

Методические рекомендации по разработке клиентского приложения.

Реализация WPF-приложения с использованием Entity Framework (Database First) и страниц для управления данными

Ниже приведен общий подход к созданию приложения с заданной функциональностью. Основное внимание уделяется созданию окон (страниц) для работы с таблицами базы данных, включая использование **Entity Framework** для взаимодействия с БД.

1. Создание структуры проекта

Шаги:

1. Создание модели данных на основе Entity Framework (Database First):

- Выбираем компонент ADO.NET Entity Data Model (Database First)
- Устанавливаем соединение с БД на вашем сервере
- Задаем имя модели
- Сгенерируется модель данных в виде диаграммы классов
- Внесите изменения в созданный класс контекста. Добавьте после конструктора метод `GetContext()`, загружающий в переменную контекста `_context` весь контекст нашей базы данных `PlatezhiEntities`. Это можно сделать для удобства, чтобы избежать повторов кода при получении контекста.

```
public partial class PlatezhiEntities : DbContext
{
    public PlatezhiEntities()
        : base("name=PlatezhiEntities")
    {
    }
    private static PlatezhiEntities _context;
    public static PlatezhiEntities GetContext()
    {
        if(_context==null)
            _context = new PlatezhiEntities();
        return _context;
    }
}
```

.....

- Проверьте правильность подключения в классе `App.config`.

2. Добавьте папки:

- `Pages` – для страниц.
- `Models` – для моделей базы данных (сгенерированы автоматически).
- `ViewModels` – для логики отображения данных (по необходимости).

2. Главное окно

Главное окно будет содержать фрейм для отображения страниц и меню для навигации.

Код `MainWindow.xaml`:

```

<Window x:Class="MyApp.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Главное меню" Height="450" Width="800">
    <DockPanel>
        <!-- Верхнее меню -->
        <Menu DockPanel.Dock="Top">
            <MenuItem Header="Категории" Click="Categories_Click"/>
            <MenuItem Header="Продукция" Click="Products_Click"/>
            <MenuItem Header="Платежи" Click="Payments_Click"/>
            <MenuItem Header="Отчет" Click="Report_Click"/>
        </Menu>
        <!-- Основное содержимое -->
        <Frame Name="MainFrame" NavigationUIVisibility="Hidden"/>
    </DockPanel>
</Window>

```

Код MainWindow.xaml.cs:

Отображены переходы на страницы Категории, Продукция, Платежи, Отчет.

```
using System.Windows;
```

```
namespace MyApp
```

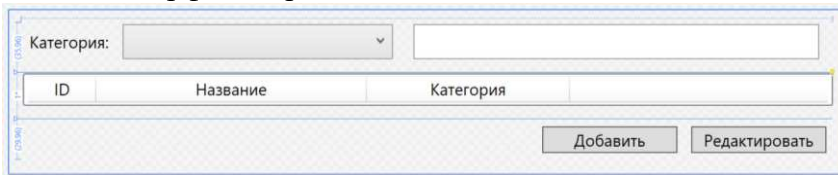
```

{ public partial class MainWindow : Window
    { public MainWindow()
        { InitializeComponent(); }
        private void Categories_Click(object sender, RoutedEventArgs e)
        { MainFrame.Navigate(new CategoriesPage()); }
        private void Products_Click(object sender, RoutedEventArgs e)
        { MainFrame.Navigate(new ProductsPage()); }
        private void Payments_Click(object sender, RoutedEventArgs e)
        { MainFrame.Navigate(new PaymentsPage()); }
        private void Report_Click(object sender, RoutedEventArgs e)
        { MainFrame.Navigate(new ReportPage()); } } }

```

3. Страница Продукции

Интерфейс страницы:



Код ProductsPage.xaml:

```
<Page x:Class=" MyApp.ProductsPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Продукция">
<Grid Margin="10">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
        <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>

    <StackPanel Orientation="Horizontal" Margin="0,0,0,10">
        <Label Content="Категория:" VerticalAlignment="Center"/>
        <ComboBox x:Name="CategoryFilterComboBox" Width="200" Margin="5,0"
ItemsSource="{Binding category}" DisplayMemberPath="category_name"
SelectedValuePath="id_category"
SelectionChanged="CategoryFilterComboBox_SelectionChanged"/>

        <TextBox x:Name="SearchTextBox" Width="300" Margin="10,0"
TextChanged="SearchTextBox_TextChanged"/>
    </StackPanel>

    <!--компонент для отображения списка продукции -->
    <ListView x:Name="ProductsListView" Grid.Row="1" Margin="0,0,0,10">
        <ListView.View>
            <GridView>
                <GridViewColumn Header="ID" DisplayMemberBinding="{Binding product_id}"
Width="50"/>
                <GridViewColumn Header="Название" DisplayMemberBinding="{Binding
product_name}" Width="200"/>
                <GridViewColumn Header="Категория" DisplayMemberBinding="{Binding
category.category_name}" Width="150"/>
            </GridView>
        </ListView.View>
    </ListView>

    <!-- Кнопки добавления/редактирования-->
    <StackPanel Orientation="Horizontal" Grid.Row="2" HorizontalAlignment="Right">
        <Button Content="Добавить" Width="100" Margin="5" Click="AddProduct_Click"/>
        <Button Content="Редактировать" Width="100" Margin="5" Click="EditProduct_Click"/>
    </StackPanel>
</Grid>
</Page>
```

Код ProductsPage.xaml.cs:

Код содержит вывод списка продукции. Название категории подключается к продукции через свойство навигации класса products.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Navigation;

namespace MyApp
{
    /// <summary>
    /// Логика взаимодействия для Producti.xaml
    /// </summary>
    public partial class ProductsPage: Page
    {
        public ProductsPage()
        {
            InitializeComponent();
            var categories = PlatezhiEntities.GetContext().category.ToList();
            categories.Insert(0, new category { id_category = 0, category_name = "Все
                категории" });

            CategoryFilterComboBox.ItemsSource = categories;
            CategoryFilterComboBox.SelectedIndex = 0;
            ProductsListView.ItemsSource =
                PlatezhiEntities.GetContext().products.ToList();

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

        }

        private void EditProduct_Click(object sender, RoutedEventArgs e)
        {

        }

        private void CategoryFilterComboBox_SelectionChanged(object sender,
            SelectionChangedEventArgs e)
        {

        }

        private void SearchTextBox_TextChanged(object sender, TextChangedEventArgs e)
        {

        }

    }
}
```

4. Добавим фильтрацию на страницу Продукции [ProductsPage](#)

Добавим метод `ApplyFilter()`, в котором будет фильтровать список продукции по выбранной категории (из комбобокса) и с названиями, начинающимися с заданной строки (из текстового поля), чтобы оба фильтра применялись одновременно. Данный метод будем вызывать при изменении выбора категории в комбобоксе и при изменении строки поиска в текстовом поле.

```
private void ApplyFilter()
{
    var selectedCategory = CategoryFilterComboBox.SelectedItem as category;
    var query = PlatezhiEntities.GetContext().products.AsQueryable();

    if (selectedCategory != null && selectedCategory.id_category != 0)
    {
        query = query.Where(p => p.category_name ==
                                selectedCategory.id_category);
    }

    if (!string.IsNullOrEmpty(SearchTextBox.Text))
    {
        query = query.Where(p =>
                                p.product_name.StartsWith(SearchTextBox.Text));
    }

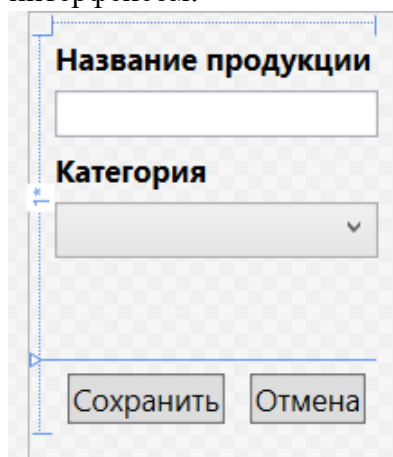
    ProductsListView.ItemsSource = query.ToList();
}

private void CategoryFilterComboBox_SelectionChanged(object sender,
    SelectionChangedEventArgs e)
{
    ApplyFilter();
}

private void SearchTextBox_TextChanged(object sender, TextChangedEventArgs e)
{
    ApplyFilter();
}
```

5. Создадим страницы добавления и редактирования продукции

Для добавления и редактирования создадим одну страницу **ProductEditPage** с интерфейсом:



Код ProductEditPage.xaml:

```
<Page x:Class=" MyApp.ProductEditPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Добавление/Редактирование продукции">
    <Grid Margin="10">
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="Auto" />
        </Grid.RowDefinitions>

        <!-- Интерфейс -->
        <StackPanel>
            <TextBlock Text="Название продукции" FontWeight="Bold" />
            <TextBox x:Name="ProductNameTextBox" Margin="0,5" />

            <TextBlock Text="Категория" FontWeight="Bold" />
            <ComboBox x:Name="CategoryComboBox" DisplayMemberPath="category_name"
                    SelectedValuePath="id_category" Margin="0,5" />

            <TextBlock x:Name="InfoTextBlock" Foreground="Red" Margin="0,10" />
        </StackPanel>

        <!-- Кнопки -->
        <StackPanel Grid.Row="1" Orientation="Horizontal"
                    HorizontalAlignment="Right">
            <Button Content="Сохранить" Margin="5" Click="SaveButton_Click" />
            <Button Content="Отмена" Margin="5" Click="CancelButton_Click" />
        </StackPanel>
    </Grid>
</Page>
```

Код ProductEditPage.xaml.cs

```
namespace MyApp
{
    /// <summary>
    /// Логика взаимодействия для ProductEditPage.xaml
    /// </summary>
    public partial class ProductEditPage: Page
    {
        private PlatezhiEntities _context; //объявляем контекст
        private products _product; //объявляем экземпляр продукции

        // в конструктор передаем выделенный продукт для редактирования или null
        public ProductEditPage(products product = null)
        {
            InitializeComponent();
            //формируем список категорий
            var categories = _context.category.ToList();
            categories.Insert(0, new category { id_category = 0, category_name =
                "Все категории" });

            CategoryComboBox.ItemsSource = categories;
            CategoryComboBox.SelectedIndex = 0;
            //выгружаем в контекст нашу БД
        }
    }
}
```

```

        _context = PlatezhiEntities.GetContext();

// Если передан продукт, заполняем поля для редактирования иначе создаем
новый
        _product = product != null ? product : new products();

        if (product != null)
        {
            ProductNameTextBox.Text = product.product_name;
            CategoryComboBox.SelectedValue = product.category_name;
        }
    }

//кнопка сохранения изменений
private void SaveButton_Click(object sender, RoutedEventArgs e)
{ // проверяем заполнены ли все поля
    if (string.IsNullOrEmpty(ProductNameTextBox.Text) ||
        CategoryComboBox.SelectedValue == null)
    {
        InfoTextBlock.Text = "Заполните все поля!";
        return;
    }

// Сохраняем данные с формы в экземпляр продукта
    _product.product_name = ProductNameTextBox.Text.Trim();
    _product.category_name = (int)CategoryComboBox.SelectedValue;

// Формируем ID только для нового продукта, на единицу больше
максимального ID в таблице Products

    int max = 0;
    if (_product.product_id == 0)
    { if (_context.products.Count() != 0)
        max = _context.products.Max(pr => pr.product_id);
        _product.product_id = max + 1;

        // Добавляем новый продукт
        _context.products.Add(_product);
    }

    // Сохраняем изменения
    _context.SaveChanges();
    MessageBox.Show("Продукт успешно сохранен!", "Информация",
        MessageBoxButton.OK, MessageBoxImage.Information);

// Закрываем страницу добавления и переходим назад на страницу списка
продукции
    NavigationService.GoBack();
}

//кнопка отмены изменений
private void CancelButton_Click(object sender, RoutedEventArgs e)
{
    NavigationService.GoBack();
}
}

```

6. Добавим на страницу Продукции ProductsPage переходы на форму ProductEditPage при нажатии на кнопок Добавить/Редактировать

Фрагмент кода ProductsPage.xaml.cs:

```
private void AddProduct_Click(object sender, RoutedEventArgs e)
{
    //переходим на страницу добавления
    NavigationService.Navigate(new ProductEditPage());
    //обновляем список продукции
    ProductsListView.ItemsSource =
        PlatezhiEntities.GetContext().products.ToList();
}

private void EditProduct_Click(object sender, RoutedEventArgs e)
{
    //находим ID выделенного продукта в списке
    var selectedProduct =
        ProductsListView.SelectedItems.Cast<products>().FirstOrDefault();
    //если ID найден, то передаем его на страницу редактирования
    if (selectedProduct != null)
    {
        NavigationService.Navigate(new ProductEditPage(selectedProduct));
        ProductsListView.ItemsSource =
            PlatezhiEntities.GetContext().products.ToList();
    }
    else
    {
        MessageBox.Show("Выберите продукт для редактирования.", "Ошибка",
            MessageBoxButton.OK, MessageBoxImage.Error);
    }
}
```