



Do U(I)Test?

Best Practices and Tips for Xamarin.UITest

Who's this gal?



Sweekriti “Sweekly” Satpathy

Chief Lego Builder - Mobile Customer Advisory Team, Microsoft

swsat@microsoft.com

@SweekritiS

Xamarin.UITest Recap

- Automated User Interface Testing Framework written in C# using NUnit
- Recently added support for Nunit 3
- Run tests against thousands of real devices using App Center
- Integrates with CI/CD pipelines
- Can be used with :
 - Xamarin Forms/Native Apps
 - Native iOS/Android Apps
 - ReactNative, Ionic

```
[TestFixture]
public class Tests
{
    [SetUp]
    public void BeforeEachTest()
    {
        // Set up app
    }
    [Test]
    public void LogInTest()
    {
        app.Tap(x => x.Id("log-in-
button"));
        // ...
    }
}
```

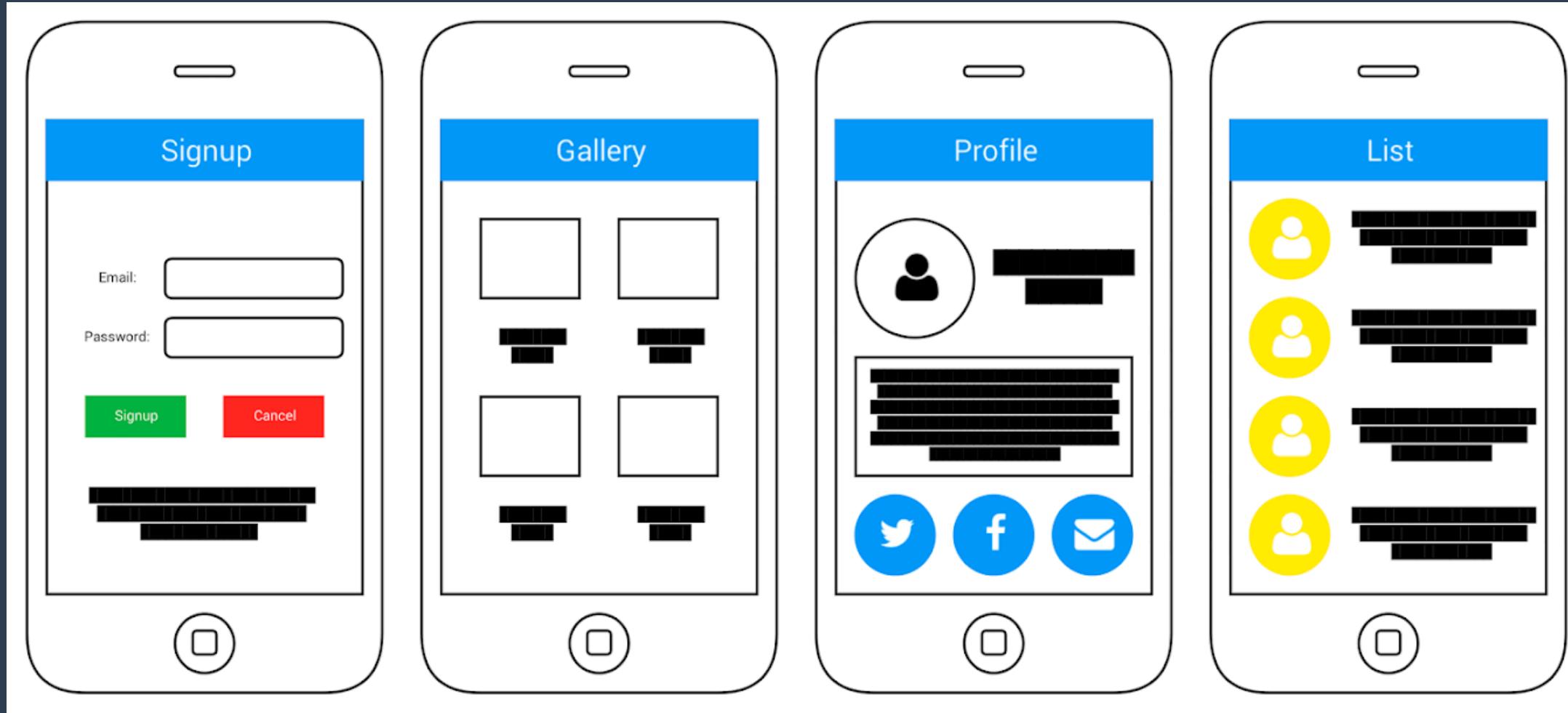
Page Object Pattern

What?

- Each page in your app is modeled by a class in your test project
- Page representations handle all interaction with their target page
- This simplifies test methods down to just calling methods on pages

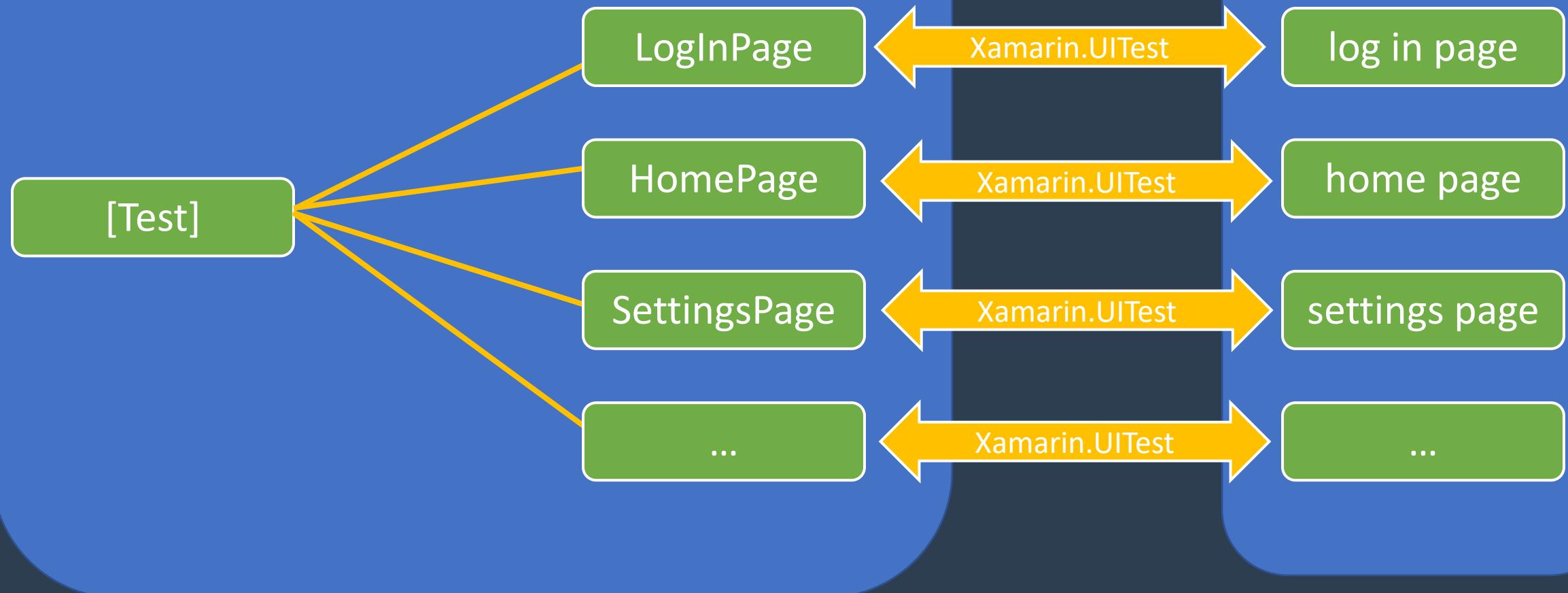
Why?

- It's intuitive and easy to implement
- Good separation of responsibility
- Also makes it fairly easy to do cross-platform tests as long as pages are similar between platforms



Visually Imagine App -> Each Page -> Each Class

Test Suite



Test using the page object pattern

```
[TestFixture]
public class Tests
{
    [SetUp]
    public void BeforeEachTest()
    {
        // Set up app
    }

    [Test]
    public void LogInTest()
    {
        app.LoginPage().Logging("usebut", "password");
    }
}
```



Let's write some Tests

Who doesn't love some Live Coding on Stage? ;)

A few other general UI Test Tips

- Test on Real Devices
- Split tests into small chunks wherever possible
 - Use backdoors to avoid repetition
- Put breakpoints -> Run Tests in Debug
- `using Query = System.Func<Xamarin.UITest.Queries.AppQuery,
Xamarin.UITest.Queries.AppQuery>;`
- Use NUnit features
 - TestCase
 - Categories – “smoke” “nightly”

A few other general UI test best practices

- Make each test self contained
 - Independent tests that start with a clean state
- Be as deterministic as possible
 - Use backdoors to set up the app
 - Avoid branching on UI state
 - Avoid branching in general (if, else and try, catch)
 - Use assertions (WaitForElement, WaitForNoElement, and Assert)
 - Take many screenshots
 - Try not to do anything randomly
 - Try not to use Thread.Sleep
 - Be careful with loops, especially the while loop



Questions?



Thank You for Listening!

Find me at the Microsoft Booth!

Email : swsat@microsoft.com

Twitter : [@Sweekriti91](https://twitter.com/Sweekriti91)

All the LINKS!

- [Code \(App + Tests\)](#)
- [Page Object Pattern \(C# Sample\)](#)
- [Page Object Pattern \(Appium Sample\)](#)
- [Page Object Pattern \(XCUITest Sample\)](#)
- [Page Object Pattern \(SpecFlow Sample\)](#)
- YouTube: [Best Practices for User Interface Automation | The Xamarin Show](#)
- Channel9: [Best Practices for User Interface Automation](#)
- Best Practices and Tips for Using Xamarin.UITest - [Blog Post](#)

