Android, iOS and Hybrid Applications

Mobile-Development

Introduction

- Fabrizio Niedda
- fabrizio@encodo.ch
- Bachelor
- Encodo Systems AG, Winterthur
- Mobile Development in Banking, Pharma, User Assistance etc.

Introduction

- Name?
- Where do you work?
- What Technologies are you experienced in?
- Any experience in Mobile Development?
- What do you want to learn about it?

General

- Project week - development of an app
- From 21:00 open session - questions, problems etc.
- One written exam after the project week
- App from the project week will be rated as well

PROGRAM

DAY 1

- Overview Mobile Development
- Xamarin Forms basics
- Setup Xamarin Sample-App
- Cross Platform Basics
- Navigation Basics

Day 2

- MVVM
- XAML for Forms
- Controls and differences to WPF
- Bindings
- Commands

Program

Day 3

- Dialogs
- Styling
- Inversion of Control (IOC)
- Testing

Program

Day 4

- Notifications
  - Local
  - PUSH/Remote

Program

Day 5

- Hybrid Applications
- Interoperability with the native part
  - Design a possible interface
  - Present your approach
- Create a small working sample

PROGRAM

DAY 6

- Basics (Block Cyphers, PK-Infrastructure)
- Mobile Security
- Biometrics

Program

Day 7

- Building & Integration
  - fastlane
  - gradle
- Testing & Deployment
  - TestFlight & AppStore
  - Beta-Track & Play Store

Program

Day 8

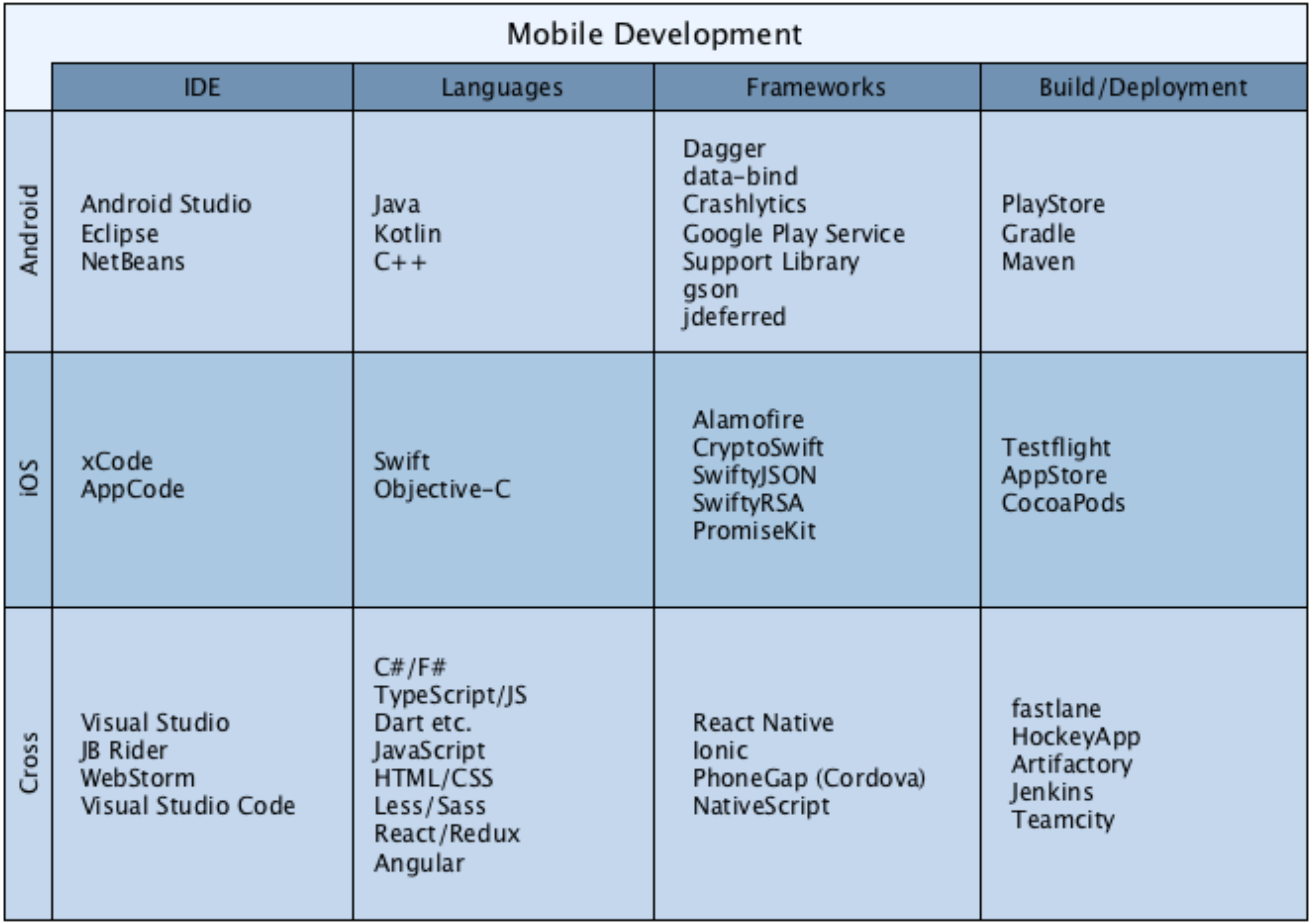
- Written exam
- Continue working on the project week app
    - Short review with every group

PROGRAM

DAY 9

- Present your apps from the project week

Overview Mobile Development



Mobile Development

	IDE	Languages	Frameworks	Build/Deployment
Android	Android Studio Eclipse NetBeans	Java Kotlin C++	Dagger data-bind Crashlytics Google Play Service Support Library gson jdeferred	PlayStore Gradle Maven
iOS	xCode AppCode	Swift Objective-C	Alamofire CryptoSwift SwiftyJSON SwiftyRSA PromiseKit	Testflight AppStore CocoaPods
Cross	Visual Studio JB Rider WebStorm Visual Studio Code	C#/F# TypeScript/JS Dart etc. JavaScript HTML/CSS Less/Sass React/Redux Angular	React Native Ionic PhoneGap (Cordova) NativeScript	fastlane HockeyApp Artifactory Jenkins Teamcity

Questions?

- Short break

SETUP

- Install Visual Studio Xamarin Tools
- One of the following:
  - Setup an emulator (Android) or simulator (iOS)
  - Setup your device to allow debugging
- Install SmartGit (https://www.syntevo.com/smartgit/)
- Provide me with your github username

SMARTGIT

- Pull vs Fetch
- Push vs Commit
- Local vs Remote
- Merge vs Rebase

- Demo
- Clone the “hfu2019docs” & “hfutodoapp” repository
- FabrizioNiedda is my account

CREATE A PROJECT

- Pick Xamarin.Forms (with .NET Standard)
- You’ll see three projects (Shared, iOS, Android)
- Shared one is a .NET Standard project (we’ll get to it)

- Walkthrough
- Everyone get it running!

Overview

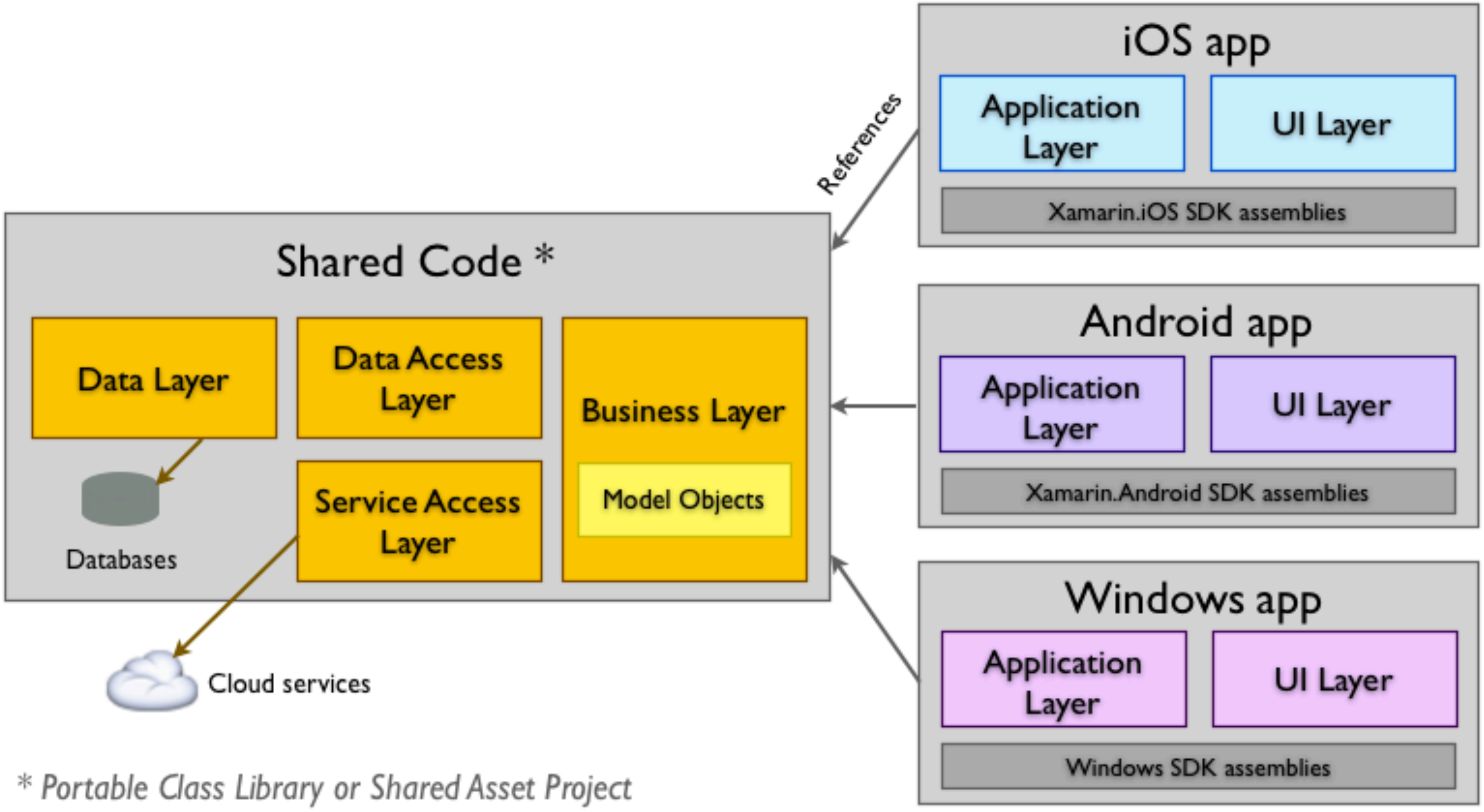
- Xamarin was created by Mono
- Use C# to develop on Android and iOS native
- Acquired and integrated by MS

OVERVIEW – XAMARIN CLASSIC

- Xamarin.iOS & Xamarin.Android (Classic)
  - They basically map the platform 1:1
  - Support new APIs in 24 hours
  - Share the “Core” libraries
  - No shared UI - unless you use a Hybrid approach

https://docs.microsoft.com/en-us/xamarin/cross-platform/app-fundamentals/building-cross-platform-applications/understanding-the-xamarin-mobile-platform

OVERVIEW – XAMARIN CLASSIC



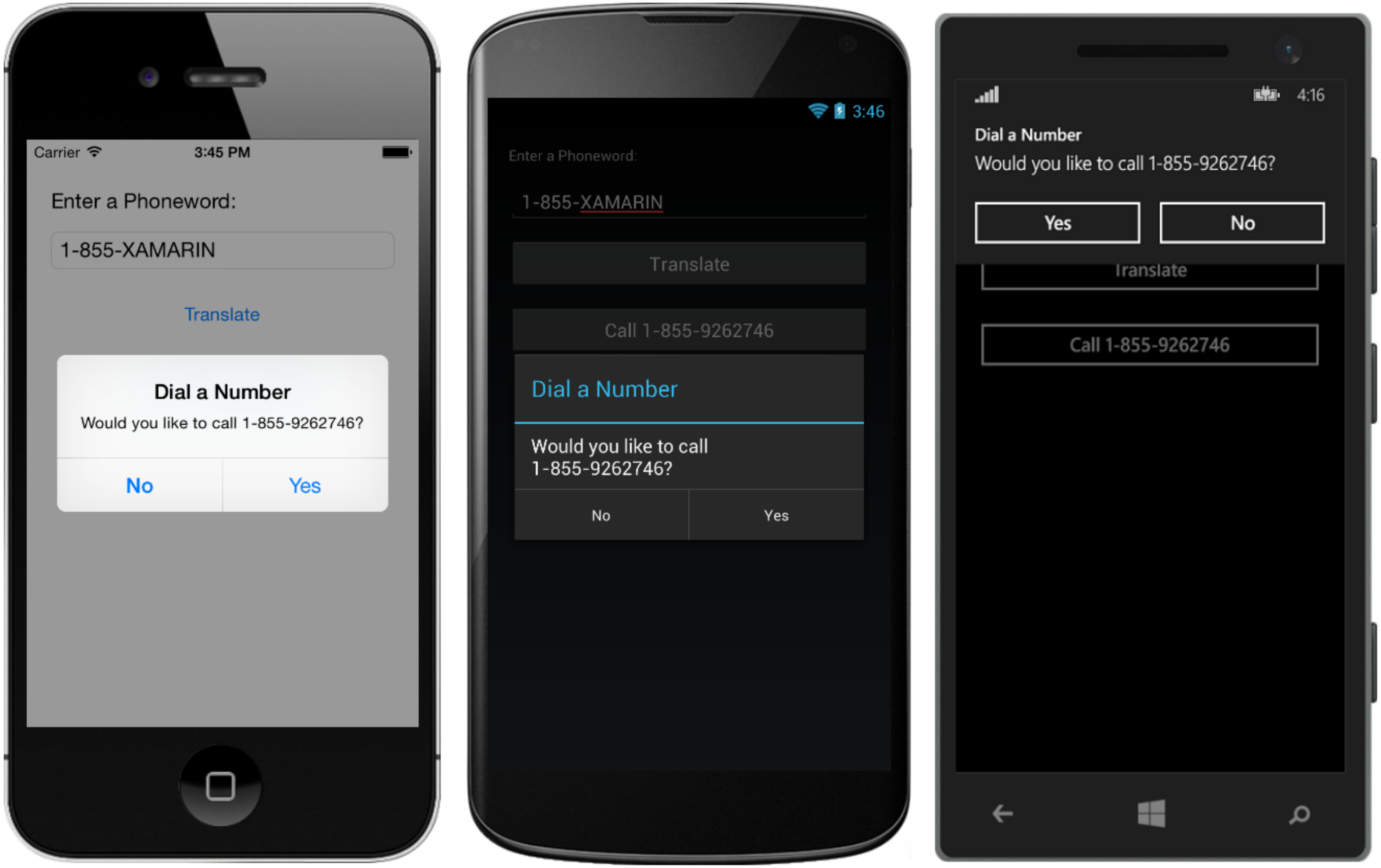
* *Portable Class Library or Shared Asset Project*

OVERVIEW – XAMARIN FORMS

- Xamarin.Forms
  - Share UI code and write it in XAML
  - Use DependencyService (IoC) to access platform features
  - Abstraction of features (Dialogs, Notifications, etc.)
  - Built in support for navigation

https://docs.microsoft.com/en-us/xamarin/xamarin-forms/

Overview – Xamarin Forms



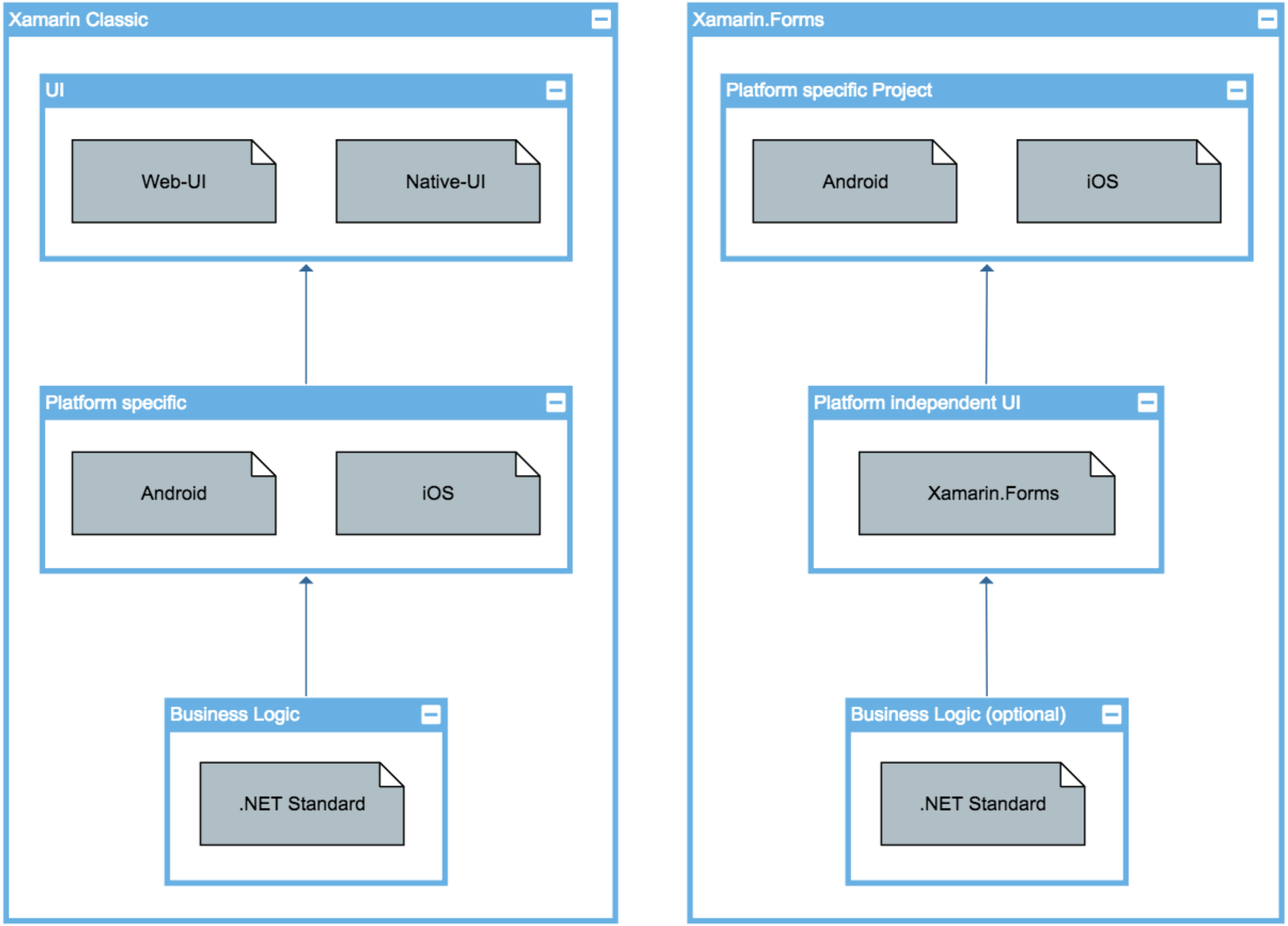
Enter a Phoneword:

1-855-XAMARIN

Dial a Number

Would you like to call 1-855-9262746?

Overview –Architecture



.NET Standard

- Version 2.0 is the latest
  - The higher the version the more APIs are available
- Think of it as an “interface” it does not contain actual code
  - Mono implements .NET Standard for iOS/Android
  - .NET Framework implements it for Windows
- You can find the definitions on github

https://github.com/dotnet/standard/blob/master/docs/versions/netstandard2.0.md

.NET Standard

- Some popular APIs:
  - File (System.IO)
  - Collections & LINQ
  - Task & async await
  - Http (Client) (System.NET)

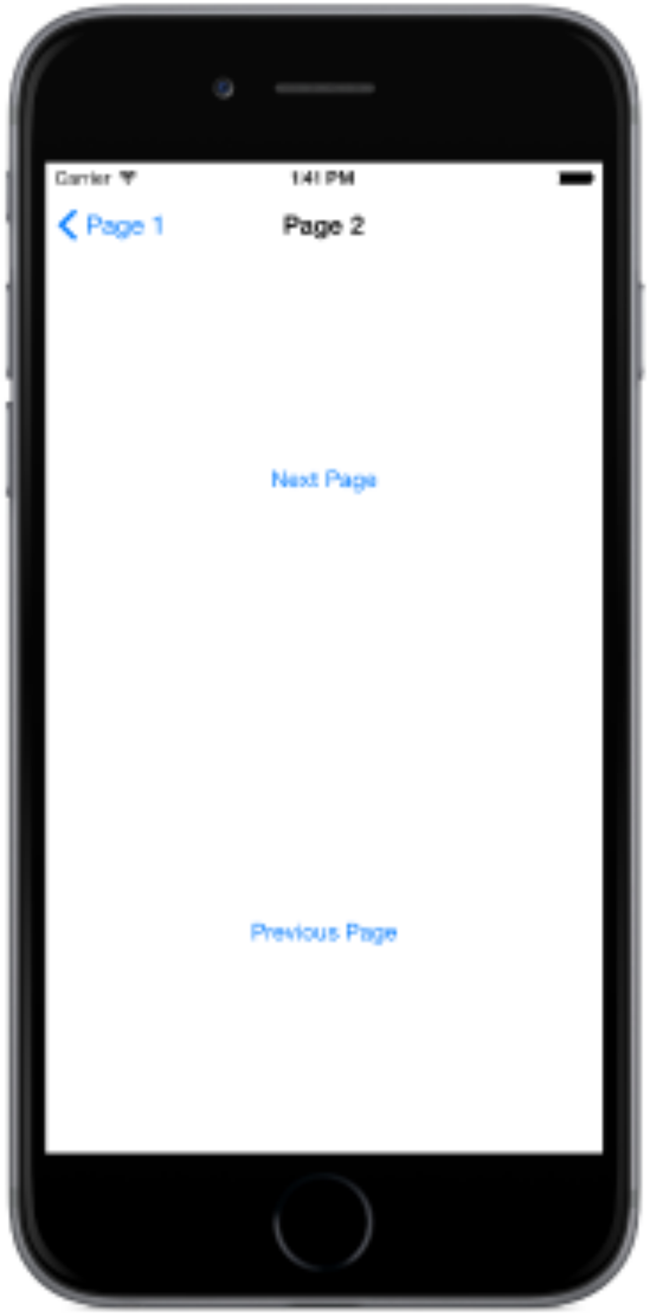
Xamarin Forms: Navigation

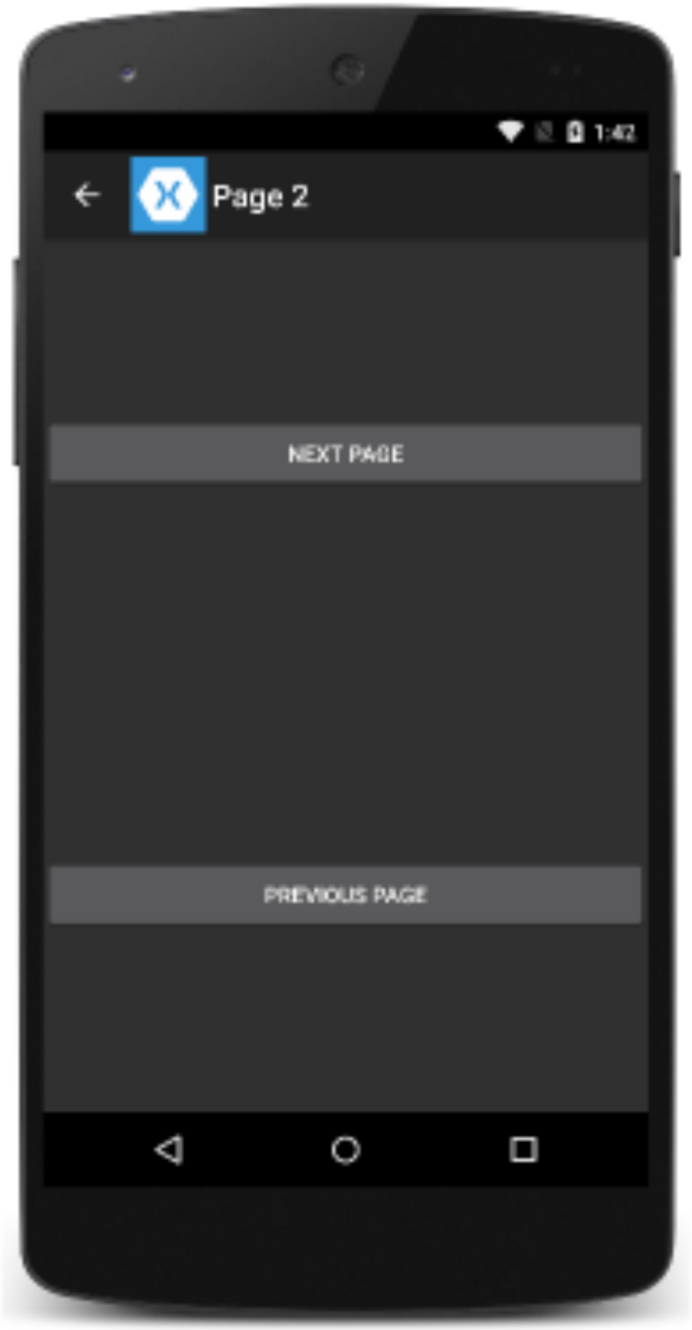
- Wrap your start page in a “NavigationPage“
- Push other pages on top of the Stack

```
// App.xaml.cs
MainPage = new NavigationPage(new MainPage());

// Navigate to the "ListPage"
this.Navigation.PushAsync(new ListPage());

// Pop the top view and return to the previous one.
this.Navigation.PopAsync();
```

https://docs.microsoft.com/en-us/xamarin/xamarin-forms/app-fundamentals/navigation/hierarchical

Navigation


Carrier
1:41 PM
Page 1
Page 2
Next Page
Previous Page


1:42
Page 2
NEXT PAGE
PREVIOUS PAGE

Navigation

- Example

Think about a Project

- It should include
  - Notifications
  - Some (fake) login scenario
  - Alerts (Yes/No Dialog)
  - Some input fields
  - A list or something similar
- TODO-App for example

Setup the Application Layers

- Setup the necessary navigation pages
- Push to github
- Invite me to your project
- Test the navigation