

ПРАКТИЧНА РОБОТА № 11. СТВОРЕННЯ MASTERDETAIL PAGE ЗАСОБАМИ XAMARIN.FORMS

Мета: набуття здобувачами освіти навичок створення проєктів зі сторінками типу FlyoutPage (MasterDetailPage).

Короткі теоретичні відомості

FlyoutPage (MasterDetailPage) – це сторінка, яка керує двома зв'язаними сторінками інформації - спливаючою сторінкою, яка представляє елементи, а також сторінкою відомостей про елементи на сторінці, що спливає (рис. 1). Якщо вибрати елемент на спливаючій сторінці, то відбувається перехід на відповідну сторінку відомостей.

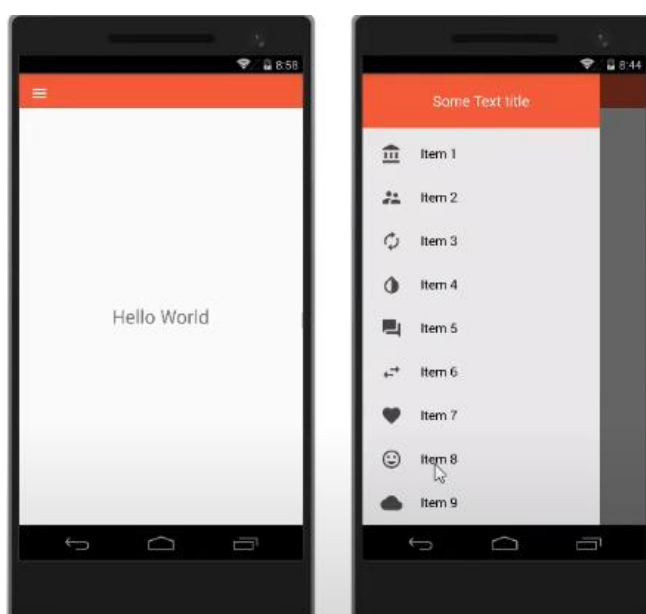


Рисунок 1 – Приклад FlyoutPage

FlyoutPage визначає такі властивості:

- Detail з типом Page визначає сторінку відомостей, що відображається для вибраного елемента на спливаючій сторінці;
- Flyout типу Page визначає спливаюче меню сторінки;
- FlyoutLayoutBehavior вказує на поведінку макета сторінок;
- IsGestureEnabled типу bool визначає чи буде жест прокручування перемикатися між спливаючим елементом та сторінки відомостей. Значення за замовчанням цієї властивості дорівнює true;
- IsPresented типу bool визначає чи відображається спливаючий елемент або сторінка відомостей. Значення за замовчуванням – false (сторінка відомостей). Для відображення спливаючої сторінки необхідно встановити значення в true.

FlyoutPage також визначає подію, що виникає IsPresentedChanged за зміною

значення властивості `IsPresented`.

Щоб створити спливаючу частину сторінки, необхідно створити `FlyoutPage` об'єкт та встановити його властивість `Detail`. Властивість `Flyout` має бути задано як об'єкт `ContentPage`, а властивість `Detail` має бути задано як об'єкт `TabbedPage` чи об'єкт `NavigationPage` `ContentPage`. Це дозволяє забезпечити узгодженість інтерфейсу користувача на всіх платформах.

`FlyoutPage` призначена для того, щоб бути кореневою сторінкою програми, та її використання як дочірня сторінка в інших типах сторінок може призвести до несподіваної й неузгодженості поведінки. Для сторінки `Flyout` має бути задана властивість `Title`, інакше виникне виняток. Значення властивості `FlyoutLayoutBehavior` впливає лише на програми, які працюють на планшетах або на робочому столі. Програми, що працюють на телефонах, завжди мають поведінку `Popover` (сторінка відомостей повністю або частково перекриває сторінку, що спливає).

Хід роботи

1. Створюємо проект `MasterDetailApp` в `Visual Studio 19`. Вибираємо шаблон `Blank`. При створенні проекту в опції `Platform` вибираємо платформу `Android` під яку буде створюватись проект (рис. 2).

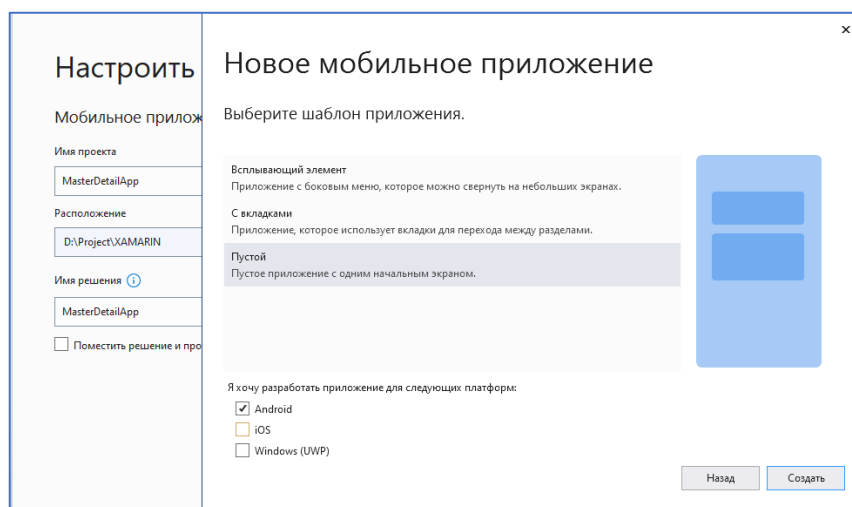


Рисунок 2 – Створення проекту `MasterDetailApp`

2. Відкриваємо файл `MainPage.xaml.cs` та дописуємо успадкування від класу `FlyoutPage` (за рекомендаціями `VStudio MasterDetailPage` застаріла технологія; рекомендують використовувати `FlyoutPage`).

Було	Стало
<pre>public partial class MainPage : ContentPage { public MainPage() { InitializeComponent(); } }</pre>	<pre>public partial class MainPage : FlyoutPage { public MainPage() { InitializeComponent(); } }</pre>

3. Відкриваємо файл MainPage.xaml та міняємо Contentpage на FlyoutPage. Додаємо Title="MasterApp". Створюємо 2 блоку макету FlyoutPage FlyoutPage.Flyout та FlyoutPage.Detail. До FlyoutPage.Flyout додаємо розмітку типу StackLayout, в середину якого поміщаємо 2 кнопки "Friends", "News". До FlyoutPage. Detail додаємо мітку з текстом "DetailPage". Відреагувати код за зразком:

```
<?xml version="1.0" encoding="utf-8" ?>
<FlyoutPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="MasterDetailApp.MainPage"
  Title="MasterApp">
  <FlyoutPage.Flyout>
    <ContentPage Title="Master" BackgroundColor="#b96fca">
      <StackLayout Padding="5, 30, 5, 5">
        <Button Text="Friends"></Button>
        <Button Text="News"></Button>
      </StackLayout>
    </ContentPage>
  </FlyoutPage.Flyout>

  <FlyoutPage.Detail>
    <ContentPage>
      <Label Text="DetailPage" VerticalOptions="Center"
HorizontalOptions="Center"></Label>
    </ContentPage>
  </FlyoutPage.Detail>
</FlyoutPage>
```

Для зручності форматування коду застосовуємо комбінацію клавіш .Ctrl+K ... Ctrl+D.

4. Запускаємо на виконання. На екрані мобільного пристрою завантажиться сторінка з текстом Detail.Page. Якщо потягнути сторінку праворуч відкривається FlyoutPage з 2 кнопками «FRIENDS», «NEWS» (рис. 3).

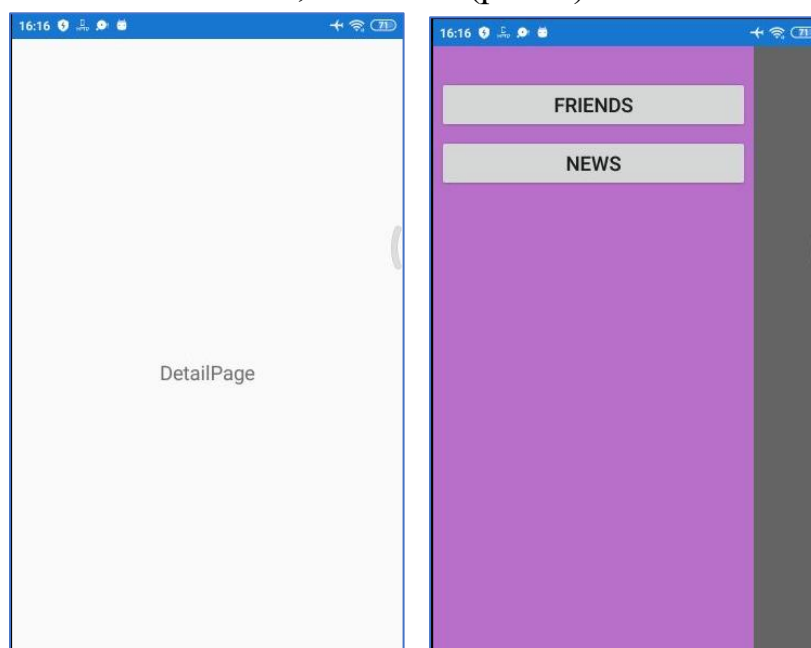


Рисунок 3 – Екран додатку MasterDetailApp

5. Додамо до проекту 2 xml сторінки Friends.xaml, News.xaml. Для цього викликаємо контекстне меню провідника рішень (Solution Explorer). Вибираємо Add -> Create element -> ContentPage та вказуємо назву сторінку. Провідник рішень (Solution Explorer) після додавання сторінок наведений на рис. 4.

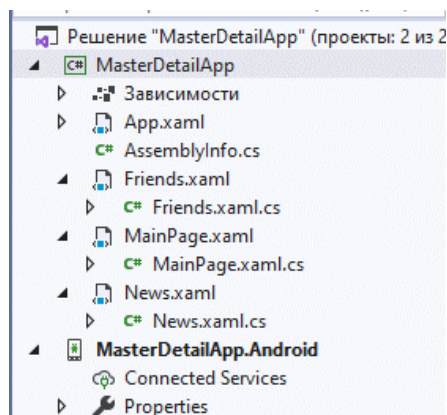


Рисунок 4 – Solution Explorer

6. Для цих сторінок додаємо атрибут Title та елемент Label.

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="MasterDetailApp.News"
             Title="News">
  <Label Text="Новини" VerticalOptions="Center" HorizontalOptions="Center" />
</ContentPage>
```

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="MasterDetailApp.Friends"
             Title="Friends">
  <Label Text="Friends of the user" VerticalOptions="Center" HorizontalOptions="Center" />
</ContentPage>
```

7. Переходимо до MainPage.xaml.cs та задаємо сторінку яка буде відображатись при старті

```
public MainPage()
{
    InitializeComponent();

    Detail = new NavigationPage(new News());
}
```

8. Додаємо обробку кнопок. Для цього у файлі MainPage.xaml задамо їм події Button1_Click, Button2_Click та у файлі MainPage.xaml.cs додамо методи, які будуть викликати сторінки «FRIENDS», «NEWS».

Фрагмент коду файлу MainPage.xaml

```
<StackLayout Padding="5, 30, 5, 5">
    <Button Text="Friends" Clicked="Button1_Click"></Button>
    <Button Text="News" Clicked="Button2_Click"></Button>
</StackLayout>
```

Фрагмент коду файлу MainPage.xaml.cs

```
private void Button1_Click(object sender, EventArgs e)
{
    Detail = new NavigationPage(new Friends());
}

private void Button2_Click(object sender, EventArgs e)
{
    Detail = new NavigationPage(new News());
}
```

9. Запускаємо додаток та дивимось на результат (рис.5). Тестуємо реакцію на натиснення кнопок

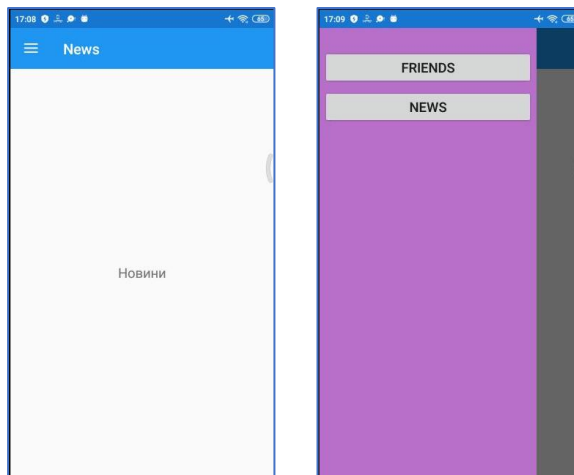


Рисунок 5 – Екран додатку MasterDetailApp після виконання п.1-8

10. Властивість `IsPresented` відповідає за показ мастер сторінки при запуску додатку. Додаємо це до коду `MainPage.xaml.cs` та змінимо колір `Toolbar`

```
public MainPage()
{
    InitializeComponent();

    Detail = new NavigationPage(new News())
    {
        BarBackgroundColor = Color.FromHex("#008000")
    };
    IsPresented = true;
}

private void Button1_Click(object sender, EventArgs e)
{
    Detail = new NavigationPage(new Friends())
    {
        BarBackgroundColor = Color.FromHex("#008000")
    };
    IsPresented = false;
}
```

```

}

private void Button2_Click(object sender, EventArgs e)
{
    Detail = new NavigationPage(new News())
    {
        BarBackgroundColor = Color.FromHex("#008000")
    };
    IsPresented = false;
}

```

11. Запускаємо додаток та дивимось на результат (рис.6). Тестуємо реакцію на натиснення кнопок

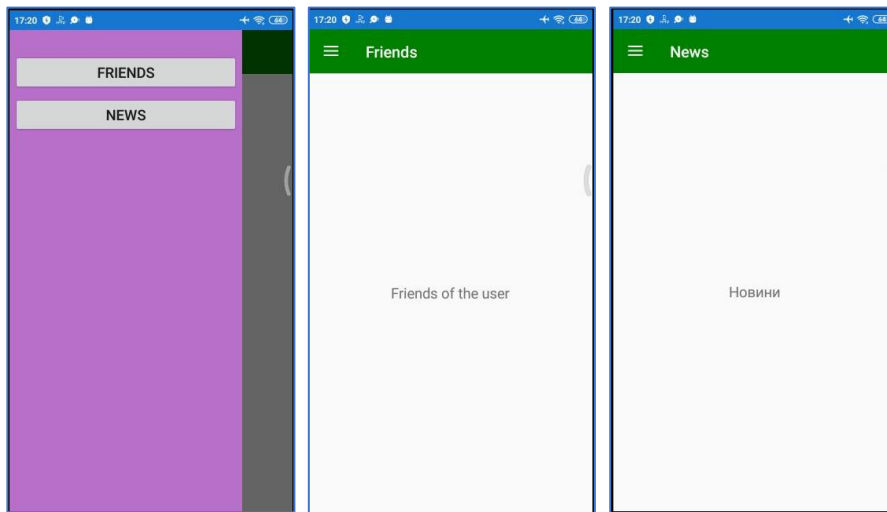


Рисунок 6 – Екран додатку MasterDetailApp після виконання п.1-10

12. Додаємо іконки до кнопок. Переходимо до файлу MainPage. Вирізаємо код кнопок та тимчасово зберігаємо його у блокноті

```

<Button Text="Friends" Clicked="Button1_Click"></Button>
<Button Text="News" Clicked="Button2_Click"></Button>

```

Створюємо новий StackLayout з параметрами Orientation="Horizontal", HorizontalOptions="FillAndExpand". Код у файлі MainPage.xaml має вигляд:

```

<FlyoutPage.Flyout>
    <ContentPage Title="Master" BackgroundColor="#b96fca">
        <StackLayout Padding="5, 30, 5, 5">
            <StackLayout Orientation="Horizontal" HorizontalOptions="FillAndExpand">

                </StackLayout>
            </StackLayout>
        </ContentPage>
    </FlyoutPage.Flyout>

```

13. Додаємо до StackLayout елемент Image, задаємо ширину, висоту та співвідношення (Aspect), щоб рисунок не спотворювався. Додаємо рисунок

friends.png в папку Drawable. Перевіряємо для рисунку властивість «Установка при сборке» має значення AndroidResource. Додаємо також елемент Label до StackLayout. Код StackLayout файлу MainPage.xaml має вигляд:

```
<StackLayout Orientation="Horizontal" HorizontalOptions="FillAndExpand">
  <Image WidthRequest="65" HeightRequest="65" Aspect="AspectFit" Source="friends.png"
    VerticalOptions="Center"> </Image>
  <Label Text="Users" VerticalOptions="Center"> </Label>
</StackLayout>
```

14. Переходимо до створення Блок – картинка. Додаємо до папки Drawable рисунки friends3.png, news2.png. Створюємо новий StackLayout з рисунком та кнопкою 1. Аналогічно новий StackLayout з рисунком та кнопкою 2. Повний код файлу MainPage.xaml наведений на рис. 7.

```
<StackLayout Orientation="Horizontal" HorizontalOptions="FillAndExpand">
  <Image WidthRequest="24" HeightRequest="24" Aspect="AspectFit"
    Source="friends3.png"
    HorizontalOptions="Start"></Image>
  <Button Text="Friends" Clicked="Button1_Click"
    HorizontalOptions="FillAndExpand"></Button>
</StackLayout>

<StackLayout Orientation="Horizontal" HorizontalOptions="FillAndExpand">
  <Image WidthRequest="24" HeightRequest="24" Aspect="AspectFit" Source="news2.png"
    HorizontalOptions="Start"></Image>
  <Button Text="News" Clicked="Button2_Click"
    HorizontalOptions="FillAndExpand"></Button>
</StackLayout>
```

15. Секцію коду, що не використовується можна закоментувати або видалити. Запускаємо додаток та дивимось на результат (рис.8). Тестуємо його функціональність.



```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <FlyoutPage xmlns="http://xamarin.com/schemas/2014/forms"
3   xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4   x:Class="MasterDetailApp.MainPage"
5   Title="MasterApp">
6   <FlyoutPage.Flyout>
7     <ContentPage Title="Master" BackgroundColor="#b96fca">
8     <StackLayout Padding="5, 30, 5, 5">
9       <StackLayout Orientation="Horizontal" HorizontalOptions="FillAndExpand">
10        <Image WidthRequest="65" HeightRequest="65" Aspect="AspectFit" Source="friends.png" VerticalOptions="Center"> </Image>
11        <Label Text="Users" VerticalOptions="Center"> </Label>
12      </StackLayout>
13    </StackLayout>
14    <StackLayout Orientation="Horizontal" HorizontalOptions="FillAndExpand">
15      <Image WidthRequest="24" HeightRequest="24" Aspect="AspectFit" Source="friends3.png"
16        HorizontalOptions="Start"></Image>
17      <Button Text="Friends" Clicked="Button1_Click" HorizontalOptions="FillAndExpand"></Button>
18    </StackLayout>
19    <StackLayout Orientation="Horizontal" HorizontalOptions="FillAndExpand">
20      <Image WidthRequest="24" HeightRequest="24" Aspect="AspectFit" Source="news2.png"
21        HorizontalOptions="Start"></Image>
22      <Button Text="News" Clicked="Button2_Click" HorizontalOptions="FillAndExpand"></Button>
23    </StackLayout>
24  </ContentPage>
25 </FlyoutPage.Flyout>
26 </FlyoutPage>
```

Рисунок 7 – Повний код файлу MainPage.xaml додатку MasterDetailApp

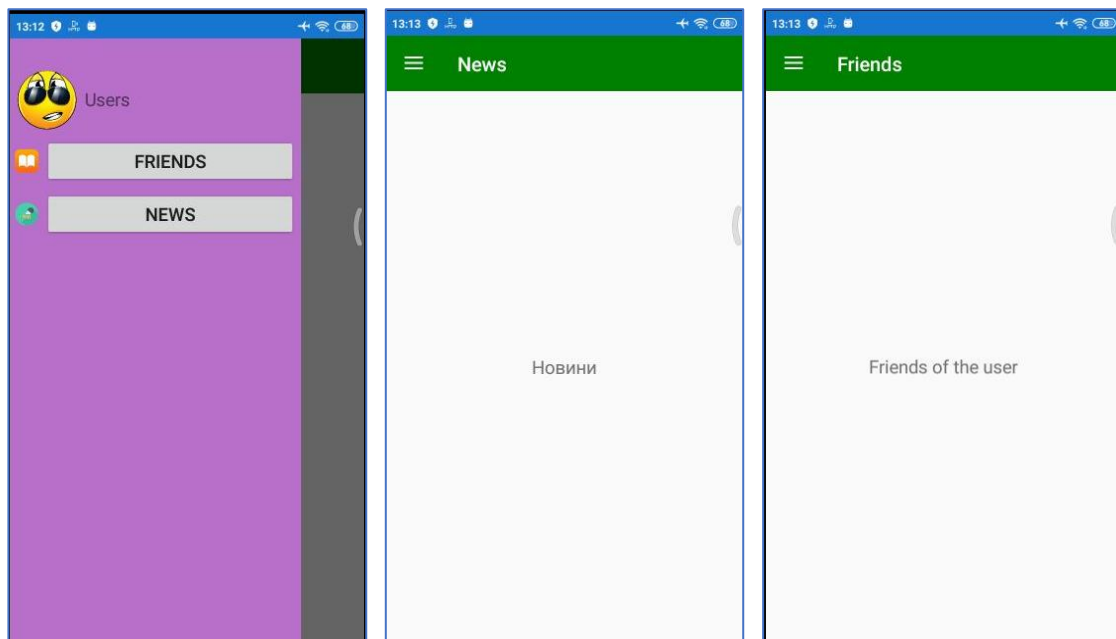


Рисунок 8 – Результати роботи додатку MasterDetailApp

Контрольні питання

1. Які способи переходу між сторінками надає платформа Xamarin Forms?
2. Як здійснюється перехід між сторінками в CarouselPage?
3. Як здійснюється перехід між сторінками в TabbedPage?
4. Як здійснюється перехід між сторінками в FlyoutPage?
5. Як здійснюється перехід між сторінками в NavigationPage?
6. Які основні властивості можна задавати FlyoutPage?
7. Як створити спливаючу частину сторінки FlyoutPage?
8. Як налаштувати проект, щоб при запуску відображалась мастер сторінка?

Список використаних джерел

1. Xamarin.Forms FlyoutPage. URL: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/app-fundamentals/navigation/flyoutpage>.
2. FlyoutPage. URL: <https://docs.microsoft.com/en-us/dotnet/maui/user-interface/pages/flyoutpage>.
3. Master-detailpage. URL: <https://www.infobrother.com/Tutorial/Xamarin/xamarin-masterdetailpage.php>.
4. MasterDetailPage In Xamarin Forms Application. URL: <https://www.c-sharpcorner.com/article/masterdetailpage-in-xamarin-forms-application>.