

Sem vložte zadání Vaší práce.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Magisterská práce

Průvodce studenta FIT ČVUT

Bc. Jan Molnár

Vedoucí práce: Ing. Michal Havryluk

3. května 2012

Poděkování

Doplňte, máte-li komu a za co děkovat. V opačném případě úplně odstráňte tento příkaz.

Prohlášení

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

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

V Praze dne 3. května 2012

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2012 Jan Molnár. 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

Jan Molnár. *Průvodce studenta FIT ČVUT: Magisterská práce*. Praha: ČVUT v Praze, Fakulta informačních technologií, 2012.

Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

Keywords Nahradte seznamem klíčových slov v angličtině oddělených čárkou.

Abstrakt

V několika větách shrňte obsah a přínos této práce v češtině. Po přečtení abstraktu by se čtenář měl mít čtenář dost informací pro rozhodnutí, zda chce Vaši práci číst.

Klíčová slova Nahradte seznamem klíčových slov v češtině oddělených čárkou.

Obsah

Úvod	17
1 Popis problému, specifikace cíle	19
1.1 Popis řešeného problému	19
1.2 Popis uživatele a jeho potřeb	20
1.3 Vymezení cílů diplomové práce	20
1.4 Popis struktury DP ve vztahu k vytyčeným cílům	21
1.5 Serverová aplikace	21
1.5.1 Rešeršní zpracování existujících ontologií	22
1.6 Mobilní aplikace	23
1.6.1 Rešeršní zpracování existujících mobilních aplikací	23
2 Analýza a návrh implementace	27
2.1 Serverová aplikace	27
2.1.1 Potřeby uživatelů	27
2.1.2 Případy užití	28
2.1.3 Funkční požadavky	31
2.1.4 Nefunkční požadavky	32
2.1.5 Návrh řešení	33
2.1.6 Návrh bezpečnosti	33
2.1.7 Návrh nasazení	33
2.1.8 Použité technologie	33
2.1.9 Použité knihovny	34
2.1.10 Použité nástroje	35
2.2 Mobilní aplikace	35
2.2.1 Potřeby uživatelů	35
2.2.2 Prototypy	35
2.2.3 Případy užití	35
2.2.4 Funkční požadavky	35
2.2.5 Nefunkční požadavky	36
2.2.6 Návrh řešení	36
2.2.7 Návrh bezpečnosti	36
2.2.8 Návrh nasazení	36

2.2.9	Použité technologie	37
2.2.10	Použité knihovny	38
2.2.11	Použité nástroje	39
3	Realizace	41
3.1	Serverová aplikace	41
3.1.1	Řešené problémy	41
3.1.2	Implementace bezpečnosti	41
3.2	Mobilní aplikace	41
3.2.1	Řešené problémy	41
3.2.2	Implementace bezpečnosti	42
3.2.3	Ideální implementace	42
4	Testování	45
4.1	Serverová aplikace	45
4.1.1	Testování programátorem	45
4.1.2	Integrační testování	45
4.1.3	Zátěžové testování	45
4.1.4	Akceptační testování	45
4.2	Mobilní aplikace	45
4.2.1	Testování programátorem	45
4.2.2	Testování funkcionalit	45
4.2.3	Integrační testování	46
4.2.4	Testování systémové integrace	46
4.2.5	Zátěžové testování	46
4.2.6	Akceptační testování	46
4.2.7	Testování použitelnosti	46
4.2.8	Komparativní testování	46
	Závěr	47
	Literatura	49
	A Seznam použitých zkratk	51
	B Instalační a uživatelská příručka	53
	C Obsah přiloženého CD	55
	D TODO	57

Seznam obrázků

Seznam tabulek

Úvod

Když jsem se v květnu roku 2011 začal zabývat diplomovou prací, neměl jsem v tom, co budu dělat, vůbec jasno. Jedním z mála tehdejších cílů bylo vytvoření aplikace, ze které bude mít co největší počet lidí užitek. V rámci bakalářské práce jsem se zabýval doménou navigace studentů po Fakultě elektrotechnické ČVUT [8], proto jsem, mezi jinými tématy, zvažoval i pokračování v obdobné doméně na Fakultě informačních technologií – jedná se o oblast, ve které lze studentům přinést mnoho užitečného.

Nejprve jsem zkontaktoval pana docenta Vitvara, kterého téma oslovilo a začali jsme společně tvořit zadání diplomové práce. Dohodli jsme se na struktuře práce, použitých technologiích a některých dalších detailech. Mezi požadavky se objevila tvorba mobilní aplikace pro přístup k informacím, těmi se pan docent nezabývá, proto mi doporučil další pokračování práce pod panem inženýrem Havrylukem, jehož jsou primární doménou. Této nabídky jsem využil.

Shodná doména bakalářské a diplomové práce se potvrdila být prospěšnou – ačkoliv jsem z bakalářské práce v diplomové mimo získaného přehledu o doméně nakonec nic jiného nevyužil, právě ten se ukázal být jako velmi přínosný – od začátku jsem měl přehled o problematických místech a těch, které tehdy nebylo možné realizovat, a mohl se na ně zaměřit a vyřešit tentokrát podstatně lépe. Bakalářská práce se zabývala pouze navigací do určitého bodu, diplomová jde podstatně více do hloubky i do šířky a přináší celého průvodce postaveného na solidních základech.



Popis problému, specifikace cíle

Tato kapitola se zabývá problematikou zpracovávanou diplomovou prací, kontextem, do kterého je zasazena, rozvržením práce a končí řešerší již zpracovaných obdobných témat.

1.1 Popis řešeného problému

Ačkoliv máme možnost čerpat z mnoha zdrojů, ne vždy nalezneme to, co hledáme. Důvodů je samozřejmě mnoho: Známe velmi omezenou množinu zdrojů, neumíme se v nich zorientovat, nedokážeme je pro jejich komplexitu zpracovat, nevíme, které zdroje jsou ověřené, aktuální... Nic není samozřejmě ztraceno – může se tu objevit někdo, kdo je zmapuje, zagreguje, přetřídí, ohodnotí a pod jednotným rozhraním poskytne. Přesně tento úkol – umožnit studentům snadný přístup k informacím – je cílem diplomové práce.

Fakulta informačních technologií Českého vysokého učení technického v Praze, stejně tak, jako většina ostatních obdobných institucí, nabízí studentům nepřehledné množství informací. Některé se nacházejí na webových stránkách fakulty (<http://fit.cvut.cz/>), univerzity (<http://www.cvut.cz/>), jiné na serverech edux (<https://edux.fit.cvut.cz/>) a komponenta studia (<https://kos.cvut.cz/>), další pak pod Správou účelových zařízení (<http://agata.suz.cvut.cz/jidelnický/>) nebo třeba na webu Centra informačních a poradenských služeb ČVUT (<http://www.cips.cvut.cz/>).

Zorientovat se ve zdrojích nějakou dobu trvá, v těch nejdůležitějších zpravidla ne markantní, přesto existuje mnoho zdrojů, používaných méně často, na které nemusí potenciální uživatel vůbec narazit – jedním z nich je třeba seznam akcí ČVUT (<http://akce.cvut.cz/>). Velmi by proto pomohl nějaký průvodce, který by dostupné informace seskupil a koherentně poskytl.

1.2 Popis uživatele a jeho potřeb

Práce se zaměřuje především na studenty Fakulty informačních technologií Českého vysokého učení technického v Praze, přesto z ní mohou mít užitek i ostatní zainteresované strany – vyučující, neakademičtí pracovníci, zájemci o studium, návštěvníci. . . Kdokoliv pohybující se po dejvickém kampusu nebo hledající nějakou informaci vztaženou k fakultě.

Cílová skupina uživatelů chce zpravidla všechno a hned, nezabývá se proto rozsáhlými předpisy / návody a k informacím se buď dostane během krátké doby z nějakého vhodného zdroje, nebo vůbec. Vhodným zdrojem může být i to, co by pro většinovou populaci vhodné nebylo – studenti Fakulty informačních technologií mají nadprůměrně rozsáhlé znalosti v oblasti informačních technologií, proto může být vhodným zdrojem i komputerovaná řešení.

K předchozí potřebě se váží i nároky na použitelnost výsledného řešení – ačkoliv jsou studenti informačních technologií na ledač zvyklí a zvládnou tak využívat i nestandardní nástroje, mají na druhou stranu profesionální odpor k vyložení špatným řešením. Na přístupnost není třeba brát u této cílové skupiny zřetel.

Studenti zjišťují informace často až na poslední chvíli – například umístění učebny, ve které mají výuku, zjišťují až několik minut před jejím zahájením. V té době nemají možnost využít plnohodnotný počítač, ale musejí se spolehnout na omezené mobilní zařízení, které zpravidla mívají u sebe.

1.3 Vymezení cílů diplomové práce

Cílem diplomové práce je vytvořit průvodce sloužícího studentům jako jednotné rozhraní pro přístup k nejčastěji používaným / nejdůležitějším informacím a usnadnit jim tak určitou – nestudijskou – oblast studentského života. Práci je vhodné rozdělit do dvou částí – mobilní aplikaci, kterou budou studenti využívat, a serverovou aplikaci, která bude sloužit jako univerzální zdroj dat, i pro mobilní aplikaci.¹

Před započítáním implementace je třeba přibližně určit oblasti a rozsahy informací obsažených v průvodci. Následně je potřeba k požadovaným informacím nalézt vhodné zdroje. Neexistující zdroje je třeba nahradit simulovanými zdroji. Přichází na řadu analýza, návrh a realizace nejprve serverové, poté mobilní aplikace. Testování je vhodné provádět po celou dobu.

Mobilní aplikace by měla být dostupná velkému počtu studentů, měla by tedy být multiplatformní a fungovat (jak plyne z označení) i na mobilních zařízeních. Využití aplikace se předpokládá až v následujících letech, není tedy

¹ Pojmy *mobilní* a *serverová aplikace* se vyskytují napříč celou prací, až na výjimky všude reprezentují aplikaci sloužící studentům jako průvodce (*mobilní aplikace*) a aplikaci shromažďující data z nestrukturovaných zdrojů a poskytujících je unifikovaným způsobem (*serverová aplikace*).

třeba podchytávat stará a méně schopná zařízení – s jejich podporou se již nepočítá. Specifickým zdrojem poskytovaných informací by měla být navigace.

Serverová aplikace by měla být přístupná pouze administrátorům a uživatelům by sloužilo čistě jen vhodné rozhraní pro získávání požadovaných informací. Informace mají být v systému uloženy dle, na systému nezávislého, vhodného modelu.

Práce nemá za cíl plně obsáhnout všechny zpracovávané domény, tím je kvalitní zpracování základu každé domény, které pak lze dle již hotových částí doplnit do celého, pro praktické nasazení potřebného, rozsahu. Postačuje tedy vytvoření částí odpovídajících reprezentativnímu vzorku.

1.4 Popis struktury DP ve vztahu k vytyčeným cílům

Diplomová práce má být členěna do tří hlavních částí:

- Analýza s návrhem.
- Realizace.
- Testování.

Chronologické uspořádání částí je o trochu komplikovanější – začíná se analýzou, která přechází v návrh, po dokončení se vše realizuje, po celou dobu se průběžně testuje. Po ukončení této iterace se začíná nová, se širším záběrem. Tento postup se opakuje až do vytvoření aplikace.

Ve fázi analýzy a návrhu dochází nejprve k průzkumu zpracovávané domény, vytvoření uživatelských scénářů a vytyčení požadavků na vytvářenou aplikaci, později je navržena struktura aplikace, navržena bezpečnost a jsou voleny použité technologie.

Realizace by měla být pouhým mechanickým převedením navrženého systému do reálného světa, bohužel se v praxi vyskytují neočekávané problémy, které je třeba během realizace vyřešit. Vzhledem k akademickému účelu této práce je vhodné uvést obě řešení – nakonec implementované i to zamýšlené ideální.

Testování je třeba provádět na mnoha úrovních – od kontroly zdrojového kódu, přes integrační testování, testování systémové integrace, zátěžové testování, až po testování použitelnosti.

1.5 Serverová aplikace

Systém, který se skrývá pod pojmem *serverová aplikace* (viz vysvětlení v poznámce 1 na straně 20), je velmi specifický a proto pravděpodobně nemá

odpovídající obdoby. Jeho kruciální část, model reprezentující informace potřebné pro průvodce, ale obdob má velmi mnoho, proto byla rešerše prováděna hlavně v této oblasti.

1.5.1 Rešeršní zpracování existujících ontologií

Doposud bylo vytvořeno velké množství univerzitních ontologií – jedná se totiž o oblíbené téma domácích úkolů předmětů zabývajících se sémantickým webem, z toho ale plyne i druhá častá vlastnost těchto ontologií – nebývají příliš propracované a odzkoušené. Kvalitních ontologií je tedy pouze malý zlomek.

1.5.1.1 Univ-Bench

Univ-Bench (<http://swat.cse.lehigh.edu/projects/lubm/>) je, pravděpodobně, nejznámější ontologií modelující univerzitu – jedná se totiž o ontologii využitou v The Lehigh University Benchmark (LUBM) – nástroji pro srovnávací testování úložišť sémantických dat [5]. Ontologie se snaží být co nejrealističtější, vzhledem ke svému účelu se ale nemůže dále rozvíjet. Autorem je Zhengxiang Pan z Lehigh University. Aktuální verze pochází z roku 2004.

1.5.1.2 Academic Institution Internal Structure Ontology

Academic Institution Internal Structure Ontology (AIISO) (<http://purl.org/vocab/aiiso/schema>) je, pravděpodobně, nejpropracovanější ontologií modelující univerzitu. Ve spojení s *Participation* (<http://purl.org/vocab/participation/schema>), Friend of a friend (FOAF) (<http://xmlns.com/foaf/0.1/>) a AIISO Roles (<http://purl.org/vocab/aiiso-roles/schema>) popisuje téměř celou doménu univerzity [9]. Ontologie je uvolněna pod licencí Creative Commons (attribute). Autory jsou Rob Styles a Nadeem Shabir z Talis Information Ltd. Aktuální verze pochází z roku 2008.

1.5.1.3 Higher Education Reference Ontology

Higher Education Reference Ontology (HERO) (<http://sourceforge.net/projects/heronto/>) je další ontologií mající za cíl pokrýt celou doménu jakékoliv univerzity. Ontologie je uvolněna pod Adaptive Public License. Autorkou je Leila Zemmouchi-Ghomari. Aktuální betaverze, aktivně vyvíjená, pochází z roku 2012 [13].

1.5.1.4 University Ontology

University Ontology (<http://purl.org/weso/uni>) je univerzitní ontologií vycházející z AIISO, FOAF a An organization ontology (Org). Je uvolněna pod licencí Creative Commons (attribute). Autorem je Jose Emilio Labra Gayo z

výzkumné skupiny WESO na Universidad de Oviedo. Aktuální verze pochází z roku 2011 [4].

1.5.1.5 University Ontology

University Ontology (<http://www.cs.umd.edu/projects/plus/SHOE/onts/univ1.0.html>) náleží do projektu Simple HTML Ontology Extensions (SHOE), který má poněkud širší záběr. Autorem je Jeff Heflin z Lehigh University (SHOE, jako celek, ale patří pod katedru informatiky na University of Maryland). Aktuální verze pochází z roku 2000 a v současné době už není udržovaná [6].

1.6 Mobilní aplikace

Na poli mobilních aplikací sloužících jako průvodce studenta je konkurence podstatně větší, než tomu bylo u serverové. Mnoho jich je dokonce napojených na externí zdroj dat, žádná z objevených ale nepracuje přímo se sémantickým úložištěm.

1.6.1 Rešeršní zpracování existujících mobilních aplikací

Práce zpracovávanou doménou navazuje na bakalářskou práci *Mobilní navigační systém pro FEL ČVUT* [8], ta ale není jediným zdrojem, který byl prozkoumán. Ostatní uvedené aplikace zpravidla směřují jinými směry, proto jim nebude věnováno tolik prostoru.

Jako důsledek relativně nedávného prudkého vzestupu počtu chytrých telefonů se za uplynulé roky objevilo mnoho aplikací zabývajících se podobnými tématy, jako tato práce.² Ne vždy se jedná o studentovy průvodce po univerzitě, problém ale řeší obdobný. Nabídka jejich funkcí bývá vzájemně podobná, proto zde uvedu pouze výběr.

V první části této sekce se budu věnovat pracím obdobným této, později se budou objevovat práce stále více odlišné, přesto ale v některých ohledech inspirativní.

1.6.1.1 Mobilní navigační systém pro FEL ČVUT

Bakalářská práce *Mobilní navigační systém pro FEL ČVUT* [8] je zaměřená především na multiplatformnost a použitelnost. Bylo tedy nutné v některých oblastech (včetně funkcionality) podstoupit určité kompromisy. Aplikace vyhledá místnost na základě zadaného označení a přehledným plánem k ní uživatele navede. Označením může být oficiální název, přezdívka, jméno sídlícího pracovníka, či jiný identifikátor. Dále aplikace vizualizuje body zájmu, jako

²Zpracování těchto aplikací bývá ovšem jiné, jsou zaměřené na užitek a sofistikovanost zpracování zpravidla nehraje roli.

jsou občerstvení, výtahy, toalety. . . Práce poskytuje užitečného pomocníka, po čistě programové stránce ale moc zajímavá není.³

1.6.1.2 Průvodce ČVUT

Nyní již zastaralá mobilní aplikace *Průvodce ČVUT FEL* (<http://lr.czechian.net/j2me/>⁴) vytvořená v J2ME poskytovala vyhledávání místností s následným textovým popisem cesty, případným minimalistickým plánkem a dopravním spojením z frekventovaných lokalit.

1.6.1.3 ČVUT navigátor

Projekt *ČVUT navigátor* (<http://navigator.fit.cvut.cz/>) je tématem několika akademických prací. Vypadá slibně, ale ani v době dokončování této práce není známo více, než cíle vytyčené v době založení projektu.

1.6.1.4 Ostatní průvodci ČVUT

Mezi další, v souvislosti s touto prací méně významné, průvodce patří tištěný *Průvodce prváka po ČVUT* [2] nebo například specializovaný *Průvodce prváka po koleji Podolí* (<http://pruvodce.pod.cvut.cz/>).

1.6.1.5 Vlastní průvodci

Mobilní průvodci univerzitami, tak jako se děje i v jiných oblastech mobilních aplikací, se v poslední době začínají velmi rozmáhat. Velké množství univerzit nabízí svým studentům na vlastní půdě vytvořené aplikace, ať již studenty, nebo výpočetními centry. Tyto aplikace bývají šité na míru potřebám univerzit, na druhou stranu ale nemusí být tolik odladěné, jako to bývá u větších projektů. Nabídka funkcionalit sahá od minimalistických až po velmi rozsáhlé. S podporovanými platformami je to podobné – od jedné až po univerzální aplikace.

1.6.1.6 campusM

Univerzity jako Imperial College London, London School of Economics, University College London, Trinity College Dublin a pár desítek dalších využívají komerční aplikace z dílen společnosti *campusM* (<http://www.campusm.com/>).

³Cílem této, diplomové, práce je naopak vytvoření robustní ontologie a samotná mobilní aplikace slouží pouze jako ukázka implementace za využití moderních mobilních technologií, takže ke kompromisům dojde v rozdílných oblastech. Návaznost na bakalářskou práci se bude projevovat převážně v poučení se z předchozích chyb ve shodné zpracovávané doméně.

⁴Aplikace *Průvodce ČVUT FEL* byla, pravděpodobně kvůli neaktualizovanému obsahu a celkovému nedokončení, stažena a nyní již není veřejně dostupná. Nachází se na přiloženém Compact Disc (CD) v adresáři *data/Průvodce ČVUT FEL*.

Řešení je nabízeno pro různé mobilní platformy a nabízí následující funkcionality:

- Vyhledání cesty.
- Lokalizace přátel.
- Novinky a události.
- Upozornění studentů.
- Integrace se systémem knihovny.
- Integrace s univerzitním adresářem.
- Přístup k rozvrhům.
- Informace o předmětech.

1.6.1.7 guidebook

guidebook (<http://guidebook.com/>) je typickým představitelem aplikace pro podporu určité události v širším slova smyslu – může sem spadat i průvodce školou. Ve webovém rozhraní pořadatel událost vytvoří a upraví podle specifických potřeb, na základě toho jsou vytvořeny aplikace pro různé mobilní platformy a ty pak jsou nabízeny návštěvníkům dané události. *guidebook* nabízí obdobné funkcionality výše uvedeným a je kompletně zdarma pro omezený počet uživatelů. Aplikace je pro diplomovou práci inspirativní záběrem a profesionalitou zpracování.

1.6.1.8 Ostatní, neuniverzitní, průvodci

V této široce zaměřené podpodsekci zmíním hlavně průvodce z jednotlivých akcí – ti totiž představují nejmohutnější podmnožinu. V poslední době se stává, hlavně na technicky zaměřených odborných konferencích,⁵ zvykem poskytnout účastníkům mobilní aplikaci disponující informacemi o konferenci – navozuje to profesionální dojem, přináší více možností (ankety, upomínky...) a jsou tu mimo jiné i taková pozitiva, jako možnost aktualizace v případě nenadálé události. Další mobilní průvodci se vyskytují například na hudebních festivalech nebo po městech.

⁵Na těchto konferencích má většina účastníků u sebe zařízení schopné s touto aplikací pracovat.

1.6.1.8.1 Příklady funkcionalit: Žádná z nalezených aplikací nenabízí všechny níže uvedené funkcionality, poměrně velká část se k tomu ale blíží.

- Registrace účastníků.
- Zobrazování harmonogramu.
- Sestavení vlastního programu s upomínkami.
- Zobrazování plánů.
- Zobrazování informací o přednášejících, vystavovatelích, účinkujících. . .
- Zobrazování doplňujících materiálů (slidy a podrobnosti k přednáškám, texty písní u koncertů. . .).
- Interakce se sociálními sítěmi.
- Získávání zpětné odezvy (otázky přednášejícím, ankety. . .).
- Vytvoření prostoru pro reklamu sponzorů.

1.6.1.8.2 Příklady aplikací: Po úvaze jsem se rozhodl odkázat pouze na několik reprezentantů – aplikací je velmi mnoho a vzhledem k podobnosti by nemělo přínos uvádět všechny. Jako ukázkou předkládám jednu svobodnou a několik komerčních aplikací:

iosched (<http://code.google.com/p/iosched/>) je svobodnou aplikací vyvíjenou pro konferenci Google I/O. Nabízí všechny výše uvedené funkcionality vyjma: registrace, sestavování vlastního programu a prostoru pro reklamu, což má svá opodstatnění. Aplikace je napsána v Javě a je určena pro běh v mobilních telefonech se systémem Android. *iosched* je pro diplomovou práci inspirativní záběrem zpracování a možností poučení se z otevřeného zdrojového kódu.

Google Maps Floor Plans (<http://maps.google.com/help/maps/floorplans/>) je dalším, v tomto případě již nesvobodným, nástrojem společnosti Google poskytujícím veřejnosti možnost tvorby plánů vnitřních prostor budov. Tyto plány budou v budoucnu – *Google Maps Floor Plans* je stále ve vývoji – zaneseny do mapových podkladů *Google Maps for Android*. Aplikace je tedy pro diplomovou práci inspirativní hlavně v oblasti navigace uvnitř budov.

Analýza a návrh implementace

Analýza s návrhem probíhaly nezávisle na dvou místech – po krátké analýze celku a určení vzájemných komunikačních rozhraní byly provedeny analýza s návrhem serverové aplikace a až po jejím dokončení začaly analýza s návrhem mobilní aplikace. V průběhu času jsem se k analýze a návrhu ještě několikrát vrátil a vylepšil / rozšířil to, co se ne vždy na poprvé povedlo.

Analýza se skládala z identifikace potřeb uživatelů, vytvoření případů užití, ze kterých byly odvozeny funkční požadavky, ke kterým byly doplněny nefunkční požadavky. Poté byl vytvořen návrh řešení, zapracován návrh bezpečnosti a celý koncept byl podložen návrhem nasazení. Nakonec byly voleny technologie.

2.1 Serverová aplikace

Začal jsem pracovat nejprve na serverové aplikaci – jednak na ní měla být ta mobilní závislá (sama na žádné jiné části práce závislá není) a také má užší vazbu ke zpracovávané doméně.

2.1.1 Potřeby uživatelů

Tato část se věnuje potřebám uživatelů serverové aplikace, zejména uživatelů rozhraní, přes které poskytuje informace. Potřeb je mnoho – proto zde uvedu pouze reprezentativní vzorek.

2.1.1.1 Potřeba zjistit informaci

Potřeba zjistit informaci je základní potřebou, kvůli které se mají uživatelé obracet na systém. Tato potřeba může vzniknout z velkého množství příčin – od hledání informací spjatých s průchodem studiem, až například po zjišťování jídelního lístku v menze.

2.1.1.2 Potřeba ověřit informaci

Často je cílová skupina uživatelů v situaci, kdy má požadovanou informaci, není si jí ale jistá a potřebuje ji ověřit / vyvrátit. Bez jistoty pravdivosti informace je její hodnota podstatně nižší.

2.1.1.3 Potřeba ověřených informací

Velmi podstatnou potřebou je potřeba ověřených informací. Ačkoliv je název velmi podobný té předchozí, jedná se o dvě diametrálně odlišné potřeby, předchozí se bez naplnění této neobejde. Jakmile jsou informace poskytovány, je nutné, aby byly ověřené – pravdivé.

2.1.1.4 Potřeba aktuálních informací

Potřeba aktuálních informací opět souvisí s již zmíněnými potřebami – sebepravdivější jídelní lístek z předchozího dne mi při výběru dnešního oběda bohužel moc nepomůže.

2.1.1.5 Potřeba kompletních informací

Někdy nestačí jen pravdivé a aktuální informace, naopak vědomí záruky pravdivosti a aktuálnosti může i uškodit – pokud totiž informace nejsou kompletní, může dojít k jejich misinterpretaci a pod vlivem záruk nedojde ke kritickému posouzení. Proto je zde i potřeba kompletních informací.

2.1.1.6 Potřeba nalézt, co nehledám

Na první pohled tato potřeba nedává smysl, v praxi se ale často vyskytují informace, které bychom měli znát a o jejich existenci nemáme ani ponětí. Je proto důležité ve vhodné chvíli sdělit i nepožadované informace, které jsou s velkou pravděpodobností pro adresáta důležité.

2.1.2 Případy užití

Případy užití (*use cases*) jsou takové sekvence kroků, které reprezentují danou činnost prováděnou aktérem na systému a vedou od jejího zahájení až po naplnění.

V serverové aplikaci byli identifikováni celkem tři aktéři:

- Uživatel – student / softwarová komponenta (*mobilní aplikace*).
- Správce – administrativní pracovník fakulty.
- Plánovač – softwarová komponenta.

Roli správce by šlo dále dělit na další role, vzhledem k velmi nízkému rozsahu činností každé z nich (zpravidla jedna činnost), je vše zahrnuto pod tímto souhrnným názvem.

Uživatel se pouze dotazuje nad systémem a zjišťuje informace. Správce manuálně upravuje informace v systému, může vynutit aktualizaci a jinak spravuje systém. Plánovač automaticky obsluhuje aktualizace obsažených informací.

2.1.2.1 UC1: Vyžádání informací

Aktéři:

- Uživatel.

Systém:

- Serverová aplikace.

Vstupní podmínky:

- Uživatel zná jazyk dotazovacího rozhraní systému.
- Uživatel má přístup k dotazovacímu rozhraní systému.

Hlavní scénář:

- Uživatel vytvoří dotaz.
- Uživatel odešle dotaz na rozhraní systému.
- Systém dotaz přijme.
- Systém dotaz zpracuje.
- Systém odešle odpověď.
- Uživatel přijme odpověď.
- Uživatel zpracuje odpověď.

Alternativní scénáře:

- V případě neplatného dotazu je navraceno chybové hlášení.
- V případě chyby serveru je navraceno chybové hlášení.

Poznámky:

- V případě nenalezení informace je navracena prázdná množina.
- Pokud je uživatelem softwarová komponenta, student ji obsluhující nemusí znát jazyk dotazovacího rozhraní – může s komponentou komunikovat jiným způsobem a ta si požadavky do požadovaného jazyka převádět.

2.1.2.2 UC2: Aktualizace informací

Aktéři:

- Plánovač.

Systém:

- Serverová aplikace.

Vstupní podmínky:

- Plánovač je řádně nastaven a spuštěn.

Hlavní scénář:

-

Poznámky:

- Případné chyby jsou zaznamenány do logu.
- Aktualizaci může obdobným způsobem vynutit i správce.

2.1.2.3 UC3: Správa informací

Aktéři:

- Správce.

Systém:

- Serverová aplikace.

Vstupní podmínky:

- Správce má přístup do systému.
- Správce zná jazyk rozhraní správy systému.

Hlavní scénář:

-

Poznámky:

- Případné chyby jsou zaznamenány do logu.

2.1.3 Funkční požadavky

Funkčními požadavky (*functional requirements*) jsou myšleny takové požadavky, které jsou zaměřeny na jednotlivé funkcionality. Pro serverovou aplikaci mezi ně patří především:

1. Aplikace bude čerpat informace o názvech budov z předstíraného adaptéru.
2. Aplikace bude čerpat informace o umístění budov z předstíraného adaptéru.
3. Aplikace bude čerpat informace o názvech místností z předstíraného adaptéru.
4. Aplikace bude čerpat informace o umístění místností z předstíraného adaptéru.
5. Aplikace bude čerpat informace o rozvrzích místností z KOSapi (<http://kosapi.fit.cvut.cz/>). **TODO** Vynechám – chybí API.
6. Aplikace bude čerpat informace o rozvrzích studentů z formátu iCalendar.
7. Aplikace bude čerpat informace o kontakech studentů z Usermap ČVUT (<http://usermap.cvut.cz/>).
8. Aplikace bude čerpat informace o rozvrzích vyučujících z KOSapi (<http://kosapi.fit.cvut.cz/>). **TODO** Vynechám – chybí API.
9. Aplikace bude čerpat informace o kancelářích vyučujících z Usermap ČVUT (<http://usermap.cvut.cz/>).
10. Aplikace bude čerpat informace o kontakech vyučujících z Usermap ČVUT (<http://usermap.cvut.cz/>).
11. Aplikace bude čerpat informace o kancelářích neakademických pracovníků z Usermap ČVUT (<http://usermap.cvut.cz/>).
12. Aplikace bude čerpat informace o kontaktech neakademických pracovníků z Usermap ČVUT (<http://usermap.cvut.cz/>).

13. Aplikace bude čerpat informace o akcích ČVUT z Kalendáře akcí ČVUT (<http://akce.cvut.cz/>).
14. Aplikace bude čerpat informace o harmonogramu akademického roku z předstíraného adaptéru (na základě <http://fit.cvut.cz/>).
15. Aplikace bude čerpat informace o datech MAR dostupných zdrojů (prozatím Orlík a Masarykova kolej, snaha o T9).
16. Aplikace bude čerpat informace o jídelničkách menz z Jídelníčků menz (<http://agata.suz.cvut.cz/jidelnicky/>).
17. Aplikace bude čerpat informace o otvírací době menz z Jídelníčků menz (<http://agata.suz.cvut.cz/jidelnicky/>).
18. Aplikace bude čerpat informace o otvírací době NTK z Národní technické knihovny (<http://www.techlib.cz/>).
19. Aplikace bude čerpat informace o otvírací době Vydavatelství průkazů (<http://www.cvut.cz/informace-pro-studenty/prukazy>). **TODO** Vynechám – chybí API.
20. Aplikace bude čerpat informace o počtu lidí ve frontě ve Vydavatelství průkazů (<http://ke.customer.decent.cz/a021/mon/>).
21. Vytvořte / nalezněte / sestavte z dílčích částí ontologii reprezentující výše uvedené informace.
22. Informace získané z jednotlivých zdrojů převádějte dle výše požadované ontologie pro další zpracování do RDF.
23. Aplikace uloží získané sémantické informace do databáze.
24. Aplikace umožní vykonávat SPARQL dotazy nad uloženými daty.

2.1.4 Nefunkční požadavky

Nefunkčními požadavky (*non-functional requirements*) jsou myšleny takové požadavky, které jsou zaměřeny na dílo jako celek, nikoliv na jednotlivé funkcionality. Pro serverovou aplikaci mezi ně patří především:

1. Aplikaci naprogramujte v *JavaScriptu*.
2. Pro běh serveru využijte *Node.js*.
3. Sémantická data ukládejte do takového úložiště, aby nad ním bylo možné vykonávat dotazy přímo (případně přes nezávislého prostředníka) a nebylo nutné přistupovat přes vlastní aplikaci.

4. Práce nemusí nutně obsáhnout celou zpracovávanou doménu, je ale žádané, aby kvalitně zpracovala hlavní scénáře využití a nebránila dalšímu rozšiřování.
5. Zveřejněte aplikaci a její data pod svobodnými licencemi z rodin *GNU*, *Creative Commons* a kompatibilními, aby byl umožněn další rozvoj nezávislý na původním autorovi.

2.1.5 Návrh řešení

TODO Diskuse alternativ. Struktura.

TODO Volba implementačního prostředí.

2.1.6 Návrh bezpečnosti

TODO Injekce kódu. Pouze povolené URL. Ošření vstupů.

2.1.7 Návrh nasazení

TODO Deployment diagram.

2.1.8 Použité technologie

Mnoho použitých technologií bylo dáno zadáním práce, byl mi tak do značné míry usnadněn jejich výběr. Na druhou stranu nejsou dané technologie dokonalé, takže by často bývala práce s jinými snazší.

Celá serverová aplikace je napsána v *JavaScriptu*, zprovozněna na *Node.js*, datovým úložištěm se stalo *MongoDB*.

2.1.8.1 JavaScript

Využití *JavaScriptu* bylo pro svoje pravděpodobné vzrůstající nasazení v budoucnosti a své specifické vlastnosti dáno jako nefunkční požadavek.

2.1.8.2 Node.js

Node.js (<http://nodejs.org/>) je platforma postavená nad běhovým prostředím *JavaScriptu* z *Chrome*. Slouží ke tvorbě rychlých, škálovatelných síťových aplikací. *Node.js* využívá událostmi řízený neblokovací I/O model, který ho činí nenáročným a efektivním, vhodným pro datově náročné *real-time* aplikace běžící v distribuovaném prostředí [12].

Použití *Node.js* bylo vynuceno nefunkčním požadavkem, jednalo se ale o velmi cennou zkušenost – je novou technologií a ačkoliv je jeho dokumentace na slušné úrovni, není zatím vytvořena dostatečně velká báze příkladů a nejlepších praktik, takže jsem se k nim musel postupně dostat a při té příležitosti se toho hodně naučil. Navíc, zdá se, se budou nabité vědomosti v budoucnu hodit.

2.1.8.3 MongoDB

MongoDB (<http://www.mongodb.org/>) je škálovatelnou, vysoce výkonnou, otevřenou NoSQL databází napsanou v C++ [11].

Celé prostředí serverové aplikace je umístěno v serverové edici GNU/Linuxové distribuce *Ubuntu* (<http://www.ubuntu.com/>) a ta je pro snadnou přenositelnost nainstalována do kontejneru virtuálního stroje virtualizačního řešení *VirtualBox* (<http://www.virtualbox.org/>).

2.1.9 Použité knihovny

Mimo standardního JavaScriptu lze v Node.js využívat i specifická rozšíření a množství modulů třetích stran. Z těch jsem využil:

- *date-utils* (<https://github.com/JerrySievert/node-date-utils>) – mikro framework doplňující funkcionalitu JavaScriptového objektu *Date*. Zvažováno bylo i použití *dateformat* (<https://github.com/felixge/node-dateformat>), ten je ale pro projekt méně vhodný. Oba moduly jsou stále ve vývoji.
- *express* (<http://expressjs.com/>) – renomovaný framework pro tvorbu webových aplikací založený na middleware frameworku *connect* (<http://www.senchalabs.org/connect/>).
- *iCalendar* (<https://github.com/tritech/node-icalendar>) – knihovna pro práci s formátem *iCalendar*. Modul je stále ve vývoji. Zvažovány byly i další knihovny, například *ical* (<https://github.com/peterbraden/node-ical>), ty ale implementovaly ještě méně možností, než použitá.
- *iconv* (<https://github.com/bnoordhuis/node-iconv>) – knihovna pro obsluhu systémového příkazu *iconv* sloužícímu k převádění kódování.
- *jquery* (<https://github.com/coolaj86/node-jquery>) – wrapper umožňující využít renomovanou JavaScriptovou knihovnu *jQuery* (<http://jquery.com/>) v prostředí Node.js. Je zde využít *jsdom* (<https://github.com/tmpvar/jsdom>) – JavaScriptová implementace DOM.
- *rdfstore* (<https://github.com/antoniogarrote/rdfstore-js>) – JavaScriptová implementace *RDF* grafového úložiště s podporou pro *SPARQL*. Data mohou být (a v implementaci DP jsou) persistentně ukládána prostřednictvím Node.js modulu-ovladače *mongodb* (<https://github.com/christkv/node-mongodb-native>) do NoSQL databáze *MongoDB* (<http://www.mongodb.org/>). Modul je stále ve vývoji.

- *soap* (<https://github.com/milewise/node-soap>) – **TODO** Pouze pokud se objeví podpora poskytnutého WSDL.

2.1.10 Použité nástroje

2.2 Mobilní aplikace

2.2.1 Potřeby uživatelů

2.2.1.1 Potřeba určit cílovou pozici

2.2.1.2 Potřeba určit aktuální pozici

TODO Seznam a popis jednotlivých potřeb uživatelů.

2.2.2 Prototypy

TODO Pojednání o vytvořených prototypech, možná nakonec pod něco zařadím, nebude mít velký rozsah.

2.2.3 Případy užití

TODO Různé případy užití v textové formě.

2.2.4 Funkční požadavky

Funkčními požadavky (*functional requirements*) jsou myšleny takové požadavky, které jsou zaměřeny na jednotlivé funkcionality. Pro mobilní aplikaci mezi ně patří především:

1. Aplikace umožní pokročilému uživateli zadat vlastní SPARQL dotaz. Ten se odešle na server, kde se zpracuje, a výsledek se navrátí uživateli.
2. Pro méně znalé uživatele SPARQL a pro uživatele minimalizující použití klávesnice vytvořte předpřipravené dotazy pro základní scénáře použití aplikace.
3. Implementujte funkcionalitu umožňující určení aktuální pozice a její předání dalšímu zpracování.
4. Aplikace dokáže na mapě vizualizovat určitou pozici.
5. Aplikace dokáže vizualizovat HTML data nebo data jako HTML.
6. Aplikace bude poskytovat nápovědu pro snazší použití.
7. Aplikace bude postavená nad daty reprezentovanými výše uvedenou ontologií.

2.2.5 Nefunkční požadavky

Nefunkčními požadavky (*non-functional requirements*) jsou myšleny takové požadavky, které jsou zaměřeny na dílo jako celek, nikoliv na jednotlivé funkcionality. Pro mobilní aplikaci mezi ně patří především:

1. Aplikaci vytvořte v HTML5, aby byla na současných, obzvláště mobilních, zařízeních multiplatformní.
2. Aplikace bude funkční v prohlížečích na nejnovějších verzích OS Android, iOS a Symbian.
3. Aplikace bude funkční v majoritních desktopových prohlížečích.
4. Při vývoji se zaměřte i na použitelnost, aplikaci by měl být schopný, po krátkém zaškolení, používat jakýkoliv běžný student Fakulty informačních technologií.
5. Snažte se nabízet alternativy k zadávání dlouhých textů, aby byla usnadněna obsluha uživatelům mobilních zařízení bez hardwarových klávesnic.
6. Práce nemusí nutně obsáhnout celou zpracovávanou doménu, je ale žádané, aby kvalitně zpracovala hlavní scénáře využití a nebránila dalšímu rozšiřování.
7. Zveřejněte aplikaci a její data pod svobodnými licencemi z rodin GNU, *Creative Commons* a kompatibilními, aby byl umožněn další rozvoj nezávislý na původním autorovi.
8. Aplikace umožní využití off-line. **TODO** Vynechám – muselo by se složitě indexovat.

2.2.6 Návrh řešení

TODO Diskuse alternativ.

TODO Volba implementačního prostředí.

TODO Struktura.

2.2.7 Návrh bezpečnosti

TODO Injekce JS, ošetření vstupů, white-listing URL, elementů, atributů, zakázání cizích zdrojů...

2.2.8 Návrh nasazení

TODO Deployment diagram.

2.2.9 Použité technologie

Níže uvádím soupis použitých technologií s odůvodněním, proč byly vybrány a v jakém kontextu použity.

2.2.9.1 HTML5

Většina technologií tvořících mobilní aplikaci by se dala zahrnout do pojmu HTML5 (v širším smyslu významu). Pomocí JavaScriptu se manipuluje s HTML strukturou, která je prezentována za pomoci CSS, klíčová část aplikace – mapa – je tvořena SVG (o tom ještě dále). Technologie byly vybrány pro jejich výbornou portovatelnost – téměř všechna dnešní zařízení je implementují podobně (nebo alespoň mají takové výkonnostní parametry, že na nich lze provozovat přidanou vrstvu abstrahující od skutečné implementace).

2.2.9.2 SVG

Mapa použitá v aplikaci je implementována prostřednictvím SVG. Rozhodnutí bylo učiněno na základě identifikace a porovnání možných technologií:

- Statické obrázky s předgenerovanými proměnnými částmi.
- Statické obrázky s dynamicky pozicovanými proměnnými částmi.
- Dynamicky kreslené obrázky na *canvas*.
- Statické SVG s dynamickými úpravami.

První zmíněné řešení bylo zavrženo kvůli nízké flexibilitě – není možné dopředu vygenerovat všechny varianty obrázků, které by pokrývaly možné situace – i kdyby to možné bylo, množství obrázků by bylo velmi vysoké⁶ a pro omezenou paměťovou kapacitu cílových zařízení nevhodné.

Druhý přístup, dynamické pozicování, byl zavržen z důvodů optimalizací rozvržení prvků mobilními prohlížeči – není zaručené, že se prvek (například ukazatel nalezené místnosti) skutečně umístí, kam má.

Nová vlastnost HTML5, kreslení prostřednictvím JavaScriptu na *canvas* element, byla, narozdíl od předchozích možností, zvažována až do finálního rozhodnutí – ačkoliv se jedná o řešení čistě v režii již používaných technologií (JavaScript, HTML), jeho využití jsem z několika důvodů zavrhl:

- Neumožňuje nakreslený obrázek nijak upravit (například posunout ukazatel na mapě), ale musí se vždy překreslit, což je náročnější na zdroje.
- Kvůli principu dynamického generování obrázků JavaScriptem nelze kostru připravit pomocí běžných grafických editorů.

⁶Pro pokrytí všech situací by byly třeba zvlášť obrázky s vyznačením kterékoliv místnosti, kterékoliv bodu zájmu, kterékoliv výchozí pozice... Vzhledem k rozsáhlosti mapy by tak počet obrázků byl enormní.

- Subjektivně mi přijde práce s mapou jako SVG obrázkem reprezentovaným XML dokumentem majícím DOM přehlednější, než s méně strukturovatelným JavaScriptem.

Reprezentaci mapy jsem tedy svěřil SVG z následujících důvodů:

- Snadné vytvoření mapy prostřednictvím běžného grafického editoru.
- Možnost znovupoužití mapy standardního grafického formátu i v jiných projektech.
- Práce s mapou (něpř. vyznačování pozic) se provádí pohodou úpravou jejího DOM.
- I po úpravách se stále jedná o týž objekt – není nutné ho překreslovat.

Nutno podotknout, že jsou implementace kreslení na *canvas* a SVG na různých mobilních zařízeních na různých úrovních, ani jedna ovšem nemá výrazně navrch [1] [3].

2.2.9.3 PHP

Z důvodů popsaných v podsekcí 3.2.1.4 (str. 42) bylo nutné vytvořit podpůrnou serverovou aplikaci – proxy. Po úvaze jsem se rozhodl pro využití jazyka PHP, který je podporovaný jako jediný nástroj pro tvorbu dynmického obsahu na fakultním webovém serveru <http://webdev.fit.cvut.cz/>. Tuto kombinaci, vzhledem k její podpoře ze strany fakulty a určení práce pro fakulní nasazení, považuji i přes jinde využitý JavaScript za vhodnou.

2.2.10 Použité knihovny

V práci byla, po úvaze, nakonec využita i práce třetích stran – svobodné knihovny *jQuery* a *HTML Purifier*. Jejich krátký popis a odůvodnění následují:

2.2.10.1 jQuery

A

2.2.10.2 HTML Purifier

Po několika vlastních implementacích nástroje na pročištění a ošetření HTML vstupů pocházejících od třetích stran jsem sáhl k implementaci již prověřené – oblast ošetření aplikace před škodlivými vstupy má z hlediska bezpečnosti vysokou prioritu a neustále se objevující nové hrozby nakonec vedly k tomu, že jsem se rozhodl nespolehat na své vlastní implementace a využít *know-how* tvůrců svobodné PHP knihovny HTML Purifier (<http://htmlpurifier.>

org/) – ta odstraní případná nebezpečí za využití pokročilého auditovaného *whitelistu* a v případě potřeby navíc zajistí validitu kódu. Knihovna je využita v rámci proxy zmíněné v předchozím odstavci.

2.2.11 Použité nástroje

Pro tvorbu aplikace byly použity převážně následující nástroje. Majorita jich je uvolněných pod svobodnými licencemi, zpravidla GNU GPL nebo kompatibilními. Jedinou výjimkou jsou některé mobilní prohlížeče, které byly použity k testování aplikace.

2.2.11.1 Kate

Většina práce, včetně mobilní aplikace, byla vytvořena v textovém editoru Kate (<http://kate-editor.org/>). Jedná se o velmi pokročilý nástroj pro práci s textem z prostředí GNU/Linuxového desktopového prostředí KDE. Kate nabízí zvýrazňování syntaxe všech v diplomové práci použitých formátů, poskytuje velmi užitečnou práci s regulárními výrazy, umožňuje pracovat s mnoha soubory najednou a, co je asi nejpodstatnější, plně se integruje do KDE, takže ním například lze prostřednictvím KDE Input/Output (KIO) transparentně pracovat se vzdálenými soubory. Kate neumožňuje automatické doplňování kódu, nabízí ale již použitá slova kdekoli v dokumentu, což je vlastnost v některých případech i lepší. Kate byl zvolen na základě velice kladných zkušeností pro napsání kódu v JavaScriptu, HyperText Markup Language (HTML), Cascading Style Sheets (CSS) i úpravy mapy v Scalable Vector Graphics (SVG).

2.2.11.2 Inkscape

Grafické podklady práce byly vytvořeny v editoru vektorové grafiky Inkscape (<http://www.inkscape.org/>). Nástroj implementuje téměř kompletní specifikaci formátu SVG verze 1.1, je multiplatformní a svobodný. Inkscape byl vybrán kvůli předchozím výborným zkušenostem s prací v něm a vhodnosti pro tvorbu mapových podkladů.

2.2.11.3 Kompare

V případě potřeby důkladného porovnání souborů byl použit Kompare (<http://www.caffeinated.me.uk/kompare/>). Nástroj umožňuje velmi přehledně porovnat mimo jednotlivých souborů i celé jejich adresáře. Kompare neporovnává soubory sám, ale je nadstavbou nad konzolovou aplikací diff či obdobnou.

2.2.11.4 Webové prohlížeče

Aplikace byla průběžně testována v různých mobilních prohlížečích:

- Mozilla Firefox (<http://www.mozilla.org/firefox>).
- Chromium (<http://chromium.org/>).
- rekonq (<http://rekonq.kde.org/>).
- Prohlížeč platformy Android (<http://www.android.com/>).
- Google Chrome for Android (<https://www.google.com/intl/en/chrome/android/>).
- Opera Mobile (<http://www.opera.com/mobile>).
- Internet Explorer Mobile (<http://www.microsoft.com/windowsphone/en-us/features/default.aspx#internet-explorer>). Před dokončením byla navíc otestována panem inženýrem Havrylukem, vedoucím práce, v následujících prohlížečích:
- Nokia Browser for Symbian (<http://browser.nokia.com/>).
- Prohlížeč platformy BlackBerry (<http://us.blackberry.com/smartphones/features/internet.jsp>).
- Safari (<http://www.apple.com/safari/>).

2.2.11.5 Git

Práce byla průběžně verzována systémem pro správu verzí Git (<http://git\discretionary{-}{-}{scm.com/>), čímž bylo zajištěné pohodlné zálohování jak ve smyslu možnosti návratu k některé z minulých verzí, tak i ve smyslu vzdáleného duplikování na serveru GitHub (<http://github.com/>). Pozitivním vedlejším efektem tohoto přístupu je možnost sledování vývoje prostřednictvím veřejného repozitáře.

Realizace

Během implementace se objevila celá řada zádrhelů, z nichž většina souvisela s doposud ne zcela zdokumentovanými a odlazenými technologiemi, přesto byla velmi přínosná – neustálým hledáním různých řešení a alternativ jsem nutně narazil na mnoho dalších zajímavých postřehů a nápadů. Kapitola se zabývá strukturou aplikací, řešeními problémy, volbou použitých technologií, použitými knihovnami, nádtroji a potenciální aplikací v ideálním světě.

3.1 Serverová aplikace

Aplikace bě

3.1.1 Řešené problémy

TODO Optimalizace, komplikace...

3.1.2 Implementace bezpečnosti

3.2 Mobilní aplikace

Pod pojmem *mobilní aplikace* se skrývá aplikace pevně nezávislá na výše uvedené *serverové*. Jejím hlavním cílem je sloužit studentovi FIT ČVUT jako průvodce, zároveň má ale plnit i účel ukázky využití SPARQL endpointu vytvořeného v *serverové* části.

3.2.1 Řešené problémy

3.2.1.1 Změna velikosti displeje

chrome css v px

3.2.1.2 Chyby prohlížečů v implementaci specifikací

Při práci jsem několikrát narazil na situaci, kdy vše v jednom prohlížeči fungovalo, ve druhém nikoliv, a přesto bylo vše implementované správně podle specifikace. V takovém případě pak bývala chyba na straně prohlížeče, který špatně implementoval určitou specifikaci. Situace nastávala převážně u hodně specifických případů. Příkladem může být chybějící podpora pro nastavení transformace SVG elementu JavaScriptem v prohlížeči Google Chrome [7], tento problém ale lze obejít vlastním sestavením hodnoty celého atributu.

3.2.1.3 Rozdílné DOM u HTML a SVG

SVG je, co se týče návrhu DOM, komplikovanější, než HTML – zatímco u HTML stačí pro přístup k hodnotě atributu

3.2.1.4 Zpracování Cross-Origin zdrojů

3.2.2 Implementace bezpečnosti

TODO Proxy. jQuery...

3.2.3 Ideální implementace

Finální implementace se od té ideální, co se týče omezení vzniklých neideálním světem, moc neliší. Vzhledem k cílení aplikace na současná nová zařízení bylo možné realizovat téměř vše, co bylo naplánováno – v uplynulých letech bylo odvedeno mnoho práce na jednotných standardech napříč zařízeními, a co víc, tyto standardy byly implementovány a z velké části se i dodržují. Problémy se proto vyskytly spíše na poli stále ještě v některých ohledech omezených možnostech mobilních zařízení.

3.2.3.1 Malá velikost displeje

Ačkoliv je toto omezení u mobilních zařízení zřejmé, nic to nemění na faktu, že se s ním musí neustále počítat a je třeba vybírat vhodné komponenty a ty vhodně rozvrhovat. Mým cílem proto bylo vizualizovat jen to nejnutnější.

3.2.3.2 Různé typy polohovacích zařízení

Různá zařízení poskytují různé ovládací prvky využívající různé principy, aplikaci je ale nutné přizpůsobit pro všechny najednou. Jiné je ovládání pro myš (touchpad, trackpad), jiné pro klávesy a jiné pro dotyková zařízení. I přes malý displej je třeba u mobilních dotykových zařízení vytvářet ovládací prvky natolik velké, aby bylo možné je používat prsty.

3.2.3.3 Optimalizace rozvržení prvků

Mobilní zařízení nabízejí pro lepší využití malých displejů mnoho různých optimalizací rozvržení zobrazovaných prvků. Zpravidla jsou tyto vlastnosti velmi užitečné – komprimují nevyužitý místo, ve specifických případech, jako je třeba přesné pozicování ukazatele, ale může docházet k obtížně řešitelným situacím. Jediné místo aplikace, které se bez přesného rozmístění neobejde, jsou mapy. Problém jsem se nakonec rozhodl řešit využitím SVG mapových podkladů, ve kterých prohlížeče, zdá se, zatím rozvržení neoptimalizují.

3.2.3.4 Implementace SVG

Implementace SVG je v desktopových prohlížečích na velmi dobré úrovni již nějakou dobu,⁷ v nedávné minulosti se ale podpora značně zlepšila i na poli mobilních zařízení [1].

U mobilních zařízení nepřekvapuje chybějící implementace některých neklíčových možností SVG, jako je třeba průhlednost v případě Opery Mobile. V této oblasti v současné době rozsáhlému nasazení nepřeje SVG nepodporující hojně využívaný nativní webový prohlížeč Androidu verze 2, od verze 3 je podpora zavedena. Problém lze obejít využitím jiného dostupného prohlížeče.

Překvapením byla pomalá, ačkoliv jinak velmi rozsáhlá, implementace SVG v prohlížeči Mozilla Firefox – po některých operacích, v mém případě to byla třeba změna hodnot atributu `viewBox` SVG elementu (posunutí mapy), prohlížeč vyžaduje překreslení celého obrázku, což je velmi zdlouhavé a zabraňuje to plynulému posunu mapy z aktuální na novou pozici, byť by to bylo vhodné pro lepší uživatelské zorientování se. Tento nedostatek je známý již delší dobu [10].

3.2.3.5 Absence popisků

Velmi užitečnou vlastností dostupnou u desktopových aplikací jsou popisky zobrazující se po najetí myši nad podporovaný prvek. Dá se tímto způsobem velmi dobře implementovat kontextová nápověda, která patří mezi nejpodstatnější zdroje pomoci uživateli. U dotykových mobilních zařízení ale bohužel popisky zpravidla zobrazovat nejdou – najetím, tedy dotykem prstu, se rovnou vyvolá akce prvku, aniž by se popisek mohl zobrazit.

3.2.3.6 Dynamické načítání lokálních souborů

Z bezpečnostních důvodů došlo v minulosti v prohlížečích k zamezení načítání JavaScriptových souborů z jiných zdrojů, než ze kterých stránka pochází. To v některých případech⁸ přerostlo v zamezení dynamického načítání jakýchkoliv

⁷Posledním významným desktopovým prohlížečem podporujícím SVG se stal s dlouhým odstupem verze 9.0 Internet Explorer [1], předtím pro něj ale byly dostupné vhodné plugíny.

⁸Příkladem je Google Chrome.

lokálních souborů – jejich zdroj nemá žádnou hodnotu (`null`), takže je nelze načíst. Aby byla zachována možnost offline využití aplikace na větší skupině zařízení, byla aplikace přepsána na jediné možné řešení – explicitní uvedení všech potenciálně vkládaných souborů, což značně omezilo modularitu aplikace.

3.2.3.7 Otevírání lokálních souborů

Omezení zmíněná v předchozím odstavci stupňuje až do extrémů mobilní prohlížeč Chrome Beta, který nepovolí ani přímé otevření lokálního souboru. **TODO** Ověřit, možná nefunguje celé Chrome...

Testování

4.1 Serverová aplikace

TODO Testování sys., integrace netřeba - běží na jednom místě. Komparativní testování nemá vůči čemu probíhat. Použitelnost netřeba řešit.

4.1.1 Testování programátorem

TODO Kód jsem opětovně revidoval a opravoval chyby. Testování vstupů. Review...

4.1.2 Integrační testování

TODO Po zařazení nové funkcionality byly dotčené části programu otestovány.

4.1.3 Zátěžové testování

4.1.4 Akceptační testování

TODO Bylo prováděno vedoucím práce.

4.2 Mobilní aplikace

4.2.1 Testování programátorem

TODO Kód jsem opětovně revidoval a opravoval chyby.

4.2.2 Testování funkcionalit

TODO Splnění požadavků bylo testováno mnou i vedoucím práce.

4.2.3 Integrační testování

TODO Po zařazení nové funkcionality byly dotčené části programu otestovány.

4.2.4 Testování systémové integrace

TODO Kompatibilita.

4.2.5 Zátěžové testování

TODO Má tu smysl? Proxy? SPARQL?

4.2.6 Akceptační testování

TODO Bylo prováděno vedoucím práce.

4.2.7 Testování použitelnosti

4.2.7.1 Cíl testu

4.2.7.2 Charakteristika účastníků

4.2.7.3 Nastavení testu

4.2.7.4 Seznam testovaných úloh

4.2.7.5 Průběh testu

4.2.7.6 Shrnutí testu

4.2.8 Komparativní testování

TODO Porovnání mobilní aplikace s konkurencí.

Závěr

ZÁVĚR

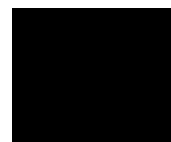
- TODO** Zhodnocení splnění cílů DP.
- TODO** Doporučení pro další pokračování.
- TODO** Shrnutí výsledků a vlastního přínosu.

Literatura

- [1] Deveria, A.: When can I use. . . . 2012,
<http://caniuse.com/>, stav z 1. 5. 2012.
- [2] *Průvodce prváků po ČVUT*. IAESTE ČVUT Praha, 2011.
- [3] Firtman, M.: Mobile HTML5. 2012,
<http://mobilehtml5.org/>, stav z 1. 5. 2012.
- [4] Gayo, J. E. L.: University Ontology. 2011,
<http://purl.org/weso/uni>, stav z 10. 3. 2012.
- [5] Guo, Y.: Lehigh University Benchmark.
<http://swat.cse.lehigh.edu/projects/lubm/>, stav z 1. 5. 2012.
- [6] Hendler, J.; Heflin, J.; Luke, S.: SHOE. 2001,
<http://www.cs.umd.edu/projects/plus/SHOE/>, stav z 1. 5. 2012.
- [7] Laag, L.: SVG transform attribute does not work when manipulated from JavaScript. Září 2010,
<http://code.google.com/p/chromium/issues/detail?id=55010>,
stav z 1. 5. 2012.
- [8] Molnár, J.: *Mobilní navigační systém pro FEL ČVUT*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Květen 2010.
- [9] Styles, R.; Shabir, N.: Academic Institution Internal Structure Ontology (AIISO). 2008,
<http://purl.org/vocab/aiiso/schema>, stav z 1. 5. 2012.
- [10] Bugzilla. 2012,
<https://bugzilla.mozilla.org/>.
- [11] MongoDB. 2012,
<http://www.mongodb.org/>.
- [12] Node.js. 2012,
<http://nodejs.org/>.

LITERATURA

- [13] Zemmouchi-Ghomari, L.: Higher Education Reference Ontology (HERO). 2012,
<http://sourceforge.net/projects/heronto/>, stav z 1. 5. 2012.



Seznam použitých zkratek

AIISO Academic Institution Internal Structure Ontology.

CD Compact Disc.

CSS Cascading Style Sheets.

ČVUT České vysoké učení technické v Praze.

DOM Document Object Model.

DP diplomová práce.

FIT Fakulta informačních technologií.

FOAF Friend of a friend.

GNU GNU's Not Unix!.

GPL General Public License.

HERO Higher Education Reference Ontology.

HTML HyperText Markup Language.

J2ME Java 2 Platform, Micro Edition.

KDE K Desktop Environment.

KIO KDE Input/Output.

Kate KDE Advanced Text Editor.

LUBM The Lehigh University Benchmark.

A. SEZNAM POUŽITÝCH ZKRATEK

MAR měření a regulace.

NTK Národní technická knihovna.

OS operační systém.

Org An organization ontology.

PHP Hypertext Preprocessor.

RDF Resource Description Framework.

SHOE Simple HTML Ontology Extensions.

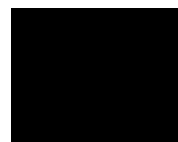
SPARQL SPARQL Protocol and RDF Query Language.

SVG Scalable Vector Graphics.

XML Extensible Markup Language.

PŘÍLOHA

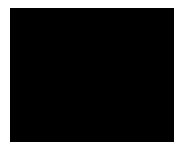
B



Instalační a uživatelská příručka

PŘÍLOHA

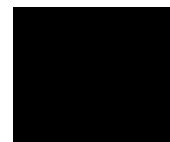
C



Obsah přiloženého CD

PŘÍLOHA

D



TODO

23 Bylo prováděno vedoucím práce.

6 Deployment diagram.

12 Diskuse alternativ.

3 Diskuse alternativ. Struktura.

32 Doporučení pro další pokračování.

15 Injekce JS, ošetření vstupů, white-listing URL, elementů, atributů, zakázání cizích zdrojů...

5 Injekce kódu. Pouze povolené URL. Ošetření vstupů.

24 Kód jsem opětovně revidoval a opravoval chyby.

21 Kód jsem opětovně revidoval a opravoval chyby. Testování vstupů. Review...

27 Kompatibilita.

28 Má tu smysl? Proxy? SPARQL?

17 Optimalizace, komplikace...

19 Ověřit, možná nefunguje celé Chrome...

22 Po zařazení nové funkcionality byly dotčené části programu otestovány.

D. TODO

- 9** Pojednání o vytvořených prototypch, možná nakonec pod něco zařadím, nebude mít velký rozsah.
 - 30** Porovnání mobilní aplikace s konkurencí.
 - 7** Pouze pokud se objeví podpora poskytnutého WSDL.
 - 18** Proxy. jQuery...
 - 10** Různé případy užití v textové formě.
 - 8** Seznam a popis jednotlivých potřeb uživatelů.
 - 33** Shrnutí výsledků a vlastního přínosu.
 - 25** Splnění požadavků bylo testováno mnou i vedoucím práce.
 - 14** Struktura.
 - 20** Testování sys., integrace netřeba - běží na jednom místě. Komparativní testování nemá vůči čemu probíhat. Použitelnost netřeba řešit.
 - 4** Volba implementačního prostředí.
 - 0** Vynechám – chybí API.
 - 11** Vynechám – muselo by se složitě indexovat.
 - 31** Zhodnocení splnění cílů DP.
- (Shodné položky jsou vynechány.)