

Vývoj mobilních aplikací pomocí frameworku Xamarin

Roman Jašek

Microsoft Most Valuable Professional
Riganti s.r.o.

roman.jasek@riganti.cz

Tradiční přístup k vývoji aplikací



iOS App

Objective-C
Swift

Xcode



Android App

Java
Kotlin

Android Studio

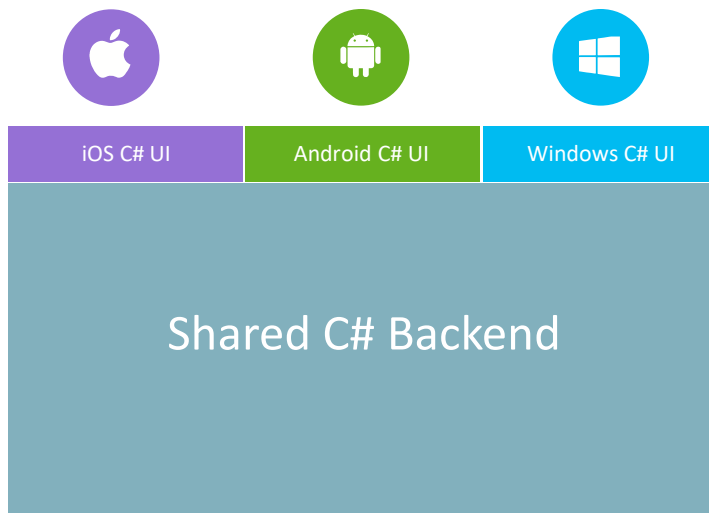


Windows App

C#

Visual Studio

Xamarin



Xamarin

- Společná logika pro všechny platformy
- UI psáno v C#
- Možnost využít vývoj podobný vývoji pro specifické platformy

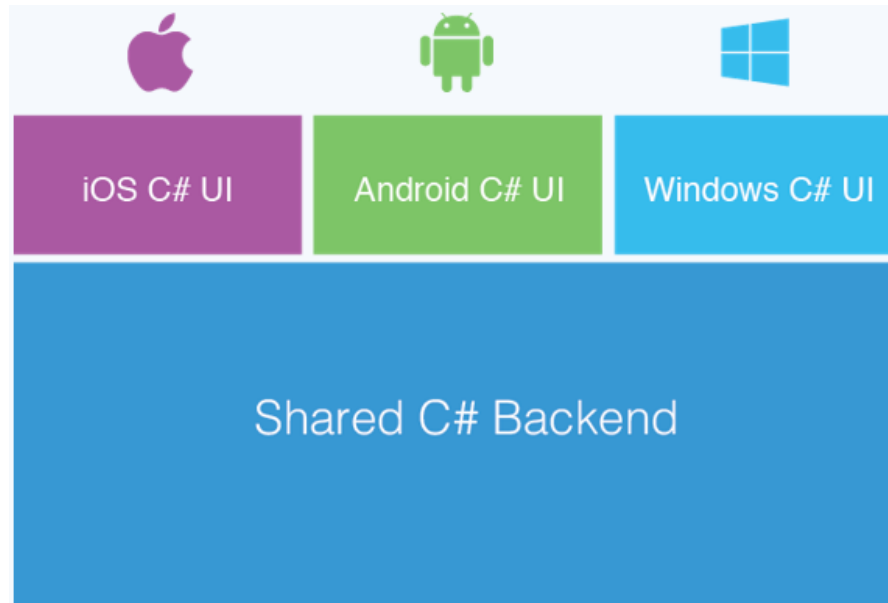
Xamarin Android

- Celý vývoj je možný ve Visual Studiu
- Podpora pro .axml
- Integrovaný designer
- Kód psaný v C#
- Možnost debugovat na zařízeních

Xamarin iOS

- Je nutno mít někde Mac
- Vývoj probíhá ve Visual Studiu
- Mac na síti zbuildí a spustí aplikaci
- iOS Simulátor běží na Macu, ale existuje Xamarin iOS Simulator, který ho „streamuje“ do Windows
- Možnost debugovat na zařízení

Platformy - old



Platformy - updated

Oficiální podpora*



UWP

Komunita



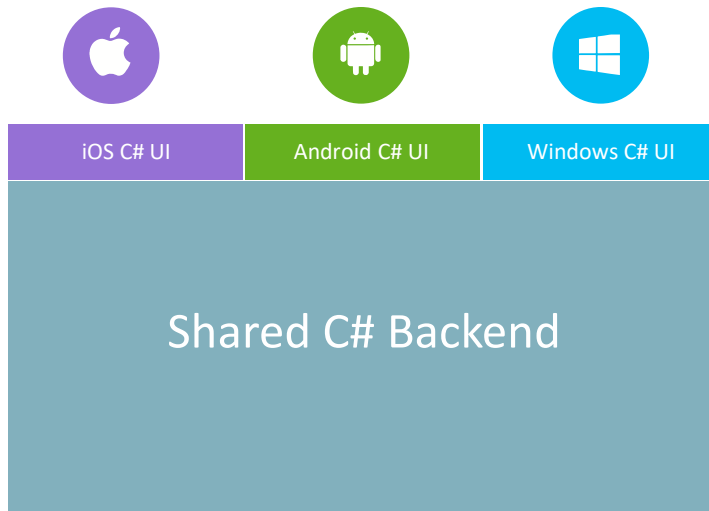
WPF



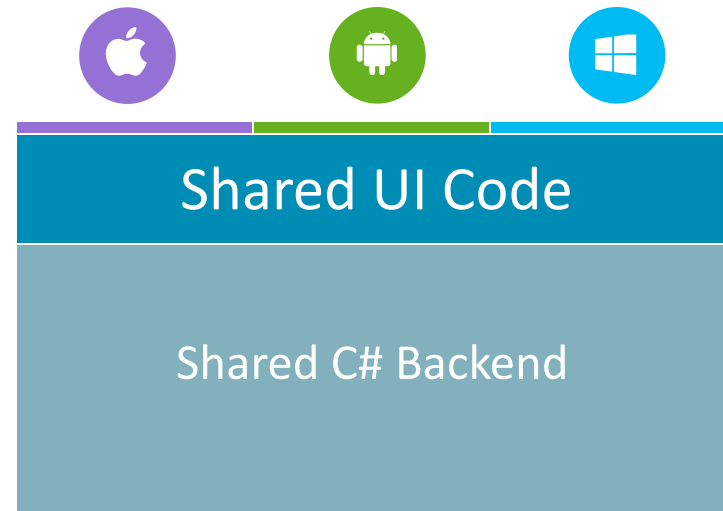
ASP.NET Core

*Tizen je podporován Samsungem

Xamarin + Xamarin.Forms

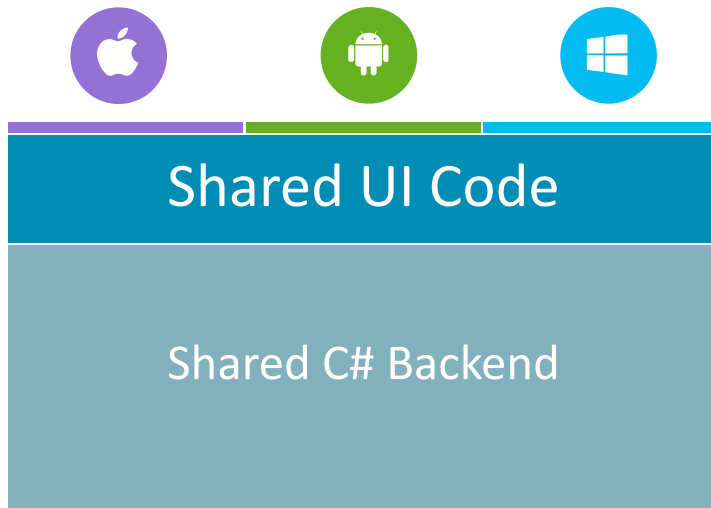


Xamarin



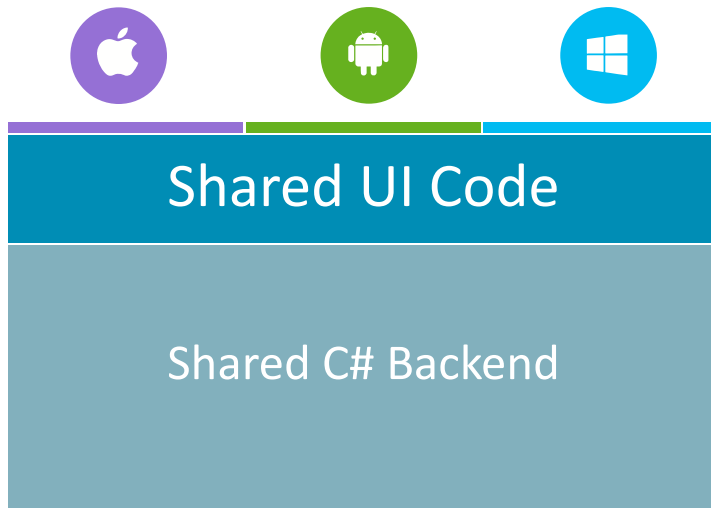
Xamarin.Forms

Xamarin Forms



- Společná logika pro všechny platformy
- UI psáno v C#
- Možnost využít vývoj podobný vývoji pro specifické platformy
- **Společné uživatelské rozhraní pro všechny platformy**

Aktuální stav



- ✓ 40+ stránek, layoutů, komponent (XAML)
- ✓ Data binding
- ✓ Navigace
- ✓ Animace
- ...

XAML

- Formát pro serializaci hierarchie objektů
- Nejen pro popis uživatelského rozhraní
- Mapování .NET namespaces na XML namespaces
- Možné propojení s code-behind třídou

Kde všude XAML potkáme?

- Windows Presentation Foundation (WPF)
 - Poprvé v .NET 3.0
- Silverlight
- Windows Phone
- Universal Windows Platform (UWP)
- Windows Workflow Foundation (WF)
 - Neslouží pro popis UI
- Xamarin.Forms

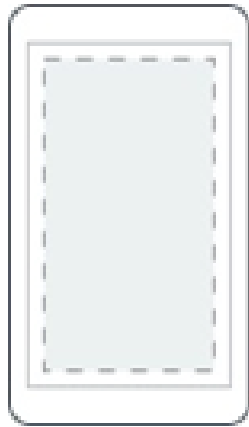
XAML

- `x:Class` ... třída s code-behindem
- `UserControl` ... z čeho tato třída dědí
- `xmlns:x` ... speciální namespace pro účely XAMLu (povinný)
- `xmlns` ... namespace s vestavěnými komponentami

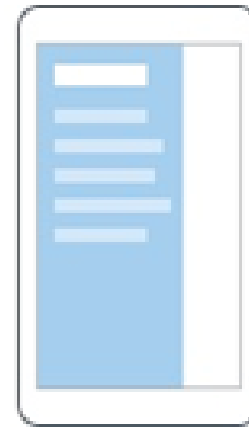
```
<?xml version="1.0" encoding="utf-8" ?>  
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"  
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  
             x:Class="XamarinCvut.Forms.Views.Page1">  
</ContentPage>
```

Stránky

- Content Page
Jednoduchý obsah
- MasterDetail Page
Seznam + detail položky



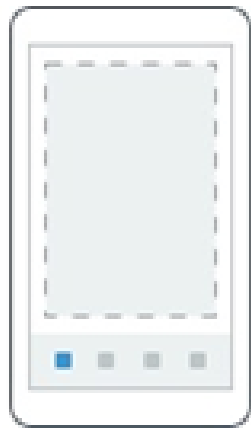
Content



MasterDetail

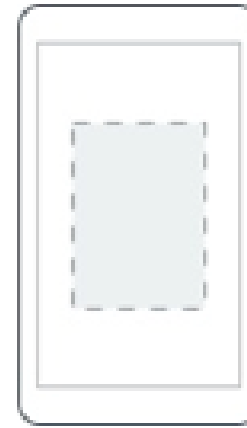
Stránky

- TabbedPage
Záložky



Tabbed

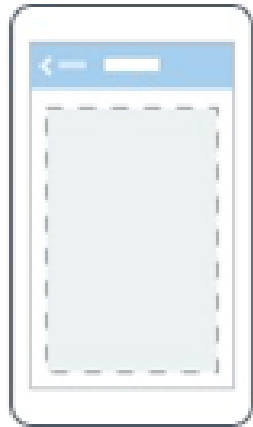
- CarouselPage
Stránky vedle sebe



Carousel

Stránky

- NavigationPage
Aby fungovala navigace



Navigation

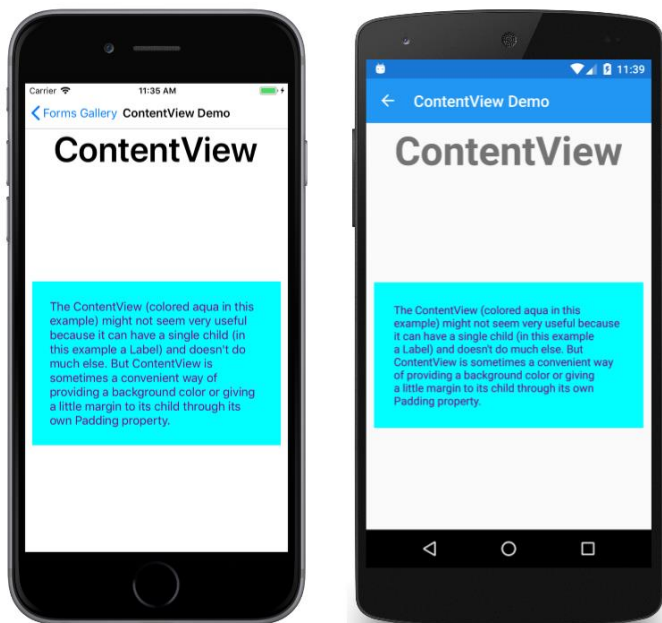
Stránky

- První, co se zobrazí, je v souboru App.cs
 - Defaultně MainPage

Layouts – jedna komponenta

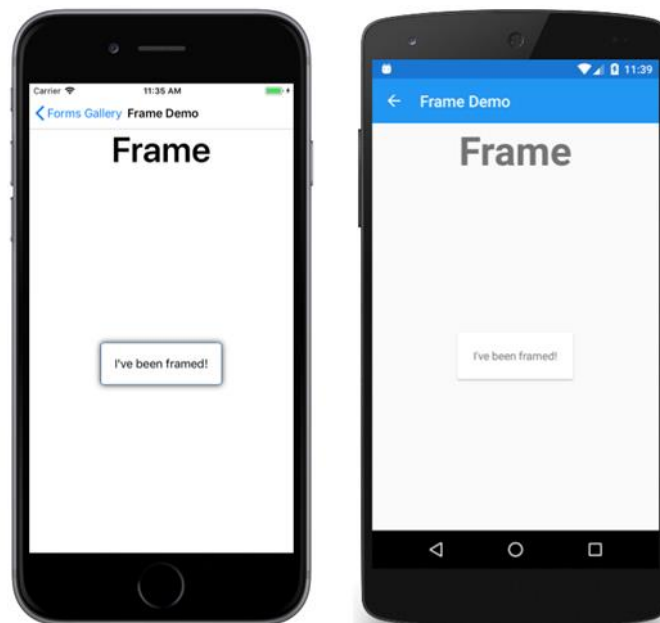
- ContentView

Jeden obsah, bázová třída



- Frame

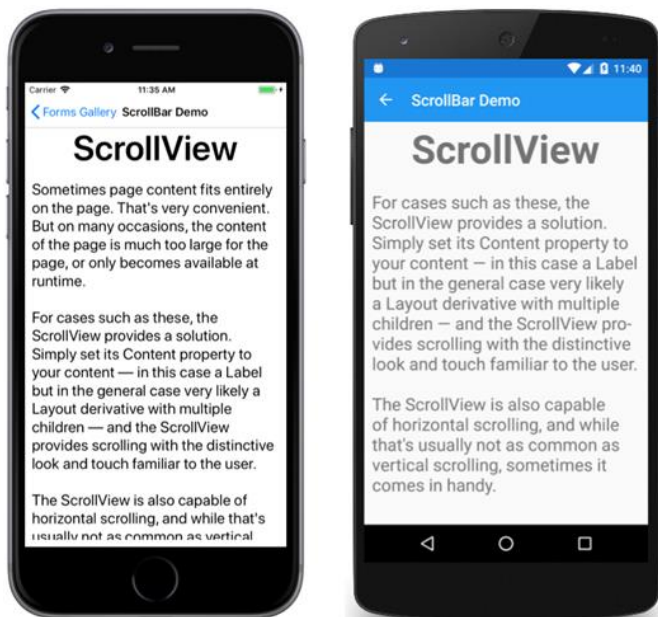
Možnost přidat rámeček



Layouts – jedna komponenta

- ScrollView

Pokud se nevejde, scrolluje se



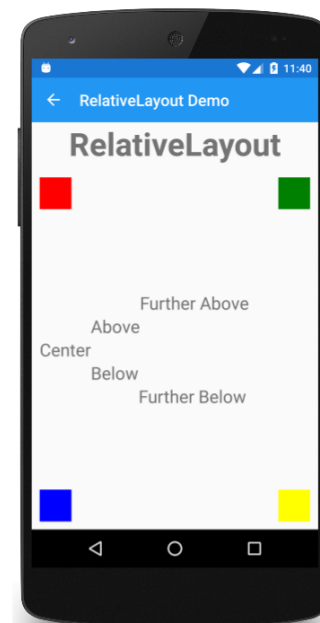
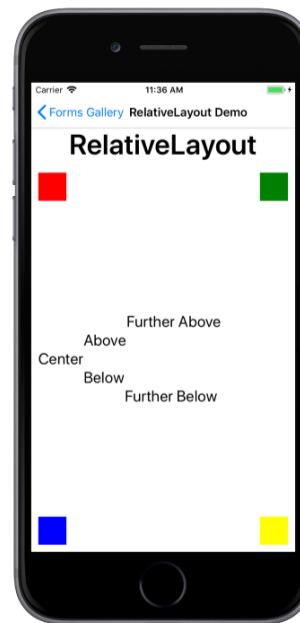
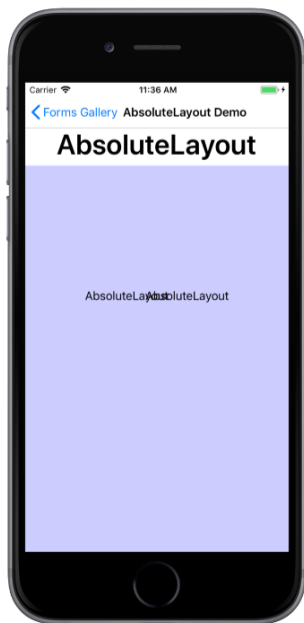
Layouts – více komponent

- **RelativeLayout**

Absolutní pozicování komponent

- **RelativeLayout**

Rozmístění pomocí constraintů



Layouts – více komponent

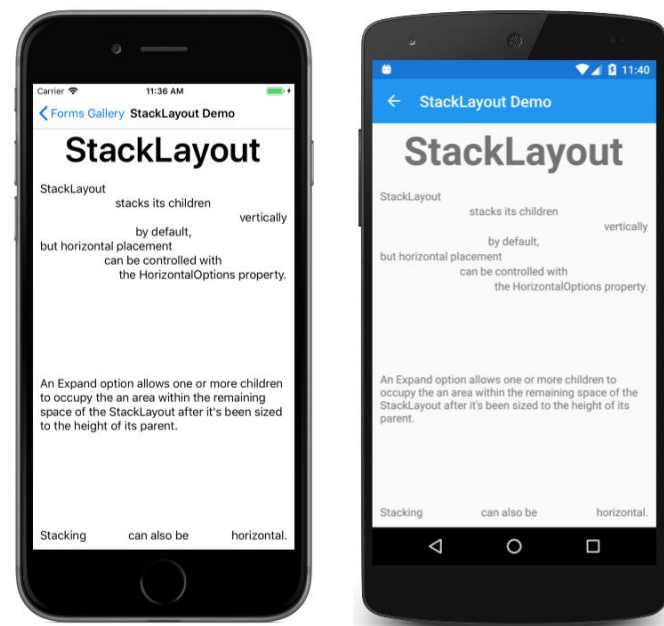
- Grid

Tabulkový layout



- StackLayout

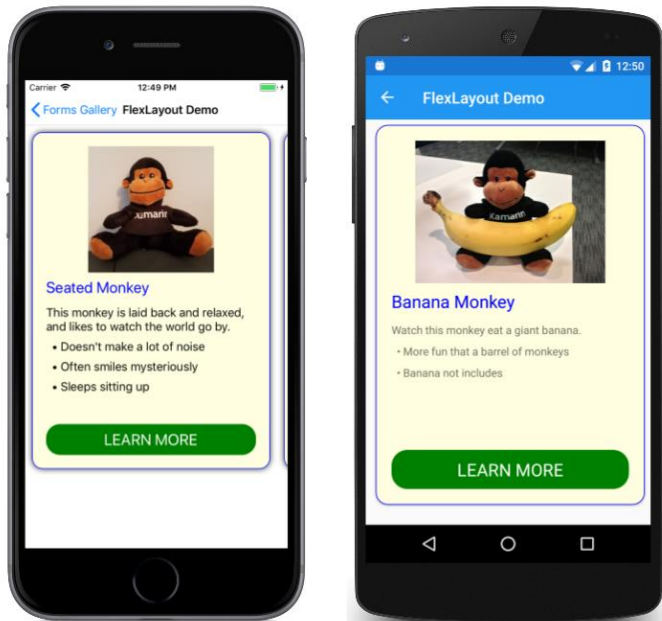
Pod sebe nebo vedle sebe



Layouts – více komponent

- FlexLayout

Obdoba webového Flexboxu



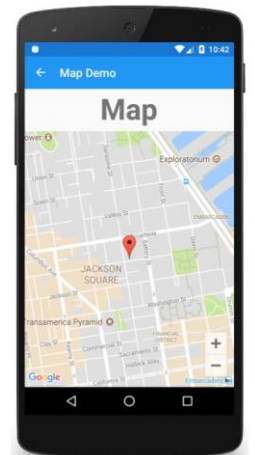
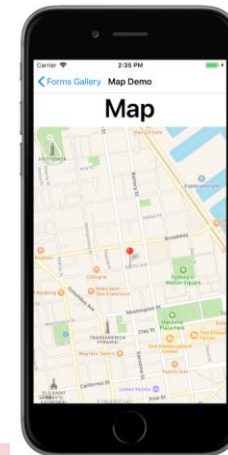
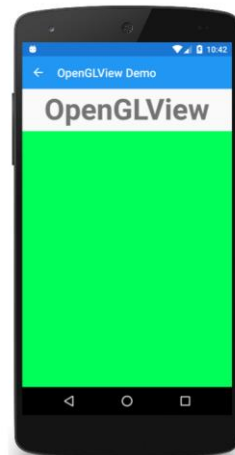
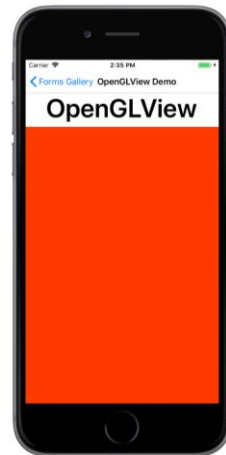
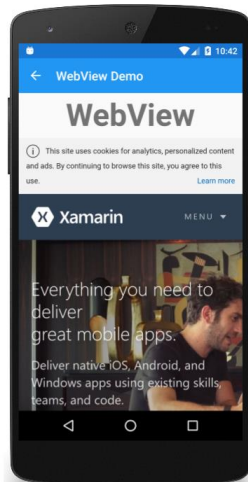
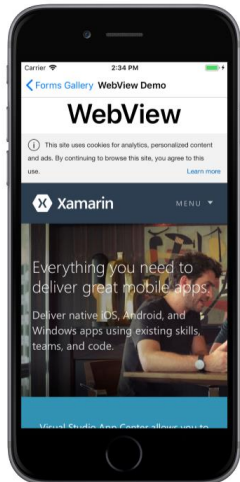
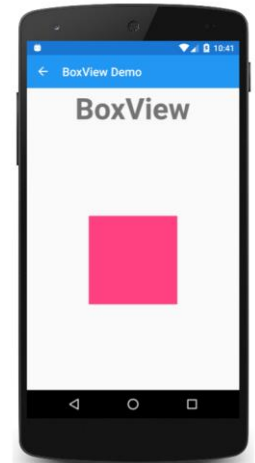
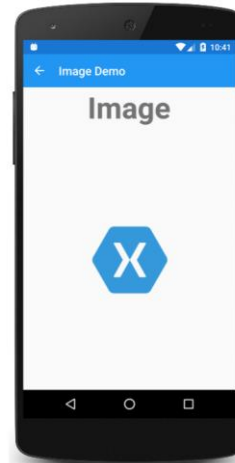
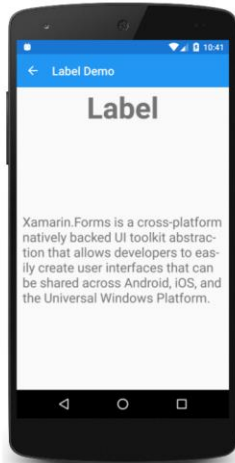
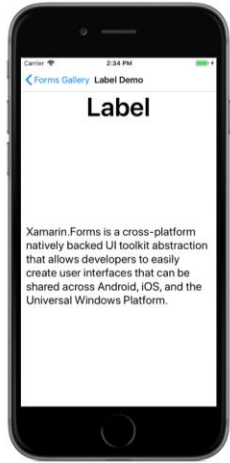
Grid

- Skoro stejné jako ve WPF
- RowDefinitions, ColumnDefinitions
 - Width / Height = 150 | 3* | Auto
- Grid.Row, Grid.Column, Grid.RowSpan...
- ColumnSpacing, RowSpacing – je potřeba nastavit

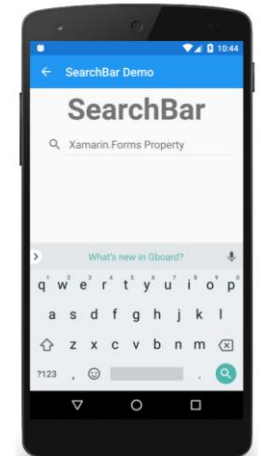
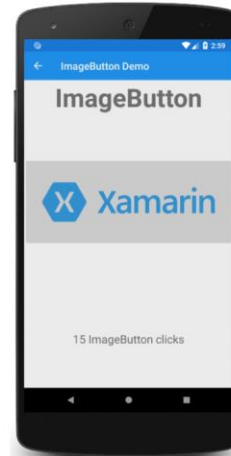
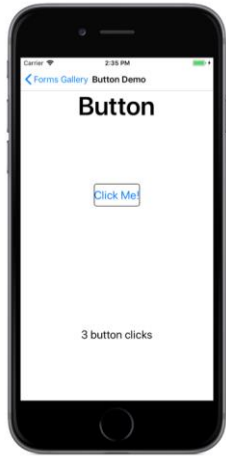
StackLayout

- Obdoba StackPanelu ve WPF
- Řadí komponenty pod sebe nebo vedle sebe
- Orientation = Horizontal | Vertical
- Spacing – rozestup jednotlivých komponent
- HorizontalOptions a VerticalOptions
 - Start, Center, End a Fill
 - ...AndExpand – komponenta může dostat víc místa, než nutně potřebuje, pokud je to místo k dispozici

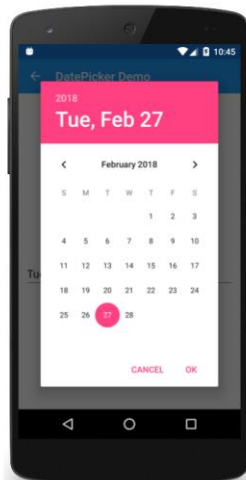
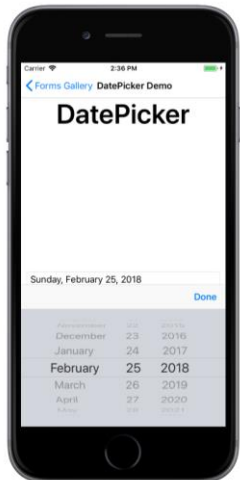
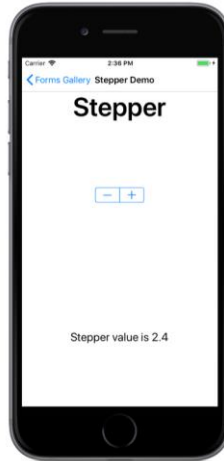
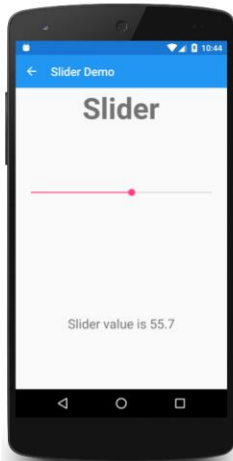
Prezentace obsahu



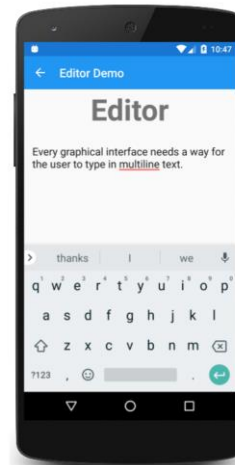
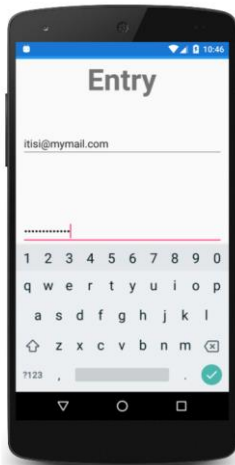
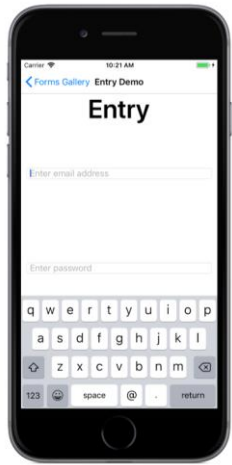
Akce



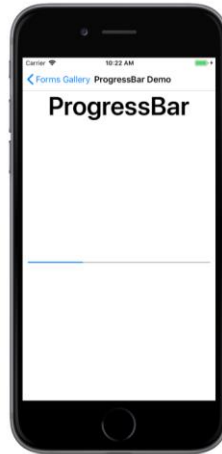
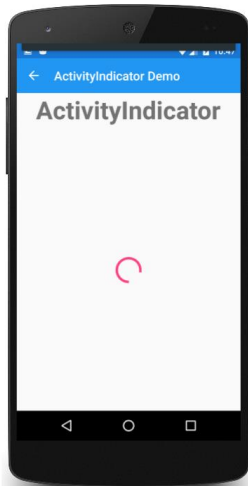
Nastavování hodnot



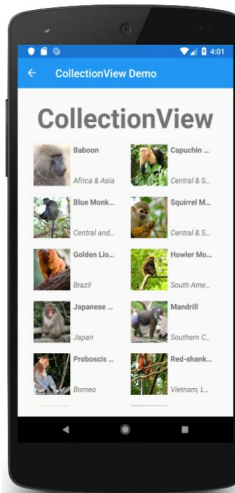
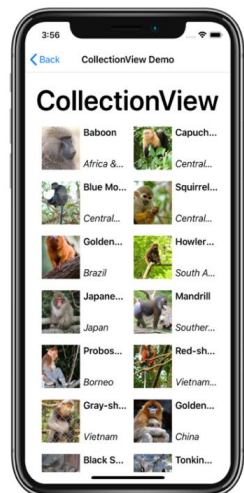
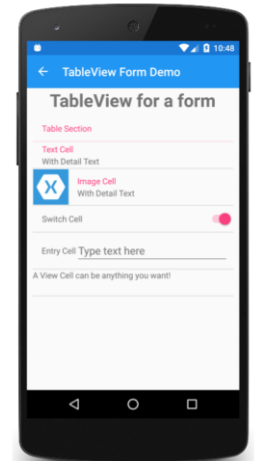
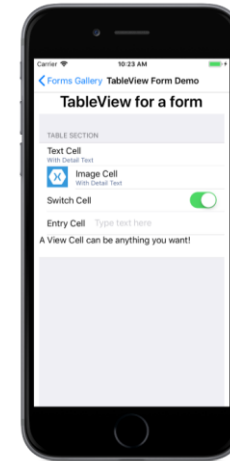
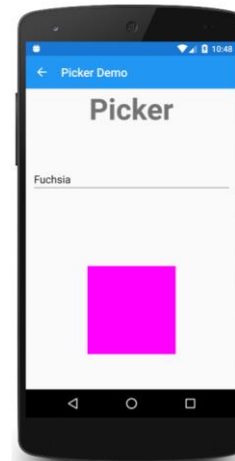
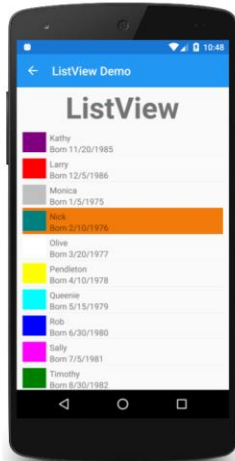
Editace textu



Indikace aktivity



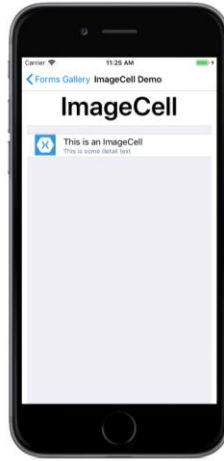
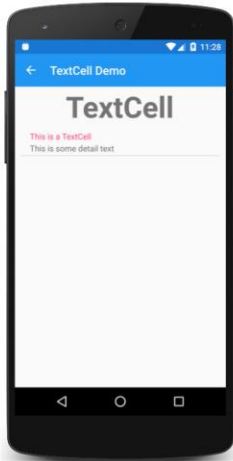
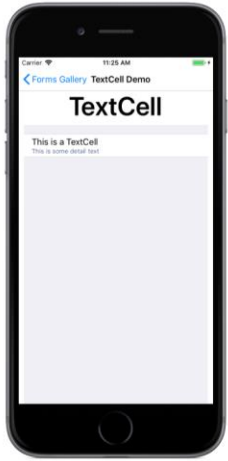
Kolekce



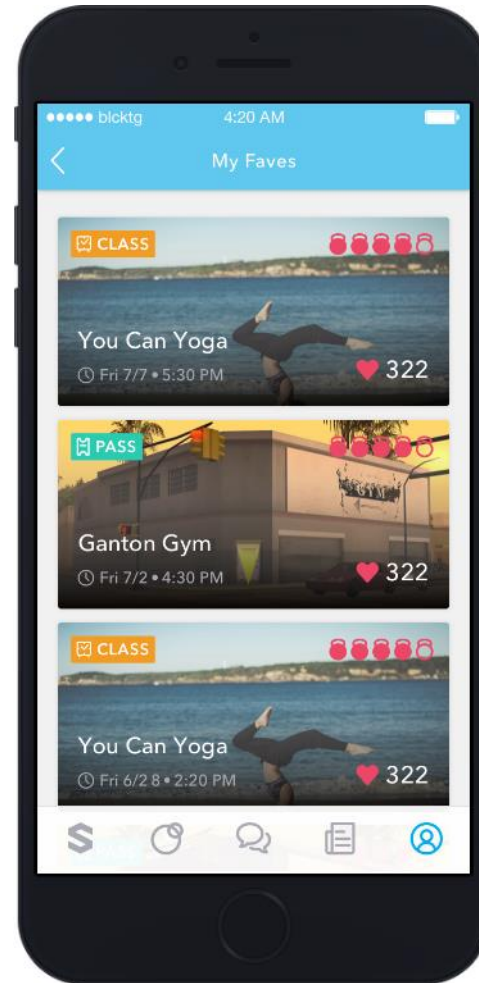
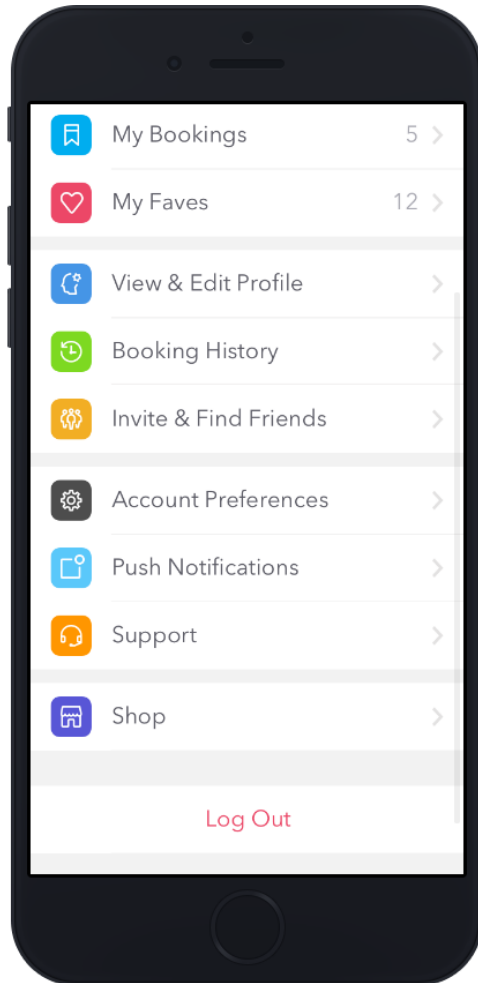
Kolekce - Cells

- TextCell, ImageCell, SwitchCell, EntryCell...
- ViewCell (dovnitř lze dát cokoliv)
 - ViewCell.ContextActions ... kontextové menu
- ListView
 - ItemsSource="{Binding: ...}"
 - IsPullToRefreshEnabled = true | false
 - zavolá RefreshCommand
 - HasUnevenRows = true | false
 - různé výšky řádků

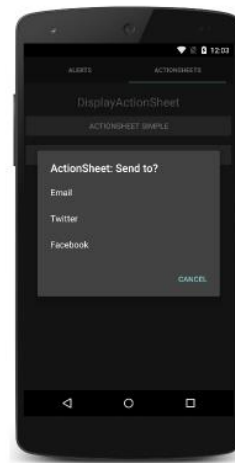
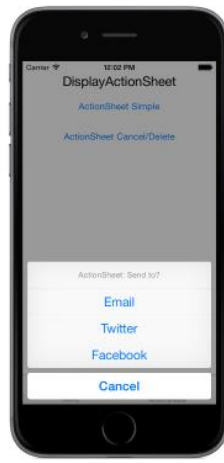
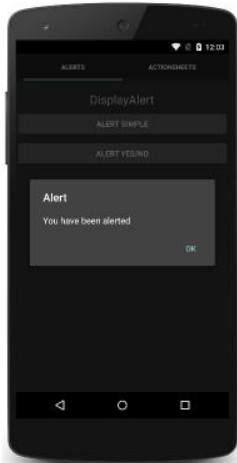
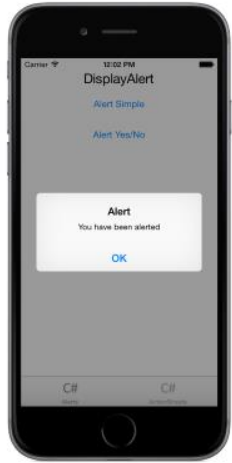
Cells



ViewCell



Pop-ups



Komponenty třetích stran



Rozdíly – názvy

WPF	Xamarin Forms
StackPanel	StackLayout
TextBox	Entry
ListBox	ListView
CheckBox	Switch
ProgressBar	ActivityIndicator
Grid	Grid
Label	Label
Button	Button
Image	Image
Date/TimePicker	Date/TimePicker

Rozdíly – Data Binding

Windows	Xamarin.Forms
DataContext	BindingContext
{Binding Property}	{Binding Property}
ItemsSource	ItemsSource
ItemTemplate	ItemTemplate
DataTemplate	DataTemplate

```
<Label Text="{Binding Color.R,  
        Converter={StaticResource intConverter},  
        ConverterParameter=255,  
        StringFormat='R={0:X2}'}" />
```

Platformně specifické

- Možnost změnit hodnotu parametru pro konkrétní platformu přímo v XAMLu

```
<Button.BorderWidth>  
  <OnPlatform x:TypeArguments="system:Double">  
    <On Platform="Android" Value="5"/>  
    <On Platform="iOS, UWP" Value="10"/>  
  </OnPlatform>  
</Button.BorderWidth>
```

Platformně specifické

- Custom Renderer
- Když něco nejde v XAML
- Ve Forms projektu vyrobit třídu
`public class BoxViewWithRadius : BoxView`
- V Android/iOS projektu vyrobit Renderer
`[assembly: ExportRenderer(typeof(BoxViewWithRadius),
typeof(BoxViewWithRadiusRenderer))]`
`public class BoxViewWithRadiusRenderer : BoxRenderer`
- V XAMLu použití třídy z Forms projektu
- Implementace pro každou platformu zvlášť

Platformně specifické

- Podpora pro iOS a Android
 - 100% podpora pro API při releasu
- Zápis velmi podobný Java/Swift, stejné názvy tříd atd.
- Z existujícího kódu na jinou platformu je možné jednoduše přepsat do Xamarinu
- **Xamarin.Essentials** – multiplatformní přístup k těmto API

Data binding

- Mezi code-behind a XAML
- Mezi ViewModelem a View
- `Text="{Binding Path=Operand1}"`
- Mezi komponentami ve View
- `Text="{Binding Source={x:Reference Name=DisplayAlertButton}, Path=Text}"`

Směr bindingu

- Vlastnost **Mode**
 - **OneTime**
 - jen jednou na začátku
 - **OneWay**
 - jedním směrem – ze zdroje do cíle
 - **TwoWay**
 - obousměrně – změna v cíli změní i zdroj
 - **OneWayToSource**
 - jedním směrem – od cíle k zdroji
- Zdroj – vlastnost, na kterou bindujeme
- Cíl – komponenta, která má {Binding}

Binding - Notifikace o změnách

- OneWay a TwoWay
 - reaguje na změny zdroje
 - zdroj o nich musí dát nějak vědět
 - objekt implementuje INotifyPropertyChanged
 - `public event PropertyChangedEventHandler PropertyChanged`
- Je vhodné přidat metodu `OnPropertyChanged` s parametrem `s` a atributem `[CallerMemberName]`

PropertyChanged.Fody

- Automatické doplnění volání PropertyChanged eventu
- Attributes:
 - AlsoNotifyFor
 - DependsOn

INotifyCollectionChanged

- `ObservableCollection<T>`
 - implementuje toto rozhraní
- Vlastní kolekce
 - podporu přidáme pomocí `INotifyCollectionChanged`
- Existující kolekce
 - např. napsat kolem ní "wrapper"

Data Binding Intellisense

`xmlns:d=http://schemas.microsoft.com/expression/blend/2008`

`xmlns:mc=http://schemas.openxmlformats.org/markup-compatibility/2006`

`mc:Ignorable="d"`

`d:DataContext="{d:DesignInstance test:MainPageViewModel}"`

Může být problematické

- Možnost řešení pomocí Locator-u (ViewModelLocator)
- Využití BindingContext v XAMLu

MVVM Frameworky

- MVVM Cross
- Simple MVVM
- MVVM Light
- Catel
- ReactiveUI

Tip - načtení dat z webové služby

- Využití frameworku Swagger
- Vytvoření Web API v .Net
- Vygenerování klientského kódu
- Přiložení kódu do projektu jako dll

Xamarin Forms 4 – Novinky

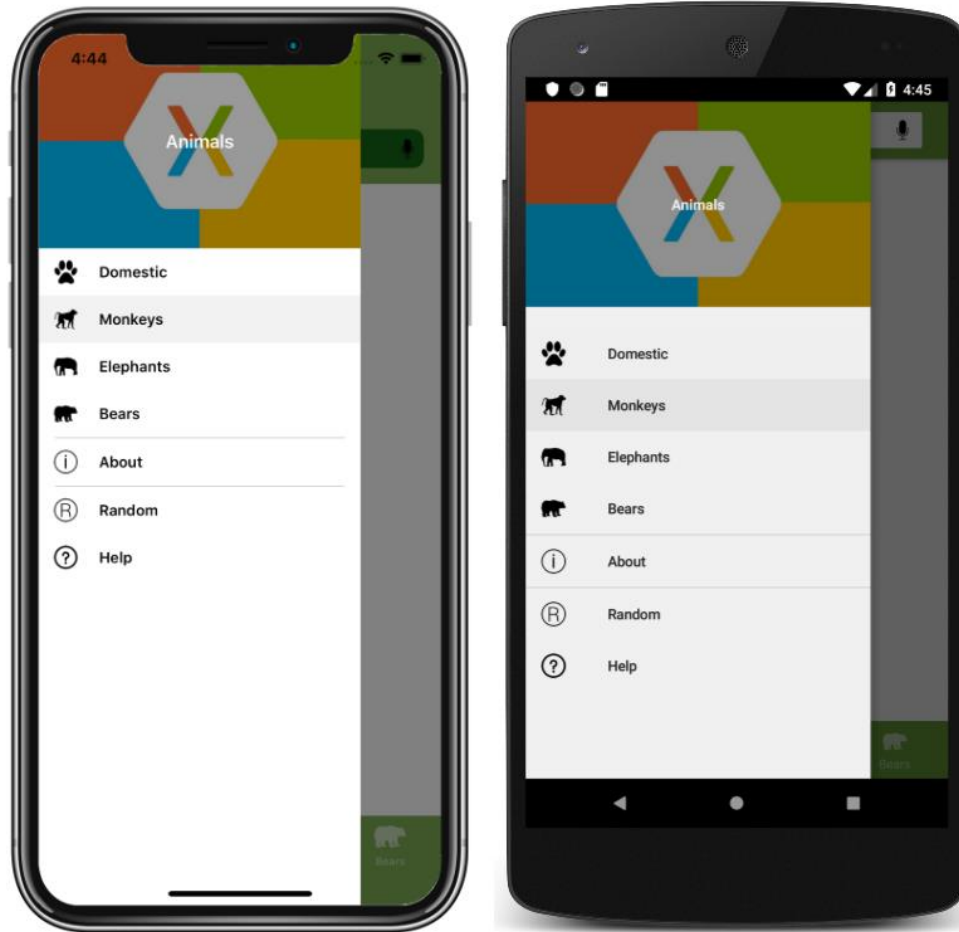
Shell

- Nový způsob definice organizace stránek
- Nový přístup k navigaci pomocí URI
- Jedno místo na definici struktury aplikace
- Integrované vyhledávání

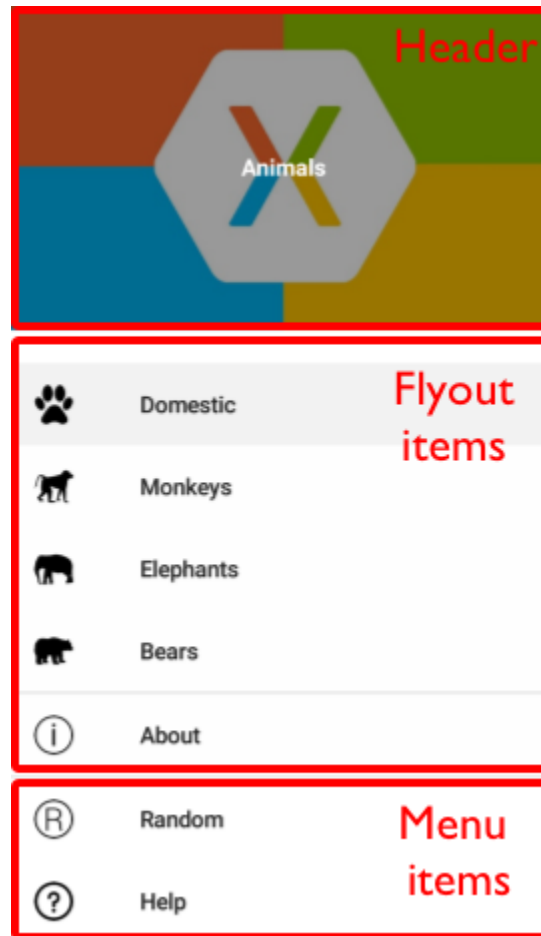
Shell – Flyout

- Náhrada za MasterDetail page
- Vyjížděcí menu vlevo
- Možnost nastavit hlavičku
- Možnost přepsat vykreslování jednotlivých položek

Shell - Flyout



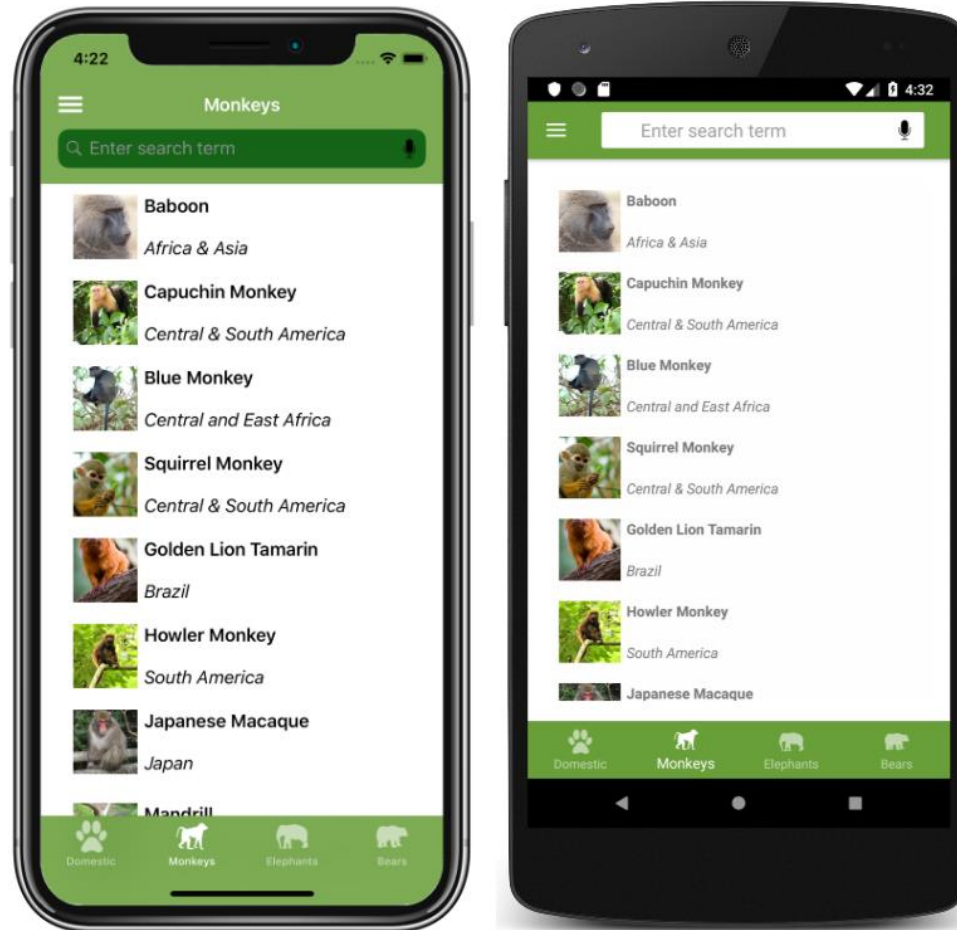
Shell - Flyout



Shell - Tabs

- Náhrada za Tabbed page
- Taby jsou ve spodní části
- Podpora pro text a ikonu

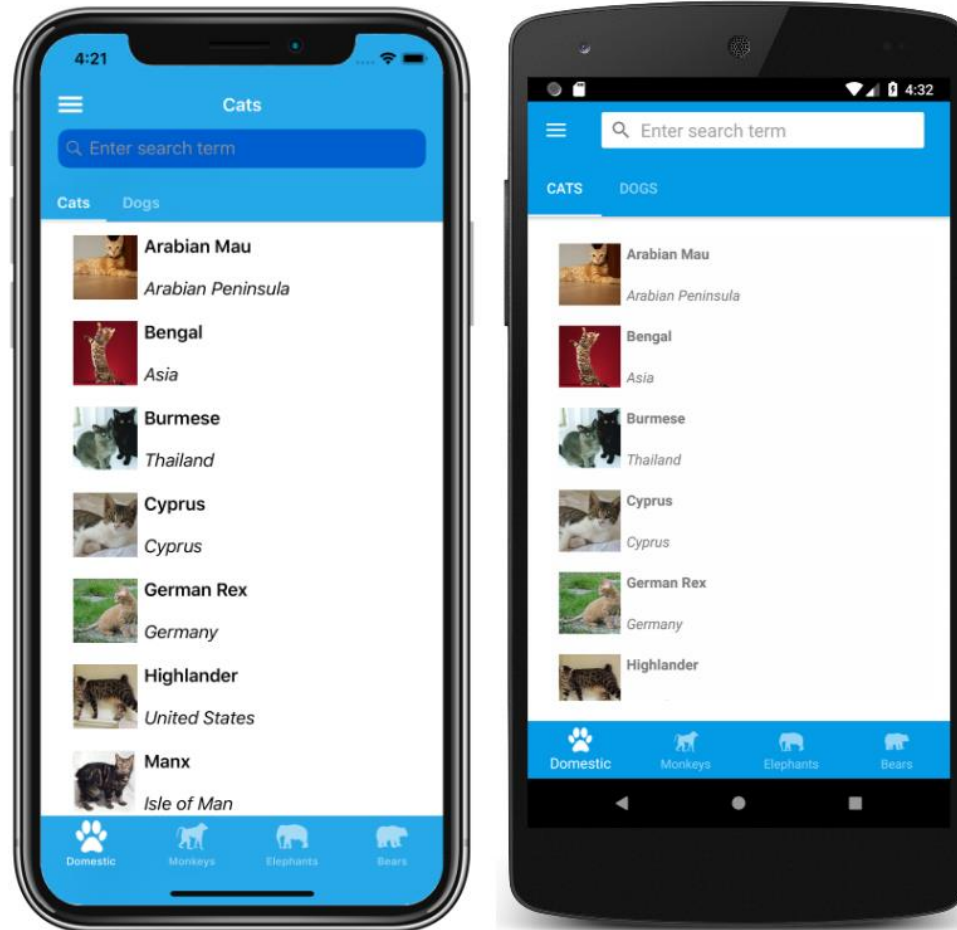
Shell - Tabs



Shell – Více obsahů

- Víc stránek v jednom tabu
- Taby v horní části stránky
- Text, možnost upravit barvu

Shell – Víc obsahů



CollectionView

- Nový způsob zobrazování kolekcí
- Flexibilnější a optimalizovanější než ListView
- Vertikální i horizontální orientace
- Není potřeba Cell
- Podpora pro Grid zobrazení
- Podpora pro prázdné zobrazení
- Je potřeba zapnout experimentální flag

```
global::Xamarin.Forms.Forms.SetFlags("CollectionView_Experimental");
```

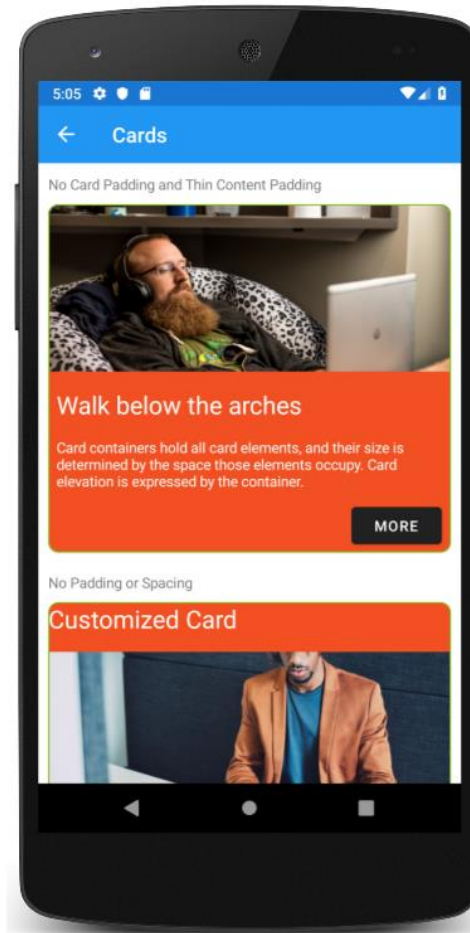
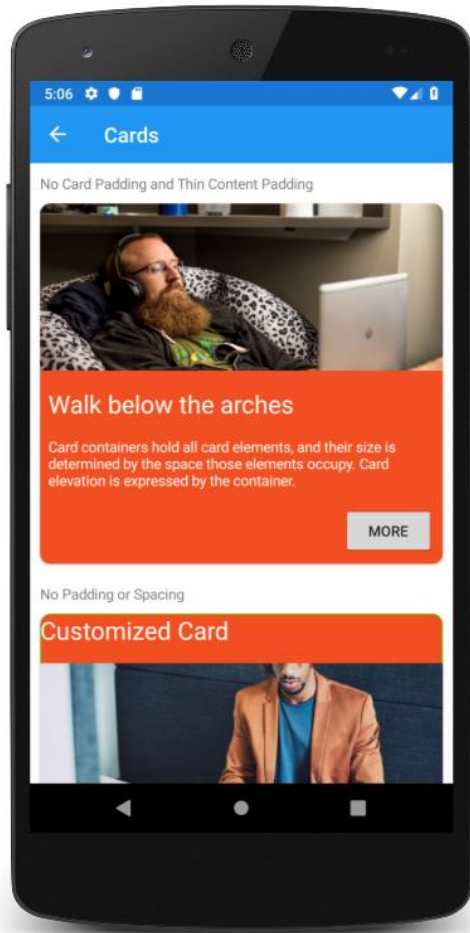
Visual

- „Opinionated design system“
- Identicky vypadající UI pro Android a iOS
- Možnost jednoduchého nastavení defaultního vzhledu
- Možnosti
 - Material design
 - Chystají se: Fluent, Fabric, Cupertino
 - Vlastní Visual

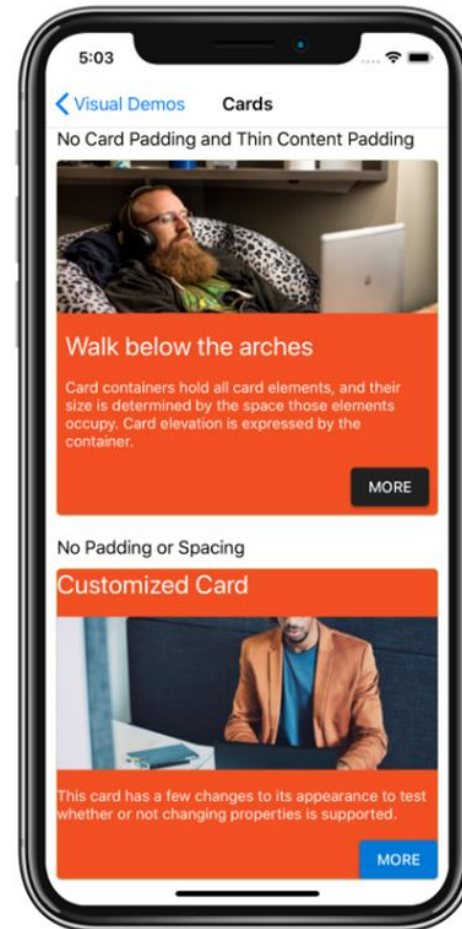
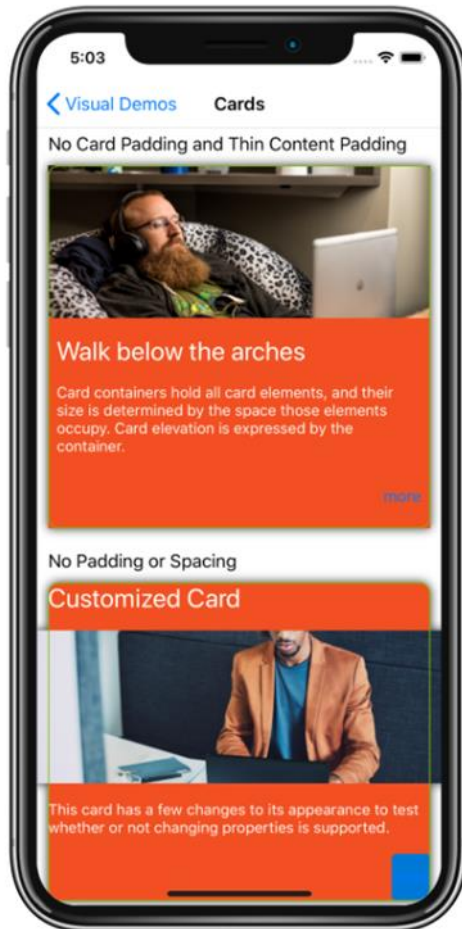
Xamarin Forms 4 – Novinky

- Menší změny
 - ImageButton
 - Maximální počet řádek v Label
 - Fast Renderers
 - ...

Visual – příklad Android



Visual – příklad iOS



Visual

- Nuget balíček Xamarin.Forms.Visual.Material
- Inicializace na úrovni platformy
FormsMaterial.Init();

- Nastavení v XAML:

```
<ContentPage Visual="Material">
```

```
...
```

```
</ContentPage>
```

- Nastavení je hierarchické

Tipy

XAML Compilation

XAML

XAMLC



Compile-time

Parsed & turned into
IL

Parsed and inflated

Runtime



XAML Compilation

- Rychlejší načítání
- Menší velikost aplikací
- Chyby v XAML již při kompilaci

Font Awesome

- Přidat font do platformě specifického projektu
- Přidat do ResourceDictionary do App.xaml
- Možnost použít v textu
- Možnost použít jako obrázek pomocí FontImageSource
- Pro Tip – vygenerovat třídu s názvy pro jednodušší použití:
<https://andreinitescu.github.io/IconFont2Code/>

Xamarin Essentials

- Hromada užitečných napojení
- Přímo od tvůrců Xamarinu
- Práce se soubory, SMS, gyroskopem, akcelerometrem...

Xamarin + Swagger

- Klientský kód pro komunikaci s API
- Vygenerování z OpenAPI definice
- Není nutno řešit infrastrukturní kód pro komunikaci
- Pro C# generátor AutoRest

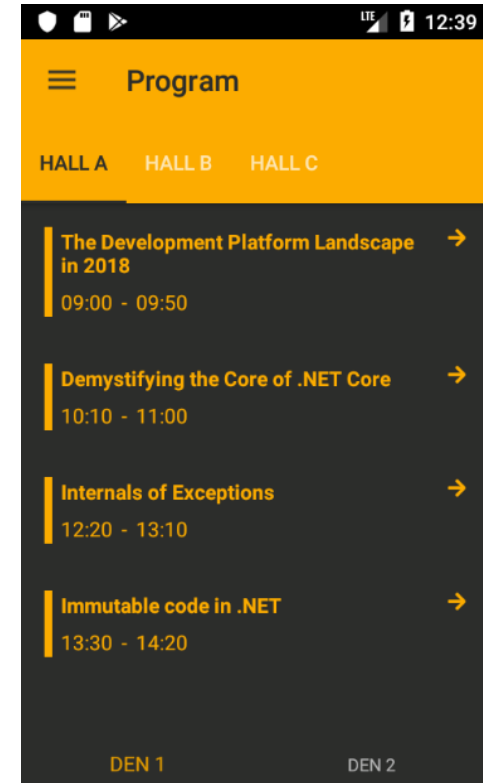
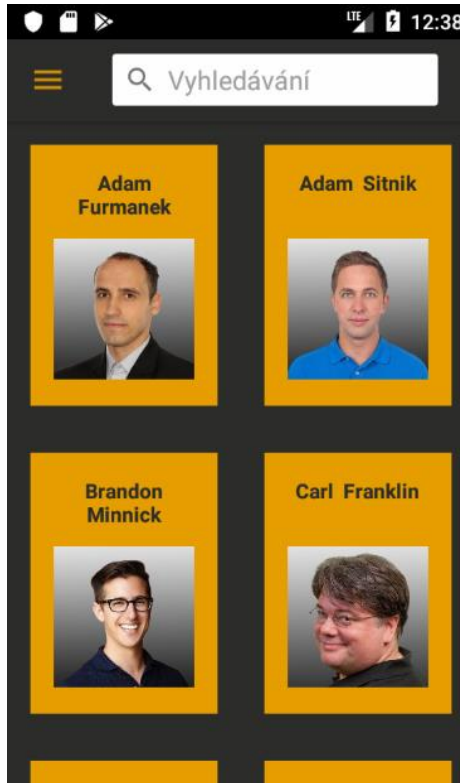
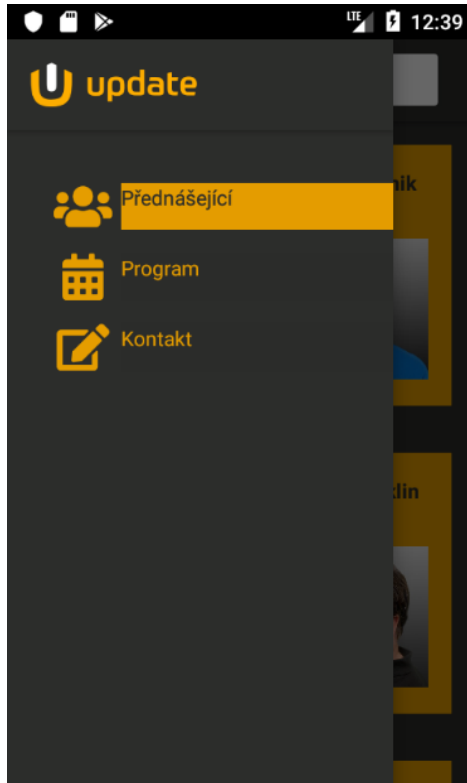
XAML Hot Reload

- Možnost vidět změny v kódu přímo v běžící aplikaci
- Není potřeba žádné knihovny, stačí spustit v debug módu
- Funguje pro úpravy v XAML kódu
- Projeví se při uložení souboru
- Funguje v emulátoru, simulátoru i zařízení
- Aktuálně v Preview – je možné se přihlásit

Výhody Xamarin Forms

- XAML
- .Net
- Dobrá dokumentace
- Sdílení kódu – 85%+

Demo



<https://github.com/jasho/xconferenceapp>

Pozvánka – SQL Bootcamp 2019

- 20. 8. – 21. 8.
- Brno
- 2 tracky
- Cca 24 přednášek
- Již se dá registrovat
- Hlasování o přednáškách již brzy

Pozvánka - WUG Days 2019

- Víkend 14. 9. – 15. 9.
- Brno
- 4 tracky
- Cca 48 přednášek
- Již se dá registrovat
- Hlasování o přednáškách již brzy