

Sem vložte zadání Vaší práce.



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

## **Návrh platformy pro rychlý vývoj multiplatformních mobilních aplikací**

*Jan Skála*

Vedoucí práce: Ing. Jiří Hunka

10. května 2015



---

## Poděkování

Děkuji Ing. Jiřímu Hunkovi a Ing. Tomáši Kolínovi za odborné vedení, ochotu a poskytnutí cenných rad při tvorbě bakalářské práce.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 10. května 2015

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2015 Jan Skála. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Skála, Jan. *Návrh platformy pro rychlý vývoj multiplatformních mobilních aplikací*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.



---

# Abstrakt

Práce se zabývá možnostmi současných multiplatformních mobilních aplikačních frameworků. Poskytuje souhrn vlastností existujících frameworků a několik z nich podrobně srovnává.

Srovnání je založeno na implementaci jednoduché aplikace v každém frameworku. Součástí srovnání jsou obecné vlastnosti jako podporované mobilní platformy, obsažené komponenty a použité technologie, ale také vzhled a rychlost výsledné aplikace a náročnost vývoje z pohledu webového vývojáře. Výsledky srovnání poskytují informace nutné pro výběr frameworku vhodného pro vývoj konkrétní aplikace.

Na srovnání navazuje tvorba platformy, která má za cíl usnadnit počáteční fázi vývoje mobilní aplikace. Platforma umožňuje výběr komponent a nástrojů potřebných pro vývoj, stará se o jejich stažení z internetu a integraci do projektu. Platforma obsahuje uživatelské rozhraní s ukázkami jednotlivých komponent pro usnadnění jejich výběru. Pro demonstraci použití této platformy je vytvořena vzorová mobilní aplikace.

**Klíčová slova** mobilní aplikace, multiplatformní mobilní vývoj, tvorba platformy, hybridní frameworky

# Abstract

The thesis summarizes the abilities of today's multiplatform mobile application frameworks.

It provides information about large amount of these frameworks and compares several of them in many categories. The thesis helps with the selection of the suitable framework for specific application. The conclusions are based on personal experiences from the implementation of sample application in each framework.

The next step is the creation of development platform. The main goal of the platform is helping developer with the early phase of the mobile application development. The platform allows developer to choose, download and initialize the required components and development tools. The platform contains the graphical user interface with the examples of each component to help developer with the choice.

**Keywords** mobile applications, multiplatform mobile development, platform creation, hybrid frameworks

---

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Přiblížení problematiky</b>	<b>5</b>
2.1 Mobilní aplikace . . . . .	5
2.2 Aplikační framework . . . . .	7
2.3 Platforma . . . . .	8
2.4 Webové technologie . . . . .	9
2.5 Apache Cordova . . . . .	11
<b>3 Výběr a srovnání frameworků</b>	<b>13</b>
3.1 Existující frameworky . . . . .	13
3.2 Výběr a popis vybraných frameworků . . . . .	16
3.3 Specifikace kritérií a srovnávací metodika . . . . .	27
3.4 Testovací aplikace . . . . .	30
3.5 Srovnávací tabulka . . . . .	32
3.6 Zhodnocení výsledků . . . . .	35
<b>4 Tvorba platformy</b>	<b>41</b>
4.1 Cíle platformy . . . . .	41
4.2 Technologie . . . . .	41
4.3 Architektura . . . . .	42
4.4 Implementace . . . . .	42
4.5 Dokumentace . . . . .	44
4.6 Požadavky a instalace . . . . .	45
4.7 Použití . . . . .	45
4.8 Rozšíření a modifikace . . . . .	47
4.9 Vzorová aplikace . . . . .	48
4.10 Postup při tvorbě mobilní aplikace . . . . .	48

<b>Závěr</b>	<b>51</b>
<b>Literatura</b>	<b>53</b>
<b>A Seznam použitých zkratk</b>	<b>65</b>
<b>B Obsah přiloženého CD</b>	<b>67</b>

---

## Seznam obrázků

2.1	Princip obchodu s aplikacemi . . . . .	5
2.2	Podíl mobilních platforem na trhu . . . . .	9
2.3	Princip HTML elementu . . . . .	10
2.4	Počáteční element s atributem . . . . .	10
2.5	Architektura aplikace na platformě Cordova . . . . .	12
3.1	Struktura aplikace ve frameworku Titanium . . . . .	19
3.2	Struktura s ohledem na platformu ve frameworku Titanium . . . . .	19
3.3	Vzhled aplikací v různých frameworkcích. . . . .	31
3.4	Graf s celkovými výsledky frameworků ve srovnání . . . . .	36
3.5	Graf s jednotlivými kategoriemi . . . . .	37



---

# Úvod

Aplikace pro mobilní telefony a tablety v posledních několika letech prodělávají výrazný rozmach. Pro největší platformy co do počtu prodaných zařízení, tedy Android, iOS a Windows Phone, jsou nyní ke stažení celkem více než dva miliony aplikací a další stále přibývají [1]. Hlavním problémem s vývojem a udržováním aplikací pro více platforem současně je jejich roztržitost. Každá zmíněná platforma pro vývoj využívá jiný programovací jazyk a mimo Androidu vyžaduje i specifický operační systém.

Řešením tohoto problému jsou multiplatformní frameworky, které umožňují ucelený vývoj pro více platforem současně. Tyto frameworky jsou poměrně nové, velmi rychle se rozvíjejí, je jich velké množství a není jednoduché si mezi nimi vybrat.





## Cíl práce

Cíle mé bakalářské práce jsou rozděleny do tří vzájemně navazujících částí.

Základním cílem je poskytnout přehled o možnostech multiplatformního vývoje mobilních aplikací. Součástí přehledu je seznam existujících frameworků a jejich obecné vlastnosti.

Navazujícím cílem je doporučit vhodný framework pro vývoj různých typů aplikací. K tomu slouží podrobné srovnání několika vybraných frameworků.

Konečným cílem je vytvoření platformy, která usnadní vytvoření projektu s integrací požadovaných komponent a nástrojů pro vývoj mobilní aplikace. Jednou z obsažených komponent bude framework, který na základě srovnání nejlépe odpovídá požadavkům zadavatele.



## Přiblížení problematiky

### 2.1 Mobilní aplikace

Nejprve je třeba definovat pojem mobilní aplikace. Mobilní aplikace je aplikace určená pro přenosná zařízení jako jsou mobilní telefony a tablety [2].

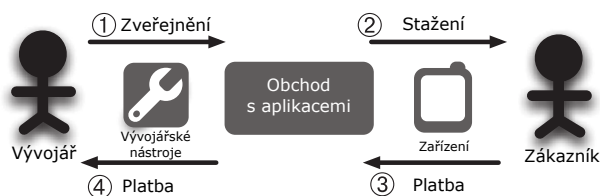
#### 2.1.1 Distribuce mobilních aplikací

Pro distribuci mobilních aplikací se používají specializované elektronické obchody. Pro Android je to nejčastěji obchod Google Play, pro iOS Apple AppStore a pro platformu Windows Phone obchod Windows Store.

Princip distribuce aplikací skrze tyto obchody je jednoduchý. Vývojář svoji vytvořenou aplikaci zveřejní v konkrétním obchodu, kde ji uvidí uživatel. Uživatel za aplikaci zaplatí, stáhne si ji do zařízení a může ji používat. Peníze, které uživatel zaplatil, přejdou k provozovateli obchodu, který si část z nich nechá a zbytek vyplatí vývojáři [3].

Grafické znázornění lze vidět na obrázku 2.1.

23



Obrázek 2.1: Princip obchodu s aplikacemi [3, str. 2]

### 2.1.2 Typy mobilních aplikací

Aplikace můžeme rozdělit do několika kategorií, které se vzájemně liší jak z vývojového, tak z uživatelského hlediska. Srovnání různých typů aplikací publikoval Jim Cowart [4]. Přeložené shrnutí této práce je obsahem dalších odstavců.

- Nativní aplikace

Aplikace jsou vyvíjeny a optimalizovány zvlášť pro každou platformu. V případě iOS je nutno využít programovací jazyk Objective C a vyvíjet na operačním systému Mac OS. Aplikace na Windows Phone se programují v jazyce C# nebo XAML pod operačním systémem Windows 8 nebo novějším. Android pro změnu používá technologii Java. Znaky tohoto typu vývoje jsou následující:

- Nevýhodou je nutnost mít kvalitní vývojáře pro každou platformu, déle trvající vývoj a horší údržba napříč platformami. Důsledkem je vyšší cena za vývoj nativních multiplatformních aplikací.
- Výhodou je naopak vysoký výkon aplikací na rozdíl od ostatních přístupů a vzhled, ovládací prvky a chování přizpůsobené zbytku systému.

- Webové aplikace

Aplikace vyvíjená ve webových technologiích a umístěná na webový server. K aplikaci se poté přistupuje zadáním adresy přes webový prohlížeč. Tato aplikace může být optimalizovaná pro mobilní zařízení, může mít vzhled i ovládací prvky přizpůsobené menší velikosti displeje a dotykovému ovládání.

- Nevýhodou je rychlost odezvy, která je závislá na internetovém připojení a nemožnost používat aplikace v offline režimu. Dalším omezením je nemožnost využívat hardwarové prostředky zařízení a také není možné distribuovat aplikaci přes obchody s aplikacemi, jako je Apple App Store a Google Play.
- Výhodou je naopak ucelený vývoj, jednoduchá údržba a přístupnost z jakéhokoliv zařízení a jakékoliv platformy.

- Hybridní aplikace

Aplikace psaná ve webových technologiích JavaScript, HTML a CSS, ale zabalená spolu s jednoduchým webovým prohlížečem do balíčků, které se dají nainstalovat na jednotlivých zařízeních na odpovídajících platformách.

- Nevýhodou je nižší rychlost aplikace oproti nativnímu přístupu, která je navíc ovlivněna různými implementacemi tzv. WebView komponenty, jež slouží pro interpretaci aplikace na zařízeních.
  - Výhodou je možnost distribuovat aplikace přes obchody a rychlý a jednotný vývoj a údržba pro všechny platformy.
- Kompilované multiplatformní aplikace

Aplikace psané v jednom společném prostředí i jazyku se v závislosti na cílové platformě zkompilují na nativní aplikaci.

    - Výhodou je rychlost výsledné aplikace a přístup k nativním funkcím zařízení.
    - Nevýhodou potom to, že se aplikace většinou nechovají na všech platformách stejně, musí se psát část kódu pro každou platformu zvlášť a vývojář se musí naučit pracovat s API v daném prostředí, které poskytuje abstrakci nad cílovou platformou.

## 2.2 Aplikační framework

Český výraz rámeček se pro aplikační framework příliš nepoužívá, proto jsem se rozhodl použít přejaté slovo framework s českým skloňováním.

Nejobecněji lze framework definovat jako nástroj, který slouží pro urychlení a zkvalitnění vývoje aplikace [5].

Framework může řešit několik různých problémů, které se vyskytují při vývoji softwaru [6]:

- Zjednodušuje práci s komplexními technologiemi.
- Seskupuje k sobě více jednotlivých komponent, které jsou užitečnější pohromadě.
- Nutí vývojáře k dodržování konvencí a jednotné struktury.
- Umožňuje snáze testovat a opravovat kód.

Framework může být [6]:

- Obal pro API a jiné objekty a sjednocení jejich používání. Důsledkem je zjednodušení práce, konzistence rozhraní a rozšíření funkcionality.
- Architektura. Určuje vztahy a závislosti mezi komponenty a strukturu aplikace.
- Metodologie. Určuje způsob interakce mezi komponenty.

Aplikační frameworky mají více podob, mohou se zaměřovat na všechny uvedené i jiné problémy, na jejich podmnožinu, nebo pouze na jeden specifický problém (například na vhodné uživatelské rozhraní). Sám programátor má možnost si vybrat, co z frameworku použije a může si určité části upravit [6]. Je dokonce možné používat najednou více frameworků, pokud jsou kompatibilní.

### 2.3 Platforma

Na pojem platforma je z hlediska počítačové vědy několik pohledů.

#### 2.3.1 Běhová platforma

Běhová platforma je soubor technologií umožňující běh jiných technologií a programů. Běhová platforma může označovat [7]:

- typ procesoru a/nebo ostatního hardwaru, na kterém běží operační systém a ostatní programy,
- typ operačního systému a jiného softwaru, který umožňuje běh jiných aplikací,
- kombinace typu hardwaru a operačního systému, který na něm běží.

#### 2.3.2 Vývojová platforma

Pojem platforma může být použit i v kontextu vývoje aplikací. Obecně je vývojová platforma skupina technologií, která umožňuje vývoj aplikací, procesů nebo dalších technologií [8].

##### 2.3.2.1 Rozdíl mezi frameworkem a vývojovou platformou

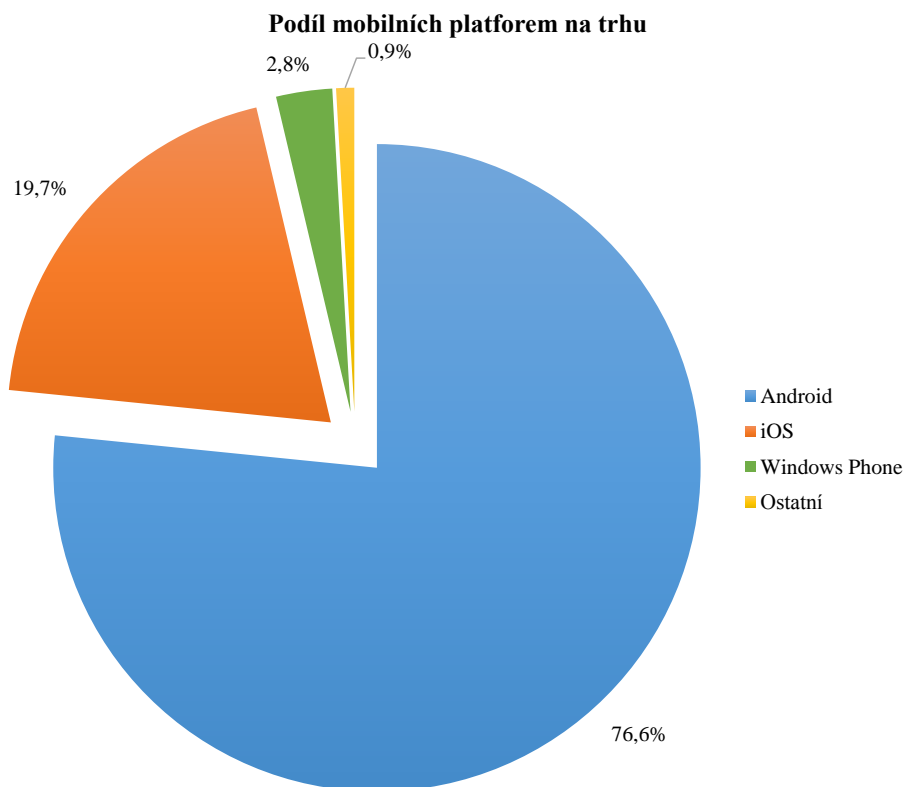
Zajímavý je rozdíl mezi frameworkem a vývojovou platformou. Podle firem a projektů v oblasti vývoje softwaru je vývojová platforma balíček obsahující framework a další nástroje, například vývojové prostředí, analytický software, software pro návrh uživatelského rozhraní a další. Takto rozdělené pojmy má na svých webových stránkách například projekt Ionic, firma Sencha nebo Xamarin.

Není možné vždy určit jednoznačné rozdělení, kdy se ještě jedná o framework a kdy se už jedná o platformu [9].

#### 2.3.3 Mobilní platforma

Kombinace typu hardwaru a operačního systému nám určuje mobilní platformu. Podle průzkumu trhu od firmy IDC jsou třemi nejrozšířenějšími plat-

formami Android, iOS a Windows Phone [10]. Graf 2.2 ukazuje jejich podíl na trhu.



Obrázek 2.2: Podíl mobilních platforem na trhu [10]

## 2.4 Webové technologie

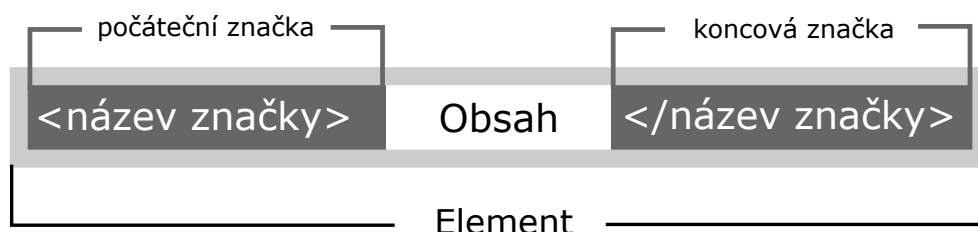
Kvůli dalšímu srovnání hybridních frameworků z pohledu vývojáře je potřeba přiblížit několik webových technologií. Účelem tohoto seznámení je poskytnout pouze základní přehled a definovat dále použité pojmy, nikoliv poskytnout kompletní informace o dané technologii.

### 2.4.1 HTML, CSS a JavaScript

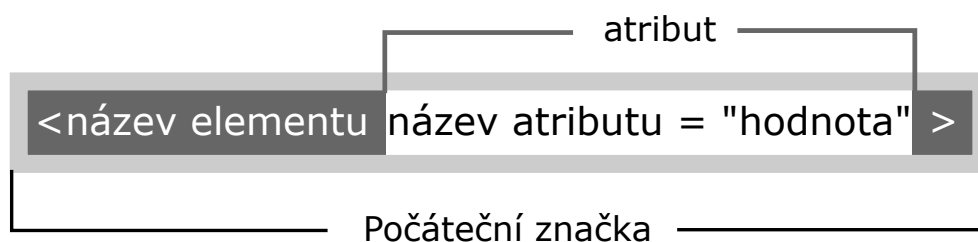
Technologie používané pro tvorbu webových stránek a aplikací. HTML je jazyk určený k popisu struktury webových stránek, zatímco CSS k definici jejich vzhledu [11]. JavaScript se stará o dynamické změny jejich vzhledu a obsahu [12], například na základě uživatelské interakce.

### 2.4.1.1 Základní princip a struktura

Hlavní část HTML dokumentu je tvořena strukturou HTML elementů [13]. Elementy jsou tvořeny počáteční a koncovou značkou, obsahem a případně různými atributy, jak je ukázáno na obrázcích 2.3 a 2.4.



Obrázek 2.3: Princip HTML elementu [14]



Obrázek 2.4: Počáteční element s atributem [14]

### 2.4.1.2 HTML5 a CSS3

Nejnovější specifikace HTML vyšla teprve v listopadu 2014 [15], nicméně moderní webové prohlížeče podporují řadu jeho vlastností už delší dobu [16].

Mezi nové klíčové prvky HTML5 pro tvorbu mobilních aplikací patří:

- možnost ukládání dat do lokálního úložiště,
- geolokace [17],
- vlastní elementy [18],
- vlastní datové atributy [19].

Stejně jako HTML5, i dosud nedokončená specifikace CSS3, poskytuje několik nových vlastností, které jsou významné pro vývoj mobilních aplikací ve webových technologiích.

Mezi tyto vlastnosti CSS3 patří podpora [20]:



- přechodů,
- transformací,
- animací.

### 2.4.2 AJAX

AJAX je zkratka pro asynchronní JavaScript a XML. Nejedná se o programovací jazyk, nýbrž o novou metodu použití jazyků stávajících [21]. Tato metoda dovoluje JavaScriptovému kódu webové aplikace komunikovat se serverem. To v konečném důsledku umožňuje odesílat data na server, přijímat data a ihned aktualizovat zobrazované informace bez potřeby načítat znovu celou stránku [22].

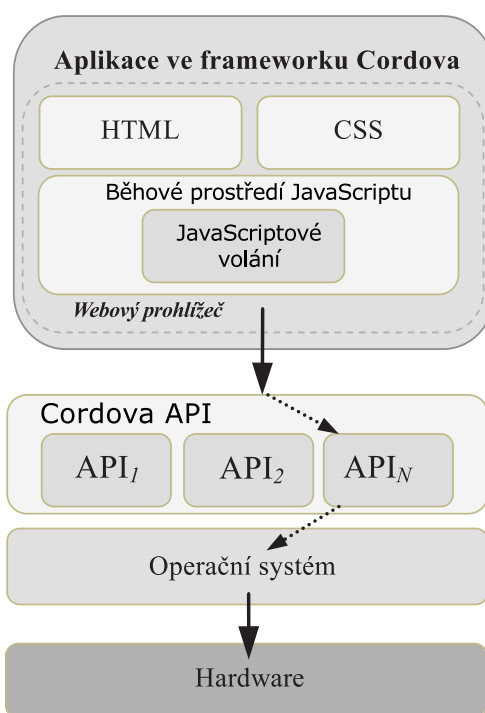
## 2.5 Apache Cordova

Apache Cordova je open source vývojová a běhová platforma, která umožňuje tvorbu a běh hybridních mobilních aplikací.

Skládá se z [23]:

- API, které umožňuje vývojáři přistupovat k nativním funkcím zařízení z JavaScriptového kódu. Toto API je konzistentní napříč podporovanými platformami.
- Nástrojů pro zabalení webových aplikací do specifických formátů pro každou platformu.
- Obal pro naši aplikaci - tzv. "WebView Wrapper". Jedná se o nativní aplikaci, která přes celou svou plochu zobrazí webovou stránku. K tomu využívá komponenty ("WebView"), které jsou implementovány v každé platformě.

Architekturu aplikace na platformě Cordova lze vidět na obrázku 2.5.



Obrázek 2.5: Architektura aplikace na platformě Cordova [24]

## Výběr a srovnání frameworků

Základním požadavkem na multiplatformní mobilní aplikaci je možnost její distribuce v obchodě s aplikacemi. Tento požadavek splňují pouze hybridní a multiplatformní kompilované aplikace z rozdělení výše. Na vývoj těchto aplikací jsou zaměřeny hybridní a jiné multiplatformní mobilní frameworky, kterým se bude věnovat další část práce.

### 3.1 Existující frameworky

Multiplatformních mobilních frameworků existuje obrovské množství (minimálně několik desítek). Tabulky 3.1 a 3.2 uvádějí některé z nich a stručně představují jejich klíčové znaky.

Tabulka 3.1: Hybridní frameworky

Framework	Podporované platformy	Licence	Poslední verze s datem vydání
Famo.us	iOS, Android, Kindle, Firefox OS	Open Source - Mozilla Public License 2.0	4.2. 2015 v0.3.5
Framework 7	iOS	Open Source - MIT License	22.2. 2014 v1.0.2
Chocolate Chip-UI	iOS, Android Windows Phone 8	Open Source - MIT License, Komerční licence s podpůrnými nástroji	6.1. 2015 v3.8.0

### 3. VÝBĚR A SROVNÁNÍ FRAMEWORKŮ

Tabulka 3.1: Hybridní frameworky

Framework	Podporované platformy	Licence	Poslední verze s datem vydání
ImpactJS	Všechny prohlížeče s podporou HTML5	Komerční	28.7. 2014 v1.24
Ionic	iOS 6+, Android 4.0+, Android 2.3 *, Firefox OS Windows Phone	Open Source - MIT License	15.12. 2014 v1.0.0-beta.14
Jo	iOS, Android, Windows Phone 8, BlackBerry 10, Tizen	Open Source	25.11. 2013 v0.5.0
jQuery Mobile	iOS 4+, Android, Windows Phone, BlackBerry	Open Source - MIT License	31.11. 2014 v1.4.5
Kendo UI	iOS 6+, Android 2.3+, BlackBerry 10+, Windows Phone 8.0+	Komerční licence, Open Source s omezeními - Apache v2.0 License	11.2. 2015 Q3 2014 SP2
Onsen UI	iOS 6+, Android 2.3.4+	Open Source - Apache v2.0 License	1.12. 2014 v1.2.1
OpenUI5	iOS 6+, Android 4.0+, Windows Phone 8.1*	Open Source Apache v2.0 License	16.2. 2015 v1.26.7
DevExtreme Mobile	iOS 5+, Android 4+, Windows Phone 8, Tizen	Komerční	18.2. 2015 v14.2.4
Ratchet	iOS 7.1+, Android, Windows Phone*	Open Source - MIT License	2.4. 2014 v2.0.2
Sencha Touch	iOS 6+, Android 2.3+, Windows Phone 8.1, BlackBerry 10, Tizen	Freeware Open Source - GPL v3	22.10. 2014 v2.4.1

Tabulka 3.1: Hybridní frameworky

Framework	Podporované platformy	Licence	Poslední verze s datem vydání
The M Project	iOS 6+, Android 4+	Open Source - MIT License	12.12. 2013 v2.0
Zepto.js	iOs 5+, Android 2.3+	Open Source - MIT License	12.12. 2014 v1.1.6

Data pocházejí z oficiálních stránek frameworků ([25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]).

Tabulka 3.2: Multiplatformní kompilované frameworky

Framework	Programovací jazyk	Podporované platformy	Licence	Poslední verze s datem vydání
Marmelade	C++/Lua/ Objective-C/HTML5	iOS 5.1+, Android 2.2+, BlackBerry 10, Windows Phone 8+	Freeware s omezeními, komerční plná verze	6.2. 2015 - v7.5.2
MoSync	C/C++ nebo HTML5, CSS a JavaScript	iOS 5.1 - 6.1, Android 2.2+, Windows Phone 7.5 - 7.8	Open Source-GPL, Freeware, komerční s podporou	30.5.2013 - v3.3.1

### 3. VÝBĚR A SROVNÁNÍ FRAMEWORKŮ

Tabulka 3.2: Multiplatformní kompilované frameworky

Framework	Programovací jazyk	Podporované platformy	Licence	Poslední verze s datem vydání
RhoMobile	JavaScript, HTML, CSS a Ruby	iOS 6+, Android 2.3+, Windows Phone 8	Open source* - MIT Licence, plná komerční	18.4. 2014 - v4.1.1
RubyMotion	Ruby	iOS 4.3+, Android 3.2+	Komerční	11.12.2014 - v3.0
Titanium	JavaScript, XML	iOS, Android, Windows Phone, BlackBerry	Freeware, komerční s podporou	12.1. 2015 - v3.5.0
Trigger.io	HTML, CSS, JavaScript	iOS, Android	Komerční	18.2. 2015 - v2.2.9
Xamarin	C#	iOS, Android, Windows Phone	Komerční	3.2. 2015 <sup>1</sup>

([40, 41, 42, 43, 44, 45, 46])

## 3.2 Výběr a popis vybraných frameworků

### 3.2.1 Obecné požadavky

Kvůli velkému množství frameworků musí být na základě obecných požadavků sestaven užší výběr frameworků, které se podrobně porovnají. Požadavky jsou zadány následovně.

- možnost umístit výslednou aplikaci do obchodů s aplikacemi,
- vývoj v programovacím jazyku C# nebo Javascript, HTML a CSS,

<sup>1</sup>Jednotlivé části (Xamarin.iOS, Xamarin.Android, Xamarin.Form, ...) se vydávají zvlášť, uvedeno poslední vydání.

- aktivní vývoj - poslední verze maximálně z listopadu 2014,
- plná podpora platforem Android a iOS.

Umístit výslednou aplikaci do jednotlivých obchodů lze se všemi zmíněnými frameworky. U hybridních frameworků se o zabalení aplikací do odpovídajícího formátu může postarat nástroj Cordova. U kompilovaných multiplatformních frameworků se o to postarají nástroje samotného frameworku. Všechny hybridní frameworky už z definice používají programovací jazyky HTML, CSS a JavaScript. U ostatních frameworků je v tomto ohledu větší variabilita.

### 3.2.2 Výběr frameworků pro srovnání

Dle tabulek 3.1 a 3.2 splňuje výše zmíněné podmínky několik frameworků.

Těchto frameworků je pro praktické srovnání stále příliš mnoho, proto bylo vybráno několik frameworků z této skupiny na základě dostupných srovnání. Jedním z nich je seriál článků s několika hybridními frameworky a jejich popisem od Grgura Grisogona [47] a dalším je stručné srovnání sedmi multiplatformních frameworků od Dragana Gaiće [48].

#### 3.2.2.1 Vybrané frameworky

- Appcelerator Titanium,
- Ionic,
- jQuery Mobile,
- Kendo UI,
- Onsen UI,
- Sencha Touch,
- Xamarin.

### 3.2.3 Popis vybraných frameworků

#### 3.2.3.1 Appcelerator Titanium

Open source aplikační framework Appcelerator Titanium obsahuje dvě části - Titanium API a framework Alloy.

Titanium API abstrahuje aplikační rozhraní cílové platformy a umožňuje psaní jednotného JavaScriptového kódu pro všechny podporované platformy. Přes toto API se přistupuje k funkcím zařízení i k uživatelskému rozhraní [49].

### 3. VÝBĚR A SROVNÁNÍ FRAMEWORKŮ

---

Rozšířením Titanium API je framework Alloy, který poskytuje rozdělení aplikace do MVC architektury, nabízí datové úložiště v offline režimu a umožňuje definování struktury uživatelského rozhraní v jazyku XML a jeho vzhledu v jazyku TSS [50]. Zápis Titanium Style Sheet (TSS) souboru je podobný formátu JSON a slouží jako náhrada CSS pro použití spolu s vlastní XML definicí [51].

Tento přístup na rozdíl od hybridního přístupu znemožňuje použití jiných JavaScriptových knihoven pracujících s uživatelským rozhraním.

Kombinace Titanium API, frameworku Alloy, freewareových Titanium SDK nástrojů a vývojového prostředí Titanium Studio a dalších, už komerčních nástrojů, dohromady tvoří platformu Appcelerator [52].

V Titaniu se dají vytvořit hybridní, webové a z části nativní aplikace. Jaká aplikace vznikne, můžeme rozhodnout až při finálním sestavení. Užitečná je možnost vytvořit z části nativní aplikaci. Nativní je sice jen uživatelské rozhraní, ale i to poskytuje výhodu v chování a vzhledu korespondujícímu se zbytkem systému jako u klasických nativních aplikací. Nativní uživatelské rozhraní lze použít pouze pro platformy iOS a Android, na pozadí stále běží JavaScriptový interpret a v něm aplikační kód [53].

Framework se snaží sjednotit vývoj pro všechny tyto typy aplikací, ale hlavně kvůli rozdílům v přístupu u iOS a Androidu je nutné psát alespoň z části rozdílný kód s ohledem na platformu [54].

Pokud to aplikace vyžaduje, umožňuje framework psát rozdílný kód nebo jeho část pro každou z platform. Tímto způsobem lze cílové platformě přizpůsobit aplikační logiku, strukturu uživatelského rozhraní i použité styly [55].

Struktura aplikace je důsledně dodržována. Každá část MVC architektury má vlastní složku ve struktuře frameworku, navíc je zde složka pro styly. Soubory vztahené k jednomu logickému celku mají stejné jméno, odlišují se podle složky, ve které se nacházejí. Pokud budeme mít část např. se samostatným menu aplikace, vypadala by struktura souborů takto 3.1.

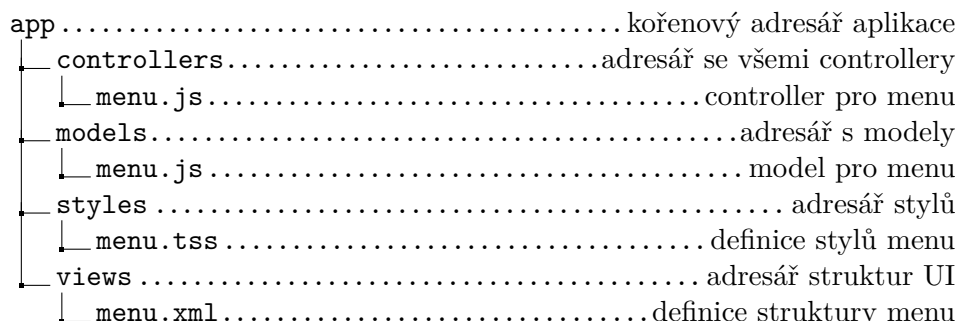
Platformně závislý kód může být rozdělen do dalších podadresářů. Pokud bychom chtěli jiný styl pro platformu Android a jinou aplikační logiku pro iOS, bude struktura jako na obrázku 3.2.

Není-li přítomen adresář a soubor pro specifickou platformu, je použit soubor nadřazený [56].

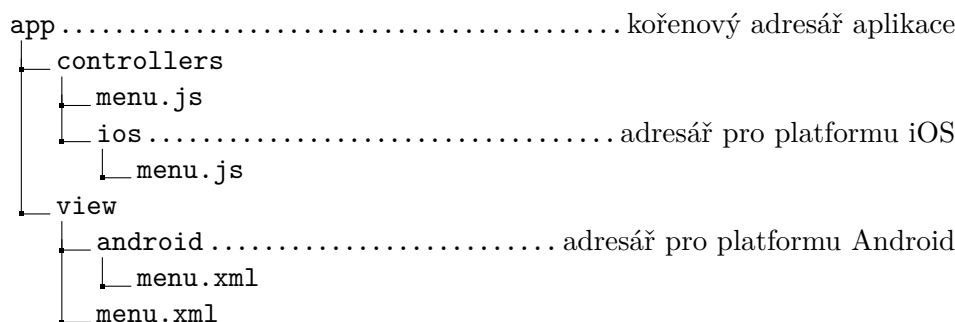
Dalším způsobem odlišení aplikace v závislosti na platformě je testování pomocí speciálních proměnných frameworku Alloy. Pro představu proměnná `OS_ANDROID` bude mít hodnotu `true`, pokud je platformou Android [57].



Obrázek 3.1: Struktura aplikace ve frameworku Titanium



Obrázek 3.2: Struktura s ohledem na platformu ve frameworku Titanium



### 3.2.3.2 Ionic

Framework Ionic se zaměřuje na tvorbu hybridních mobilních aplikací a to hlavně z pohledu uživatelského rozhraní. Snaží se o to, aby hybridní aplikace vypadala a reagovala stejně jako aplikace nativní [58]. Podporuje platformy iOS a Android, ale do budoucna je ohlášena i podpora Windows Phone a Firefox OS [59]. Ionic je šířen jako open source pod MIT licencí [60].

Ionic je založen na oblíbeném MVC frameworku JS, který potřebuje pro svou plnou funkčnost [61]. AngularJS poskytuje další funkcionalitu. Seznam funkcí je k nalezení na webových stránkách projektu [62]. Patří mezi ně:

- Usnadnění rozdělení aplikace do zvolené architektury. Podporované jsou architektury MVC, MVVM, MVP [63]. S tím souvisí snadné definování kontrolerů, služeb a zobrazovacích vrstev.
- Definování a použití šablon v jazyku HTML s dalším rozšířením pro dynamické zobrazování obsahu na základě informací z modelu a z controllerů [64].
- Direktivy pro změnu chování HTML elementů. Tyto direktivy se píšou jako atributy v rámci HTML elementů. Umožňují například jednoduchý

### 3. VÝBĚR A SROVNÁNÍ FRAMEWORKŮ

---

výpis několika položek z pole (direktiva `ng-repeat`), zobrazení elementu při splnění určité podmínky (direktiva `ng-if`), namapování zadaných hodnot na model (`ng-model`) a mnoho dalších. Mimo změny chování existujících elementů umožňují AngularJS direktivy také tvorbu vlastních HTML elementů [65].

- Předávání dat mezi jednotlivými vrstvami aplikace pomocí tzv. Scope objektů [62].
- Správa závislostí (Dependency Injection) - jednotlivé části aplikace si mohou vyžádat externí knihovny a další části aplikace, pokud je vyžadují pro svou činnost [66].

Mimo frameworku AngularJS využívá Ionic také nástroj Cordova. Tento nástroj je obsažen v Ionic CLI nástrojích. Ionic CLI umožňuje z příkazové řádky vytvořit projekt ve frameworku Ionic, vytvořit lokální server pro testování a zobrazit aplikaci v internetovém prohlížeči, na fyzickém zařízení nebo na emulátoru [67].

Samotná nadstavba Ionic je hlavně kombinací CSS frameworku a Javascriptové UI knihovny [68]. Obě součásti se dají intuitivně použít společně. Pro tento účel využívá Ionic vlastních HTML elementů. Výhodou tohoto přístupu je zjednodušení zápisu a zlepšení čitelnosti kódu. Všechny tyto elementy začínají předponou `ion-` a zapisují se přímo do HTML souboru, respektive do šablon AngularJS [69]. Mezi vlastní Ionic elementy patří postranní menu (`ion-side-menus`), vertikálně a/nebo horizontálně posuvná plocha (`ion-scroll`), uživatelská navigace formou karet (`ion-tabs`) a další [70].

Ionic obsahuje jedno grafické téma, které je přizpůsobené pro dotykové displeje s malou úhlopříčkou. Toto téma je inspirované vzhledem iOS7, ale nejedná se o jeho dokonalou kopii. Téma je určeno pro snadnou modifikaci vývojářem [71].

Vzhled aplikace může být změněn přímo v CSS, nebo také v jazyku SASS [58]. SASS je jazyk, který rozšiřuje možnosti CSS. Oproti CSS umožňuje používání proměnných, zanořování elementů, dědičnost, importování jiných SASS souborů a výpočty uvnitř kódu pomocí operátorů `+`, `-`, `*`, `/` a `%` [72]. Z vytvořených SASS souborů se vygeneruje klasické CSS [73].

Celé uživatelské rozhraní aplikace ve frameworku Ionic se dá vytvořit ve webovém nástroji Ionic Creator, který je volně k dispozici a momentálně se nachází v BETA verzi. Pro přístup k nástroji je třeba se zaregistrovat na webových stránkách projektu [74].

Framework je také připraven na odlišení platformě závislého kódu. Běžová platforma se dá zjistit v rámci JavaScriptového kódu díky objektu

`ionic.Platform`, který obsahuje potřebné metody [75]. Platformu lze odlišit i z CSS, slouží k tomu CSS třídy `.platform-ios`, `.platform-android` a `.platform-browser` [76].

Součástí materiálů pro vývojáře je také soubor mnoha interaktivních vzorových aplikací na webu Codepen. Kód těchto aplikací se dá upravovat přímo z webového prohlížeče a jeho změna je ihned viditelná v běžící aplikaci [77].

### 3.2.3.3 jQuery Mobile

Mobilní framework poprvé vydaný v roce 2010 [78, str. 16] je rozšířením oblíbené knihovny jQuery [78, str. 20].

jQuery Mobile podporuje velké množství platforem včetně Androidu, iOS a Windows Phone [79]. Framework je vhodný pro tvorbu webových aplikací určených k prohlížení z mobilních zařízení a spolu s nástrojem Cordova může být použit pro tvorbu hybridních mobilních aplikací.

Vývojář může využít zabudovaný CSS framework, který zjednodušuje implementaci vzhledu uživatelského rozhraní aplikace. Obsahuje předpřipravené CSS třídy s často používanými definicemi vzhledu, jako jsou zahnuté rohy elementu, stínování, různé ikony a jejich umístění [80].

Dále obsahuje komponentu s názvem `Grid Layout`. Tato komponenta umožňuje pozicování elementů do mřížky, určení jejich vzájemné polohy a velikosti a chování při změně velikosti okna, potažmo o přizpůsobení velikosti elementů pro různé displeje [81].

Poslední částí CSS frameworku v rámci jQuery Mobile je jednotné vzhledové téma vhodné pro mobilní zařízení. V základu obsahuje dvě barevné varianty, tmavší a světlejší. Barevná témata si ale může vývojář vytvořit vlastní, k tomuto účelu slouží webová aplikace ThemeRoller. Témata se označují jedním písmenem abecedy a ta předpřipravená mají označení `a` a `b` [82].

Dále framework obsahuje UI elementy přizpůsobené pro dotykové ovládání. Mezi tyto elementy patří vyskakovací kalendář pro výběr data, tabulky, karty s různým obsahem, posuvný seznam a elementy pro použití ve formulářích [83].

K určení role jednotlivých elementů používá jQuery Mobile datový atribut `data-role`. [78, str. 26]. Elementy mohou mít roli stránky (`data-role='page'`), dialogového okna (`data-role='dialog'`), vyskakovacího okna (`data-role='popup'`), a jiné [84].

K zobrazení jednotlivých částí aplikace se používá klasický odkaz na cílový element `<a href='#idElementu'>`. Tento odkaz dále musí obsahovat datový atribut `data-rel`, který popisuje roli cíle. Při klepnutí na odkaz se framework postará o správnou reakci, tedy např. zobrazení vyskakovacího okna, přechodu na jinou stránku a podobně [85].

Všechny stránky aplikace mohou být uloženy v jednom HTML souboru, případně mohou být uloženy všechny zvlášť a framework se postará o jejich načtení pomocí technologie AJAX v případě jejich potřeby [78, str. 35].

#### 3.2.3.4 Sencha Touch

Sencha Touch je aplikační framework pro tvorbu hybridních mobilních aplikací. Ze všech zvolených frameworků působí nejobsáhlejším dojmem a zahrnuje nástroje a funkce pro tvorbu celé mobilní aplikace od uživatelského rozhraní včetně vzhledu a běžných UI komponent, přes MV\* architekturu, po zobrazování grafiky, komunikaci se serverem a mnoho dalšího. Kompletní možnosti lze najít v oficiální dokumentaci [86], která je velmi kvalitní, obsahuje množství příkladů a videí a působí přehledně.

Samotný framework je založen na jádru podnikového JavaScriptového frameworku ExtJS od stejné společnosti. Navíc je na rozdíl od samotného ExtJS [87] zdarma [88].

Aplikace ve frameworku Sencha Touch je kolekcí 5 typů objektů [89].

- Model reprezentuje datové objekty aplikace.
- View je zodpovědný za zobrazování dat.
- Controller se stará o interakci aplikace s uživatelem.
- Store zodpovídá za načtení dat do aplikace a zpřístupnění těchto dat komponentám (např. posuvný seznam).
- Profile umožňuje přizpůsobení aplikace vzhledem k cílové platformě a typu zařízení.

Framework je silně objektově orientovaný a vnitřní struktura používá velké množství abstrakce. Abstrakce zahrnuje také uživatelské rozhraní, jež se většinou nepíše v jazyku HTML, ale v JavaScriptu, ze kterého poté framework vytvoří samotné HTML [90].

Třída `Ext.Loader` se stará o správu závislostí [47].

Framework podporuje platformy iOS, Android, Windows Phone, Black Berry a Tizen [91]. Pro každou z těchto platform má vlastní vzhledové téma, které se snaží danou platformu napodobit. Pokud vývojář neurčí jinak, použije aplikace vytvořená v tomto frameworku právě tyto styly podle aktuální platformy zařízení, na kterém je spuštěna [92]. Stejně jako ve frameworku Ionic je pro stylování využit jazyk SASS.

Firma Sencha k frameworku nabízí také některé nástroje a další komponenty. Nástroj Sencha CMD nabízí funkce usnadňující vývoj, které se spouštějí z příkazové řádky. Mezi tyto funkce patří generování kódu a zabalení aplikace pro použití v zařízení a distribuci [93]. Dále je to aplikace Sencha Architect pro tvorbu uživatelského rozhraní, a komponenty pro zobrazení grafů nebo tabulek. Kromě Sencha CMD jsou tyto nástroje a komponenty placené. Kompletní balík nástrojů jde zakoupit dohromady pod názvem Sencha Touch Bundle [94].

### 3.2.3.5 Kendo UI

Kendo UI je framework určený pro tvorbu webových a hybridních aplikací.

Existuje ve dvou verzích. Verze Core, která je šířena jako open source pod licencí Apache v2 a komerční verze Pro.

Framework využívá knihovnu jQuery, podporuje Twitter Bootstrap a umožňuje integraci do frameworku AngularJS [39].

Obsahuje více než 70 UI komponent (tzv. widgetů), patří mezi ně běžně používané komponenty jako tlačítka, menu, posuvný výběr hodnoty, výběr data a času a mnoho dalších.

Některé komponenty jsou určeny speciálně pro mobilní zařízení, jiné se přizpůsobí podle velikosti displeje.

Kendo UI Pro obsahuje oproti verzi Core navíc některé pokročilé widgety, jinak jsou tyto 2 verze totožné.

Komponenty obsažené pouze v edici Pro jsou [95]:

#### **Barcode**

Komponenta pro generování čárových kódů. Podporuje několik standardů kódování i zobrazení kontrolního součtu.

#### **Charts**

Pokročilé zobrazení grafů. Výsledkem může být graf sloupcový, paprskový, prstencový, spojnicový, souřadnicový nebo jiný.

#### **Diagram**

Interaktivní schémata se schopností přibližování, oddalování a editace uživatelem. Tato schémata také možno exportovat ve formátu PDF, SVG nebo PNG.

#### **Editor**

Textový editor přístupný přímo z webového prohlížeče. Podporuje formátování textu a odstavců, tvorbu tabulek, vkládání obrázků i jiných souborů a přizpůsobení jednotlivých stylů. Z textu je vytvořen kód HTML.

### 3. VÝBĚR A SROVNÁNÍ FRAMEWORKŮ

---

#### **Gauges**

Komponenta pro znázornění hodnot, zobrazení postupu nebo rozpětí.

#### **Grid**

Interaktivní zobrazení tabulek. Položky v tabulce mohou být rozděleny na několik stránek, dají se řadit podle jednotlivých kritérií, lze je mazat, přidávat a editovat. Výsledná tabulka se dá exportovat do PDF nebo Excelu.

#### **Map**

Komponenta s mapou. Mapu lze posouvat, přibližovat a oddalovat. Mapa může mít více vrstev a dají se na ni zobrazovat další informace pomocí symbolů.

#### **QR Code**

Generování a zobrazení tzv. QR kódu pro předávání informací.

#### **Scheduler**

Interaktivní kalendář, ve kterém se dají zobrazit události. Tyto události je možné přidávat, odebírat, editovat i přesouvat. Kalendář má také různé pohledy (denní, týdenní, měsíční...).

#### **Stock Chart**

Burzovní graf s časovou osou pro výběr časového rozpětí, ze které budou zobrazeny jednotlivé hodnoty a jejich vývoj v grafu.

#### **TreeView**

Zobrazení stromové struktury (např. adresářová struktura). Uživatel může jednotlivé uzly zabalit/rozbalit a je přítomna i funkce pro přesouvání jednotlivých položek tažením myši.

#### **Upload**

Nástroj pro synchronní a asynchronní výběr a nahrávání souborů na server.

Tyto komponenty jsou určeny jak pro klasické počítače, tak pro mobilní zařízení [95].

Framework kromě widgetů obsahuje i podporu pro MVVM architekturu, šablony, validaci dat, jazykovou lokalizaci a grafické efekty a animace.

Základem pro použití MVVM přístupu z Kendo UI je JS metoda `kendo.observable` a datový atribut `data-bind`. Tímto způsobem může být předáván obsah elementu, jeho atribut, styl nebo funkce reagující na určitou událost [96].

Šablony reprezentují části stránky s přiřazenými daty. Definují se v elementu `<script>` s atributem `type="text/x-kendo-template"`. Uvnitř šablon mohou být použity výrazy ohraničené znakem `#`.

Existují 3 typy těchto výrazů [97].

- `#= proměnná #` ..... Vloží do HTML kódu hodnotu proměnné.
- `#: proměnná #` ..... Také vloží hodnotu proměnné, ale pokud proměnná obsahuje HTML kód, nebude tento kód interpretován a bude zobrazen jako řetězec.
- `# JavaScriptový kód #` ..... Provede vložený JS kód

S těmito výrazy může být efektivně kombinován klasický HTML kód, zobrazení dat a logika, kterou chceme mít v zobrazovací vrstě [97].

Role jednotlivých elementů se stejně jako u frameworku jQuery Mobile určuje atributem `data-role`.

Samotná mobilní aplikace je složena z jednotlivých pohledů, pohledy se označují atributem s hodnotou `data-role="view"`.

Tyto pohledy jsou velmi podobné stránkám z jQuery Mobile, může jich být více v jednom HTML souboru, nebo mohou být zvlášť. Pokud má být zobrazen pohled, který je v jiném souboru, je načten technologií AJAX.

Pokud v aplikaci není žádný element, který je označen jako pohled, bude automaticky vytvořen element `<div data-role="view">`, který ohraničí ostatní obsah [98].

### 3.2.3.6 Onsen UI

OnsenUI je open source hybridní framework podobný frameworku Ionic. Stejně jako on je založen na frameworku AngularJS, používá vlastní HTML elementy a CSS komponenty.

CSS komponenty umožňují úpravu vzhledu klasických HTML elementů pro mobilní zařízení. Elementům stačí přiřadit správnou CSS třídu. Takto může být změněn vzhled tlačítek, zaškrtačích polí, přepínačů a dalších. Další elementy jako `div` a `span` s příslušnou CSS třídou mohou být použity pro záhlaví/zápatí stránky, pro zobrazení notifikací a různých vyskakovacích oken [99].

HTML elementy začínají předponou `ons-` a umožňují snadné přidávání OnsenUI komponent. Tyto komponenty mají upravenou funkčnost a vzhled pro mobilní zařízení a patří mezi ně [100]:

- `ons-page`.....element určující jednotlivé stránky aplikace

### 3. VÝBĚR A SROVNÁNÍ FRAMEWORKŮ

---

- `ons-carousel`..... přepínání obsahu gestem prstu
- `ons-dialog`..... vyskakovací okno
- `ons-button`..... tlačítko s přizpůsobeným vzhledem
- `ons-switch`..... přepínač se stavy vypnuto a zapnuto
- `ons-list`..... posuvný seznam položek
- `ons-modal`..... překrytí obrazovky jiným obsahem, využívané např. při čekání na načtení dat z internetu
- `ons-sliding-menu`..... menu, které se vysune z kraje obrazovky
- `ons-tab`..... přepínatelné záložky
- `ons-popover`..... bubila s textem, která se zobrazí u určeného elementu
- `ons-toolbar`..... záhlaví s textem a tlačítky

Na stránkách projektu jsou k dispozici šablony pro různé typy stránek. Ve webové aplikaci Theme Roller mohou být vybrány konkrétní vyžadované komponenty, které budou zahrnuty do balíčku, pro omezení velikosti výsledné aplikace. Ve stejné aplikaci se dají tyto komponenty také vzhledově upravovat [101].

Framework Onsen UI může být použit pro tvorbu webových aplikací, pro tvorbu hybridních aplikací s nástrojem Cordova, nebo s webovým vývojovým prostředím Monaca od stejných vývojářů [102].

#### 3.2.3.7 Xamarin

Framework Xamarin přistupuje k vývoji multiplatformních mobilních aplikací zcela jinak než ostatní srovnávané frameworky.

Umožňuje vývoj na platformě .NET v programovacím jazyku C#. Xamarin je založen na projektu Mono, který je zaměřen na multiplatformní vývoj a provoz aplikací v různých technologiích, mezi které patří i C#. Mono se skládá ze 2 částí [103].

- Mono C# Compiler,
- Mono Runtime.

Xamarin abstrahuje veškerou funkcionalitu cílových platforem a umožňuje tvorbu kompilovaných aplikací. Tyto aplikace jsou vzhledem, rychlostí a chováním k nerozeznání od aplikací nativních [46].



Xamarin je jediný ze zmíněných frameworků, který pro vývoj komerčních aplikací vyžaduje placení pravidelných poplatků. Nabízí sice i verzi zdarma, ta je však přístupná pouze organizacím s méně než 5 zaměstnanci a obsahuje výrazná omezení. Levnější z korporátních licencí poté stojí \$ 999 za rok pro jedno zařízení a jednoho uživatele [104].

K vývoji se může použít buď vývojové prostředí Visual Studio s patričným doplňkem, nebo vlastní Xamarin Studio [105]. Toto IDE je velmi dobré, nabízí chytré doplňování syntaxe, umožňuje tvorbu uživatelského rozhraní v grafickém editoru a testování aplikace na zařízení nebo emulátoru.

Pro vývoj aplikace pro platformu iOS je nutný běh pod operačním systémem MacOS, naopak pro vývoj aplikace pro Windows Phone je zapotřebí vývojové prostředí Visual Studio a operační systém Windows 8+. Pro vývoj jedné aplikace a její distribuci pro více platform je tedy zapotřebí více než jedna licence frameworku [105].

Samotný vývoj se podobá vývoji nativních aplikací. Pro každou cílovou platformu se vytvoří vlastní projekt. Pro sdílení kódu napříč platformami se využívá tzv. Shared Projects [105]. Uživatelské rozhraní a jeho vzhled se může vytvořit pro každou platformu zvlášť, případně se může použít API s názvem `Xamarin.Form`, které umožňuje tvorbu společného uživatelského rozhraní, které se pouze přizpůsobí daným platformám. Toto API není přístupné ve verzi zdarma, a proto nebylo vyzkoušeno [106].

Do aplikace se dají přidat i další knihovny. Je možné použít knihovny v jazyku C# i knihovny specifické pro dané platformy v jazycích Java a Objective-C [107].

### 3.3 Specifikace kritérií a srovnávací metodika

Do srovnání frameworků jsou zahrnuta kritéria jako podporované platformy, licenční podmínky a schopnost využít hardwarové prostředky zařízení.

Výsledky frameworků v těchto kategoriích jsou určeny na základě údajů z dokumentací, které poskytují vývojáři daných frameworků.

Dále se zkoumá podpora základních komponent využívaných u mobilních aplikací, tedy komponenty formuláře, posuvné seznamy s klikatelnými položkami, vysouvací menu a podpora formátu SVG.

Pro tuto část kritérií a část následující byly implementovány zkušební aplikace ve všech srovnávaných frameworkích a podpora těchto komponent v nich byla otestována.

### 3. VÝBĚR A SROVNÁNÍ FRAMEWORKŮ

---

Poslední hodnocenou částí frameworků je srovnání zkušebních aplikací z uživatelského hlediska. Kritérii této části jsou pocitová rychlost z používání aplikace a její vzhled.

Hodnotí se pouze to, co daný framework podporuje v základu a dá se použít bez dodatečných úprav. U frameworků, které nemají vlastní nástroj pro tvorbu instalovatelné mobilní aplikace, je pro tento účel použit nástroj Cordova.

V rámci poskytnutí dalších informací jsou do tabulky zahrnuty i údaje, které se nehodnotily. Mezi tato kritéria patří technologie nutné pro vývoj a cena za korporátní licenci.

Každému hodnocenému kritériu je přiřazena procentuální hodnota udávající jeho váhu. Pokud kritérium umožňuje více než 2 možné hodnoty výsledku (ano/ne), je kritérium hodnoceno číslem od 1 do 5. Výsledné procentuální hodnocení je určeno na základě těchto hodnot.

U kritérií s možnými hodnotami ANO/NE je výsledek jednoznačný (100 % nebo 0 %). U kritérií s 5 možnými hodnotami je poté výsledek určen procentem z váhy. Nejlepší hodnocení je značeno číslem 1 (100 %), nejhorší číslem 5 (0 %). Průběh je lineární.

#### 3.3.1 Seznam a upřesnění kritérií

Následující seznam poskytuje kompletní přehled o srovnávaných kritériích a jejich váze vzhledem k celkovému hodnocení.

Je-li to nutné, pak je dané kritérium více upřesněno.

1. Podpora platforem
  - a) Android
  - b) Windows Phone
  - c) iOS
2. Podpora základních UI komponent
  - a) Postranní vysouvací menu - menu aplikace vysouvané stisknutím tlačítka a/nebo tažením prstu
  - b) Vertikálně posuvný seznam s klikatelnými položkami
  - c) Vyskakovací okno s textem a tlačítkem
3. Komponenty formuláře
  - a) Pole pro zadávání textu
  - b) Zaškrtávací pole

- c) Tlačítko
  - d) Výběr z více položek
  - e) Pole pro zadání hesla
4. Platformě závislý vzhled
- a) Možnost mít jiný vzhled pro jednotlivé platformy
  - b) Základní styly pro jednotlivé platformy obsaženy ve frameworku
5. Přístup k HW zařízení
- a) Možnost ukládat soubory na lokální úložiště
  - b) Geolokace podle bezdrátové sítě a GPS
  - c) Zjištění stavu internetového připojení
  - d) Ovládání fotoaparátu
  - e) Získání údajů z akcelerometru
  - f) Přístup ke kompasu
  - g) Řízení vibrací zařízení
6. Podpora SVG
- a) Zobrazení
  - b) Přibližování a oddalování
  - c) Posouvání
  - d) Klikání na jednotlivé části
7. Technologie pro vývoj
- a) Náročnost z pohledu webového vývojáře - vystačí-li si vývojář s programovacími jazyky HTML, CSS a JS a nebude-li se muset učit nic dalšího
  - b) Kvalita a dostupnost dokumentace - zda je dokumentace kompletní, aktuální a přehledně řazená
  - c) Licence zdarma pro komerční účely
8. Výkon
- a) Rychlost a reakce aplikace - rychlost přechodů grafického uživatelského rozhraní aplikace. Hodnotí se podle chování vzorové aplikace, což je možné díky tomu, že se frameworky rychlostně liší velmi výrazně a rozdíl.

### 3.4 Testovací aplikace

#### 3.4.1 Cíl

Cílem je vytvořit v každém frameworku aplikaci, ve které budou vyzkoušeny vyžadované komponenty a otestovány některé funkce frameworků. Výsledné aplikace by měly být co nejvíce podobné, aby se daly vzájemně porovnat.

#### 3.4.2 Implementace

Každý srovnávaný framework přistupuje k tvorbě aplikace odlišně. Při implementaci aplikací byla snaha tyto odlišnosti zaznamenat a vyvinout aplikaci na míru jednotlivým frameworkům.

Aplikace jsou z uživatelského hlediska rozděleny na podstránky, které se liší obsahem. Mezi jednotlivými podstránkami se přechází přes vysouvací menu, které se zobrazí při klepnutí na tlačítko v pravé horní části obrazovky. Pokud to framework podporuje, lze menu vysunout i gestem prstu (tažením zleva doprava).

Pokud ve frameworku není přítomno vysouvací menu, je jako menu použita další podstránka.

Aplikace obsahují následující podstránky.

##### **Hlavní stránka**

Vstupní bod aplikace. Obsahuje označení frameworku a přístup k hlavnímu menu.

##### **Formulář**

Stránka s formulářem, který obsahuje textové pole, pole s heslem, přepínač, výběr, tlačítko a další komponenty.

##### **Vertikálně posuvný seznam**

Stránka s posuvným seznamem položek. Po kliknutí na jednotlivou položku se ukáže její detail.

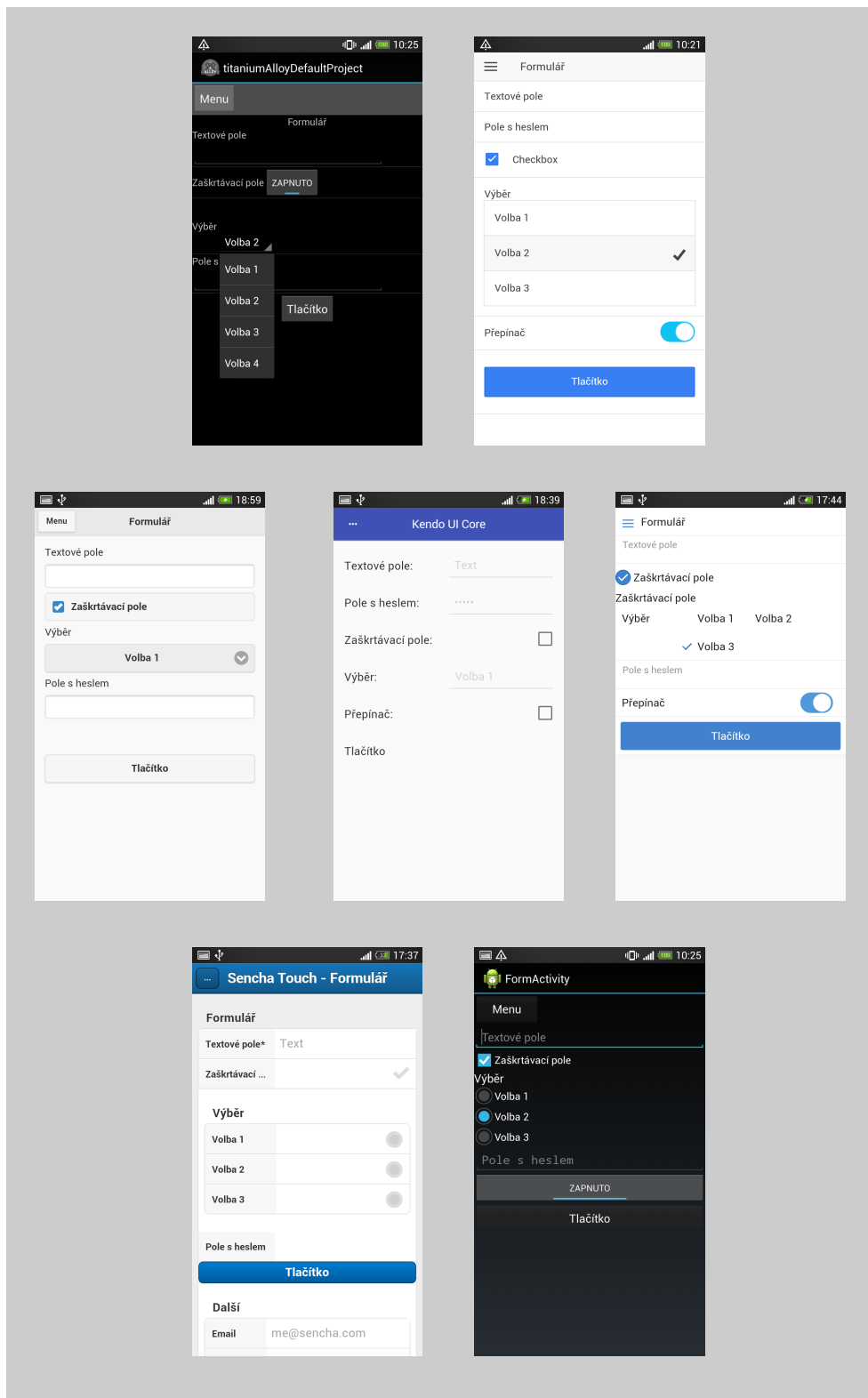
##### **SVG**

Stránka s obrázkem ve formátu SVG s funkcemi přibližování/oddalování, posuvu a klikání na jednotlivé části.

#### 3.4.3 Vzhled

Aplikace slouží také k porovnání základních vzhledů, které poskytují jednotlivé frameworky. Na obrázku 3.3 lze vidět vzhled vzorových aplikací. Postupně jsou zobrazeny frameworky Appcelerator Titanium, Ionic, jQuery Mobile, Kendo UI Core, Onsen UI, Sencha Touch a Xamarin.

### 3.4. Testovací aplikace



Obrázek 3.3: Vzhled aplikací v různých frameworkech.

## 3.5 Srovnávací tabulka

Tabulka 3.3: Srovnávací tabulka

Kritérium	Váha [%]	Appcelerator	Ionic	jQuery Mobile	Kendo UI	Onsen UI	Sencha Touch	Xamarin
<i>Podpora platforem</i>								
Android	15 %	✓	✓	✓	✓	✓	✓	✓
iOS	15 %	✓	✓	✓	✓	✓	✓	✓
Windows Phone	15 %	✓	* <sup>2</sup>	✓	✓	✗	✓	✓
<b>Výsledky sekce</b>	45 %	100 %	83 %	100 %	100 %	67 %	100 %	100 %
<i>Podpora základních komponent</i>								
Postranní menu	2 %	✗	✓	✓	✓	✓	✗ <sup>3</sup>	✗
Vertikálně posuvný seznam položek	2 %	* <sup>4</sup>	✓	✓	✓	✓	✓	✓
Vyskakovací okno s textem a tlačítkem	2 %	✓	✓	✓	✓	✓	✓	✓
<b>Výsledky sekce</b>	6 %	50 %	100 %	100 %	100 %	100 %	67 %	67 %

<sup>2</sup>Uvedena částečná podpora s možnou plnou podporou v budoucích verzích [59].

<sup>3</sup>Existuje plugin doplňující tuto funkcionalitu.

<sup>4</sup>Pouze pro platformy Android a iOS.

Tabulka 3.3: Srovnávací tabulka

Kritérium	Váha [%]	Appcelerator	Ionic	jQuery Mobile	Kendo UI	Onsen UI	Sencha Touch	Xamarin
<i>Formulář</i>								
Textové pole	1 %	✓	✓	✓	✓	✓	✓	✓
Zaškrtačací pole	1 %	✓	✓	✓	✓	✓	✓	✓
Tlačítko	1 %	✓	✓	✓	✓	✓	✓	✓
Výběr ze seznamu	1 %	✓	✓	✓	✓	✓	✓	✓
Pole s heslem	1 %	✓	✓	✓	✓	✓	✓	✓
<b>Výsledky sekce</b>	5 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
<i>Odlišení platform</i>								
Možnost mít jiný vzhled pro jednotlivé platformy	3 %	✓	✓	✓ <sup>5</sup>	✓	✓ <sup>5</sup>	✓	✓
Obsaženy styly pro jednotlivé platformy	3 %	✗	✗	✗	✓	✗	✓	✗
<b>Výsledky sekce</b>	6 %	50 %	50 %	50 %	100 %	50 %	100 %	50 %

<sup>5</sup>Rozlišením tzv. `User Agent` proměnné v CSS Media Query.

### 3. VÝBĚR A SROVNÁNÍ FRAMEWORKŮ

Tabulka 3.3: Srovnávací tabulka

Kritérium	Váha [%]	Appcelerator	Ionic	jQuery Mobile	Kendo UI	Onsen UI	Sencha Touch	Xamarin
<i>Podpora nativních funkcí</i>								
Lokální úložiště	1 %	✓	✓	✓	✓	✓	✓	✓
GPS	1 %	✓	✓	✓	✓	✓	✓	✓
Stav sítě a připojení k internetu	1 %	✓	✓	✓	✓	✓	✓	✓
Fotoaparát	1 %	✓	✓	✓	✓	✓	✓	✓
Akcelerometr	1 %	✓	✓	✓	✓	✓	✓	✓
Kompas	1 %	✓	✓	✓	✓	✓	✓	✓
Vibrace	1 %	✓	✓	✓	✓	✓	✓	✓
<b>Výsledky sekce</b>	7 %	100 %	100 %	100 %	100 %	100 %	100 %	100 %
<i>Podpora SVG</i>								
Zobrazení	1 %	✗	✓	✓	✓	✓	✓	✗
Posouvání	1 %	✗	✓	✗ <sup>6</sup>	✗	✗	✗	✗
Přibližování a oddalování	1 %	✗	✓	✗ <sup>6</sup>	✗	✗	✗	✗
Klikání	1 %	✗	✓ <sup>7</sup>	✓	✓	✓	✓	✗
<b>Výsledky sekce</b>	4 %	0 %	100 %	50 %	50 %	50 %	50 %	0 %

<sup>6</sup>Existuje plugin doplňující tuto funkcionalitu.

<sup>7</sup>Klikání lze použít, ale ne zároveň s přibližováním.



Tabulka 3.3: Srovnávací tabulka

Kritérium	Váha [%]	Appcelerator	Ionic	jQuery Mobile	Kendo UI	Onsen UI	Sencha Touch	Xamarin
<i>Vývoj</i>								
Náročnost	5 %	5	3	2	2	3	4	5
Kvalita a dostupnost dokumentace	2 %	3	2	1	1	2	1	3
Zdarma pro komerční účely	5 %	✓	✓	✓	✓	✓	✓	✗
<b>Výsledky sekce</b>	12 %	50 %	75 %	90 %	90 %	75 %	69 %	8 %
<i>Výkon</i>								
Rychlost výsledné aplikace	15 %	2	2	5	2	2	4	1
<b>Výsledky sekce</b>	15 %	75 %	75 %	0 %	75 %	75 %	25 %	100 %

## 3.6 Zhodnocení výsledků

### 3.6.1 Grafy s výsledky

Celkové výsledky srovnání jsou znázorněny na grafu 3.4. Výsledky frameworků v jednotlivých kategoriích lze vidět na grafu 3.5.

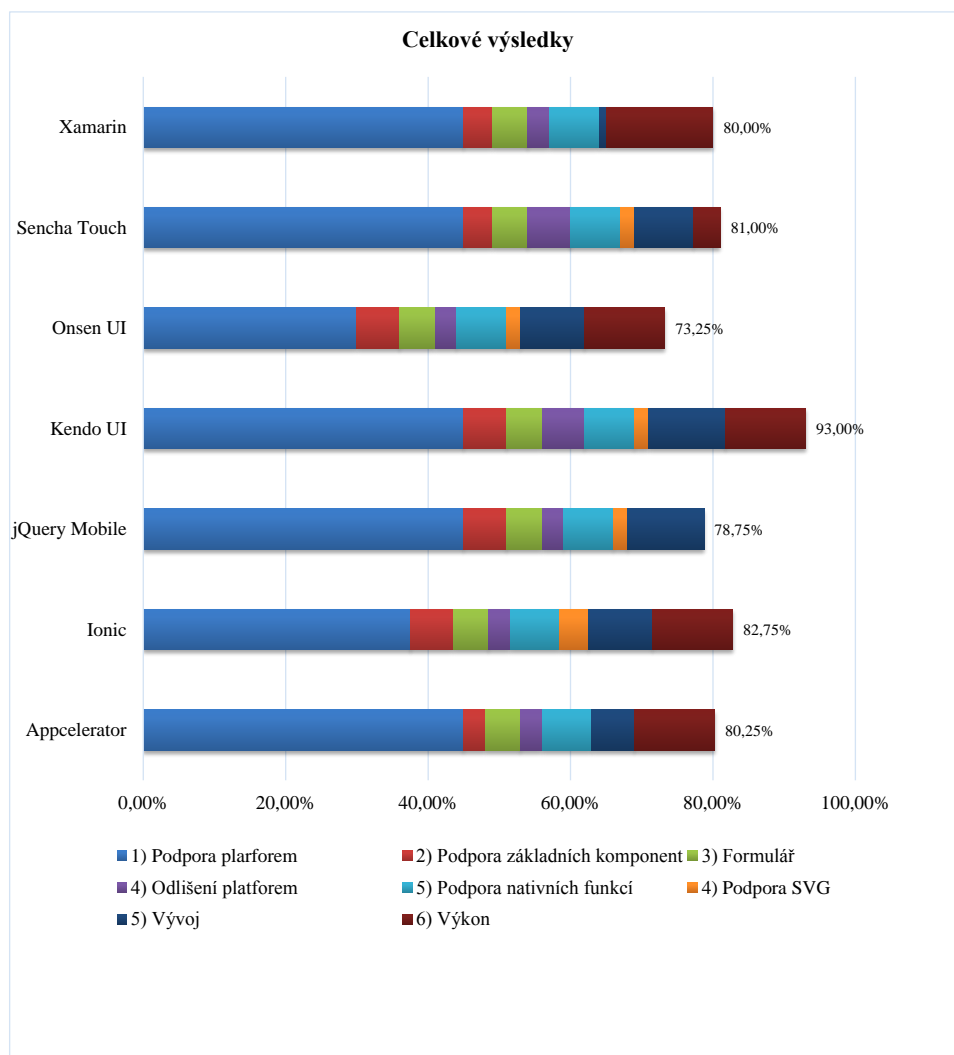
### 3.6.2 Stručný souhrn

Ze srovnání vyšel nejlépe framework Kendo UI. Hlavními vlastnostmi, kterými se odlišuje od konkurenčních frameworků, jsou plná podpora platformy Windows Phone, vlastní grafické styly pro každou platformu a rychlost srovnatelná s nejrychlejšími hybridními frameworky.

Všechny testované frameworky podporují většinu základních komponent, jediná problematická komponenta je vysouvací menu.

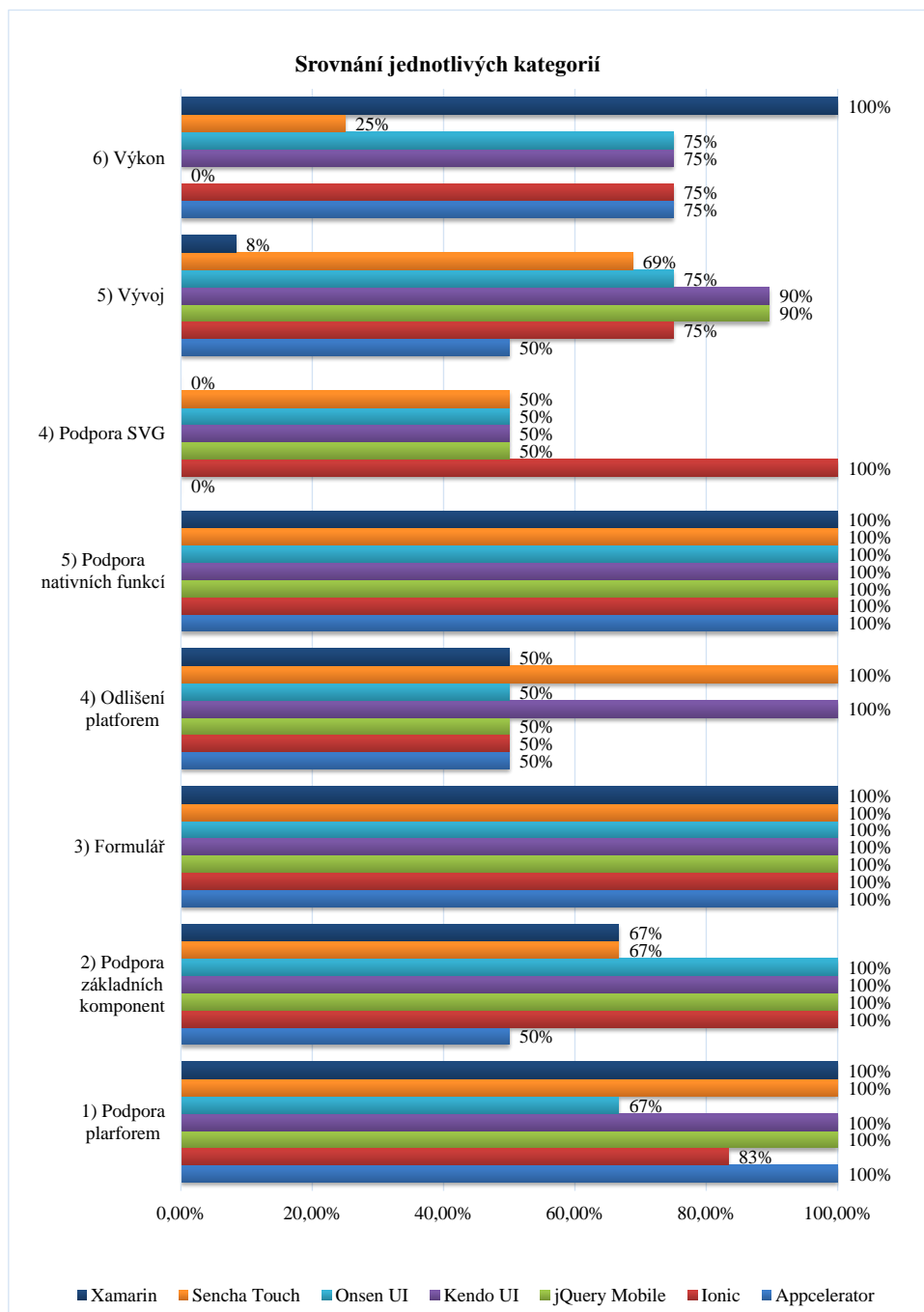
### 3. VÝBĚR A SROVNÁNÍ FRAMEWORKŮ

---



Obrázek 3.4: Graf s celkovými výsledky frameworků ve srovnání

### 3.6. Zhodnocení výsledků



Obrázek 3.5: Graf s jednotlivými kategoriemi

V hodnocení jsou částečně zvýhodněny všechny hybridní frameworky. Na rozdíl od ostatních totiž nabízejí minimálně jeden předpřipravený grafický styl a zobrazení ve webovém prohlížeči jim dává alespoň částečnou podporu formátu SVG.

Nicméně z výsledků je patrné, že jsou si frameworky možnostmi velmi podobné a mobilní aplikace se dá snadno vyvinout v kterémkoliv z nich. Některé frameworky jsou vhodnější pro některý typ aplikace, jiné frameworky zase pro jiný, proto je každý framework zhodnocen zvlášť v následující sekci práce.

#### 3.6.3 Zhodnocení jednotlivých frameworků

##### 3.6.3.1 Appcelerator Titanium

Framework Appcelerator Titanium najde uplatnění u aplikací, u kterých je vyžadován nativní vzhled (pouze pro iOS a Android) a nelze z nějakého důvodu (cena za licenci, programovací jazyk) použít framework Xamarin.

Z teoretického pohledu by framework měl mít rychlejší odezvu na uživatelskou interakci než hybridní frameworky. Tento předpoklad se ale při testování nepotvrdil.

Dále framework trochu sráží samotná technologie. Vývojář se musí speciálně naučit jazyk TSS a specifickou definici XML. Dále je zde z mého pohledu méně přehledná dokumentace, která si občas protiřečí (někde je uváděn vývoj v jazyce CSS, jinde nutnost TSS) a některé stránky nejsou delší dobu aktualizovány a jsou proto určeny pro starou verzi frameworku.

Framework v základu nemá komponentu pro vysouvací menu, existuje sice několik volně dostupných rozšíření, které řeší tuto problematiku, ale nepodařilo se mi najít žádné, které podporuje všechny platformy.

##### 3.6.3.2 Ionic a Onsen UI

Tyto dva hybridní frameworky jsou si velice podobné, jak z technologického hlediska, tak z pohledu vývoje, podporovaných komponent i výsledné aplikace.

Výsledné aplikace jsou velmi rychlé, vývoj není náročný, jsou přítomny všechny základní komponenty a vzhled výsledných aplikací je jednoduchý, ale účelný.

Jediným problémem je žádná (Onsen UI), nebo pouze částečná podpora (Ionic) platformy Windows Phone.

To je také jeden z mála vzájemných rozdílů, dalším je podpora posouvání a přibližování SVG obrázků pouze ve frameworku Ionic. Z porovnání statistik

Ionicu [108] a OnsenUI [109] na stránce GitHub také usuzují, že má Ionic větší komunitní podporu.

Oba frameworky jsou velmi vhodné pro tvorbu multiplatformních mobilních aplikací. Pokud není vyžadována podpora platformy Windows Phone a není potřeba rychlost srovnatelná s nativní aplikací (Xamarin), lze jako základ vývoje aplikace doporučit jeden z těchto frameworků.

#### 3.6.3.3 jQuery Mobile

jQuery Mobile se oproti ostatním frameworkům vyznačuje podporou velkého množství platforem (mimo iOS, Androidu a WP).

Přes svou vzájemnou podobnost s frameworkem Kendo UI toho nabízí celkově méně. Ze srovnávaných frameworků je také jQuery Mobile výrazně nejpomalejší.

Framework bych doporučil pouze pro tvorbu velmi jednoduchých aplikací, případně pro rozšíření webových stránek pro zobrazení v mobilních zařízeních.

#### 3.6.3.4 Kendo UI

Framework, který na základě daných kritérií vyšel ze srovnání nejlépe. Vyznačuje se podporou velkého množství komponent, mnoha různými grafickými styly včetně stylů pro Windows Phone, Android a iOS, rychlostí srovnatelnou s frameworky Ionic a Onsen UI a podporou tří největších mobilních platforem. V placené verzi poté dalšími pokročilými komponenty, které u ostatních frameworků nemají obdobu.

Vývoj aplikace je podobný frameworku jQuery Mobile. Na rozdíl od něj je ale výsledná aplikace výrazně rychlejší a Kendo UI nabízí také více možností.

Kendo UI je vhodné pro tvorbu aplikací, kde není potřeba výkon nativní aplikace.

#### 3.6.3.5 Sencha Touch

Sencha Touch se vyznačuje velkým množstvím podpůrných funkcí a metod, které umožňují použití frameworku bez dalších rozšiřujících knihoven.

Výhodou je integrace frameworku ExtJS s jeho architekturou, systémem tříd a celkovým přístupem. Přítomnost grafických stylů pro jednotlivé platformy a jejich celková podpora.

### 3. VÝBĚR A SROVNÁNÍ FRAMEWORKŮ

---

Aplikace vytvořená ve frameworku Sencha Touch je ale výrazně pomalejší, než obdobná aplikace u frameworků Ionic, Onsen UI, Kendo UI a Xamarin.

Framework je vhodný pro rozsáhlé projekty, kde usnadní členění aplikace a bude nutit vývojáře dodržovat stejné konvence. Je ale nutné, aby vývojáři měli zkušenosti s frameworkem ExtJS.

#### 3.6.3.6 Xamarin

Z pohledu chování a rychlosti výsledné aplikace vítěz srovnání. Aplikace vytvořené pomocí tohoto frameworku jsou rychlostně i vzhledově stejné jako aplikace nativní. Výhodou je také výborné vlastní vývojové prostředí.

Framework na rozdíl od hybridních frameworků neobsahuje grafické styly a nutí vývojáře pro tvorbu určité části kódu pro každou platformu zvlášť.

Největší nevýhodou tohoto frameworku je jeho cena, ostatní frameworky se dají používat ve verzi zdarma, ale za tento je nutno platit částku \$ 999 za rok za korporátní licenci na jedno zařízení. Vzhledem k tomu, že pro vývoj současně pro platformy Windows Phone a iOS potřebujeme zařízení s operačním systémem Windows 8+ a Mac OS, náklady ještě vzrostou.

Pokud je požadován výkon a vzhled nativní aplikace, je tento framework ze všech srovnávaných jedinou volbou. Další možností je vývoj každé aplikace pro každou platformu zvlášť.

---

# Tvorba platformy

## 4.1 Cíle platformy

Cílem vytvořené platformy je urychlit začátek vývoje mobilních hybridních aplikací. Konkrétně platforma usnadní výběr, stažení a instalaci komponent a dalších nástrojů přímo na míru vyvíjené aplikaci.

## 4.2 Technologie

Zásadní vliv pro výběr technologií měl požadavek na snadné rozšíření a modifikaci platformy vývojářem (uživatelé platformy) během vývoje vlastní aplikace.

Na základě předpokladu, že vývojář využívající platformu pro vývoj hybridních aplikací bude seznámen s programovacími jazyky HTML, CSS a JavaScript, byly tyto technologie využity i pro vývoj samotné platformy.

Tyto technologie samy o sobě neumožňují tvorbu takové aplikace. Základní vlastností webových aplikací, které jsou interpretovány v klasickém webovém prohlížeči, je jejich izolovanost. Tyto aplikace mohou pouze omezeně přistupovat k souborům na klientském počítači a nemohou spouštět jiné aplikace a procesy. Bez zmíněných schopností se výsledná platforma neobejde.

Řešením tohoto problému je platforma `Node.js`. Tato platforma umožňuje spouštění skriptů v jazyce JavaScript z příkazové řádky. Mimo to obsahuje modul pro práci se souborovým systémem, modul pro spouštění jiných procesů, modul NPM a spoustu dalších. NPM je správce balíků (modulů), umožňuje stahovat, instalovat a celkově spravovat moduly z příkazové řádky nebo z JavaScriptového kódu. Tyto moduly mohou být různé nástroje, aplikace, nebo celé frameworky [110].

Dalším problémem je nemožnost spouštět Node.JS skripty z klasického webového prohlížeče. Tento problém řeší nástroj NW.js (dříve node-webkit). NW.js je kombinace webového prohlížeče (založeném na prohlížeči Chromium) a běhového prostředí Node.js [111]. Umožňuje tedy tvorbu desktopových aplikací s plným přístupem k danému zařízení s použitím webových technologií.

### 4.3 Architektura

Architekturu platformy lze označit jako MVC, je tedy rozdělena na 3 základní části.

#### Model

Veškerá data potřebná pro běh platformy jsou ukládána do souborů ve formátu JSON. Mezi tyto informace patří nainstalované a požadované moduly (komponenty) pro konkrétní projekt a centrální seznam se všemi dostupnými moduly s dalšími potřebnými informacemi.

#### View

Zobrazovací vrstva se liší podle způsobu použití platformy.

Platformu lze použít přímo z příkazové řádky. Při tomto použití veškerá interakce s vývojářem probíhá přes textový terminál. Vývojář do něj zadává příkazy a vidí na něm veškeré informace v textové podobě.

Dalším způsobem je použití platformy skrz plnohodnotné grafické uživatelské rozhraní. GUI platformy umožňuje zobrazení a výběr komponent. Uživatelské rozhraní dále obsahuje tlačítka pro vytvoření projektu se zadanými komponentami a změnu závislostí v existujících projektech.

#### Controller

Veškerá aplikační logika, tedy Node.JS skripty uspořádané do adresářové struktury. V domovském adresáři platformy jsou skripty, které může vývojář spustit z příkazové řádky. Příkazy se řídí přímo informacemi z modelu, vývojář se tedy může obejít i bez uživatelského rozhraní. Tyto nástroje umožňují instalaci samotné platformy a vytvoření a modifikace projektů.

### 4.4 Implementace

#### 4.4.1 Použité externí moduly

Při implementaci platformy byly použity tyto externí moduly.

#### Fs-extra

NPM modul pro práci se souborovým systémem.



**Kendo-ui-core**

Framework popsaný v předchozí části práce. Konkrétně byla použita komponenta Kendo UI Panelbar pro zobrazení seznamu komponent a komponenta Kendo Menu pro hlavní menu aplikace v grafickém uživatelském rozhraní.

**AngularJS**

Framework použit pro oddělení logiky, struktury a NodeJS funkcionality v grafickém uživatelském rozhraní aplikace.

**Bootstrap**

Použití některých komponent pro lepší přehlednost grafického rozhraní.

**4.4.2 Struktura platformy**

Vytvořená platforma je logicky rozdělena do adresářové struktury. Tato struktura je poměrně rozsáhlá a pro pochopení funkce platformy a pro usnadnění tvorby modifikací je nutno přiblížit její jednotlivé části. Kořenový adresář platformy obsahuje následující soubory a adresáře.

```
platform.....kořenový adresář platformy
├── add.js .... skript pro přidání závislosti do projektu z příkazové řádky
├── remove.js . skript pro odstranění závislosti .2 create.js . skript pro
    vytvoření projektu uživatelem
├── reload.js .. stažení vyžadovaných závislostí a smazání nepotřebných
├── install.js ..... skript pro instalaci platformy
├── package.json .. soubor s NPM závislostmi a informacemi o projektu
├── package_default.json ..... package.json použitý u nového projektu
├── project_modules.json ..... informace o vyžadovaných a stažených
    modulech
├── default_modules.json .....project_modules.json pro nový projekt
├── available_modules.json .....soubor se všemi dostupnými moduly
├── index.html .....vstupní bod GUI
├── git_modules/ .....adresář s moduly staženými z GIT repozitáře
├── node_modules/ .....adresář s Node.JS moduly
├── gui/ .....adresář s uživatelským rozhraním
└── scripts/ ..... adresář s podpůrnými skripty
```

Samotná logika platformy je rozdělena do několika NodeJS modulů a umístěna do samostatného podadresáře.

```
scripts .....adresář s aplikační logikou
├─ installer.js .....skript sjednocující instalace modulů
├─ logger.js .....skript s výpisem do konzole
├─ modules.js .....hlavní skript pro správu modulů
├─ processes.js .....spouštění dalších procesů
├─ project.js .....informace o projektu
├─ util.js .....pomocné skripty
├─ installers .....adresář se skripty pro jednotlivé repozitáře
│   └─ gitInstaller.js .....skript pro GIT repozitář
│   └─ npmInstaller.js .....skript pro GIT repozitář
│   └─ commandsInstaller.js .....skript pro instaci přes další příkazy
```

Grafické uživatelské rozhraní platformy má svou vlastní strukturu jako webová aplikace. Pro rozšiřování platformy o další komponenty je nutné tuto strukturu dále přiblížit.

```
├─ index.html .....hlavní stránka a vstupní bod aplikace
├─ gui .....adresář s uživatelským rozhraním
│   └─ app.js .....hlavní JS soubor aplikace, slouží k její inicializaci
│   └─ js .....adresář se všemi JavaScriptovými soubory
│       └─ controllers .....adresář se soubory AngularJS controllerů
│       └─ services .....adresář se službami a další podpůrnou logikou
├─ styles .....CSS styly uživatelského rozhraní
├─ templates .....podstránky jednotlivých modulů
│   └─ #module_1# .....adresář s jedním konkrétním modulem
│       └─ #module_1#.html .....hlavní stránka modulu
│       └─ submodules .....adresář s podmoduly
│           └─ #submodule_1#.html .....první podmodul
│           └─ ...
└─ ...
```

## 4.5 Dokumentace

Dokumentace zdrojového kódu platformy je psána formou anotací. Z těchto anotací je poté vygenerována dokumentace pomocí nástroje JSDoc.

Další části dokumentace jako požadavky na systém, postup instalace a základní uživatelská příručka pro použití z příkazové řádky, jsou součástí této práce a jsou uvedeny v dalších odstavcích.

Použití platformy z grafického uživatelského rozhraní je triviální a uživatelskou příručku nevyžaduje.

## 4.6 Požadavky a instalace

### 4.6.1 Požadavky na systém

Pro správný chod platformy musí systém splňovat určité požadavky, jejíž kompletní seznam je obsahem této sekce.

- Softwarové požadavky
  - Nainstalovaná platforma Node.JS, včetně registrace umístění do odpovídající systémové proměnné. Ověřit lze zadáním příkazu `node -help` z příkazové řádky.
  - Pro instalaci samotné platformy (pokud není stažen kompletní balíček), případně pro stažení modulů z GIT repozitáře je nutný nainstalovaný GIT a funkční příkaz `git` (umístění binárního souboru zaregistrováno v `PATH` proměnné systému)
- Hardwarové požadavky
  - Pro instalaci platformy a přidávání modulů do jednotlivých projektů je zapotřebí připojení zařízení k internetu.

### 4.6.2 Instalace

Jsou 2 možnosti pro zprovoznění platformy.

- Zkopírování kompletního balíčku, který se nemusí instalovat, ale je mnohem větší než distribuční balíček.
- Zkopírování distribučního balíčku, stažení a instalace potřebných modulů příkazem `node install` v kořenovém adresáři platformy.

## 4.7 Použití

Platforma se dá použít z příkazové řádky, případně z grafického uživatelského rozhraní (mimo vlastní instalace). Příkazy příkazové řádky je vždy nutno provádět v kořenovém adresáři platformy.

### 4.7.1 Příkazy

`node install`

Instalace samotné platformy. Skládá se ze stažení potřebných modulů z internetu a jejich inicializace.

`npm start`

Spuštění grafického rozhraní platformy, přes které lze provádět úkony odpovídající následujícím příkazům.

`node create #project_path#`

Vytvoření nového projektu se zadaným názvem.

`node add #project_path# #module_name_1# [#module_name_2#] [...]`

Přidání modulů do souboru `project_module.json` a jejich označení pro budoucí stažení.

`node remove #project_path# #module_name_1# [#module_name_2#] [...]`

Odstranění modulu z adresáře projektu a označení tohoto modulu za nepotřebný v souboru `project_module.json`.

`node reload #project_path#`

Stažení a inicializace modulů, které jsou označeny v souboru `project_module.json` do adresáře projektu a smazání nepotřebných modulů ze souborového systému.

### 4.7.2 Dostupné moduly

#### **Kendo UI Core**

Mobilní hybridní framework od společnosti Telerik, popsáný v předchozí části práce.

#### **AngularJS**

JavaScriptový framework vyvíjený Googlem [112].

#### **jQuery**

Populární knihovna pro dynamické změny struktury, vzhledu a chování webových stránek [113].

#### **Bootstrap**

Framework pro tvorbu responzivního webu [114].

#### **RequireJS**

Knihovna pro dynamické načítání JS souborů a dalších modulů [115].

#### **CoffeeScript**

Programovací jazyk, ze kterého se vygeneruje JavaScript [116].

#### **JShint**

Nástroj pro detekci chyb a potenciálních problémů v JavaScriptovém kódu [117].

#### **Karma**

Testovací prostředí [118].

#### **Less**

Preprocesor rozšiřující jazyk CSS [119].

**Sass**

Rozšíření pro jazyk CSS [120].

**Uglify-JS**

Nástroj pro automatizovanou kompresi aplikace [121].

**Grunt**

Automatizace úloh při vývoji [122].

## 4.8 Rozšíření a modifikace

### 4.8.1 Přidání modulu

Platforma je snadno rozšiřitelná o další moduly. Záznam o modulu stačí přidat do souboru `available_modules.json`. Záznam je objekt ve formátu JSON s následující strukturou.

**name**

Název modulu, který je použit pro vytvoření odpovídajícího adresáře a pro stažení modulu z internetu.

**type**

Typ repozitáře. Pokud nebudou přítomna další rozšíření, jsou podporovány pouze typy `npm` a `git`.

**address**

Adresa pro stažení modulu typu `git`.

**postInstall**<sup>8</sup>

Pole příkazů, které budou postupně provedeny po stažení modulu z internetu.

Příkazy jsou ve formě textu a budou spuštěny jako samostatné procesy. Pokud je v příkazu uveden výraz `#path#`, bude před provedením příkazu nahrazen cestou ke kořenovému adresáři odpovídajícího modulu.

**submodules**<sup>8</sup>

Pole samostatně použitelných podmodulů.

### 4.8.2 Přidání repozitáře

Další možností rozšíření je přidání dalšího typu repozitáře. To je možné vytvořením souboru `#název_repozitáře#Installer.js` v adresáři `.../scripts/installers`. Tento soubor musí obsahovat veřejnou (přiřazenou v NodeJS objektu `module.exports`) metodu `install`. Metoda `install` má následující argumenty.

---

<sup>8</sup>Nepovinný parametr.

### **projectPath**

cesta k projektu

### **module**

objekt s informacemi o instalovaném modulu

### **void callback(bool success)**

funkce volaná po úspěšné instalaci s argumentem `true`, případně `false` při chybě.

## 4.9 Vzorová aplikace

Vzorová aplikace má dva hlavní účely. Jedním z nich je demonstrace použití platformy. Druhým poté usnadnění práce vývojářů mobilních aplikací.

Projekt vzorové aplikace byl vytvořen s pomocí platformy, která také spravuje moduly aplikace. Základem aplikace jsou frameworky AngularJS a Kendo UI Core, které jsou spolu kompatibilní. Pro zabalení aplikace je použit nástroj Cordova.

Aplikace se skládá z velkého množství podstránek. Každá podstránka zobrazuje jednu nebo více komponent frameworku Kendo UI Core v prostředí frameworku AngularJS. Mezi těmito podstránkami se přechází pomocí postranního menu.

Spolu s dokumentací může aplikace sloužit jako šablona, která se bude postupně upravovat při vývoji prototypu konkrétní mobilní aplikace. Pro vygenerování dokumentace ze zdrojového kódu byl stejně jako u platformy použit nástroj JSDoc.

Pokud z nějakého důvodu nebude tato aplikace použita jako základ koncové aplikace, může vývojář použít alespoň části kódu.

Tato aplikace může také sloužit k demonstraci schopností frameworku Kendo UI, nebo obecně možností hybridního vývoje mobilních aplikací. V tomto případě může usnadnit rozhodnutí, zda při vývoji použít tento přístup, nebo se vydat jinou cestou.

## 4.10 Postup při tvorbě mobilní aplikace

Nejdříve se musí zvolit přístup k vývoji mobilní aplikace. Pokud bude zvolen nativní, nebo multiplatformní kompilovaný přístup, je nutno zvolit specifický postup.

Pro hybridní a webové mobilní aplikace lze tento postup zobecnit díky použití vytvořené platformy.

1. Zkopírování/instalace platformy (`node install`)
2. Založení projektu (`node create #path#`)
3. Výběr a přidání komponent (`node add #path# #module1# [#module2#]`  
...)
4. Stažení a inicializace komponent (`node #path# reload`)
5. Zvolení struktury aplikace
6. Použití vyhovujícího kódu z GUI platformy a/nebo ze vzorové aplikace
7. Implementace specifických částí a úprava existujícího kódu
8. Sestavení aplikace pro použití ve webovém prohlížeči nebo přímo v zařízení





---

## Závěr

Využití multiplatformních mobilních frameworků je jedním ze způsobů vývoje mobilních aplikací. Tyto frameworky se neustále vyvíjejí a jejich použitelnost stoupá. Z těchto důvodů by bylo žádoucí na ně přihlídnout při analyzování vhodného způsobu vývoje mobilní aplikace.

Cílem první části práce bylo demonstrovat možnosti těchto frameworků a porovnat je tak, aby bylo možné vybrat konkrétní framework pro tvorbu aplikace se specifickými požadavky. Tyto frameworky se liší použitými technologiemi, podporovanými platformami, obsaženými komponenty, licenčními podmínkami i rychlostí a vzhledem výsledné aplikace. Při výběru konkrétního frameworku je tedy nutné zvážit veškeré jeho vlastnosti.

Navazující část práce měla za cíl vytvořit platformu, která usnadní a urychlí počáteční fázi vývoje mobilní aplikace. Platforma umožňuje vybrat a spravovat různé komponenty a integrovat je do projektu. Jednou z komponent je i framework, který podle zadaných kritérií vyšel ze srovnání nejlépe.

Práce může být v budoucnu rozšířena o další frameworky. Případně může sloužit jako výchozí bod pro určení směru a rychlosti vývoje těchto frameworků.



---

## Literatura

- [1] KLAIS, Brian. Research: How Many Apps Are in Each App Store?. *Pure Oxygen Labs* [online]. 5.3.2013, 25.9.2014 [cit. 2015-03-28]. Dostupné z: <http://pureoxygenlabs.com/how-many-apps-in-each-app-store/>
- [2] VISWANATHAN, Priya. What is a Mobile Application?. *About Tech: Explore more in tech* [online]. ©2015 [cit. 2015-05-05]. Dostupné z: <http://mobiledevices.about.com/od/glossary/g/What-Is-A-Mobile-Application.htm>
- [3] HOLZER, Adrian a Jan ONDRUS. Mobile application market: A developer's perspective. *Telematics and Informatics* [online]. 2011, vol. 28, issue 1, s. 22-31 [cit. 2015-03-28]. DOI: 10.1016/j.tele.2010.05.006. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0736585310000377>
- [4] COWART, Jim. When to Go Native, Mobile Web or Cross-Platform/Hybrid. *Tech.pro* [online]. 2014 [cit. 2015-03-28]. Dostupné z: <http://tech.pro/blog/1355/when-to-go-native-mobile-web-or-cross-platformhybrid>
- [5] Why should I use a framework?. SENSIOLABS. *Symfony* [online]. ©2014 [cit. 2015-01-12]. Dostupné z: <http://symfony.com/why-use-a-framework>
- [6] CLIFTON, Marc. Code of practice for project management for construction and development. *Code Project: For those who code* [online]. 3.11.2003 [cit. 2015-03-28]. Dostupné z: <http://www.codeproject.com/Articles/5381/What-Is-A-Framework>
- [7] Platform Definition. *The Linux Information Project* [online]. 19.1.2005, 9.1.2007 [cit. 2015-03-30]. Dostupné z: <http://www.linfo.org/platform.html>

- [8] JANSSEN, Cory. Platform. In: *Techopedia* [online]. ©2010-2015 [cit. 2015-03-28]. Dostupné z: <http://www.techopedia.com/definition/3411/platform>
- [9] JAMISON, Scott, Susan HANLEY a Chris BORTLIK. *Essential Share-Point 2013: practical guidance for meaningful business results, free sample chapter*. United States of America: Addison-Wesley, 2014, 745 pages. ISBN 03-218-8411-6. Dostupné z: <http://ptgmedia.pearsoncmg.com/images/9780321884114/samplepages/0321884116.pdf>
- [10] Smartphone OS Market Share, Q4 2014. INTERNATIONAL DATA CORPORATION. *IDC: Analyze the future* [online]. ©2015 [cit. 2015-03-30]. Dostupné z: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- [11] HTML & CSS: What is HTML?. WORLD WIDE WEB CONSORCIUM. *W3C* [online]. ©2015 [cit. 2015-03-30]. Dostupné z: <http://www.w3.org/standards/webdesign/htmlcss#whathtml>
- [12] HAZAËL-MASSIEUX, Dominique. JAVASCRIPT WEB APIS. WORLD WIDE WEB CONSORCIUM. *W3C* [online]. ©2015 [cit. 2015-03-30]. Dostupné z: <http://www.w3.org/standards/webdesign/script.html>
- [13] HTML5: A vocabulary and associated APIs for HTML and XHTML. WORLD WIDE WEB CONSORCIUM. *W3C* [online]. 28.10.2014 [cit. 2015-03-30]. Dostupné z: <http://www.w3.org/TR/html5/syntax.html>
- [14] HTML/Training/Tag syntax. WOLD WIDE WEB CONSORCIUM. *W3C* [online]. 2011 [cit. 2015-03-30]. Dostupné z: [http://www.w3.org/community/webdev/wiki/HTML/Training/Tag\\_syntax](http://www.w3.org/community/webdev/wiki/HTML/Training/Tag_syntax)
- [15] BRIGHT, Peter. HTML5 specification finalized, squabbling over specs continues: It only took 15 years. *Ars Technica* [online]. 29.11. 2014 [cit. 2015-03-28]. Dostupné z: <http://arstechnica.com/information-technology/2014/10/html5-specification-finalized-squabbling-over-who-writes-the-specs-continues/>
- [16] IRISH, Paul, Divya MANIAN a YAKURA. *HTML5 & CSS3 READINESS* [online]. [2009-2013] [cit. 2015-03-28]. Dostupné z: <http://html5readiness.com/>
- [17] HTML5 Introduction. REFSNES DATA. *W3schools.com* [online]. ©1999-2015 [cit. 2015-03-30]. Dostupné z: [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

- 
- [18] BIDELMAN, Eric. Custom Elements: defining new elements in HTML. GOOGLE. *HTML5 Rocks* [online]. 28.8.2013, 18.12.2013 [cit. 2015-03-30]. Dostupné z: <http://www.html5rocks.com/en/tutorials/webcomponents/customelements/>
- [19] BEWICK, Chris. HTML5 Custom Data Attributes (data-\*). *Html5 Doctor: Helping you implement HTML5 today* [online]. 27.5.2010 [cit. 2015-03-30]. Dostupné z: <http://html5doctor.com/html5-custom-data-attributes/>
- [20] BOS, Bert. CSS SPECIFICATION. WORLD WIDE WEB CONSORTIUM. *W3C* [online]. ©1994-2015, 26.3.2015 [cit. 2015-03-30]. Dostupné z: <http://www.w3.org/Style/CSS/current-work>
- [21] AJAX Introduction. REFSNES DATA. *W3schools.com* [online]. ©1999-2015 [cit. 2015-03-28]. Dostupné z: [http://www.w3schools.com/ajax/ajax\\_intro.asp](http://www.w3schools.com/ajax/ajax_intro.asp)
- [22] MURRAY, Greg. Asynchronous JavaScript Technology and XML (Ajax) With the Java Platform. *Oracle* [online]. 9.6.2005, 2006 [cit. 2015-03-28]. Dostupné z: <http://www.oracle.com/technetwork/articles/javaee/ajax-135201.html>
- [23] Apache Cordova Documentation: Overview. THE APACHE SOFTWARE FOUNDATION. *Apache Cordova* [online]. [©2012-2013] [cit. 2015-03-30]. Dostupné z: [http://cordova.apache.org/docs/en/4.0.0/guide\\_overview\\_index.md.html](http://cordova.apache.org/docs/en/4.0.0/guide_overview_index.md.html)
- [24] CORRAL, Luis, Alberto SILLITTI a Giancarlo SUCCI. Mobile Multiplatform Development: An Experiment for Performance Analysis. *Procedia Computer Science* [online]. 2012, vol. 10, s. 736-743 [cit. 2015-03-28]. DOI: 10.1016/j.procs.2012.06.094. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1877050912004516>
- [25] FUCHS, Thomas. *Zepto.JS* [online]. ©2010-2014 [cit. 2015-03-04]. Dostupné z: <http://zeptojs.com>
- [26] M-WAY SOLUTIONS GMBH. *The M Project: Create cross-platform mobile web apps based on Backbone.js that look stunning native* [online]. ©2011-2013 [cit. 2015-02-21]. Dostupné z: <http://www.the-m-project.org/>
- [27] SENCHA INC. *Sencha: Rapidly design, develop, and manage cross-platform web applications* [online]. ©2015 [cit. 2015-02-05]. Dostupné z: <http://www.sencha.com/>

- [28] *Ratchet: Build mobile apps with simple HTML, CSS, and JS components*. Build mobile apps with simple HTML, CSS, and JS components. [online]. [2013] [cit. 2015-05-10]. Dostupné z: <http://goratchet.com/>
- [29] DEVELOPER EXPRESS INC. *DevExtreme: Cross-platform HTML 5/JS tools for next-generation mobile and web development* [online]. ©2011-2015 [cit. 2015-02-15]. Dostupné z: <http://js.devexpress.com/>
- [30] SAP SE. *OpenUI5: Code once. Run on any device.* [online]. [2014] [cit. 2015-02-13]. Dostupné z: <http://openui5.org/>
- [31] THE JQUERY FOUNDATION. *JQuery Mobile: A Touch-Optimized Web Framework* [online]. ©2015 [cit. 2015-02-10]. Dostupné z: <http://jquerymobile.com/>
- [32] IDANGERO.US. *Framework7: Full Featured HTML Framework For Building iOS Apps* [online]. ©2015 [cit. 2015-02-10]. Dostupné z: <http://www.idangero.us/framework7>
- [33] A GLOBO COMPANY. *ChocolateChip-UI: Create Mobile Apps with Native Look & Feel For Web or Hybrid* [online]. ©2015 [cit. 2015-02-10]. Dostupné z: <http://chocolatechip-ui.com/>
- [34] *Impact: The Awesomest way To Create Even More Awesome HTML5 Games!* [online]. ©2015 [cit. 2015-02-10]. Dostupné z: <http://impactjs.com/>
- [35] FAMOUS INDUSTRIES, Inc. *Famo.us: The Ultimate Web Platform for Developers and Designers* [online]. ©2014 [cit. 2015-02-10]. Dostupné z: <https://famo.us/>
- [36] DRIFTY CO. *Ionic: Create amazing apps* [online]. ©2013-2015 [cit. 2015-02-10]. Dostupné z: <http://ionicframework.com/>
- [37] BALMER, JR., Dave. *Jo: The simple open source app framework for HTML5* [online]. ©2010-2013 [cit. 2015-02-10]. Dostupné z: <http://joapp.com/>
- [38] ASIAL CORPORATION. *Onsen UI: The Answer to Cordova UI Development* [online]. ©2013-2015 [cit. 2015-05-06]. Dostupné z: <http://onsen.io/>
- [39] TELERIK. *Kendo UI: Everything for building web and mobile apps with HTML5 and JavaScript* [online]. ©2002-2015 [cit. 2015-04-08]. Dostupné z: <http://www.telerik.com/kendo-ui>
- [40] MARMALADE TECHNOLOGIES LTD. *Marmelade: Native performance. Any device.* [online]. ©2015 [cit. 2015-03-10]. Dostupné z: <https://www.madewithmarmalade.com/>

- 
- [41] MOSYNC AB. *MoSync: App development made easy* [online]. ©2004-2013 [cit. 2015-05-10]. Dostupné z: <http://www.mosync.com/>
- [42] ZEBRA TECHNOLOGIES, Inc. *RhoElements: Write Native Enterprise Mobile Apps with Web Skills* [online]. ©2015 [cit. 2015-03-11]. Dostupné z: <http://rhomobile.com/products/rhoelements/>
- [43] HIPBYTE. *RubyMotion: Write cross-platform apps for iOS, Android and OS X in Ruby* [online]. ©2012-2015 [cit. 2015-03-06]. Dostupné z: <http://www.rubymotion.com/>
- [44] APPCELERATOR INC. *Appcelerator: Build great mobile experiences faster* [online]. ©2008-2015 [cit. 2015-03-08]. Dostupné z: <http://www.appcelerator.com/>
- [45] *Trigger.io: The simplest way to build amazing mobile apps* [online]. [2011] [cit. 2015-05-10]. Dostupné z: <https://trigger.io/>
- [46] XAMARIN INC. *Xamarin* [online]. ©2015 [cit. 2015-05-08]. Dostupné z: <https://xamarin.com/>
- [47] GRISOGONO, Grgur. 5 Best Mobile Web App Frameworks: Sencha Touch. *Modus Create* [online]. 3.3. 2014 [cit. 2015-05-06]. Dostupné z: <http://moduscreate.com/5-best-mobile-web-app-frameworks-sencha-touch/>
- [48] GAIĆ, Dragan. Top 7 Mobile application HTML5 frameworks. *Gajotres* [online]. 28.2.2015 [cit. 2015-03-28]. Dostupné z: <http://www.gajotres.net/top-7-mobile-application-html5-frameworks/>
- [49] Titanium Platform Overview. APPCELERATOR INC. *Appcelerator Documentation* [online]. ©2008-2015 [cit. 2015-03-30]. Dostupné z: [http://docs.appcelerator.com/titanium/latest/#!/guide/Titanium\\_Platform\\_Overview](http://docs.appcelerator.com/titanium/latest/#!/guide/Titanium_Platform_Overview)
- [50] Appcelerator Alloy: Jumpstart and accelerate the development of quality applications. APPCELERATOR INC. *The Appcelerator Platform* [online]. ©2008-2015 [cit. 2015-03-30]. Dostupné z: <http://www.appcelerator.com/platform/alloy/>
- [51] Alloy Styles and Themes. APPCELERATOR INC. *Appcelerator Documentation* [online]. ©2008-2015 [cit. 2015-03-30]. Dostupné z: [http://docs.appcelerator.com/titanium/latest/#!/guide/Alloy\\_Styles\\_and\\_Themes](http://docs.appcelerator.com/titanium/latest/#!/guide/Alloy_Styles_and_Themes)
- [52] Titanium Mobile Development Environment: Create bold, beautiful and transformative mobile experiences. APPCELERATOR INC. *The Appcelerator Platform* [online]. ©2008-2015 [cit. 2015-03-30]. Dostupné z: <http://www.appcelerator.com/titanium/>

- [53] POWELL, Jason. What Titanium Appcelerator REALLY is and How it Works!. *Forum One* [online]. 6.7.2012 [cit. 2015-03-30]. Dostupné z: <http://forumone.com/insights/what-titanium-appcelerator-really-and-how-it-works/>
- [54] TRAEGER, Peter. Four Ways To Build A Mobile Application, Part 4: Appcelerator Titanium. *Smashing Magazine* [online]. 10.3.2014 [cit. 2015-03-30]. Dostupné z: <http://www.smashingmagazine.com/2014/03/10/4-ways-build-mobile-application-part4-appcelerator-titanium/>
- [55] Alloy Concepts: Platform-Specific Resources. APPCELERATOR INC. *Appcelerator Documentation* [online]. ©2008-2015 [cit. 2015-03-30]. Dostupné z: [http://docs.appcelerator.com/titanium/3.0/#!/guide/Alloy\\_Concepts-section-34636240\\_AlloyConcepts-Platform-SpecificResources](http://docs.appcelerator.com/titanium/3.0/#!/guide/Alloy_Concepts-section-34636240_AlloyConcepts-Platform-SpecificResources)
- [56] Alloy Concepts. APPCELERATOR INC. *Appcelerator Documentation* [online]. ©2008-2015 [cit. 2015-03-30]. Dostupné z: [http://docs.appcelerator.com/titanium/3.0/#!/guide/Alloy\\_Concepts](http://docs.appcelerator.com/titanium/3.0/#!/guide/Alloy_Concepts)
- [57] Alloy Controllers: Conditional Code. APPCELERATOR INC. *Appcelerator Documentation* [online]. ©2008-2015 [cit. 2015-03-30]. Dostupné z: [http://docs.appcelerator.com/titanium/3.0/#!/guide/Alloy\\_Controllers-section-34636384\\_AlloyControllers-ConditionalCode](http://docs.appcelerator.com/titanium/3.0/#!/guide/Alloy_Controllers-section-34636384_AlloyControllers-ConditionalCode)
- [58] Overview: CSS/SASS. DRIFTY CO. *Ionic* [online]. ©2014 [cit. 2015-03-31]. Dostupné z: <http://ionicframework.com/docs/overview/#css-sass>
- [59] Ionic Documentation Overview: Browser Support. DRIFTY CO. *Ionic: Create amazing apps* [online]. ©2013-2015 [cit. 2015-05-05]. Dostupné z: <http://ionicframework.com/docs/overview/#browser-support>
- [60] Ionic Documentation Overview: Licence. DRIFTY CO. *Ionic: Create amazing apps* [online]. ©2013-2015 [cit. 2015-05-05]. Dostupné z: <http://ionicframework.com/docs/overview/#license>
- [61] Welcome to Ionic: How to get the most out of Ionic. DRIFTY CO. *Ionic* [online]. ©2014 [cit. 2015-03-31]. Dostupné z: <http://ionicframework.com/docs/guide/preface.html>
- [62] Conceptual Overview. GOOGLE. *AngularJS: Developer Guide* [online]. ©2010-2015 [cit. 2015-03-31]. Dostupné z: <https://docs.angularjs.org/guide/concepts>



- 
- [63] RUEBBELKE, Lukas. AngularJS Application Architecture. *Egghead.io* [online]. [cit. 2015-05-05]. Dostupné z: <https://egghead.io/series/angularjs-application-architecture>
- [64] Templates. GOOGLE. *AngularJS: Developer Guide* [online]. ©2010-2015 [cit. 2015-03-31]. Dostupné z: <https://docs.angularjs.org/guide/templates>
- [65] Creating Custom Directives. GOOGLE. *AngularJS: Developer Guide* [online]. ©2010-2015 [cit. 2015-03-31]. Dostupné z: <https://docs.angularjs.org/guide/directive>
- [66] Dependency Injection. GOOGLE. *AngularJS: Developer Guide* [online]. ©2010-2015 [cit. 2015-03-31]. Dostupné z: <https://docs.angularjs.org/guide/di>
- [67] Ionic CLI: The powerful command line utility. DRIFTY CO. *Ionic* [online]. ©2014 [cit. 2015-03-31]. Dostupné z: <http://ionicframework.com/docs/cli/>
- [68] JavaScript: AngularJS Extensions. DRIFTY CO. *Ionic: Create amazing apps* [online]. ©2013-2015 [cit. 2015-05-05]. Dostupné z: <http://ionicframework.com/docs/api/>
- [69] Ionic Concepts: The bigger picture of an Ionic App. DRIFTY CO. *Ionic: Create amazing apps* [online]. ©2013-2015 [cit. 2015-05-05]. Dostupné z: <http://ionicframework.com/docs/concepts/structure.html>
- [70] Javascript: Extend Ionic even further with the power of AngularJS. DRIFTY CO. *Ionic: Create amazing apps* [online]. ©2013-2015 [cit. 2015-05-05]. Dostupné z: <http://ionicframework.com/docs/api/>
- [71] BRADLEY, Adam. Where does the Ionic Framework fit in?. DRIFTY CO. *The Official Ionic Blog: The next generation HTML5 hybrid app development SDK* [online]. 28.10. 2013 [cit. 2015-05-05]. Dostupné z: <http://blog.ionic.io/where-does-the-ionic-framework-fit-in/>
- [72] Sass Basics. CATLIN, Hampton, WEIZENBAUM a Chris EPPSTEIN. *SASS: CSS with superpowers* [online]. ©2006-2015 [cit. 2015-03-31]. Dostupné z: <http://sass-lang.com/guide>
- [73] Sass (Syntactically Awesome StyleSheets). CATLIN, Hampton, Natalie WEIZENBAUM a Chris EPPSTEIN. *SASS: CSS with superpowers* [online]. 24.3.2015 [cit. 2015-03-31]. Dostupné z: [http://sass-lang.com/documentation/file.SASS\\_REFERENCE.html](http://sass-lang.com/documentation/file.SASS_REFERENCE.html)
- [74] DRIFTY CO. *Ionic Creator: Build better Ionic apps, faster.* [online]. [2014-12-15] [cit. 2015-05-05]. Dostupné z: <http://creator.ionic.io/>

- [75] Ionic.Platform: Utility in module ionic. DRIFTY CO. *Ionic: Create amazing apps* [online]. ©2013-2015 [cit. 2015-05-05]. Dostupné z: <http://ionicframework.com/docs/api/utility/ionic.Platform/>
- [76] HARTINGTON, Mike. How to use platform-ios and platform-cordova css classes?. In: *Ionic: Forum* [online]. 20.6. 2014 [cit. 2015-05-05]. Dostupné z: <http://forum.ionicframework.com/t/how-to-use-platform-ios-and-platform-cordova-css-classes/5898/2>
- [77] Ionic Framework Examples: See it in action! Checkout our examples or submit your own. DRIFTY CO. *Ionic: Create amazing apps* [online]. ©2013-2015 [cit. 2015-05-05]. Dostupné z: <http://ionicframework.com/examples/>
- [78] BAI, Giulio. *JQuery mobile first look: discover the endless possibilities offered by jquery mobile for rapid mobile web development* [online]. Birmingham, U.K.: Packt Pub., 2011, v, 198 p. [cit. 2015-03-28]. Community experience distilled. ISBN 9781849515917. Dostupné z: <http://site.ebrary.com.ezproxy.techlib.cz/lib/techlib/detail.action?docID=10482257>
- [79] JQuery Mobile 1.4 Browser Support. THE JQUERY FOUNDATION. *JQuery Mobile* [online]. ©2015 [cit. 2015-03-30]. Dostupné z: <http://jquerymobile.com/browser-support/1.4/>
- [80] Classes: Style Classes. THE JQUERY FOUNDATION. *JQuery Mobile* [online]. ©2015 [cit. 2015-03-31]. Dostupné z: <http://api.jquerymobile.com/classes/>
- [81] Grid Layout. THE JQUERY FOUNDATION. *JQuery Mobile* [online]. ©2015 [cit. 2015-03-31]. Dostupné z: <http://api.jquerymobile.com/grid-layout/>
- [82] Theme: Theming Concepts. THE JQUERY FOUNDATION. *JQuery Mobile* [online]. ©2002 [cit. 2015-03-31]. Dostupné z: <http://api.jquerymobile.com/theme/>
- [83] Demos. THE JQUERY FOUNDATION. *JQuery Mobile* [online]. ©2014 [cit. 2015-03-31]. Dostupné z: <http://demos.jquerymobile.com/1.4.5/>
- [84] Data Attributes: Data attribute reference. THE JQUERY FOUNDATION. *JQuery Mobile* [online]. ©2015 [cit. 2015-03-31]. Dostupné z: <http://api.jquerymobile.com/data-attribute/>
- [85] Introduction. THE JQUERY FOUNDATION. *JQuery Mobile* [online]. ©2014 [cit. 2015-03-31]. Dostupné z: <http://demos.jquerymobile.com/1.4.5/intro/>

- 
- [86] Touch Guides: Sencha Touch. SENCHA INC. *Sencha* [online]. 25.3. 2015 [cit. 2015-05-06]. Dostupné z: <http://docs.sencha.com/touch/2.4/index.html>
- [87] Sencha Ext JS: Buy Ext JS 5 Licenses, Maintenance & Support. SENCHA INC. *Sencha* [online]. ©2015 [cit. 2015-05-06]. Dostupné z: <https://www.sencha.com/store/extjs>
- [88] Sencha Licensing: Sencha Touch. SENCHA INC. *Sencha* [online]. ©2015 [cit. 2015-05-06]. Dostupné z: [http://www.sencha.com/legal/#Sencha\\_Touch](http://www.sencha.com/legal/#Sencha_Touch)
- [89] Intro to Applications with Sencha Touch. SENCHA INC. *Sencha* [online]. 25.3. 2015 [cit. 2015-05-06]. Dostupné z: [http://docs.sencha.com/touch/2.4/core\\_concepts/about\\_applications.html](http://docs.sencha.com/touch/2.4/core_concepts/about_applications.html)
- [90] Using Views in Your Applications. SENCHA INC. *Sencha* [online]. 25.3. 2015 [cit. 2015-05-06]. Dostupné z: <http://docs.sencha.com/touch/2.4/components/views.html>
- [91] Sencha Touch: Build Mobile Web Apps with HTML5. SENCHA INC. *Sencha* [online]. ©2015 [cit. 2015-02-06]. Dostupné z: <http://www.sencha.com/products/touch/features/>
- [92] Touch Guides: Theming in Sencha Touch. SENCHA INC. *Sencha* [online]. 25.3. 2015 [cit. 2015-05-06]. Dostupné z: <http://docs.sencha.com/touch/2.4/components/theming.html>
- [93] Sencha Cmd: Optimize Sencha Ext JS and Sencha Touch apps using Sencha Cmd. SENCHA INC. *Sencha* [online]. ©2015 [cit. 2015-02-06]. Dostupné z: <http://www.sencha.com/products/sencha-cmd/#overview>
- [94] Sencha Touch Bundle: Create mobile applications for any platform using a single source. SENCHA INC. *Sencha* [online]. ©2015 [cit. 2015-02-06]. Dostupné z: <http://www.sencha.com/products/touch-bundle/#overview>
- [95] Kendo UI Widgets and Frameworks. TELERIK. *Kendo UI: Everything for building web and mobile apps with HTML5 and JavaScript* [online]. ©2002-2015 [cit. 2015-05-06]. Dostupné z: <http://www.telerik.com/kendo-ui#more-widgets>
- [96] TELERIK. *Telerik Kendo UI: Demos and sample applications* [online]. ©2002-2015 [cit. 2015-05-06]. Dostupné z: <http://demos.telerik.com/kendo-ui/>

- [97] Kendo UI Templates Overview. TELERIK. *Telerik Kendo UI: Documentation and API reference* [online]. ©2002-2015 [cit. 2015-05-06]. Dostupné z: <http://docs.telerik.com/kendo-ui/framework/templates/overview>
- [98] Application. TELERIK. *Telerik Kendo UI: Documentation and API reference* [online]. ©2002-2015 [cit. 2015-05-08]. Dostupné z: <http://docs.telerik.com/kendo-ui/mobile/application>
- [99] CSS Components. ASIAL CORPORATION. *Onsen UI: The Answer to Cordova UI Development* [online]. ©2013-2015 [cit. 2015-05-08]. Dostupné z: <http://onsen.io/reference/css.html>
- [100] JavaScript. ASIAL CORPORATION. *Onsen UI: The Answer to Cordova UI Development* [online]. ©2013-2015 [cit. 2015-05-08]. Dostupné z: <http://onsen.io/reference/javascript.html>
- [101] ASIAL CORPORATION. *Onsen UI: Theme Roller* [online]. ©2013-2015 [cit. 2015-05-08]. Dostupné z: <http://components.onsen.io/>
- [102] Onsen UI Guide: Loading Onsen UI in Your Project. ASIAL CORPORATION. *Onsen UI: The Answer to Cordova UI Development* [online]. ©2013-2015 [cit. 2015-05-08]. Dostupné z: <http://onsen.io/guide/overview.html#LoadingOnsenUIinYourProject>
- [103] About Mono. *Mono: Cross platform, open source .NET framework* [online]. ©2015 [cit. 2015-05-06]. Dostupné z: <http://www.monoproject.com/docs/about-mono/>
- [104] Simple and transparent pricing. XAMARIN INC. *Xamarin* [online]. ©2015 [cit. 2015-05-08]. Dostupné z: <https://store.xamarin.com/>
- [105] XAMARIN INC. *Xamarin: Create native iOS, Android, Mac and Windows apps in C#*. [online]. ©2015 [cit. 2015-05-06]. Dostupné z: <https://xamarin.com/platform>
- [106] Xamarin.Forms: Build native UIs for iOS, Android and Windows from a single, shared C# codebase. XAMARIN INC. *Xamarin* [online]. ©2015 [cit. 2015-05-06]. Dostupné z: <https://xamarin.com/forms>
- [107] Frequently Asked Questions: Xamarin Platform. XAMARIN INC. *Xamarin* [online]. ©2015 [cit. 2015-05-06]. Dostupné z: <http://xamarin.com/faq>
- [108] Ionic: Contributors. GITHUB, Inc. *GitHub* [online]. ©2015 [cit. 2015-05-06]. Dostupné z: <https://github.com/driftco/ionic/graphs/contributors>

- 
- [109] Onsen UI: Contributors. GITHUB, Inc. *GitHub* [online]. ©2015 [cit. 2015-05-06]. Dostupné z: <https://github.com/OnsenUI/OnsenUI/graphs/contributors>
- [110] NEŠETRIL, Jakub. JavaScript na serveru: Začínáme s Node.js. *Zdroják.cz* [online]. 23.11.2010 [cit. 2015-05-08]. Dostupné z: <http://www.zdrojak.cz/clanky/javascript-na-serveru-zaciname-s-node-js/>
- [111] WANG, Roger et al. Node-webkit is renamed NW.js. *GitHub* [online]. 2003, 24.4.2015 [cit. 2015-05-08]. Dostupné z: <https://github.com/nwjs/nw.js/blob/master/README.md>
- [112] GOOGLE. *AngularJS: HTML enhanced for web apps!* [online]. ©2010-2015 [cit. 2015-05-09]. Dostupné z: <https://angularjs.org/>
- [113] THE JQUERY FOUNDATION. *JQuery: write less, do more.* [online]. ©2015 [cit. 2015-05-09]. Dostupné z: <http://jquery.com/>
- [114] TWITTER INC. *Bootstrap* [online]. [©2011-2015] [cit. 2015-05-09]. Dostupné z: <http://getbootstrap.com/>
- [115] *RequireJS: A JavaScript module loader* [online]. ©2011-2015 [cit. 2015-05-09]. Dostupné z: <http://requirejs.org/>
- [116] *CoffeeScript* [online]. [2009] [cit. 2015-05-09]. Dostupné z: <http://coffeescript.org/>
- [117] *JSHint: About* [online]. [2011] [cit. 2015-05-09]. Dostupné z: <http://jshint.com/about/>
- [118] ZIEGELMAYER, Friedel. *Karma* [online]. [2012] [cit. 2015-05-09]. Dostupné z: <http://karma-runner.github.io/0.12/index.html>
- [119] *Less: Getting started* [online]. [2010] [cit. 2015-05-09]. Dostupné z: <http://lesscss.org/>
- [120] CATLIN, Hampton et al. *SASS: CSS with superpowers* [online]. ©2006-2015 [cit. 2015-05-09]. Dostupné z: <http://sass-lang.com/>
- [121] BAZON, Mihai. *UglifyJS* [online]. ©2012-2014 [cit. 2015-05-09]. Dostupné z: <http://lisperator.net/uglifyjs/>
- [122] *Grunt: The JavaScript Task Runner* [online]. [2013] [cit. 2015-05-09]. Dostupné z: <http://gruntjs.com/>



## Seznam použitých zkratek

**AJAX** Asynchronous JavaScript and XML

**API** Application Programming Interface

**CLI** Command-Line Interface

**CSS** Cascading Style Sheets

**GPS** Global Positioning System

**GUI** Graphical User Interface

**HTML** Hypertext Markup Language

**IDE** Integrated Development Environment

**JSON** JavaScript Object Notation

**JS** JavaScript

**MV\*** Model-View-Whatever

**MVC** Model-View-Controller

**MVP** Model-View-Presenter

**MVVM** Model-View-ViewModel

**NPM** Node Package Manager

**PNG** Portable Network Graphics

**SASS** Syntactically Awesome Style Sheets

**SVG** Scalable Vector Graphics

**TSS** Titanium Style Sheets

## A. SEZNAM POUŽITÝCH ZKRATEK

---

**UI** User Interface

**WP** Windows Phone

**XAML** Extensible Application Markup Language

**XML** Extensible Markup Language



## Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	src	
	thesis.tex.....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
	text .....	text práce
	thesis.pdf .....	text práce ve formátu PDF
	thesis.ps .....	text práce ve formátu PS