="Дата реализации" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="8"
FontSize="20"/>
        <TextBlock Text="Проблема" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="9"
FontSize="20"/>
        <TextBlock Text="Сотрудник" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="10"
FontSize="20"/>
        <Button x:Name="SaveBtn" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.ColumnSpan="2"
Click="SaveBtn_Click" Grid.Row="11" Content="Сохранить" Width="100" Height="25"/>
        <TextBox Text="{Binding ApplicNumb}" VerticalAlignment="Center" HorizontalAlignment="Center"
Grid.Row="0" Grid.Column="1" Width="200"/>
        <TextBox x:Name="DayTb" Text="{Binding DateOfStart}" VerticalAlignment="Center"
HorizontalAlignment="Center" Grid.Row="1" Grid.Column="1" Width="200"/>
        <ComboBox x:Name="AbonentCb" Grid.Row="2" Grid.Column="1" SelectedItem="{Binding Abonent}"
DisplayMemberPath="AbonentID" VerticalAlignment="Center" HorizontalAlignment="Center" Width="200" />
        <ComboBox x:Name="ServiceCb" Grid.Row="3" Grid.Column="1" SelectedItem="{Binding Service}"
DisplayMemberPath="Name" VerticalAlignment="Center" HorizontalAlignment="Center" Width="200" />
        <ComboBox x:Name="SerViewCb" Grid.Row="4" Grid.Column="1" SelectedItem="{Binding ServiceView}"
DisplayMemberPath="Name" VerticalAlignment="Center" HorizontalAlignment="Center" Width="200" />
        <ComboBox x:Name="SerTypeCb" Grid.Row="5" Grid.Column="1" SelectedItem="{Binding ServiceType}"
DisplayMemberPath="Name" VerticalAlignment="Center" HorizontalAlignment="Center" Width="200" />
        <ComboBox x:Name="StatusCb" Grid.Row="6" Grid.Column="1" SelectedItem="{Binding Status}"
DisplayMemberPath="Name" VerticalAlignment="Center" HorizontalAlignment="Center" Width="200" />
        <TextBox Text="{Binding Opisanie}" VerticalAlignment="Center" HorizontalAlignment="Center" Grid.Row="7"
Grid.Column="1" Width="200"/>
        <TextBox x:Name="DaydTb" Text="{Binding DateOfRealiz}" VerticalAlignment="Center"
HorizontalAlignment="Center" Grid.Row="8" Grid.Column="1" Width="200"/>
        <ComboBox x:Name="ProblemCb" Grid.Row="9" Grid.Column="1" SelectedItem="{Binding Problem}"
DisplayMemberPath="Name" VerticalAlignment="Center" HorizontalAlignment="Center" Width="200" />
        <ComboBox x:Name="EmployeeCb" Grid.Row="10" Grid.Column="1" SelectedItem="{Binding Employee}"
DisplayMemberPath="Surname" VerticalAlignment="Center" HorizontalAlignment="Center" Width="200" />
    </Grid>
</Page>

```

## AddPage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;

```



```

using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using TelecomN.Modules;

namespace TelecomN.Pages
{
    /// <summary>
    /// Логика взаимодействия для AddPage.xaml
    /// </summary>
    public partial class AddPage : Page
    {
        public EjectApp EjectApps { get; set; }
        public AddPage()
        {
            InitializeComponent();
            EjectApps = new EjectApp();
            AbonentCb.ItemsSource = TelecomNevaEntities.GetContext().Abonents.ToList();
            EmployeeCb.ItemsSource = TelecomNevaEntities.GetContext().Employees.ToList();
            ProblemCb.ItemsSource = TelecomNevaEntities.GetContext().Problems.ToList();
            SerTypeCb.ItemsSource = TelecomNevaEntities.GetContext().ServiceTypes.ToList();
            ServiceCb.ItemsSource = TelecomNevaEntities.GetContext().Services.ToList();
            SerViewCb.ItemsSource = TelecomNevaEntities.GetContext().ServiceViews.ToList();
            StatusCb.ItemsSource = TelecomNevaEntities.GetContext().Status.ToList();
            StatusCb.SelectedIndex = 0;
            DataContext = EjectApps;
        }

        /*кнопка сохранить данные*/
        private void SaveBtn_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                EjectApps.AbonentID = (AbonentCb.SelectedItem as Abonent).AbonentID;
                EjectApps.EmployeeID = (EmployeeCb.SelectedItem as Employee).EmployeeID;
                EjectApps.ProblemID = (ProblemCb.SelectedItem as Problem).ProblemID;
                EjectApps.SerTypeID = (SerTypeCb.SelectedItem as ServiceType).SerTypeID;
                EjectApps.ServiceID = (ServiceCb.SelectedItem as Service).ServiceID;
                EjectApps.SerViewID = (SerViewCb.SelectedItem as ServiceView).SerViewID;
                EjectApps.StatusID = (StatusCb.SelectedItem as Status).StatusID;
                if (EjectApps.ApplicNumb != null)
                {
                    TelecomNevaEntities.GetContext().EjectApps.Add(EjectApps);
                }
                TelecomNevaEntities.GetContext().SaveChanges();
                NavigationService.GoBack();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.InnerException.Message);
            }
        }
    }
}

```