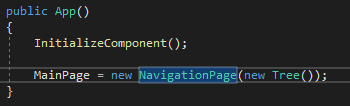
Soft 262 Task 2: View based hierarchical navigation tutorial

**How perform hierarchical navigation**

<https://docs.microsoft.com/en-gb/xamarin/xamarin-forms/app-fundamentals/navigation/hierarchical>

To set up a hierarchical navigation system, the first thing you need to do is in the app.xaml.cs declare the main page as a navigation page, this can be done by passing the page you want as the root page as a parameter to a new navigation page:

For this example, the root page is “tree”, and the child pages are branches.



Then to get to the next page you can use the function

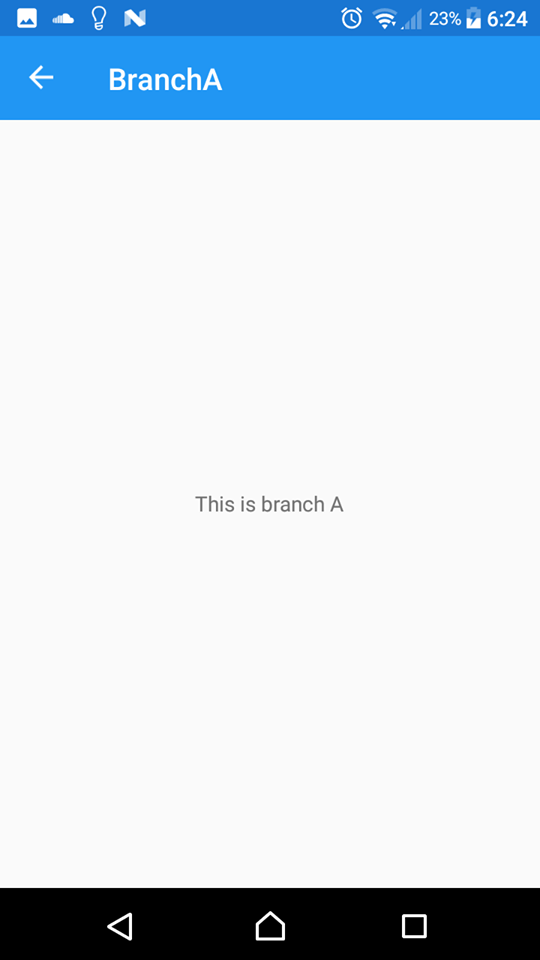


This pushes whatever page to the top of the navigation stack. The navigation stack works by pushing whatever new page you want to the top of the stack which is then the active page the user is able to see, and when the user presses the back button, the active page is popped from the stack removing it and sending the user to the previous page (the page bellow the top in the stack).

This is the code for the root page tree, it only contains the three event handlers for when three buttons are pressed, each adding a new branch page to the top of the stack.

The XAML for the tree page is just three buttons within a stack layout and a label letting you know you’re on the root page.



****

The first image is the root page tree, the second is the branch page after pressing the first button “GO TO BRANCH A”.

You can continue to push as many pages to the navigation stack as you want.

**How to pass data forward**

<https://docs.microsoft.com/en-gb/xamarin/xamarin-forms/app-fundamentals/navigation/hierarchical#passing-data-through-a-page-constructor>

One way to pass data forward is to pass data as a parameter through the page constructor, this was used in the previous example to easily show which branch button was tapped on.

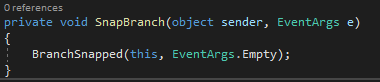


The string “A” being passed to the branch page.

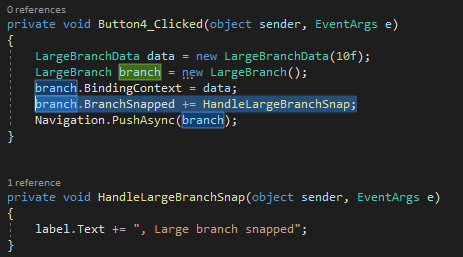
Another way to pass data forward is by using bindings, before adding a page to a

**How to pass data backwards**

One easy way to pass data back is by using events, you can have the page broadcast the event and have the parent page listen for the event.



The child page broadcasting the event when the button is clicked

****

The parent page listening for the event

**Use cases for hierarchical navigation**

Hierarchical navigation is useful for when the user needs to navigate through many different pages that may be sorted by relevance, a example of this could be the settings menu in an app, different settings for different parts of the app will be split up and the user will be able to easily navigate to the specific settings they are wanting to adjust.

Another use for hierarchical navigation could be navigating through large amounts of information, there could be different pages for different subjects, and then those subjects could be subdivided into separate topics to make it easier for the user to find the specific piece of information that they’re looking for.

**Comparison of view based and mvvm based navigation**

Mvvm navigation can handle passing values/ data a lot better compared to a view-based navigation due to being able to pass view models, this would make a mvvm model more ideal for a more complex app’s that have lots of user inputs, storing of data and data manipulation. However, the positive of using a view-based navigation system is that it’s a lot simpler and is suited to smaller, simpler applications where mvvm would be overkill.

**Code:**

<https://github.com/wmastersPlym/SOFT262Test/tree/master/Task2/SOFT262Task2>