

Zadání projektu

Vytvořte nástroj pro tvorbu a sdílení HTML5 prezentací na webu. Webový backend umožní tvorbu a editaci slidů a jejich sdílení. Pro prezentaci použijte vhodnou JavaScriptovou knihovnu, kterou rozšířte o možnost testování čtenářů prezentace. Další požadavky sesbírejte na základě vytvořeného prototypu.

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Semestrální projekt

Slide It

Radek Ježdík

Vedoucí práce: Ing. Ondřej Macek

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Softwarové inženýrství

9. ledna 2013

Poděkování

Zde můžete napsat své poděkování, pokud chcete a máte komu děkovat.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 7. 1. 2013

.....

Abstract

Translation of Czech abstract into English.

Abstrakt

Cílem projektu bylo vytvořit webový portál, který bude poskytovat nástroje pro tvorbu a sdílení prezentací postavených na technologiích HTML5. Požadavkem bylo využít potenciálu HTML5 a JavaScriptu ke zvýšení interaktivity prezentace s uživatelem.

Obsah

1	Úvod	1
2	Analýza	3
2.1	HTML5 prezentace	3
2.2	Výběr JavaScriptové knihovny	3
2.2.1	Reveal.js	3
2.2.2	Impress.js	4
2.2.3	Google I/O HTML5 slide template	4
2.2.4	Deck.js	4
2.2.5	Zhodnocení	4
2.3	Webový portál	4
2.4	Editor	4
2.5	Rešerše existující řešení	5
2.5.1	SlideShare	5
2.5.2	Prezi	5
2.5.3	Rvl.io	5
3	Případy užití	7
4	Doménový model	9
4.1	Uživatel	9
4.2	Prezentace	10
4.3	Komentář	10
5	Návrh	11
5.1	Serverová část	11
5.2	Klientská část	11
5.3	Editor	11
6	Realizace	13
6.1	Rozložení editoru	14
6.2	Chování editoru	14
6.3	Úprava Taxy! syntaxe	14
6.3.1	Markdown syntaxe	14
6.3.2	Nová syntaxe	15

7	Nasazení	17
7.1	Server	17
8	Shrnutí	19

Seznam obrázků

4.1 Diagram doménového modelu	9
---	---

Seznam tabulek

Kapitola 1

Úvod

Tento projekt má za cíl vytvořit webový portál, který uživatelům usnadní nejen snadné sdílení prezentací, ale také jejich tvorbu. Prezentace by se měly zaměřovat na podání informací a na případné testování uživatelů z nově nabytých vědomostí. Prezentace by tak měly sloužit jak přednášejícím, tak posluchačům či žákům. Pro zobrazení prezentace bude využita rodina technologií HTML5 a JavaScriptové knihovny.

Kapitola 2

Analýza

V analýze jsem zjišťoval, jak se prezentace postavené na HTML tvoří, a porovnával různé knihovny, kterou tuto funkčnost zajišťují.

2.1 HTML5 prezentace

Prezentace založené na HTML se z pohledu kódu podobají běžné webové stránce. K popisu snímků se používají sémantické značky HTML (HyperText Markup Language) a pro změnu vzhledu se používá CSS (Cascading Style Sheets). K oživení takovéto stránky a vytvoření dojmu prezentace a přechodů mezi snímky je pak potřeba JavaScript.

2.2 Výběr JavaScriptové knihovny

Pro takto vytvořené prezentace jsem se rozhodl použít už hotovou a ověřenou knihovnu. Knihoven pro zobrazení prezentací založených na (X)HTML bylo vytvořeno mnoho (viz [1]). Většina z nich je ovšem už zastaralá, nefunkční nebo neodpovídající mým představám. Za zmínku stojí několik zdařilých knihoven, které se objevily po uvedení podpory HTML5 v prohlížečích

- reveal.js
- impress.js
- Google I/O HTML5 slide template
- deck.js

2.2.1 Reveal.js

Kromě posunu snímků horizontálně, podporuje také vnořené snímky, na které se dostaneme posunem dolů nebo nahoru. Má několik zajímavých rozšíření, např. podpora syntaxe Markdown a barevné zvýraznění kódu programovacích jazyků pomocí knihovny highlight.js. Umožňuje také v novém okně zobrazit aktuální a následující snímek včetně poznámek přednášejícího.

2.2.2 Impress.js

U impress.js už z názvu vyplývá, že bude sloužit hlavně k tomu, aby oslnil posluchače. To rozhodně dělá díky bohatým transformacím při přechodu na jednotlivé snímky. Ty jsou rozloženy v prostoru, jak 2D tak i 3D, o různých velikostech a při přechodu mezi snímky se kamera efektně otáčí a přesouvá. Pro naše účely je ale až moc složitý (hlavně pro určení pozic snímků a transformací) a pro informativní prezentace se nehodí.

2.2.3 Google I/O HTML5 slide template

HTML5 šablona od Googlu je z výše zmíněných knihoven nejjednodušší co se týče vzhledu, nicméně podporuje všechny důležité funkce. Nevýhodou je, že svázána se stylem Googlu a lze jej měnit pouze přepsáním CSS pravidel.

2.2.4 Deck.js

Poslední knihovna deck.js umí vše, co umí šablona od Googlu. Je napsaná v jQuery a je velmi dobře zdokumentovaná a rozšiřitelná. Výhodou je také dvojí licence MIT nebo GPL.

2.2.5 Zhodnocení

Pro účely tohoto projektu jsem se rozhodl použít deck.js. Reveal.js má sice už v základu podporu pro Markdown a barevné zvýraznění syntaxe, ale v budoucnu pro mne mohly nastat jistá omezení při implementaci ostatní funkčnosti.

2.3 Webový portál

Pro uživatele má portál sloužit jako prostředník pro sdílení prezentací – ať už v rámci portálu mezi jeho uživateli nebo posíláním URL odkazů na prezentace v emailech, příspěvcích na Facebooku nebo Twitteru atp. – a také k tvorbě nových prezentací. V principu se tak bude podobat portálům jako YouTube, SlideShare nebo GitHub, které se zaměřují na vystavení vlastní tvorby a její sdílení.

2.4 Editor

Důležitou částí tohoto projektu byl editor pro úpravy prezentace. Jelikož je prezentace tvořena pomocí HTML, nabízí se několik možností, jak editaci pojmout:

1. psání samotného HTML kódu
2. editor WYSIWYG (What You See Is What You Get)
3. jednoduchý textový značkovací jazyk (lightweight markup language)

Psaní samotného HTML kódu bylo vyloučeno okamžitě. HTML není nijak úsporný značkovací jazyk a pro lidi neznalé HTML by byl překážkou. Na druhou stranu ale poskytuje největší možnou kontrolu nad tím, jak se má prezentace chovat či jak má vypadat. Kdyby ovšem editor měl být založen pouze na psaní HTML kódu, nebyl by vůbec potřeba. Pro mnoho lidí by pak bylo výhodnější použít nástroje, na které jsou zvyklí, a pak pouze nahrát hotovou prezentaci na web.

Konečné rozhodnutí, která z možností byla použita, je popsáno v kapitole [Realizace](#).

2.5 Rešerše existující řešení

Během analýzy jsem také hledal a zkoušel již existující projekty podobné tomuto.

2.5.1 SlideShare

SlideShare je na internetu velmi známý webový portál pro sdílení prezentací. Jeho nevýhodou ovšem je, že prezentace jsou velmi statické – bez animací a bez větší interaktivity s uživatelem (podporuje akorát kliknutí na odkaz). Další nevýhoda je, že prezentace nelze na portálu přímo tvořit, lze je pouze nahrávat ve formátech PDF, PPT a dalších z lokálního úložiště. Jakákoliv změna v prezentaci se tak musí nahrát znovu.

2.5.2 Prezi

Prezi funguje na podobném principu jako SlideShare, ale zaměřuje se daleko více na grafickou stránku a zakládá si na kreativitě uživatelů a vyjádření jejich idejí pomocí celkového pohledu na prezentaci. Prezentace se značně liší od běžných prezentací, které známe např. z Microsoft PowerPoint. Jejich použití se hodí hlavně pro vyjádření nějaké myšlenky, ale už ne tolik k podání nějakých informací.

2.5.3 Rvl.io

Tento portál se nejvíce podobá zadání tohoto projektu a možnosti jeho využití jsou prakticky stejné. Umožňuje sdílení prezentací a jejich tvorbu. Pro zobrazení prezentací používá již zmíněnou knihovnu reveal.js od stejného autora. Tento portál je velmi mladý a uveden byl po měsíci od započetí práce na tomto projektu. Služba byla v době psaní této práce stále ve fázi beta testování.

Kapitola 3

Případy užití

Následující odstavce popisují, které role se při použití systému objeví a jaké tyto role mají práva či možnosti využití systému.

Jako **nepřihlášený uživatel** chci mít možnost:

- **prohlížet veřejně dostupné prezentace**, k tomu použiji
 - vyhledávací formulář na webu
 - adresu URL, kterou jsem dostal jiným komunikačním kanálem
- **vyhledat prezentace** vyhledávacím formulářem podle názvu nebo autora, abych si mohl
 - prohlédnout hledanou prezentaci
 - přečíst komentáře pod prezentací
 - podívat se na prezentace hledaného uživatele
- **přihlásit se** do systému pomocí přihlašovacího formuláře, abych získal oprávnění přihlášeného uživatele (viz níže)
- **registrovat se** do systému pomocí registračního formuláře, abych se mohl přihlásit a získat tak oprávnění

Jako **přihlášený uživatel** chci mít možnost:

- **vytvářet** prezentace pomocí editoru
- **posílat komentáře** k prezentacím, ke kterým mám přístup, pomocí formuláře pod prezentací
- **prohlížet veřejné a mně sdílené prezentace**
- **vyhledat prezentace** vyhledávacím formulářem podle názvu nebo autora, abych si mohl (včetně toho, co může nepřihlášený uživatel):

- sledovat nebo zrušit sledování hledaného uživatele
- vidět seznam mně sdílených prezentací
- vidět seznam veřejných prezentací uživatelů, které sleduji
- odhlásit se

Jako **autor prezentace** chci mít možnost:

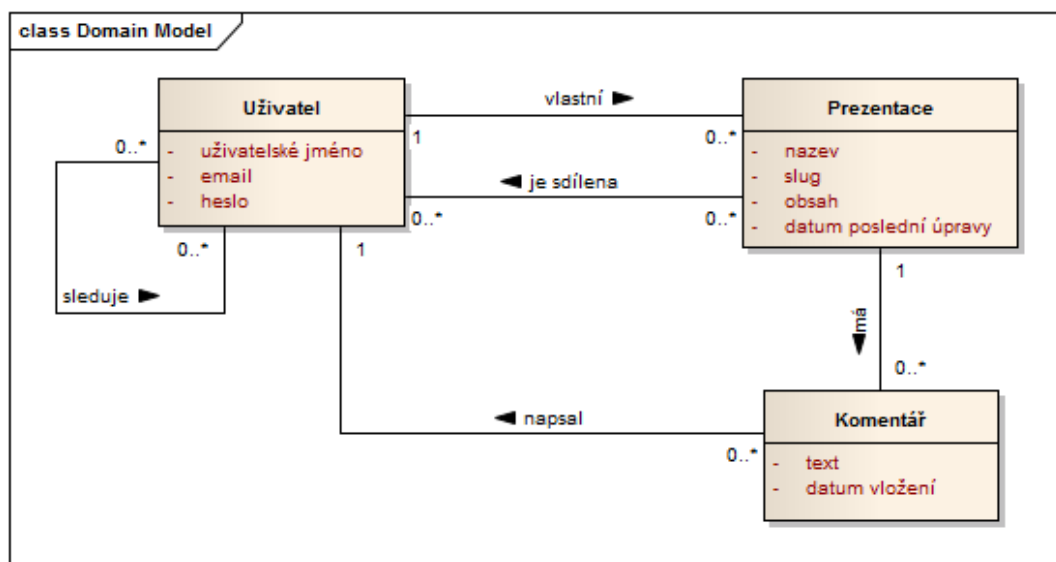
- upravit prezentaci v editoru, abych opravil chybu, či ji o něco doplnil
- smazat prezentaci
- smazat jakýkoliv komentář k vlastní prezentaci
- sdílet prezentaci ostatním uživatelům portálu

Jako **autor komentáře** chci mít možnost:

- smazat komentář

Kapitola 4

Doménový model



Obrázek 4.1: Diagram doménového modelu

4.1 Uživatel

Registrovaný a přihlášený uživatel má výhody oproti nepřihlášenému (viz výše). Pro úspěšnou registraci uživatele do systému musí nepřihlášený uživatel zadat:

- **uživatelské jméno**, pod kterým se bude prezentovat a přihlašovat. Stejně uživatelské jméno musí být v systému registrováno jenom jednou
- **e-mail**, který nebude zveřejněn a bude sloužit pro aktivaci účtu a případně bude informačním kanálem (notifikace, změna hesla, apod.)
- **heslo**, které bude chránit jeho účet (min. 4 znaky)

Uživatelé se mohou navzájem sledovat. Uživatel je pak snadněji upozorněn na nové prezentace od sledovaných uživatelů.

4.2 Prezentace

Prezentace představuje jednu prezentaci, kterou uživatel pomocí systému vytvořil (je jejím autorem). Má tyto povinné vlastnosti:

- **název**, pod kterým se daná prezentace bude zobrazovat ve výpisech, a s tím související:
- **slug** - zjednodušený název (malá písmena anglické abecedy, čísla a pomlčky), pod kterým bude prezentace dostupná přes adresu URL
- **obsah**, jenž představuje složitější objekt textů jednotlivých snímků a různých nastavení prezentace

Prezentace může být sdílena ostatním uživatelům portálu.

4.3 Komentář

Komentář je uživatelem napsaná poznámka k nějaké prezentaci. Komentář může vkládat každý přihlášený uživatel. Povinnými vlastnostmi jsou:

- **text komentáře** – obsah zprávy
- **datum a čas**, kdy byl komentář vložen

Komentář se váže k jedné prezentaci a jednomu uživateli (autorovi komentáře).

Kapitola 5

Návrh

5.1 Serverová část

Webový back-end jsem se rozhodl napsat v PHP s použitím českého frameworku Nette, s kterým mám dobré zkušenosti. Nette framework mi ušetřil spoustu času, protože všechny důležité funkce, které bych jinak musel řešit a ladit vlastními silami, zvládá elegantně již v základu. Jeho použití je jednoduché a přímočaré a navíc je velmi dobře otestovaný (mnoha uživateli i vlastními testy). Pro uchování dat byla použita databáze MySQL.

5.2 Klientská část

Front-end je pak poháněn JavaScriptem s frameworkem jQuery. Pro vzhled a grafické uživatelské rozhraní byl použit framework Bootstrap, který velmi usnadnil prototypování portálu bez nutnosti tvořit a ladit vlastní vzhled. Výhodou je i to, že Bootstrap poskytuje responsivní design a tak by se měl web chovat a zobrazovat dobře i na mobilních zařízeních.

Pro zobrazení prezentací byla použita již zmíněná knihovna deck.js.

5.3 Editor

Editoru jsem přikládal větší význam, a proto používá pár technologií navíc. Jako nejdůležitější se ukázalo použití JavaScriptového MVC frameworku AngularJS. Jeho hlavní předností je dvoucestné provázání dat mezi pohledem (view) a modelem. Každá změna v modelu se automaticky propaguje do pohledu a naopak. Snižuje se tak množství kódu, který otrocky nastavuje nebo zobrazuje data při jednotlivé změně.

Pro panel nástrojů, který uživateli ulehčí práci se zdrojovým textem ve formátu Texy!, byla použita knihovna Texyla.

Kapitola 6

Realizace

V první fázi vývoje se počítalo s použitím WYSIWYG editoru pro úpravu snímků. Existuje mnoho knihoven, které WYSIWYG editor různě implementují. Protože ale neexistuje žádný standard pro přímou editaci kódu HTML, každý prohlížeč se chová trochu jinak a zároveň každá knihovna si různé věci implementuje po svém.

Nejdříve jsem zkoušel použít pro editaci zdrojového HTML kódu prezentace knihovnu Aloha Editor. Kromě potíží se zobrazením panelu nástrojů byla největším problémem licence (pozn.: nedávno byla licence změněna na GPLv2).

Přešel jsem tedy na editor Mercury, který se zdál být vhodnější a komplexnější. I když je to JavaScriptová knihovna, byla vyvíjena v Ruby pro použití v Rails aplikacích a tak bylo hodně věcí přizpůsobeno těmto technologiím (například počítala s partial pohledy, které se posílají AJAXem ze serveru). Nicméně knihovna není na jazyce Ruby závislá a lze ji použít bez něj.

Zde se ale po chvíli objevily problémy technického rázu – prohlížeče se právě díky neexistujícímu standardu chovaly rozdílně nebo se nechovaly podle očekávání. Výsledkem pak byl například špatný HTML kód nebo kód, který se už nedal nijak pomocí editoru opravit ani smazat. Jedním z největších problémů bylo vkládání nového řádku. Prohlížeč někdy vložil značku odřádkování `
` někdy nový odstavec `<p>`. Chování tak nebylo předvídatelné. Objevily se i další chyby/bugy v použitelnosti, na které jsem se pak snažil upozornit¹ autora knihovny. Chyba dosud nebyla opravena.

Všechny tyto problémy nakonec vyústily k rozhodnutí použít značkovací jazyk, tzv. Lightweight markup language, který převádí text do HTML. V poslední době získávají tyto jazyky velkou popularitu kvůli jednoduchosti a velmi dobré podobě výsledného HTML kódu. Asi nejvíce používaný takovýto jazyk je, hlavně ve světě IT, Markdown, který byl původně napsán v Perl. Vzniklo mnoho portů této knihovny do různých jazyků, včetně PHP. Port pro PHP je ovšem velmi komplexní a zdálo se mi nemožné ho upravit pro potřeby implementace jistých funkcí do prezentace.

Rozhodl jsem se proto použít jazyk Taxy!, který je Markdownu podobný, ovšem syntaxe se občas liší. Taxy! je velmi dobře rozdělené na moduly a jednotlivé konverze se provádějí pomocí callbacků zaregistrovaných podle regulárních výrazů.

¹viz <https://github.com/jejackson/mercury/issues/253>

Největší výhodou byla ovšem možnost specifikovat CSS třídy nebo určit zarovnání textu či obrázku. To je něco, co Markdown neumí. Velkou výhodou Taxy! je také velmi dobrá podpora typografie. Například sekvenci znaků „+–“ nahradí za jediný znak „±“, jednoduchou pomlčku nahrazuje za spojovník (pokud je to ve větě potřeba), tři tečky nahradí za jediný znak trojtečky nebo vkládá nedělitelné mezery za spojky atp. Při použití v prezentacích jsem si to dokázal představit jako dobrý bonus.

6.1 Rozložení editoru

Pro rozložení editoru jsem se inspiroval u běžných nástrojů pro tvorbu prezentací (Microsoft PowerPoint, OpenOffice Impress) a rozdělil na obrazovku na náhled všech snímků po levé straně obrazovky a na editaci právě upravovaného snímku vpravo. Protože používám značkovací jazyk namísto přímé editace HTML kódu, bylo nutné dále rozdělit tuto část na textové pole, do kterého se bude psát text ve značkovacím jazyce, a na velký náhled snímku po konverzi do HTML kódu. Textové pole má navíc panel nástrojů, který dokáže vkládat značky (například nadpis, nebo kurzívu) pomocí tlačítek.

Toto rozložení se ukázalo jako ne zcela dobré. Vzhled editoru je nyní postaven na velmi jednoduchém CSS a neladí se stylem zbytku webu, protože po přidání Bootstrap frameworku, který mimo jiné upravuje globální styly prvků, vypadají náhledy jinak než ve skutečnosti. Dalším problémem jsou jisté technologické nedostatky jako například zmenšování obsahu snímku pro přizpůsobení výšce a šířce náhledu. Proto se bude editor zjednodušovat, aby na obrazovce bylo co nejméně rušivých elementů a dodržoval stejný vzhled jako zbytek webu.

6.2 Chování editoru

Ze serveru se nejdříve získá celá prezentace jako JavaScriptový objekt. HTML kód každého snímku se pak vloží do panelu s malými náhledy. Malý náhled snímku je pomocí CSS pravidla zmenšen, protože jinak by obsah byl stejně velký jako v normálním zobrazení prezentace. Po kliknutí na malý náhled se objeví v textovém poli pro úpravu snímku zdrojový text v jazyce Taxy! a ve velkém náhledu vpravo se objeví kompletní snímek s inicializovanou knihovnou deck.js včetně funkční interakce.

Změna textu snímku se automaticky posílá na server a po zkonvertování se kód HTML vrací v odpovědi. Následně se aktualizují oba náhledy. Aby se na server neposílala každá úprava snímku (např. po napsání každého písmene), byla implementovaná čekací doba 1 sekundy. Pokud po změně snímku uplyne tato doba, během které nebyla provedena žádná další úprava, pošle se požadavek na server.

6.3 Úprava Taxy! syntaxe

6.3.1 Markdown syntaxe

Protože je jazyk Markdown tak rozšířený, chtěl jsem syntaxi Taxy! pro psaní snímků co nejvíce přiblížit syntaxi Markdownu.

Syntaxe Taxy! je Markdownu velmi podobná. Jeden z nejvíce patrných rozdílů byl v syntaxi nadpisů. Proto jsem konvertor upravil a změny otestoval pomocí jednotkových testů.

6.3.2 Nová syntaxe

Zároveň bylo potřeba upravit konvertor a implementovat vlastní pravidla konverze textu do HTML. Tato úprava se týkala možnosti uživatele odpovídat na otázky během prohlížení prezentace. Jedna se pak může vyskytovat na jednom snímku.

Bylo potřeba vymyslet a implementovat syntaxi na vložení zaškrťovacího pole, které by uživatel zaškrtl jako správnou odpověď na otázku položenou na snímku.

Syntaxe byla zvolena takto:

[+] pro vložení zaškrťovacího pole ke správné odpovědi
[-] pro vložení zaškrťovacího pole ke špatné odpovědi

Pro vyhodnocení správnosti odpovědi se při konverzi automaticky vkládá vyhodnocovací tlačítko na konec snímku. Po jeho stisku uživatelem se kontroluje, že pole u správných odpovědí jsou zaškrtnutá a u špatných odpovědí nikoli. Správnost či nesprávnost je pak zvýrazněna CSS třídou u každého pole.

U snímků s otázkou lze také vložit i doplňkový text, který se zobrazí při správné resp. špatné odpovědi. Autor prezentace tak může uvést bližší informace k odpovědi na danou otázku nebo odkázat na jiný snímek prezentace či jinou stránku.

Pro vložení takového bloku textu byla použita podobná syntaxe jako pro ostatní účelové bloky textu v syntaxi Taxy! (např. blok kódu). Použití syntaxe je následující:

```
/--correct
Tento text se zobrazí při **správné** odpovědi
\--

/--incorrect
Tento text se zobrazí při **špatné** odpovědi
\--
```


Kapitola 7

Nasazení

7.1 Server

Pro citem aplikace je nutný běžící webový server, na kterém běží PHP a databáze. Pro vývoj používám hotové řešení WAMP (Windows, Apache, MySQL, PHP) jménem EasyPHP. Verze komponent:

- Apache 2.2
- PHP 5.3
- MySQL 5.5

Pro nasazení na produkční prostředí jsem se rozhodl využít cloudového systému PHP Fog¹. Pro malý počet aplikací a malý rozsah poskytuje své služby zdarma. V placených řešení je pak největší výhodou možnost škálování instancí aplikace či paměti nebo velké množství rozšíření (např. analytické nástroje či optimalizační nástroje pro databázi). Server uchovává aplikaci jako git repozitář a tak pro nahrání nové verze aplikace stačí příkaz push.

V listopadu 2012 bylo ovšem oznámeno ukončení provozu PHP Fogu a postupné odpojování bezplatných aplikací v průběhu prosince. Jako náhrada byla doporučena podobná služba AppFog² od stejné firmy.

¹[<http://phpfog.com>](http://phpfog.com)

²[<http://www.appfog.com/>](http://www.appfog.com/)

Kapitola 8

Shrnutí

Projekt byl velmi zajímavý. Vyzkoušel jsem si několik nových technologií, zejména AngularJS, který určitě budu dál sledovat a snad s ním i někdy znovu pracovat. Opět jsem ale došel k závěru, že zkoumání a použití nových a neznámých vlastností v technologiích HTML(5) není jednoduché a bezproblémové.

Jsem rád, že jsem použil Nette framework, který dobře znám a tak jsem neměl tolik problémů u serverové části. Ze začátku jsem se ovšem rozhodoval, jestli si nevyzkoušet nějakou jinou technologii, například dnes moderní Ruby on Rails nebo Node.js. Mohu ale říct, že bych si nepřál znovu dělat tento projekt v jiné technologii, neboť by mi to přineslo o to více problémů.

Více práce by si rozhodně zasloužila podoba editoru a obecná uživatelská přívětivost. Jinak ale výsledek projektu hodnotím kladně.

Literatura

[1] List of web based slide shows.

[http://en.wikipedia.org/wiki/Web-based_slideshow#List_of_web-based_slide_shows.](http://en.wikipedia.org/wiki/Web-based_slideshow#List_of_web-based_slide_shows)