ПРАКТИЧНА РОБОТА № 10. СТВОРЕННЯ NAVIGATION PAGE ЗАСОБАМИ XAMARIN.FORMS

Мета: набуття здобувачами освіти навичок створення та відлагодження кросмобільного додатку з Navigation Page засобами Xamarin.Forms у середовищі Visual Studio 2019.

Короткі теоретичні відомості

Клас NavigationPage забезпечує ієрархічну навігацію, коли користувач може переходити по сторінках вперед та назад за своїм бажанням. Цей клас реалізує навігацію на основі стека об'єктів Page за методом LIFO (останнім надійшов – першим обслужений). Для переходу з однієї сторінки на іншу програма поміщає нову сторінку у стек навігації, де вона стає активною сторінкою, як показано на рис. 1.

Рисунок 1 – Додавання нової сторінки у стек навігації

Щоб повернутися до попередньої сторінки, програма витягне поточну сторінку зі стеку навігації, а нова найвища сторінка стане активною сторінкою, як показано на рис. 2.

Рисунок 2 – Робота стеку навігації при поверненні до попередньої сторінки

Методи навігації надаються властивістю Navigation будь-яких Page похідних типів. Ці методи надають можливість надсилати сторінки в стек навігації, витягувати сторінки зі стеку навігації та виконувати маніпуляції зі стеком.

Перша сторінка, додана в стек навігації, називається кореневою сторінкою програми, що демонструється в наступному прикладі коду:

```
public App ()
{
  MainPage = new NavigationPage (new Page1Xaml ());
}
```

У результаті в стек навігації міститься екземпляр Page1Xaml ContentPage, де він стає активною сторінкою та кореневою сторінкою програми. Ці дії відображаються на рис. 3.

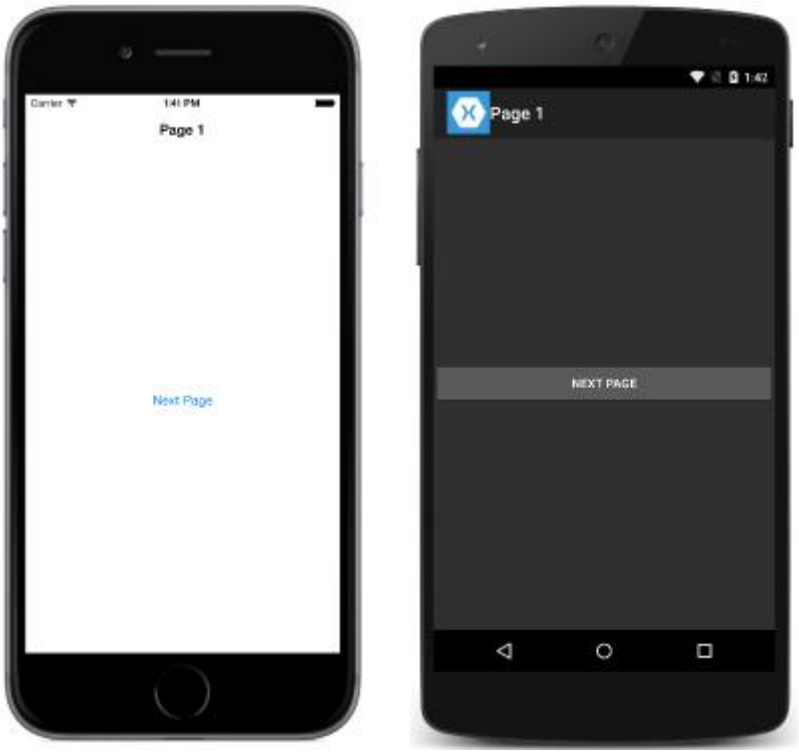

Рисунок 3 – Відображення кореневої сторінки Page1Xaml стеку навігації

Для додавання нової сторінки до стеку навігації використовується код:

```
async void NextPage (object sender, EventArgs e)
{
   await Navigation.PushAsync(new YourPageName());
}
```

Для вилучення нової сторінки зі стеку навігації використовується код:

```
async void BackToPage (object sender, EventArgs e)
{
        await Navigation.PopAsync();
}
```

Властивість Navigation надає методи InsertPageBefore та RemovePage керувати стеком шляхом вставки сторінок або їх видалення (рис. 4, рис. 5):

```
Navigation.InsertPageBefore (new MainPage (), this);
Navigation.RemovePage (new MainPage);
```

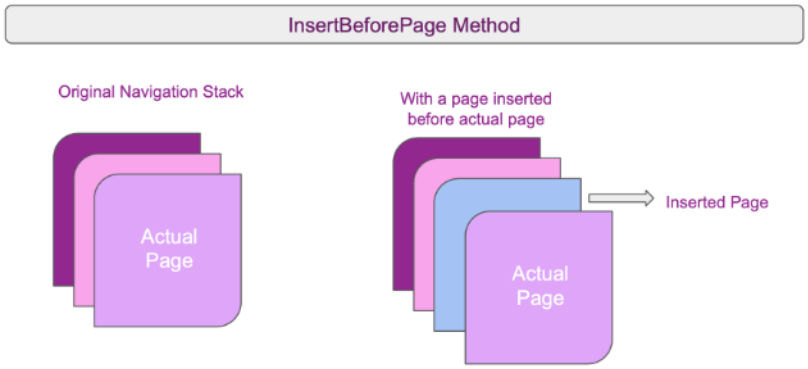

Рисунок 4 – Метод InsertPageBefore

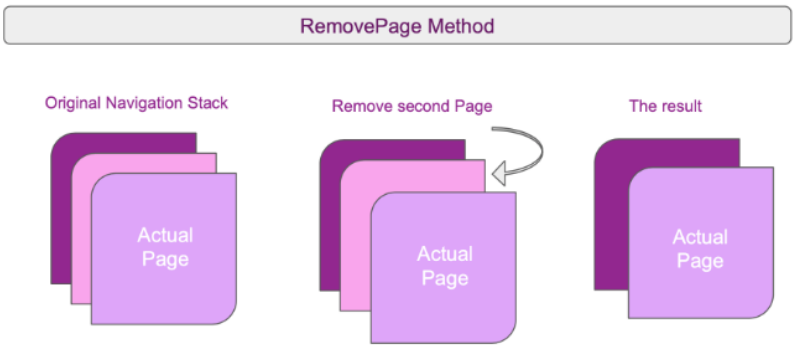

Рисунок 5 – Метод RemovePage

Метод PopToRootAsync видаляє всі сторінки, що містяться в навігаційному стеку, крім кореневої сторінки, роблячи її активною сторінкою.

```
async void ClearNavigationStack (object sender, EventArgs e)
{
        await Navigation.PopToRootAsync ();
}
```

Передати дані між сторінками стеку навігації можливо через constructor або BindingContext об’єкт.

Використовуючи конструктор, дані можемо отримати в два кроки:

Крок 1: Додайте потрібний параметр на сторінці конструктора.

```
public MainPage (string YourName)
{
    InitializeComponent ();
    Name = YourName;
}
```

Крок 2: Передача даних під час виклику сторінки, яка використовувалася раніше. (MainPage)

```
await Navigation.PushAsync(new YourPageName("MariaWhite"));
```

Розглянемо приклад використання об’єкта Binding Context. Створимо метод FillDataClicked та додамо всередину нього дані об’єкта класу Contact. Далі створений об’єкт передаємо на сторінку через об’єкт BindingContext.

```
async void FillDataClicked (object sender, EventArgs e)
{
    var contact = new Contact {
        Name     = "Merrie",
        LastName = "White",
        Country  = "USA"
  };

   var contactsPage = new ContactsPage ();
   contactsPage.BindingContext = contact;
   await Navigation.PushAsync (contactsPage);
}
```

Щоб видалити кнопку «Назад», необхідно виконати фрагмент коду у файлі XAML:

```
NavigationPage.HasBackButton="false".
```

Щоб приховати панель навігації, необхідно виконати фрагмент коду у файлі XAML:

```
NavigationPage.HasNavigationBar="false"
```

Хід роботи

1. Створюємо проект NPage в Visual Studio 19. Вибираємо шаблон Blank. При створенні проекту в опції Platform вибираємо платформу Android під яку буде створюватись проект (рис. 6).

2. У провіднику рішень (Solution Explorer) знаходимо файл App.xaml.cs та відкриваємо його. Змінюємо рядок (рис. 7)

```
MainPage = new MainPage();
```

на

```
MainPage = new NavigationPage (new NPage.MainPage());
```

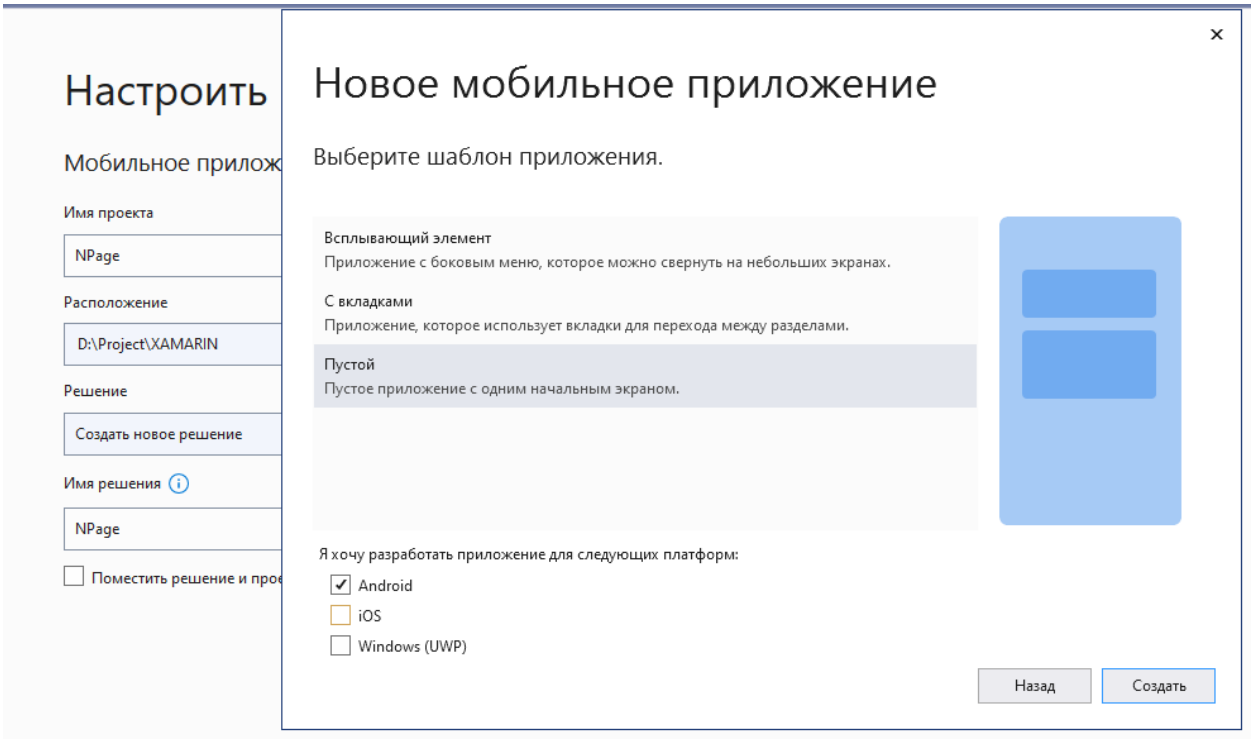

Рисунок 6 – Створення проекту NPage

Рисунок 7 – Файл App.xaml.cs. Додавання до стеку навігації MainPage

3. У провіднику рішень (Solution Explorer) переходимо до файлу MainPage.xaml. ДодаємоатрибутTitle="Main Page" та змінюємо код до такого:

```
<?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="NPage.MainPage"
             Title="Main page">
    <StackLayout Padding="12, 12, 12, 12"
                 VerticalOptions="CenterAndExpand"
                 HorizontalOptions="FillAndExpand">
        <Button Text="Open new page"
                VerticalOptions="Center"
                HorizontalOptions="FillAndExpand"
                Clicked ="Button1_Clicked">
```

```
        </Button>
    </StackLayout>
</ContentPage>
```

4. Додаємо нову ContentPage до проекту (рис .7, рис. 8).

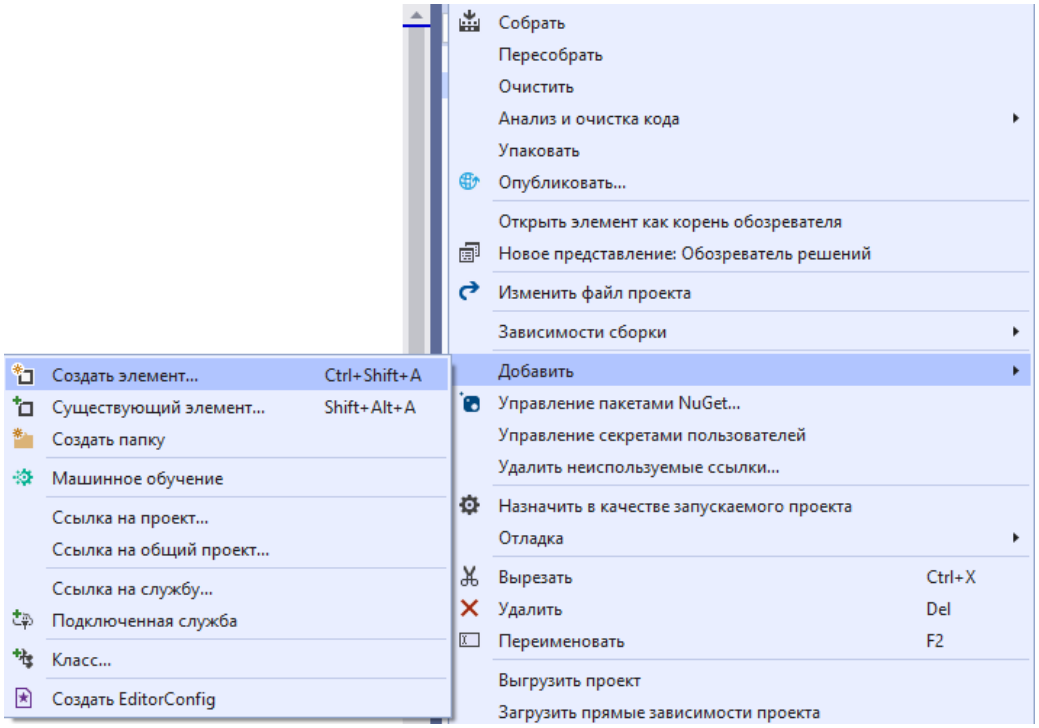

Рисунок 7 – Додавання до проекту нового елементу

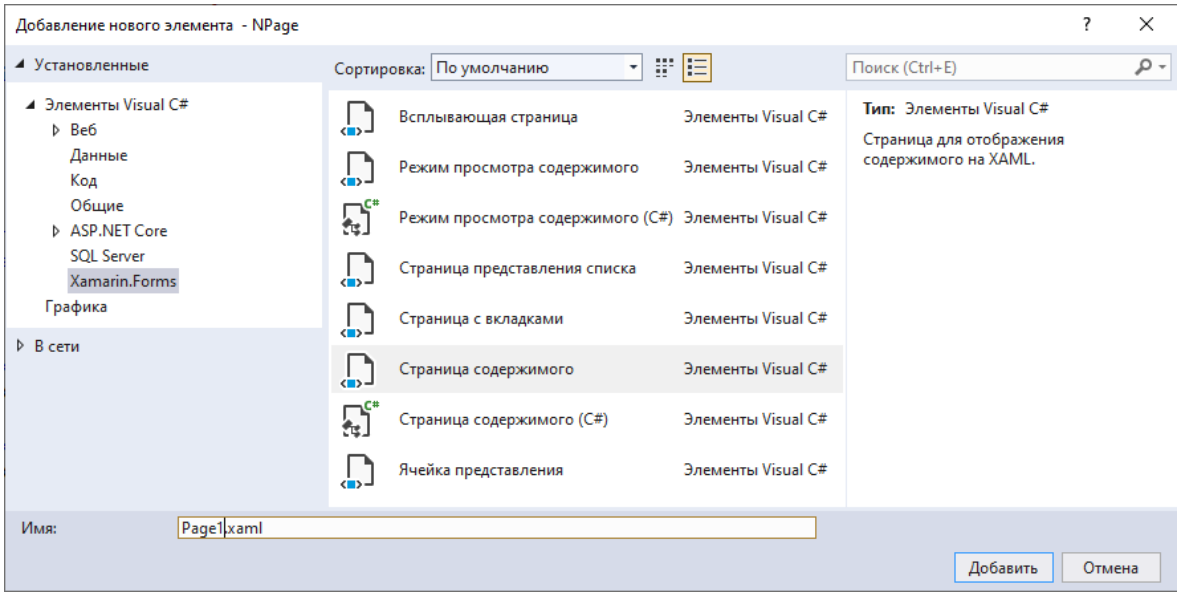

Рисунок 8 – Додавання до проекту ContentPage з ім’ям Page1

5. У провіднику рішень (Solution Explorer) відкриваємо файл MainPage.xaml.cs, у якому додаємо метод обробки кнопки Button1_Clicked на

MainPage. Цей метод буде викликати нову Navigation сторінку.

```
private async void Button1_Clicked(object sender, EventArgs e)
{
  await Navigation.PushAsync(new Page1());
}
```

6. Запускаємо проект на виконання. Спостерігаємо при запуску додатку MainPage. Натиснення кнопки «OPEN NEW PAGE» викликає нову сторінку (рис. 9).

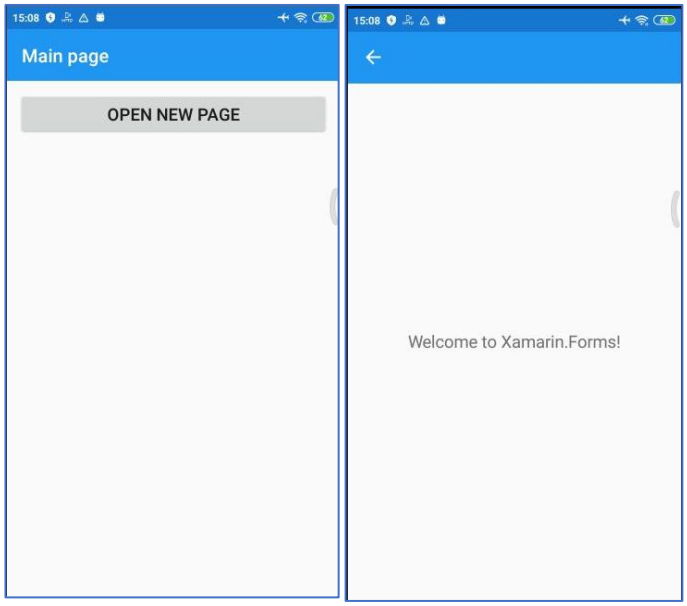

Рисунок 9 – Тестування роботи NavigationPage. MainPage, Page1

7. У провіднику рішень (Solution Explorer) відкриваємо файл Page1.xaml. Додаємо Title та до елементу розмітки StackLayout додаємо кнопку, яка буде викликати сторінку навігації Page2. Відредагувати код так:

```
<?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="NPage.Page1"
             Title="Page 1">
    <ContentPage.Content>
        <StackLayout Padding="12, 12, 12, 12"
                     VerticalOptions="FillAndExpand"
                     HorizontalOptions="FillAndExpand">
            <Button Text="Open new page"
                    VerticalOptions="End"
                    HorizontalOptions="FillAndExpand"
                  Clicked ="Button1_Clicked">
            </Button>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

8. Додаємо до проекту нову ContentPage з назвою Page2. У провіднику рішень (Solution Explorer) відкриваємо файл Page2.xaml. Додаємо Title = "Page 2" та до елементу розмітки StackLayout додаємо 2 кнопки, які будуть здійснювати навігацію до сторінки Page1 або MainPage. Відредагувати код так:

```
<?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="NPage.Page2"
             Title="Page 2">
    <ContentPage.Content>
        <StackLayout Padding="12, 12, 12, 12"
                     VerticalOptions="FillAndExpand"
                     HorizontalOptions="FillAndExpand">
            <Button Text="Back"
                    Clicked ="Button1_Clicked"
                    VerticalOptions="Center"
                    HorizontalOptions="FillAndExpand">
            </Button>
            <Button Text="On main"
                    Clicked ="Button2_Clicked"
                    VerticalOptions="Center"
                    HorizontalOptions="FillAndExpand">
            </Button>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

9. Відкриваємо файл Page1.xaml.cs та створюємо метод обробки кнопки

```
private async void Button1_Clicked(object sender, EventArgs e)
{
     await Navigation.PushAsync(new Page2());
}
```

10. Відкриваємо Page2.xaml.cs та створюємо 2 метода обробки кнопок

```
private async void Button1_Clicked(object sender, EventArgs e)
{
   await Navigation.PopAsync();//викидає сторінку зі стеку
}
private async void Button2_Clicked(object sender, EventArgs e)
{
   await Navigation.PopToRootAsync();
}
```

11. Запускаємо проект на виконання. Спостерігаємо при запуску додатку MainPage. Натиснення кнопки «OPEN NEW PAGE» викликає сторінку Page1. На

сторінці Page1 натиснення кнопки «OPEN NEW PAGE» викликає сторінку Page2 (рис. 10). На сторінці Page2 тестуємо роботу кнопок «BACK», «On MAIN».

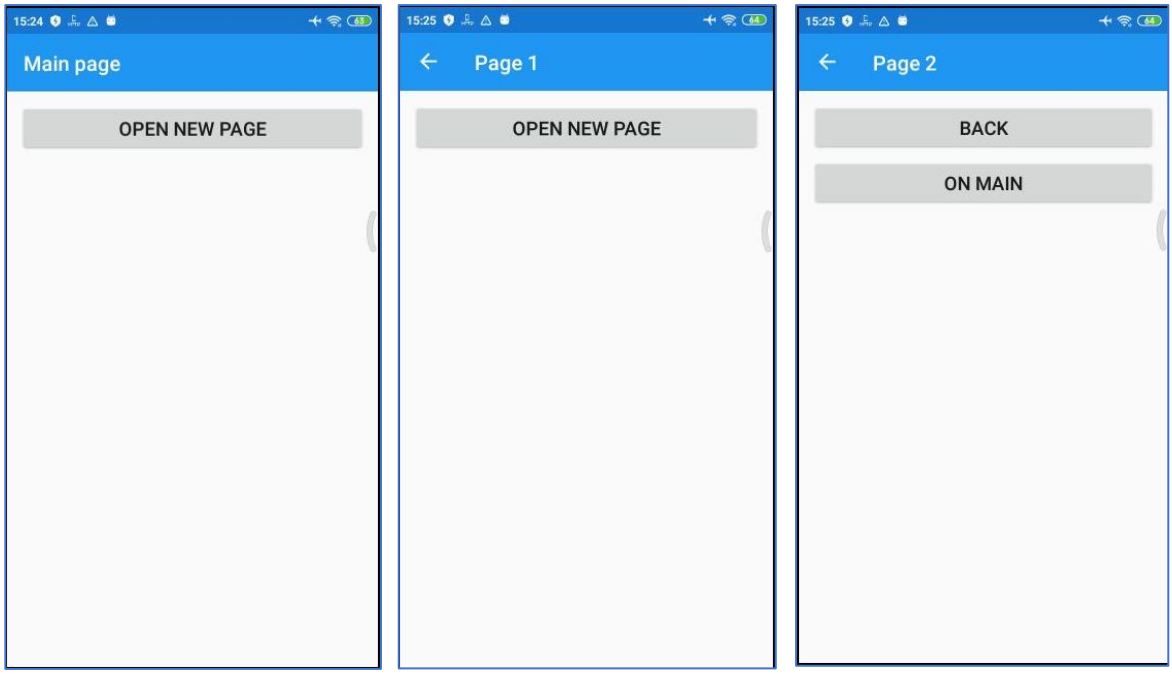

Рисунок 10 –Тестування роботи NavigationPage. MainPage, Page1

12. Робимо правку для Main.Page, Page2.xaml. Змінюємо значення атрибуту VerticalOptions = «CenterAndExpand» (рис. 11).

```
<?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="NPage.Page2"
             Title="Page 2">
    <ContentPage.Content>
        <StackLayout Padding="12, 12, 12, 12" VerticalOptions="CenterAndExpand" HorizontalOptions="FillAndExpand">
            <Button Text="Back" Clicked ="Button1_Clicked"
                VerticalOptions="Center" HorizontalOptions="FillAndExpand"></Button>
            <Button Text="On main" Clicked ="Button2_Clicked"
                VerticalOptions="Center" HorizontalOptions="FillAndExpand"></Button>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

Рисунок 11 – Зміст файлу Page2.xaml

Робимо правку для Page1.xaml. Змінюємо значення атрибуту VerticalOptions = «EndAndExpand» (рис. 12).

```
         Title="Page 1">
<ContentPage.Content>
    <StackLayout Padding="12, 12, 12, 12" VerticalOptions="EndAndExpand" HorizontalOptions="FillAndExpand">
        <Button Text="Open new page" Clicked ="Button1_Clicked"
            VerticalOptions="End" HorizontalOptions="FillAndExpand"></Button>
    </StackLayout>
</ContentPage.Content>
```

Рисунок 12 – Зміст файлу Page1.xaml

13. Запускаємо проект на виконання. Спостерігаємо при запуску додатку MainPage. Натиснення кнопки «OPEN NEW PAGE», яка знаходиться по центру,

викликає сторінку Page1. На сторінці Page1 кнопка «OPEN NEW PAGE» знаходиться внизу сторінки та викликає сторінку Page2 (рис. 13). На сторінці Page2 тестуємо роботу кнопок «BACK», «On MAIN», які розташовуються по центру сторінки.

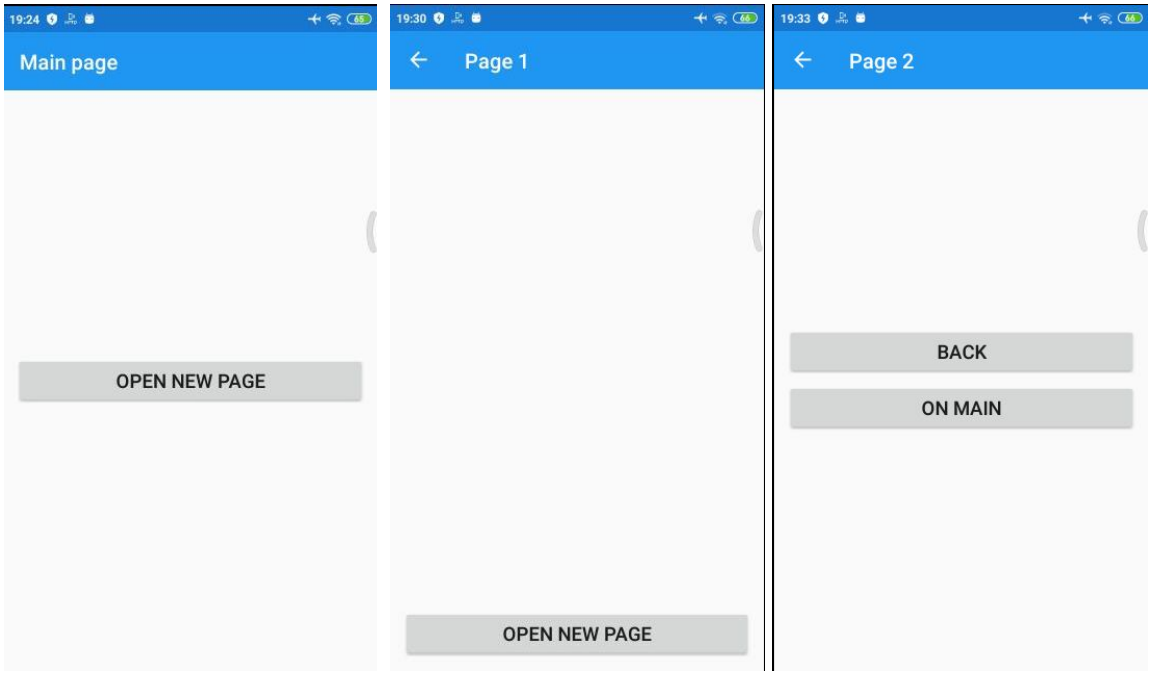

Рисунок 13 –Тестування роботи NavigationPage. MainPage, Page1. Зміна макету сторінок

14. На сторінці App.xaml.cs змінюємо колір елемента бара

```
public App()
{
    InitializeComponent();
    MainPage = new NavigationPage(new NPage.MainPage())
    {
        BarBackgroundColor = Color.FromHex("009688")
    };
}
```

15. Приховаємо на Page1 навігаційний бар. Для цього у файлі mainPage.xaml.cs у методі обробки кнопки Button1_Clicked використовуємо метод SetHasNavigationBar з параметром false

```
private async void Button1_Clicked(object sender, EventArgs e)
{
    Page1 pg1 = new Page1();
    NavigationPage.SetHasNavigationBar(pg1, false);
    await Navigation.PushAsync(pg1);
}
```

16. Запускаємо проект на виконання. Протестувати роботу мобільного додатку. Зробити screenshot усіх навігаційних сторінок при натисненні кнопок та кнопок навігації. Оформити звіт.

Контрольні питання

1. Як працює стек навігації?
2. Як додати нову сторінку до стеку навігації?
3. Як вилучити сторінку зі стеку навігації?
4. Яке призначення методів InsertPageBefore, RemovePage?
5. Як передати дані між сторінками стеку навігації через constructor?
6. Як передати дані між сторінками стеку навігації через BindingContext?
7. Яке призначення оператора away?
8. Чому застосовують асинхронне програмування при роботі зі стеком навігації?

Список використаних джерел

1. Hierarchical Navigation. URL: https://docs.microsoft.com/en-us/xamarin/xamarin-forms/app-fundamentals/navigation/hierarchical
2. Applying Simple Navigation in Xamarin Forms. URL: https://www.telerik.com/blogs/applying-simple-navigation-xamarin-forms
3. NavigationPage. URL: https://mauiman.dev/navigationpage.html