**Отчет по Главе 4: Элементы управления (Controls)**

**1. Введение**

Данная глава посвящена изучению широкого спектра элементов управления (Controls), доступных в .NET MAUI. Элементы управления являются строительными блоками любого пользовательского интерфейса, позволяя отображать информацию, принимать ввод от пользователя и обеспечивать интерактивность. Значимость этой темы огромна, так как правильный выбор и использование элементов управления напрямую влияет на функциональность, удобство использования и внешний вид разрабатываемых кроссплатформенных настольных и мобильных приложений. В рамках этой главы были рассмотрены и продемонстрированы наиболее часто используемые элементы управления.

**2. Основная часть**

**2.1. Основные концепции и принципы**

.NET MAUI предоставляет богатый набор встроенных элементов управления, наследуемых в основном от класса View. Каждый элемент имеет свой набор свойств для настройки внешнего вида и поведения, а также событий для реакции на действия пользователя. Для демонстрации была создана единая страница Chapter4ControlsPage.xaml.

**2.2. Примеры кода с пояснениями**

**2.2.1. Текстовые элементы управления (Label, Entry, Editor)**

Эти элементы предназначены для отображения и ввода текста.

* **Label**: Отображает текст. Поддерживает простое текстовое содержимое, форматированный текст (FormattedString для смешивания стилей) и отображение HTML (TextType="Html").

<Label Text="Label (Simple): This is a simple Label." />

<Label FontSize="Medium">

<Label.FormattedText>

<FormattedString>

<Span Text="Label (Formatted): You can " />

<Span Text="mix " FontAttributes="Bold" TextColor="Red" />

<Span Text="styles." TextDecorations="Underline"/>

</FormattedString>

</Label.FormattedText>

</Label>

<Label TextType="Html">

<![CDATA[

Label (HTML): <strong>Bold</strong> and <em>Italic</em>.

]]>

</Label>

* **Entry**: Поле для однострочного ввода текста. Имеет свойства Placeholder, Keyboard (для указания типа клавиатуры), IsPassword, MaxLength, IsReadOnly. События TextChanged и Completed позволяют реагировать на ввод.

<Entry Placeholder="Entry: Enter your name" TextChanged="Generic\_TextChanged" Completed="Generic\_Completed"/>

<Entry Placeholder="Entry (Password):" IsPassword="True" />

* **Editor**: Поле для многострочного ввода текста. Свойство AutoSize="TextChanges" позволяет автоматически изменять высоту поля по мере ввода.

<Editor Placeholder="Editor: Enter multi-line text..." HeightRequest="100" AutoSize="TextChanges" TextChanged="Generic\_TextChanged" Completed="Generic\_Completed"/>

**2.2.2. Кнопочные и интерактивные элементы (Button, ImageButton, CheckBox, RadioButton)**  
Элементы для выполнения действий и выбора опций.

* **Button**: Стандартная кнопка. Имеет событие Clicked. Может содержать текст и изображение (через вложение Grid или другого макета в Button.Content).

<Button Text="Simple Button" Clicked="CommonControl\_Clicked"/>

<Button Padding="10" Clicked="CommonControl\_Clicked">

<Button.Content>

<Grid ColumnDefinitions="Auto,\*" ColumnSpacing="10" VerticalOptions="Center">

<Image Source="dotnet\_bot.png" WidthRequest="30" HeightRequest="30" VerticalOptions="Center"/>

<Label Text="Button with Image" VerticalOptions="Center" Grid.Column="1"/>

</Grid>

</Button.Content>

</Button>

* **ImageButton**: Кнопка, основным содержимым которой является изображение.

<ImageButton Source="dotnet\_bot.png" WidthRequest="40" HeightRequest="40" Aspect="AspectFit" Clicked="CommonControl\_Clicked"/>

* **CheckBox**: Флажок для выбора (включено/выключено). Событие CheckedChanged.

<HorizontalStackLayout Spacing="10" VerticalOptions="Center">

<CheckBox x:Name="DemoCheckBox" CheckedChanged="DemoCheckBox\_CheckedChanged"/>

<Label Text="CheckBox" VerticalOptions="Center"/>

</HorizontalStackLayout>

* **RadioButton**: Переключатель, позволяет выбрать одну опцию из группы. Элементы группируются по GroupName. Событие CheckedChanged.

<RadioButton Content="Option Alpha" GroupName="MyGroup" Value="Alpha" CheckedChanged="DemoRadioButton\_CheckedChanged"/>

<RadioButton Content="Option Beta" GroupName="MyGroup" Value="Beta" IsChecked="True" CheckedChanged="DemoRadioButton\_CheckedChanged"/>

**2.2.3. Элементы представления и отображения (BoxView, Frame, Border, Image, WebView)**  
Элементы для визуального оформления, группировки и отображения контента.

* **BoxView**: Простой прямоугольник, который можно закрасить цветом.

<BoxView Color="CornflowerBlue" WidthRequest="100" HeightRequest="60"/>

* **Frame**: Контейнер с закругленными углами и тенью (по умолчанию), может обрамлять другие элементы.

<Frame Padding="15" CornerRadius="10" BorderColor="DarkGoldenrod" BackgroundColor="LightYellow" HasShadow="True">

<Label Text="Text inside a Frame" HorizontalOptions="Center"/>

</Frame>

* **Border**: Контейнер, позволяющий настроить границу вокруг содержимого, включая форму границы (например, RoundRectangle).

<Border Stroke="OrangeRed" StrokeThickness="2" Padding="5" BackgroundColor="Transparent">

<Border.StrokeShape>

<RoundRectangle CornerRadius="8"/>

</Border.StrokeShape>

<Entry Placeholder="Entry inside a Border"/>

</Border>

* **Image**: Отображает изображения из локальных ресурсов или по URI. Свойство Aspect управляет масштабированием.

<Image Source="dotnet\_bot.png" HeightRequest="80" Aspect="AspectFit" SemanticProperties.Description="Dot Net Bot Image"/>

<Image Source="https://aka.ms/campusplaceholder ২৩" HeightRequest="80" Aspect="AspectFill" SemanticProperties.Description="Placeholder campus image"/>

* **WebView**: Отображает веб-контент (HTML-страницы) по URI или из HTML-строки.

<WebView Source="https://dotnet.microsoft.com/apps/maui" HeightRequest="200"/>

**2.2.4. Элементы выбора (Picker, DatePicker, TimePicker)**  
Предоставляют пользователю стандартные для платформы способы выбора значений.

* **Picker**: Выпадающий список для выбора одного элемента из коллекции.

<Picker x:Name="DemoPicker" Title="Select a color" SelectedIndexChanged="DemoPicker\_SelectedIndexChanged">

<Picker.ItemsSource>

<x:Array Type="{x:Type x:String}">

<x:String>Red</x:String>

<x:String>Green</x:String>

<x:String>Blue</x:String>

</x:Array>

</Picker.ItemsSource>

</Picker>

* **DatePicker**: Позволяет выбрать дату.
* **TimePicker**: Позволяет выбрать время.

**2.2.5. Индикаторы и переключатели (Switch, Slider, Stepper, ProgressBar, ActivityIndicator)**

* **Switch**: Переключатель вкл/выкл. Событие Toggled.
* **Slider**: Ползунок для выбора значения из диапазона. Событие ValueChanged.
* **Stepper**: Элемент с кнопками "+" и "-" для пошагового изменения значения. Событие ValueChanged.
* **ProgressBar**: Отображает прогресс выполнения задачи (от 0.0 до 1.0).
* **ActivityIndicator**: Индикатор активности (например, "загрузка"). Управляется свойством IsRunning.

**2.2.6. SearchBar**  
Поле для ввода поискового запроса. События SearchButtonPressed и TextChanged.  
xml <!-- Из Chapter4ControlsPage.xaml --> <SearchBar Placeholder="Search items..." SearchButtonPressed="DemoSearchBar\_SearchButtonPressed" TextChanged="DemoSearchBar\_TextChanged"/>

**2.2.7. Элементы коллекций (CollectionView, RefreshView, SwipeView)**

* **CollectionView**: Гибкий элемент для отображения списков данных. Использует ItemTemplate для определения внешнего вида каждого элемента.
* **RefreshView**: Оборачивает прокручиваемый контент (например, CollectionView) и добавляет функциональность "потянуть для обновления". Использует Command или событие Refreshing.
* **SwipeView**: Оборачивает элемент и позволяет добавлять контекстные действия (swipe items), доступные по жесту свайпа.

<RefreshView x:Name="DemoRefreshView" Refreshing="DemoRefreshView\_Refreshing" Command="{Binding RefreshCommand}">

<CollectionView x:Name="DemoCollectionView">

<CollectionView.ItemTemplate>

<DataTemplate>

<SwipeView>

<SwipeView.LeftItems>

<SwipeItems> <SwipeItem Text="Favorite" Invoked="SwipeItem\_Invoked\_Favorite"/> </SwipeItems>

</SwipeView.LeftItems>

<!-- ... Frame с контентом элемента ... -->

</SwipeView>

</DataTemplate>

</CollectionView.ItemTemplate>

</CollectionView>

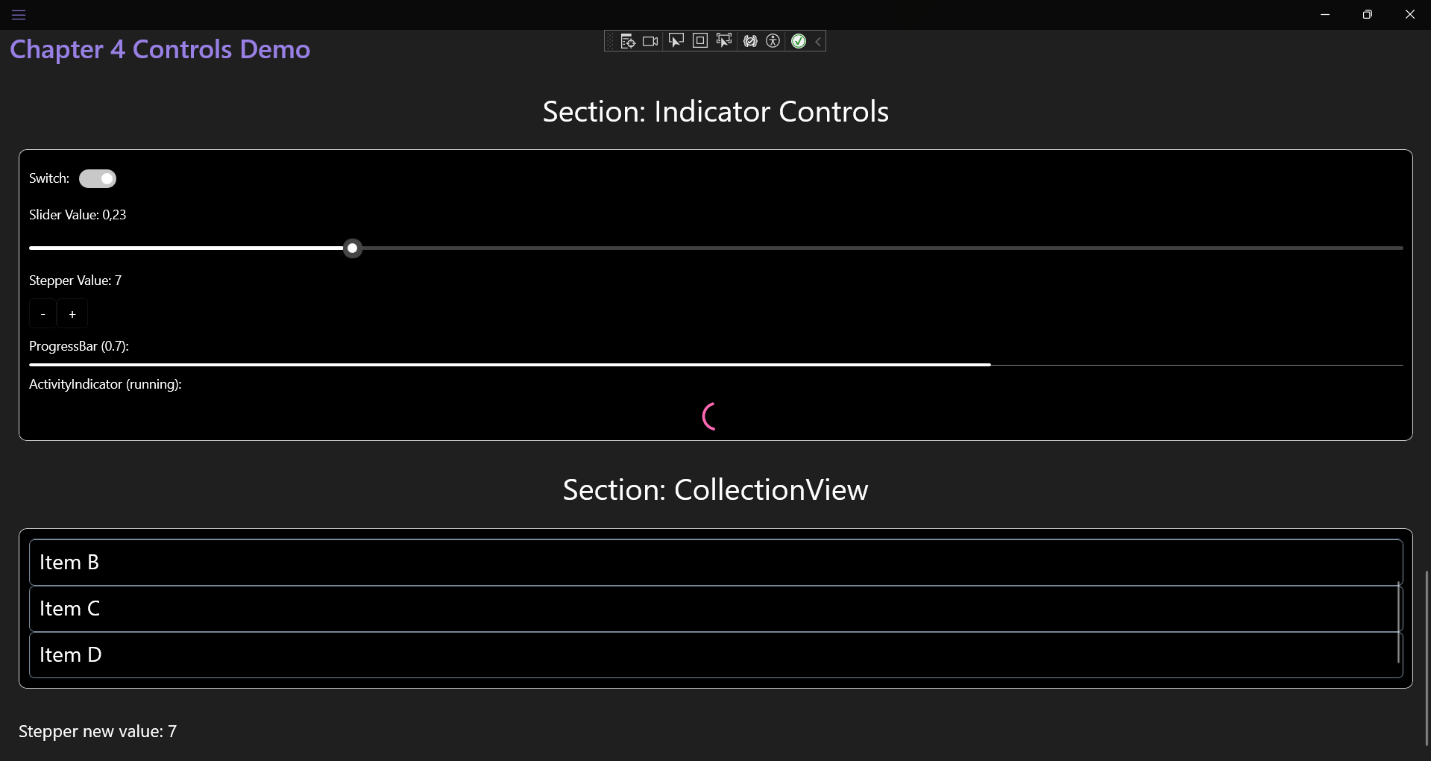
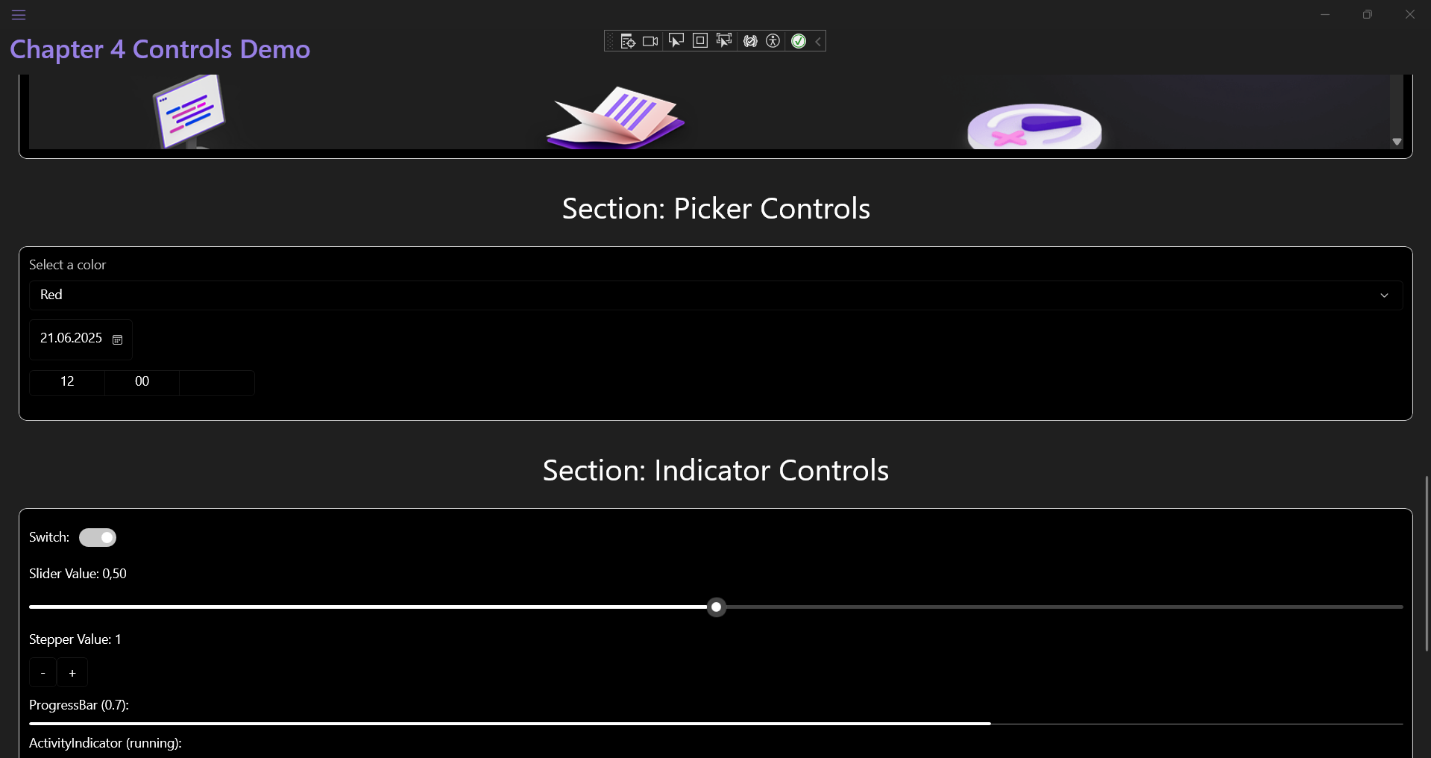
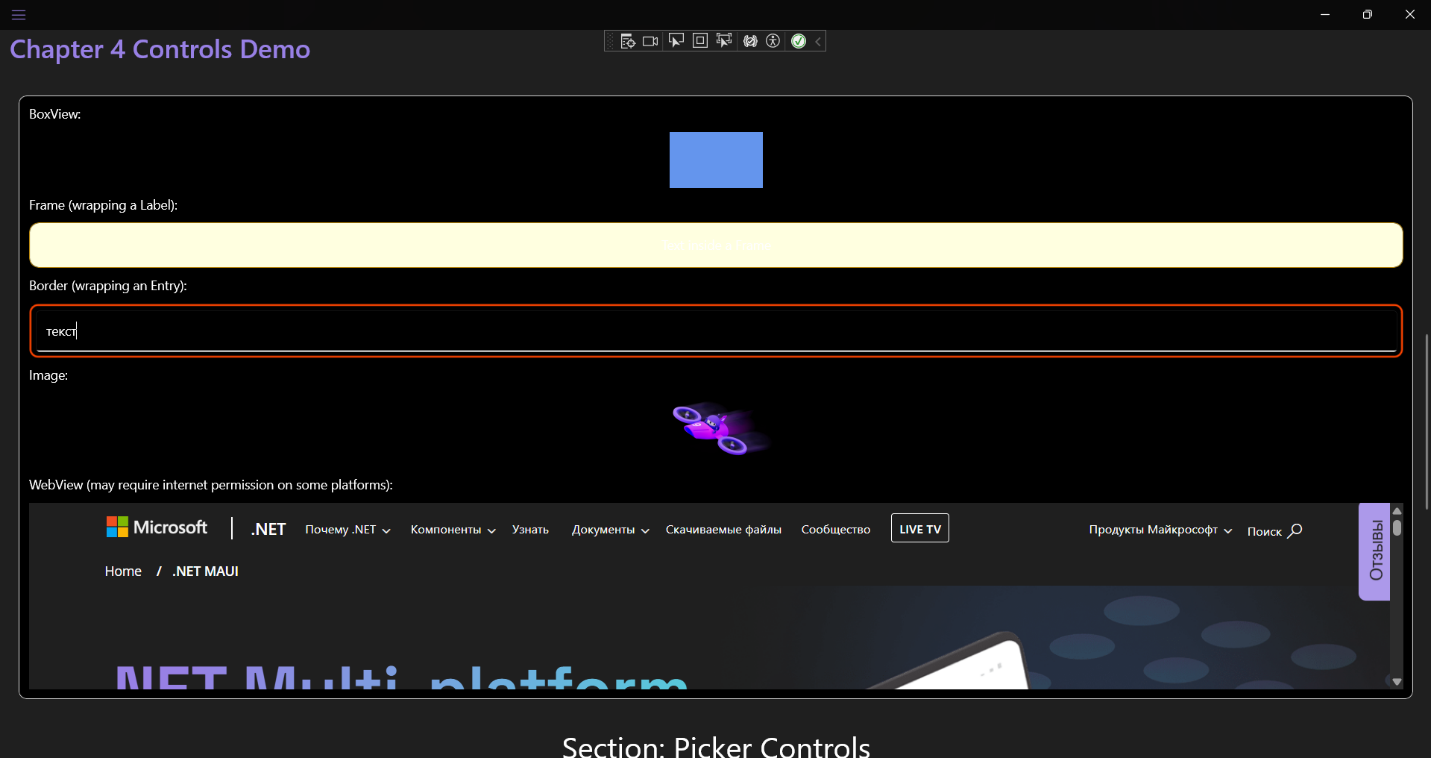
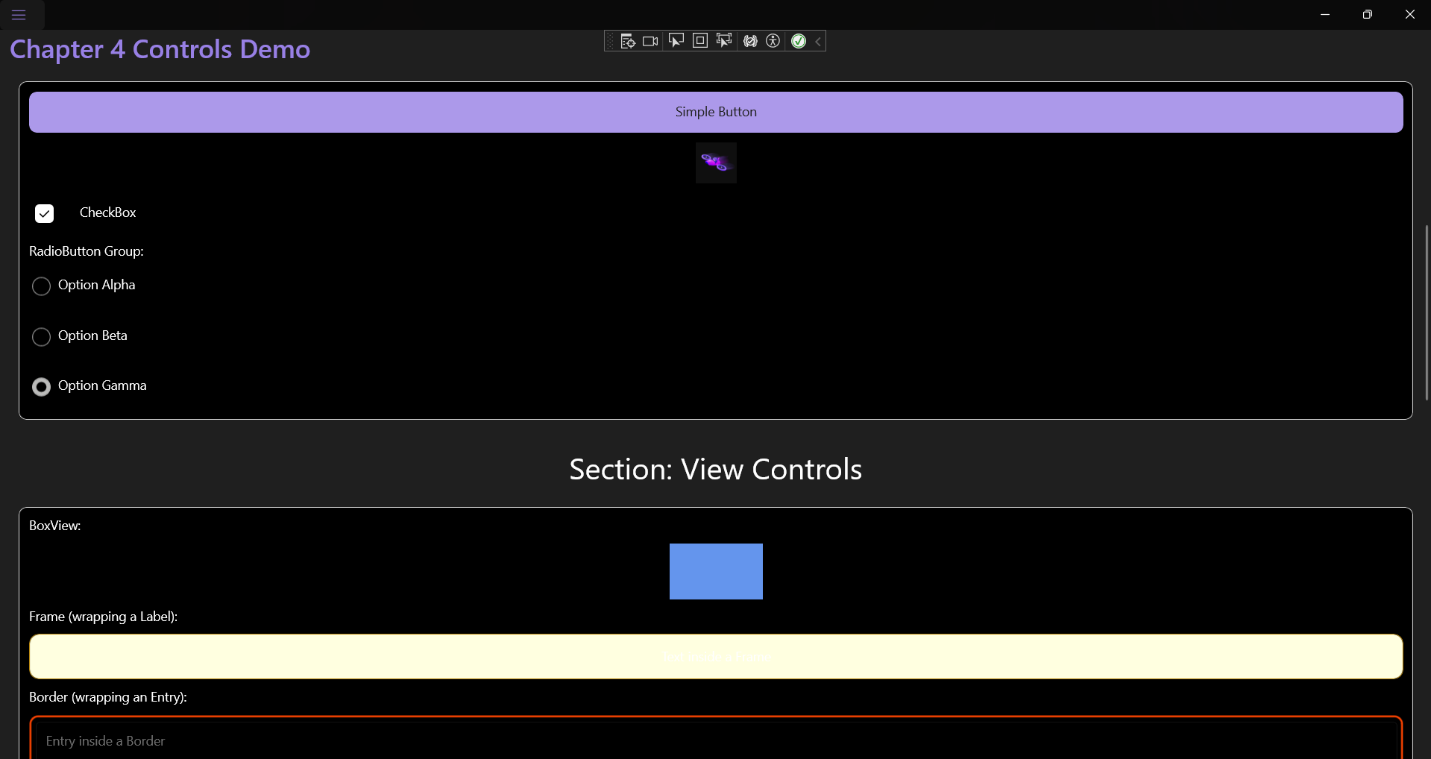
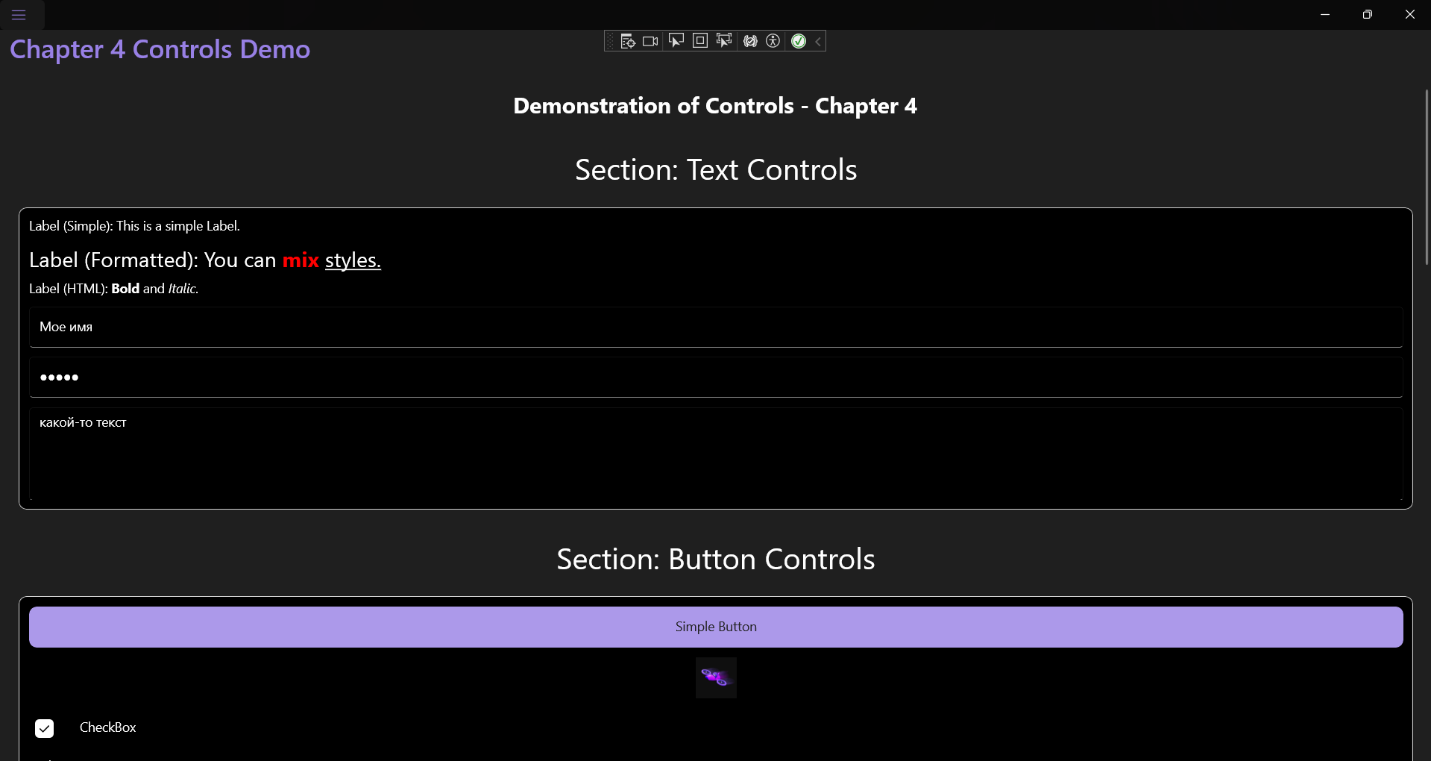
</RefreshView>

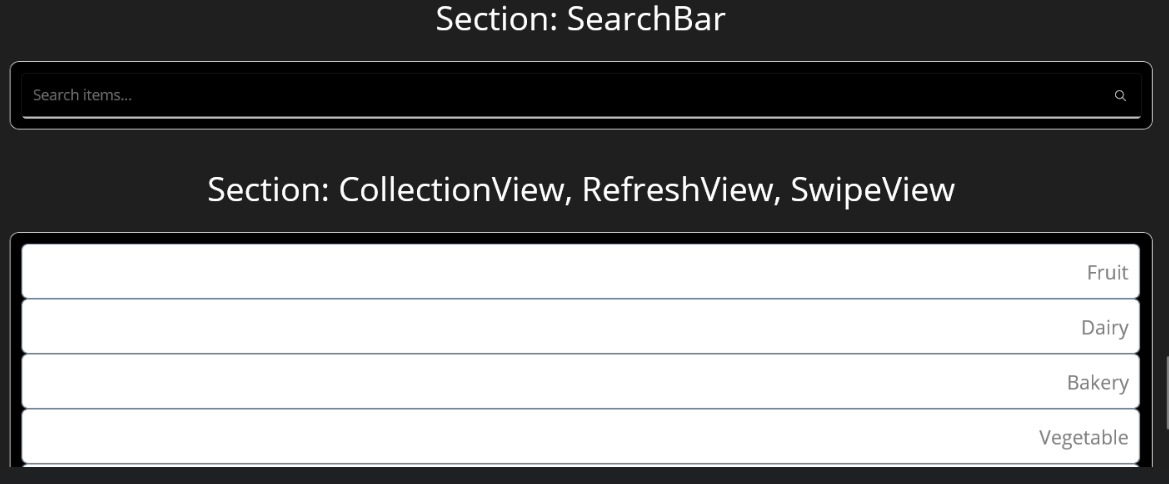
**3. Заключение**

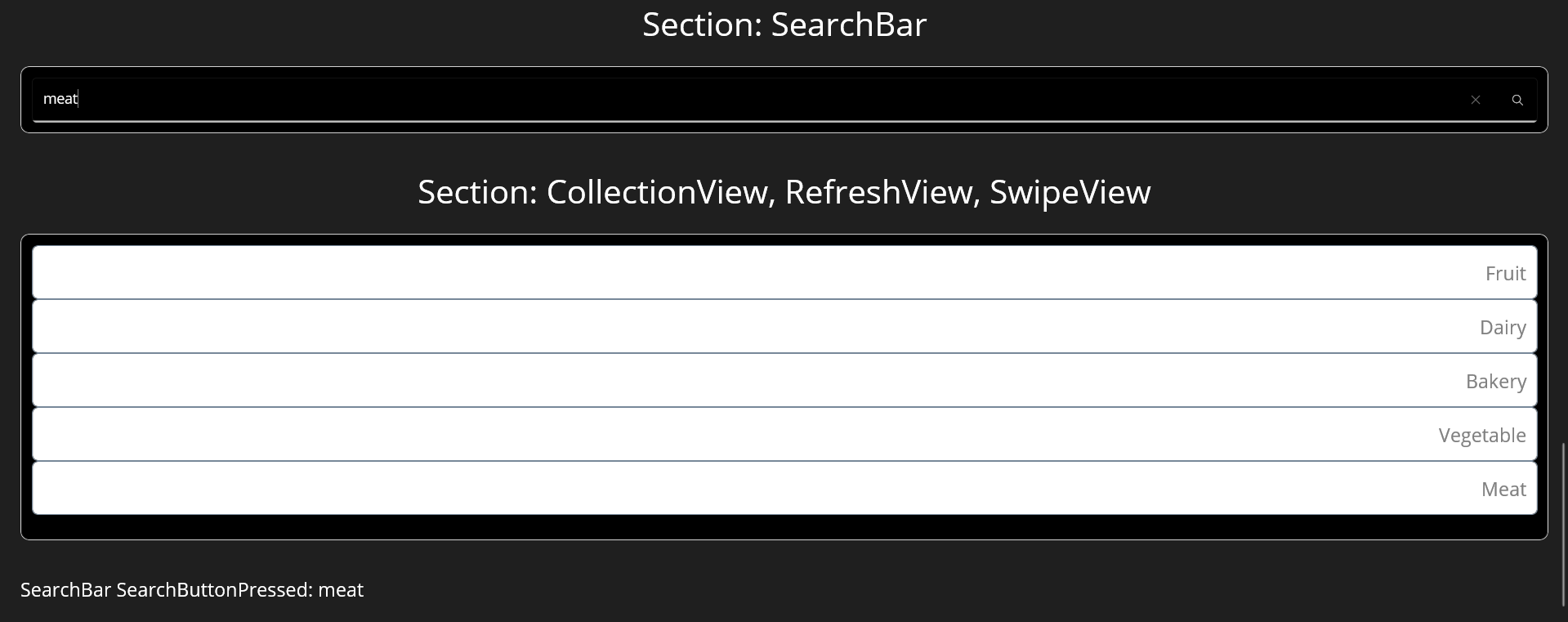
Глава 4 охватывает широкий спектр элементов управления, которые являются основой для создания интерактивных и функциональных пользовательских интерфейсов в .NET MAUI. Понимание назначения, свойств и событий каждого элемента позволяет разработчику эффективно решать задачи по представлению данных и взаимодействию с пользователем. Практическое применение этих элементов на демонстрационной странице Chapter4ControlsPage позволило наглядно увидеть их в действии и закрепить теоретические знания. Эти элементы являются универсальными и широко применимы при разработке настольных приложений.

**4. Скриншоты и примеры**

Все элементы управления были реализованы на единой странице Chapter4ControlsPage.xaml и протестированы на платформе Windows. Навигация к странице осуществлялась через AppShell.







*Рис. 4.1 - 4.7. Демонстрация различных элементов управления на странице Chapter4ControlsPage."*

**Процесс выполнения примеров:**

1. Создана новая страница Chapter4ControlsPage.xaml и связанный с ней файл Chapter4ControlsPage.xaml.cs.
2. Страница была последовательно заполнена XAML-разметкой для различных групп элементов управления, согласно материалам [metanit.com/sharp/maui/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fmetanit.com%2Fsharp%2Fmaui%2F) (глава 4).
3. В файл .xaml.cs были добавлены необходимые обработчики событий и модель данных для CollectionView.
4. Страница Chapter4ControlsPage была добавлена в навигационную структуру AppShell.xaml.
5. Проведено тестирование всех элементов управления, включая ввод текста, нажатие кнопок, выбор значений, свайп-жесты и обновление списка.  
   В процессе разработки возникали синтаксические ошибки XAML, которые были устранены путем внимательной проверки кода и сравнения с рабочими примерами. После исправления все элементы управления отображались и функционировали корректно.

**5. Документация**

* **Ссылка на GitHub с исходным кодом:** <https://github.com/Nikita-Levuskin/C-sharp-practic/tree/main/1>
* **Описание выполненных шагов:**
  1. Изучены материалы по теме "Элементы управления" (глава 4) с ресурса [metanit.com/sharp/maui/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fmetanit.com%2Fsharp%2Fmaui%2F).
  2. Создана единая страница Chapter4ControlsPage для демонстрации большинства рассмотренных элементов.
  3. Реализованы примеры использования Label, Entry, Editor, Button, ImageButton, CheckBox, RadioButton, BoxView, Frame, Border, Image, WebView, Picker, DatePicker, TimePicker, Switch, Slider, Stepper, ProgressBar, ActivityIndicator, SearchBar, CollectionView, RefreshView и SwipeView.
  4. Настроены обработчики событий для демонстрации интерактивности.
  5. Проведено тестирование функциональности на платформе Windows.
  6. Устранены возникшие в процессе разработки ошибки XAML.
  7. Подготовлены скриншоты работающего приложения.