

TODO

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

TODO

Bc. Oldřich Malec

Vedoucí práce: Ing. Jiří Hunka

22. ledna 2019

Poděkování

TODO

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl 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. Dále prohlašuji, že jsem s Českým vysokým učení technickým v Praze uzavřel licenční smlouvu o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona. Tato skutečnost nemá vliv na ust. § 47b zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů.

V Praze dne 22. ledna 2019

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2019 Oldřich Malec. 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

MALEC, Oldřich. *TODO*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019. Dostupný také z WWW: <https://gitlab.fit.cvut.cz/malecold/master-thesis>.

Abstrakt

TODO

Klíčová slova TODO

Abstract

TODO

Keywords TODO

Obsah

Úvod	1
1 Analýza požadavků	3
2 Návrh uživatelského rozhraní	5
3 Volba technologie	7
3.1 Frameworky a knihovny	7
4 Implementace	19
5 Testování	21
Závěr	23
Zdroje	25
A Seznam použitých zkratk	29
B Slovník pojmů	31

Seznam tabulek

3.1	Volba frameworku: Datum vydání	8
3.2	Volba frameworku: Zázemí	9
3.3	Volba frameworku: Licence	9
3.4	Volba frameworku: Obtížnost	10
3.5	Volba frameworku: Dokumentace	11
3.6	Volba frameworku: Testování	13
3.7	Volba frameworku: Devtools	13
3.8	Volba frameworku: Počet hvězdiček na GitHubu	14
3.9	Volba frameworku: Počet npm balíků	14
3.10	Volba frameworku: Otázky na Stack Overflow	15
3.11	Volba frameworku: Shoda s firemním stackem	15
3.12	Volba frameworku: Počet vývojářů na LinkedIn	16
3.13	Volba frameworku: Integrace se Sentry	16
3.14	Volba frameworku: Výsledky	16

Úvod

TODO

Analýza požadavků

Návrh uživatelského rozhraní

Volba technologie

Jelikož bude aplikace rozdělena na backend, kterým se zabývá můj kolega Bc. Pavel Kovář, a frontend, který je předmětem této práce, je vhodné věnovat jistou část textu volbě vhodné technologie.

Cílová platforma Aplikace je navrhována s ohledem na hardwarové vybavení skladu, ve kterém bude poprvé nasazována: zdejší skladníci jsou vybaveni mobilními telefony *Zebra TC20*, které disponují OS Android 7.1 a vestavěnou čtečkou čárových kódů. Kromě skladníků by měla být aplikace použitelná také z tabletu či stolního počítače pro účely vedoucích pracovníků. Z důvodu jednoduchosti vývoje, testování, možností aktualizací a obecně dobré zkušenosti z jiných projektů bylo hned při úvodním návrhu určeno, že aplikace bude tvořena formou webové služby, která bude na klientských zařízeních zobrazována ve WebView v jednoduchém kontejneru chovajícím se jako nativní aplikace. Řídící pracovníci budou naopak moci využít přístupu odkudkoliv, kde budou připojeni k internetu pouze pomocí běžného webového prohlížeče. Z tohoto důvodu jsou v následující rešerši zhodnocovány frameworky či knihovny, které usnadňují vývoj *webových aplikací*.

3.1 Frameworky a knihovny

V době psaní této práce patří mezi nejpopulárnější [10] [15] front-endové frameworky či knihovny Angular [3], React [29], Vue.js [37], Ember.js [12] a Backbone.js [7].

Názvosloví Pro účely tohoto textu budu na následujících řádcích používat slovo *framework*, kterým budu označovat jak frameworky, tak knihovny, z důvodu snížení opakování textu.

3.1.1 Datum vydání

Zatímco v současnosti nejčastěji porovnávanými frameworky jsou první dva zmíněné, Vue.js je z této pětky vybraných nejmladší, nabírá ale velké obliby. Ember.js a Backbone.js jsou poté lehce upozaděny z důvodu jejich stáří. Přehled prvního vydání jednotlivých frameworků je v tabulce 3.1

	Angular	React	Vue.js	Ember.js	Backbone.js
Vydání první verze	2010/2016 ^a	2013	2014	2011	2010

Tabulka 3.1: Volba frameworku: Datum vydání

^aV roce 2010 byl vydán AngularJS, který byl v roce 2016 kompletně přepsán do TypeScriptu a vydán jako Angular 2, či jednoduše *Angular*.

Datum vydání ovšem nelze objektivně ohodnotit bodovým ziskem. Na jedné straně stojí fakt, že starší framework může být vyspělejší a tudíž stabilnější atp., na straně druhé nové frameworky se často učí z chyb provedených jejich předchůdci a vyberou z nich pouze to nejlepší. Tato tabulka tedy zůstane čistě přehledová.

3.1.2 Zázemí

Zatímco Angular a React jsou vyvíjeny velkými společnostmi: Googlem, respektive Facebookem, které zná každý, Ember.js je vyvíjen společností Tilde Inc. [34], která také není žádným startupem. Vue.js a Backbone.js by se naopak daly nazvat *komunitními projekty*, neboť jsou vytvořeny převážně jedním autorem (Evan You, respektive Jeremy Ashkenas) a rozvíjeny a udržovány komunitou vývojářů.

Na první pohled by se mohlo zdát, že z tohoto hodnocení budou vycházet lépe ty frameworky, které mají za sebou stabilní firmy, neboť je tím zajištěn jejich kontinuální vývoj. Ve skutečnosti ale velké firmy *zabíjejí* své projekty poměrně často, stačí se podívat například na seznam projektů, které ukončil Google [27].

Oproti tomu komunitní projekty mohou žít dále i v případě, že jejich hlavní autor už na projektu nechce, nebo nemůže pracovat. Z toho důvodu nelze jednoznačně určit, které zázemí je pro budoucnost frameworku výhodnější, a u tabulky 3.2 se tedy opět zdržuji udělování bodů.

	Angular	React	Vue.js	Ember.js	Backbone.js
Zázemí velké společnosti	ano	ano	ne	částečně	ne

Tabulka 3.2: Volba frameworku: Zázemí

3.1.3 Licence

Licence k použití frameworku je důležitá položka při rozhodování. Naštěstí všech 5 porovnávaných frameworků je v době psaní této licencováno pod MIT licenci, která povoluje jakékoliv použití i v komerční sféře, úpravy, distribuce i použití v ne-opensource projektech. Nevýhodou této licence je nulová záruka funkčnosti či zodpovědnost autorů za potenciální spáchané škody tímto softwarem.

Licencování Reactu Facebook původně vydal svůj React pod BSD licenci spolu s dalšími patenty, avšak 24. září 2017 byl React převeden pod MIT licenci [1, 36].

Jelikož jsou všechny frameworky licencovány stejně, neprobíhá v tabulce 3.3 žádné bodování.

	Angular	React	Vue.js	Ember.js	Backbone.js
Licence	MIT	MIT	MIT	MIT	MIT

Tabulka 3.3: Volba frameworku: Licence

3.1.4 Křivka učení

Složitost frameworku je důležitá metrika, neboť má dopady zejména na ekonomickou stránku projektu. Jednoduché prvotní vniknutí do problematiky frameworku ovšem také nemusí být nutně výhodou, pokud v něm je později problémové provést některé pokročilé věci, nebo i v pokročilém stádiu zdržuje svým

nízkoúrovňovým přístupem k problémům, které jiné frameworky řeší automaticky.

Angular, React a Vue.js Přehled obtížnosti tří v současnosti nejčastěji skloňovaných frameworků přehledně shrnul Rajdeep Chandra ve své prezentaci *My experience with Angular 2, React and Vue* [16], ze které vychází hodnocení v tabulce 3.4.

Ember.js Tento framework je dle V. Lascika [25] vhodný spíše pro projekty, na kterých pracuje velké množství vývojářů, a to z důvodu své komplexnosti. Proto jej pro použití v mé vznikající aplikaci hodnotím nula body.

Backbone.js U této knihovny je důležité zmínit, že umožňuje vývojáři vytvořit si strukturu aplikace kompletně dle svého uvážení [2]. To s sebou může nést jak výhody pro zkušeného, tak nevýhody pro nezkušeného vývojáře, který v pokročilém stádiu vývoje může zjistit, že některou ze základních struktur navrhl špatně. Samotná obtížnost práce s touto knihovnou je ale poměrně nízká.

	Angular	React	Vue.js	Ember.js	Backbone.js
Obtížnost	vysoká	vyšší	nízká	velmi vysoká ^a	nízká
bodový zisk	1	2	4	1	4

Tabulka 3.4: Volba frameworku: Obtížnost

^aza předpokladu, že na projektu bude pracovat pouze velmi malé množství vývojářů

3.1.5 Oficiální dokumentace

Hlavním zdrojem ke studiu frameworku by měla být jeho oficiální dokumentace, v této sekci tedy budu hodnotit kvalitu a obsáhlost oficiálního manuálu k jednotlivým frameworkům.

Angular Jedná se o velmi obsáhlou a dobře rozdělenou dokumentaci [5], která obsahuje i řadu příkladů a ve srozumitelné stromové struktuře vývojář jednoduše najde, co potřebuje.

React Dokumentace Reactu [14] je o poznání jednodušší než ta Angularu, avšak to je způsobeno tím, že React je pouze knihovna, kdežto Angular je plnohodnotný framework. Dokumentace je rozdělena na jednodušší úvod a pokročilejší techniky, je tedy snadné s ní pracovat.

Vue.js Nejmladší z frameworků má také velmi přátelskou dokumentaci [19], která je podobně jako u Angularu velmi bohatá a stromově strukturovaná.

Ember.js Oficiální manuál Ember.js [13] je taktéž poměrně obsáhlý a strukturovou připomíná dokumentaci Angularu a Vue.js. Obsahuje velké množství ukázek kódu a je logicky strukturován.

Backbone.js Poslední ze zkoumaných frameworků má oficiální dokumentaci [8] na první pohled méně atraktivní a pro nováčka může být matoucí. Oproti ostatním dokumentacím chybí například barevné zvýraznění důležitých bodů a další grafické strukturování textu.

	Angular	React	Vue.js	Ember.js	Backbone.js
Kvalita oficiální dokumentace	velmi vysoká	vysoká	velmi vysoká	velmi vysoká	střední
<i>bodový zisk</i>	3	2	3	3	1

Tabulka 3.5: Volba frameworku: Dokumentace

3.1.6 Testování

Testování je při vývoji softwaru velkým tématem. Mnoho vývojářů nerado testuje, jiní se naopak specializují pouze na testování. V moderních aplikacích je testování z větší části řešeno automatizovanými testy, které se spouští v rámci CI. Přestože se tato práce zabývá pouze frontendem, i ten je možné testovat již od úrovně Unit testů, vyvíjej jej pomocí TDD a nakonec samozřejmě vše zkontrolovat pomocí E2E testů.

Angular Nejkomplexější z frameworků nabízí přehlednou dokumentaci [4], jak by měl být testován. Popisuje jak používat Unit testy i E2E testy a pro druhý zmíněný typ nabízí i samotný testovací framework: Protractor [28]. Jeho testovatelnost tedy hodnotím jako velmi dobrou.

React Tato knihovna ve své dokumentaci na první pohled testování příliš neprosazuje, avšak stránku dedikovanou této kratochvíli [33] lze nakonec také nalézt. Narozdíl od Angularu zde není poskytován dedikovaný framework určený přímo pro testování této knihovny. Stránka popisuje testovací frameworky, které používají různé společnosti: autoři Reactu - Facebook - zmiňují svůj Jest [23], také je ale zmiňován Enzyme [17] od Airbnb. Konkrétní příklady použití bychom tedy dále hledali v dokumentacích těchto frameworků. Celkově je pro React k dispozici více různých testovacích frameworků, které opět pokrývají jak Unit, tak E2E testování.

Vue.js Ani Vue.js se neztratí co se Unit a E2E testů týká. Přímo ve své dokumentaci [35] popisuje spouštění Unit testů pomocí vestavěných příkazů, které interně používají buďto již zmiňovaný Jest nebo Mocha [26], a dále odkazuje na vlastní kompletní testovací knihovnu [20]. Také je možné použít velké množství různých testových frameworků a aplikace ve Vue.js lze vyvíjet i pomocí TDD [24].

Ember.js Tento framework se ve své dokumentaci věnuje testování poměrně intenzivně a zasvětil mu rovnou několik stránek [18]. Jako výchozí testovací framework představuje vlastní QUnit, avšak informuje, že je možné použít i frameworky třetích stran. Testování přehledně rozděluje na *Unit*, *Container*, *Render*, a *Application* testy a také popisuje jak testovat jednotlivé komponenty. Testování Ember.js tedy také hodnotím jako velmi dobré.

Backbone.js U posledního zkoumaného frameworku na první pohled není testování moc jasné. Oficiální stránka odkazuje na *test suite* [6], což je ovšem stránka, která testy *spouští* a ne popisuje. Zdá se, že na této stránce je možné otestovat svůj prohlížeč, zda podporuje veškeré funkcionality Backbone.js. Co se psaní testů týká, rozumněji již vypadá externí stránka *Backbone.js Testing* [30], která již zmiňuje i zde dříve probírané frameworky jako *Mocha*. Ve výsledku tak bude pravděpodobně testování Backbone vypadat obdobně jako u ostatních frameworků, avšak přístup oficiální dokumentace snižuje jeho pochopení.

	Angular	React	Vue.js	Ember.js	Backbone.js
Možnosti testování a jejich popis v dokumentaci	velmi kvalitní	velmi kvalitní	velmi kvalitní	velmi kvalitní	matoucí
<i>bodový zisk</i>	2	2	2	2	1

Tabulka 3.6: Volba frameworku: Testování

3.1.7 Vývojářské nástroje

Dalším důležitým nástrojem při práci s frameworkem je možnost jeho debuggování. Framework by měl nabízet vlastní řešení, které vývojáři usnadní nalézt chybu, zjistit, jak se jeho kód chová či odladit rychlostní problémy.

Všechny ze zde porovnávaných frameworků nabízejí tyto nástroje formou doplňku do prohlížeče, konkrétně se dále budeme bavit o doplňcích pro Google Chrome.

	Angular	React	Vue.js	Ember.js	Backbone.js
Název	Augury	React Developer Tools	Vue.js devtools	Ember inspector	Backbone Debugger
Počet stažení ^a	230k	1.351k	706k	57k	9,5k
<i>bodový zisk</i>	2	4	3	1	0
Hodnocení (z 5)	3.9	4.2	4.7	4.8	4,5
<i>bodový zisk</i>	2	1	3	4	2

Tabulka 3.7: Volba frameworku: Devtools

^aStav k 24. 12. 2018

3.1.8 Počet hvězdiček na GitHubu

Počet hvězdiček na GitHubu lze velmi volně interpretovat jako oblíbenost frameworku mezi vývojáři. Z tohoto důvodu již v tabulce 3.8 hodnotím frameworky dle počtu získaných hvězdiček.

	Angular	React	Vue.js	Ember.js	Backbone.js
Počet hvězdiček na GitHubu ^a	43,6k	117,7k	122,3k	20,3k	27,3k
<i>bodový zisk^b</i>	2	4	5	0	1

Tabulka 3.8: Volba frameworku: Počet hvězdiček na GitHubu^aStav k 17. 12. 2018^bHodnocení přeskakuje bodový zisk 3, aby bylo zhodnoceno i absolutní množství hvězdiček, nejen pořadí.

3.1.9 Počet npm balíčků

Npm [9] je repozitář javascriptových komponent, na kterém jsou sdíleny jednak kompletní řešení (jako například Angular, React, Vue.js a další), ale především různé rozšiřující pluginy do těchto frameworků. Z toho důvodu budu v následující metrice hodnotit, kolik balíčků npm nabízí pro jednotlivé porovnávané frameworky.

Bodové zisky hrubě odpovídají relativnímu počtu nalezených balíčků.

	Angular	React	Vue.js	Ember.js	Backbone.js
Počet npm balíčků	26,6k	73,4k	20,6k	6,4k	1,5k
<i>bodový zisk</i>	2	3	2	1	0

Tabulka 3.9: Volba frameworku: Počet npm balíčků

3.1.10 Otázky na Stack Overflow

Stack Overflow je jedním z portálů sítě Stack Exchange, který zná prakticky každý vývojář. Kdokoliv zde může položit otázku a komunita poté odpovídá, zatímco hlasuje o kvalitě odpovědí, aby byla vybrána ta nejlepší.

Z pohledu volby frameworku může být na jednu stranu vhodné, aby bylo na této stránce hodně otázek týkajících se dané technologie, na druhou stranu to ale může znamenat i nekvalitní dokumentaci. Jelikož ale v předchozí sekci nebyla žádná dokumentace vyhodnocena jako vysloveně špatná, budu dále usuzovat, že větší množství otázek je lepší.

	Angular	React	Vue.js	Ember.js	Backbone.js
Počet otázek na Stack Overflow	146k	118k	28k	23k	21k
Počet zodpovězených otázek	86k	72k	18k	17k	16k
<i>bodový zisk</i>	<i>3</i>	<i>3</i>	<i>1</i>	<i>1</i>	<i>1</i>

Tabulka 3.10: Volba frameworku: Otázky na Stack Overflow

3.1.11 Firemní stack

Další zvolenou metrikou je, jak daná technologie zapadá do firemní stacku firmy Jagu s.r.o., ve které bude tento nový projekt realizován. Firma se specializuje především na webové aplikace a middlewary na zakázku [21], a mezi nejpoužívanější technologie patří PHP (Nette, Laravel, Symfony), dále provozuje jeden informační systém postavený na Angularu a nově také menší aplikaci ve Vue.js. Tabulka 3.11 shrnuje, jak jsou jednotlivé frameworky blízko k tomuto stacku.

	Angular	React	Vue.js	Ember.js	Backbone.js
Shoda s firemním stackem	ano	ne	ano	ne	ne
<i>bodový zisk</i>	<i>2</i>	<i>0</i>	<i>2</i>	<i>0</i>	<i>0</i>

Tabulka 3.11: Volba frameworku: Shoda s firemním stackem

3.1.12 Dostupnost vývojářů

Metrikou, kterou z hlediska udržitelnosti projektu a jeho ekonomických nákladů nelze opomenout, je dostupnost a cena vývojářů se zájmem o danou technologii.

Tato data se ale obtížněji získávají, většina statistik hovoří o nabídkách práce v dané technologii, nikoliv o počtu lidí, kteří s ní pracují. Z toho důvodu jsem se rozhodl založit tuto metriku na výsledcích vyhledávání osob v profesní síti LinkedIn - tak dokážeme zjistit alespoň hrubý počet lidí, kteří o sobě sami tvrdí, že jsou vývojáři v daném frameworku.

Bodové zisky zde hrubě reflektují relativní počet nalezených profilů.

3. VOLBA TECHNOLOGIE

	Angular	React	Vue.js	Ember.js	Backbone.js
Počet výsledků na dotaz ”<název> developer”	344k	333k	78k	21k	76k
<i>bodový zisk</i>	4	4	2	1	2

Tabulka 3.12: Volba frameworku: Počet vývojářů na LinkedIn

3.1.13 Integrace se Sentry

Sentry [32] je nástroj sloužící k automatickému i manuálnímu záznamu chyb v aplikacích. Ve firmě Jagu s.r.o. je využíván v řadě projektů a jeho nasazení bude vhodné i pro aplikaci řešenou v rámci této práce. Z toho důvodu je vhodné se podívat, jak hlubokou integraci je možné mezi jednotlivými frameworky a Sentry realizovat.

Při pohledu na přehled toho, jaké technologie Sentry podporuje v JavaScriptu [22] rychle zjišťujeme, že všech pět zde zkoumaných frameworků je oficiálně podporováno, včetně rychlého návodu na zprovoznění. Z toho důvodu neprobíhá v tabulce 3.13 žádné bodování.

	Angular	React	Vue.js	Ember.js	Backbone.js
Oficiální integrace se Sentry	ano	ano	ano	ano	ano

Tabulka 3.13: Volba frameworku: Integrace se Sentry

3.1.14 Souhrn průzkumu

V tabulce 3.14 jsou sečteny body z předchozích dílčích hodnocení.

	Angular	React	Vue.js	Ember.js	Backbone.js
bodový zisk celkem	23	25	27	14	12

Tabulka 3.14: Volba frameworku: Výsledky

Výsledky rozdělují frameworky na dvě skupiny. V té vedoucí je trojice Angular, React a Vue.js, v pozadí poté zůstávají Ember.js a Backbone.js. Na tomto místě je také vhodné znovu zmínit, že hodnocení frameworků probíhalo ve vztahu ke konkrétnímu projektu, který je cílem této práce, a také ve vztahu k firmě

Jagu s.r.o., která vývoj této aplikace zaštiťuje.

První tři frameworky jsou seřazeny poměrně těsně za sebou, avšak nejlépe vyšel ze srovnání nejmladší Vue.js, který tímto volím jako framework, ve kterém budu na práci dále pracovat.

Implementace

Testování

Závěr

TODO

Zdroje

1. ALPERT, Sophie. *Change license and remove references to PATENTS* [online] (cit. 2018-12-21). Dostupné z: <https://github.com/facebook/react/commit/b765fb25ebc6e53bb8de2496d2828d9d01c2774b#diff-9879d6db96fd29134fc802214163b95a>.
2. ANDRUSHKO, Sviatoslav. *The Best JS Frameworks for Front End* [online]. 2018 (cit. 2018-12-21). Dostupné z: <https://rubygarage.org/blog/best-javascript-frameworks-for-front-end>.
3. *Angular - One framework. Mobile & desktop.* [online]. 2018 (cit. 2018-12-16). Dostupné z: <https://angular.io/>.
4. *Angular - Testing* [online]. 2019 (cit. 2019-01-22). Dostupné z: <https://angular.io/guide/testing>.
5. *Angular - What is Angular?* [online]. 2018 (cit. 2018-12-28). Dostupné z: <https://angular.io/docs>.
6. *Backbone Test Suite* [online] (cit. 2018-12-28). Dostupné z: <http://backbonejs.org/test/>.
7. *Backbone.js* [online]. 2016 (cit. 2018-12-17). Dostupné z: <http://backbonejs.org/>.
8. *Backbone.js* [online]. 2016 (cit. 2018-12-28). Dostupné z: <http://backbonejs.org/#Getting-started>.
9. *Build amazing things* [online]. 2018 (cit. 2018-12-21). Dostupné z: <https://www.npmjs.com/>.

10. *Collection: Front-end JavaScript frameworks* [online]. 2018 (cit. 2018-12-17). Dostupné z: <https://github.com/collections/front-end-javascript-frameworks>.
11. *Comparison with Other Frameworks - Vue.js* [online]. 2018 (cit. 2018-12-14). Dostupné z: <https://vuejs.org/v2/guide/comparison.html>.
12. *Ember.js - A framework for ambitious web developers* [online]. 2018 (cit. 2018-12-17). Dostupné z: <https://www.emberjs.com/>.
13. *Ember.js Guides - Guides and Tutorials - Ember Guides* [online]. 2018 (cit. 2018-12-28). Dostupné z: <https://guides.emberjs.com/release/>.
14. *Getting Started - React* [online]. 2018 (cit. 2018-12-28). Dostupné z: <https://reactjs.org/docs/getting-started.html>.
15. GOEL, Aman. *Top 10 Web Development Frameworks in 2018* [online]. 2018 (cit. 2018-12-17). Dostupné z: <https://hackr.io/blog/top-10-web-development-frameworks-in-2018>.
16. CHANDRA, Rajdeep. *My experience with Angular 2 , React and Vue* [online]. 2018 (cit. 2018-12-21). Dostupné z: <https://medium.com/@rajrock38/my-experience-with-angular-2-react-and-vue-fb654e3ecf33>.
17. *Introduction - Enzyme* [online]. 2019 (cit. 2019-01-22). Dostupné z: <https://airbnb.io/enzyme/>.
18. *Introduction - Testing - Ember Guides* [online]. 2019 (cit. 2019-01-22). Dostupné z: <https://guides.emberjs.com/release/testing/>.
19. *Introduction - Vue.js* [online]. 2018 (cit. 2018-12-28). Dostupné z: <https://vuejs.org/v2/guide/>.
20. *Introduction | Vue Test Utils* [online]. 2019 (cit. 2019-01-22). Dostupné z: <https://vue-test-utils.vuejs.org/>.
21. *Jagu - Software na míru, webové prezentace, grafika, webhosting* [online]. 2018 (cit. 2018-12-21). Dostupné z: <https://www.jagu.cz/>.
22. *JavaScript Error Tracking* [online]. 2019 (cit. 2019-01-21). Dostupné z: <https://sentry.io/for/javascript/>.
23. *Jest - Delightful JavaScript Testing* [online]. 2019 (cit. 2019-01-22). Dostupné z: <https://jestjs.io/>.

-
24. KUROSKI, Daniel. *Working an application in Vue.js with TDD* [online]. 2018 (cit. 2019-01-22). Dostupné z: <https://vue-test-utils.vuejs.org/>.
 25. LASCIK, V. *Honest look at Ember in the middle of 2018* [online]. 2018 (cit. 2018-12-20). Dostupné z: <https://medium.com/@vlascik/honest-look-at-ember-in-the-middle-of-2018-a0dc2787e506>.
 26. *Mocha - the fun, simple, flexible JavaScript test framework* [online]. 2019 (cit. 2019-01-22). Dostupné z: <https://mochajs.org/>.
 27. OGDEN, Cody. *Killed by Google - Google Graveyard - A Google Cemetery* [online]. 2018 (cit. 2018-12-23). Dostupné z: <https://killedbygoogle.com/>.
 28. *Protractor - end-to-end testing for AngularJS* [online]. 2019 (cit. 2019-01-22). Dostupné z: <https://www.protractortest.org/>.
 29. *React - A JavaScript library for building user interfaces* [online]. 2018 (cit. 2018-12-16). Dostupné z: <https://reactjs.org/>.
 30. ROEMER, Ryan. *Backbone.js Testing* [online]. 2018 (cit. 2018-12-28). Dostupné z: <http://backbone-testing.com/>.
 31. *Routing* [online]. 2018 (cit. 2018-12-21). Dostupné z: <https://vuejs.org/v2/guide/routing.html>.
 32. *Sentry | Error Tracking Software — JavaScript, Python, PHP, Ruby, more* [online]. 2019 (cit. 2019-01-21). Dostupné z: <https://sentry.io/welcome/>.
 33. *Test Utilities - React* [online]. 2019 (cit. 2019-01-22). Dostupné z: <https://reactjs.org/docs/test-utils.html>.
 34. *Tilde Inc. - About Us* [online]. 2018 (cit. 2018-12-17). Dostupné z: <https://www.tilde.io/about-us/>.
 35. *Unit Testing - Vue.js* [online]. 2019 (cit. 2019-01-22). Dostupné z: <https://vuejs.org/v2/guide/unit-testing.html>.
 36. WOLFF, Adam. *Relicensing React, Jest, Flow, and Immutable.js* [online]. 2017 (cit. 2018-12-21). Dostupné z: <https://code.fb.com/web/relicensing-react-jest-flow-and-immutable-js/>.

ZDROJE

37. YOU, Evan. *Vue - The Progressive JavaScript Framework* [online]. 2018 (cit. 2018-12-16). Dostupné z: <https://vuejs.org/>.

Seznam použitých zkratek

API	Application Programming Interface
BSD	Berkeley Software Distribution
CI	Continuous Integration
DI	Dependency Injection
E2E	End-to-end
FIT	Fakulta informačních technologií
GUI	Graphical User Interface
MIT	Massachusetts Institute of Technology
NPM	Node Package Manager
OOP	Objektově orientované programování
OS	Operační Systém
TDD	Test-driven development

Slovník pojmů

Backend	Část aplikace, která se stará o ukládání dat, jejich zpracování a bussiness logiku. Většinou není přímo přístupná koncovému uživateli, ten k ní přistupuje přes frontend.
Dependency Injection	Technika, která umožňuje "vložení" instance objektu, který poskytuje nějakou službu, do jiného objektu, který pak může danou službu efektivně používat.
Framework	Softwarová struktura, která slouží jako podpora pro pohodlnější programování. Může obsahovat podpůrné funkce, knihovny či nástroje pro efektivnější, bezpečnější a pohodlnější vývoj softwaru.
Frontent	Část aplikace, s kterou přímo interaguje koncový uživatel či administrátor, typicky pomocí GUI. Většinou komunikuje s druhou, serverovou částí - backendem.
GitHub	GitHub je webová služba podporující vývoj softwaru za pomoci verzovacího nástroje Git.
Middleware	Software realizující integraci mezi dvěma jinými systémy, typicky pomocí API.

B. SLOVNÍK POJMŮ

TypeScript	Nadmnožina JavaScriptu, která jej rozšiřuje především o statické typování proměnných a další atributy z OOP.
WebView	Komponenta nativní Android aplikace, která zobrazuje stanovenou URL jako svůj obsah. Používá se zejména v místech, kde je žádoucí zobrazovat obsah z webu, ale je potřeba přístup k funkcím zařízení, ke kterým není možné přistupovat z běžného webového prohlížeče.

TODO přílohy