VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

VÝVOJ MULTIPLATFORMNÍ GEOLOKAČNÍ APLIKACE

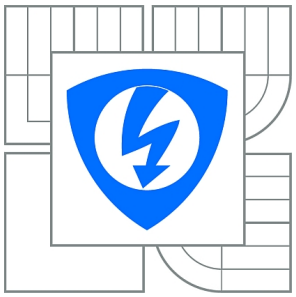
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE MAREK PINKAVA
AUTHOR

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

VÝVOJ MULTIPLATFORMNÍ GEOLOKAČNÍ APLIKACE

MULTI-PLATFORM DEVELOPMENT OF GEOLOCATION APPLICATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

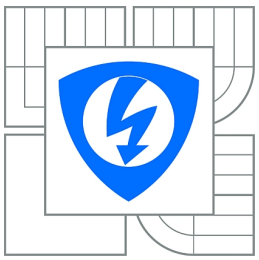
AUTOR PRÁCE — MAREK PINKAVA

AUTHOR

VEDOUCÍ PRÁCE — Ing. ALEŠ POSPÍŠIL

SUPERVISOR

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Marek Pinkava	***ID:***	125590
Ročník:	3	***Akademický rok:***	2011/2012

NÁZEV TÉMATU:

Vývoj multiplatformní geolokační aplikace

POKYNY PRO VYPRACOVÁNÍ:

Podstatou práce bude vytvořit geolokační aplikaci s využitím moderních technologií pro tvorbu webových aplikací (HTML5, CSS3, Javascript) a volně dostupných mapových API (např. Google Maps). Vytvořená aplikace bude vhodná pro využití na různých webových a mobilních platformách. Součástí práce bude hodnocení použitých technologií a aplikace bude testována v reálném provozu.

DOPORUČENÁ LITERATURA:

[1] Flanagan D., JavaScript: The Definitive Guide, O'Reilly, 2011.
[2] Svannerberg G., Beggining Google Maps API 3, Apress, 2010.
[3] Lawson B., Sharp R., Introducing HTML5, New Riders Press, 2010.

Termín zadání:	6.2.2012	***Termín odevzdání:***	31.5.2012

Vedoucí práce: Ing. Aleš Pospíšil
Konzultanti bakalářské práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Bakalářská práce se zabývá možnostmi vývoje multiplatformních geolokačních aplikací, pomocí moderních webových technologií. V úvodu práce je nabídnut souhrnný popis dostupných mobilních platforem a možnosti vývoje pro ně. Následně jsou popsány webové technologie vhodné k vývoji aplikací. Další kapitola se věnuje geolokačním metodám, jejich možnostem a omezením. Poté je popsána implementace konkrétní aplikace pomocí vybraných technologií. Výsledkem práce je vytvořená multiplatformní aplikace xStudent sloužící studentům FEKT VUT v Brně, která jim pomůže k snadné orientaci mezi důležitými místy fakulty.

KLÍČOVÁ SLOVA

Chytrý telefon, web, multiplatformní vývoj, nativní vývoj, geolokace, Google Maps API, HTML5, CSS3, Sencha Touch

ABSTRACT

This thesis deals with development possibilities of multi-platform geolocation applications using modern web technologies. The introduction is offered a summary of available mobile platforms development opportunities for them. Follows the description of web technologies suitable for application development. Another chapter is devoted with methods of geolocation and their possibilities and limitations. After is described the implementation a specific application of selected technologies. The outcome of this thesis is fully working multiplatform application developed in order to help students to easy navigation between important places of faculty.

KEYWORDS

Smartphone, web, multiplatform development, native development, geolocation, Google Maps API, HTML5, CSS3, Sencha Touch

PINKAVA, Marek *Vývoj multiplatformní geolokační aplikace*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2012. 53 s. Vedoucí práce byl Ing. Aleš Pospíšil,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Vývoj multiplatformní geolokační aplikace“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Aleši Pospíšilovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

V současné době mobilní telefony neslouží pouze k volání nebo psaní SMS (Short Message Service), jako tomu bylo dříve. S rozvojem moderních technologií se na trhu objevují telefony s dotykovými displeji, dokonalejšími fotoaparáty a výkonnějšími procesory, které pro uživatele představují malé kapesní počítače. Lidé se mohou bezdrátově připojit na internet, pomocí systému GPS používat navigaci, jednoduše fotit, natáčet a sledovat videa nebo provozovat videohovory. Nedávné uvedení nových operačních systémů jako je iOS, Android nebo Windows Phone na trh mobilních telefonů, umožnilo velké skupině vývojářů vytvářet a prodávat své aplikace. Velké množství zajímavých aplikací podpořilo velmi rychlé rozšíření těchto tzv. chytrých telefonů (angl. smartphones) mezi lidmi.

Aplikace pro mobilní operační systémy lze vyvíjet buď nativně nebo multiplatformně. Vývoj nativních aplikaci znamená pro vývojáře složité adaptování na každou jednotlivou mobilní platformu. Proto je výhodné určité druhy aplikací vyvíjet multiplatformně. Poté lze jednou napsanou aplikaci nabídnout uživatelům různých platforem. Vývoj aplikací tímto způsobem je levnější, šetří čas a takto vytvořené aplikace lze snadněji udržovat aktuální.

Uživatelé nosí mobilní telefon téměř pořád u sebe, a tak jsou pro ně zajímavé geolokační aplikace, které pracují s jejich aktuální geografickou polohou. Aplikace jim tak mohou nabízet užitečné informace či služby přímo z jejich okolí. Mohou jim nabídnout výchozí stanici při hledání dopravního spoje, najít nejbližší restauraci či obchod.

Bakalářská práce si klade za cíl vytvořit multiplatformní geolokační aplikaci s využitím moderních webových technologií. V úvodu práce jsou popsány dostupné mobilní platformy, jejich vlastnosti a možnosti vývoje pro ně. Další část práce se věnuje popisu webových technologií vhodných k multiplatformnímu vývoji aplikací. Následující kapitola se zabývá metodami geolokace uživatele. Jednotlivé metody jsou srovnány z hlediska přesnosti a dostupnosti v zařízeních. Předposlední kapitola je věnována výběru konkrétních technologií pro tvorbu multiplatformní aplikace a popisu struktury vytvořené aplikace. Poslední část práce se zabývá testováním a vlastnostmi vytvořené aplikace.

1 MOBILNÍ PLATFORMY A VÝVOJ APLIKACÍ

Každý chytrý telefon má operační systém, který je podobný tomu, jaký je u stolních počítačů. Na trhu je hned několik operačních systémů pro smatrphony. Mezi hlavní se řadí Android od firmy Google, iOS od firmy Apple, firma Microsoft nabízí systém Windows Phone, firma Nokia systém Symbian a firma RIM (Research In Motion) systém Blackberry OS. Existuje ještě několik operačních systémů, které ale nemají takové zastoupení a tvoří jen malou část trhu. Jsou to např. Bada OS , WebOS nebo systém MeeGo.

1.1 Mobilní platformy

1.1.1 Android

Systém Android začala vytvářet v roce 2003 společnost Android. Inc., kterou v roce 2005 koupila společnost Google [1], a tím získala základ pro platformu Android. Na konci roku 2007 byla založena aliance OHA (Open Handset Alliance), jenž slučuje Google a další společnosti, které poskytují mobilní služby, vyvíjí mobilní telefony a hardware. Jejím cílem bylo vytvořit otevřenou platformu pro mobilní zařízení určené pro celý segment trhu, na rozdíl od společnosti Apple, která se primárně soustředí na high-end zařízení. Tím byla představena platfrorma Android. Ihned poté byl vydán první SDK (Software Development Kit – softwarový vývojářský balík). V polovině roku 2008 se objevil první telefon s Androidem s názvem T-mobile G1. Od roku 2009 se telefony s Androidem začali objevovat ve velkém počtu. Dalším důležitým krokem po tuto platformu bylo představení Androidu 3.0 na začátku roku 2011, který je určen pro tablety.

Operační systém Android vychází z Linuxového jádra, jedná se o open source software, výrobci mobílních přístrojů tak mají možnost systém lépe optimalizovat pro konkrétní hardware a integrovat své aplikace. Pro distribuci aplikací slouží Google Play, kde mohou uživatelé najít jak placené aplikace, tak i aplikace, které jsou zdarma. Google nijak neomezuje uživatele, a aplikace tak mohou být instalovány i z jiných zdrojů.

1.1.2 iOS

Je operační systém firmy Apple, vychází z operačního systému určeného pro počítače Mac a je přizpůsoben pro mobilní zařízení. Původně byl určen pro smartphony iPhone, poté se objevil v tabletech iPad, zařízeních iPod Touch a Apple TV.

Tento systém je velmi uzavřený, neumožňuje vývojářům přístup do systému [2], aplikace je možno instalovat pouze přes oficiální AppStore, kde jsou všechny aplikace přísně kontrolovány, aby neobsahovaly bezpečnostní chyby, jejich kód nebyl škodlivý a neměli nevhodný obsah. Tyto striktní požadavky jsou často kritizovány, na druhou stranu při instalaci z AppStoru je téměř jistotou, že aplikace bude bezpečná a kvalitní.

1.1.3 Windows Phone

Nový systém firmy Microsoft je nástupcem sytému Windows Mobile [3]. Byl vydán na konci roku 2010, jedná se tedy o nejmladší systém. Je cílen na spotřebitelský trh, jeho předchůdce byl směřovám primárně na firemní segment trhu.

Obsahuje zcela přepracované rozhraní Metro, které je přizpůsobeno pro ovládání dotykem. Jelikož se jednalo o první verzi tohoto systému, neobsahoval některé důležité funkce jako například multitasking nebo neměl funkci `Copy/Paste`. Toto se již změnilo vydáním nové verze Windows Phone 7.5 s označením Mango ve třetím kvartálu roku 2011, která přináší opravy a nové funkce. Pro distribuci aplikací nabízí Microsoft Marketplace.

1.1.4 Blackberry OS

Americká firma RIM dává do telefonů značky Blackberry vlastní operační systém Blackberry OS [4], který je založený na platformě Java. Jeho hlavním cílem je firemní segment trhu. Tento systém je nejvíce rozšířený v Americe.

V roce 2011 představilo RIM nový systém se jménem Playbook OS, který je založený na platformě QNX.

1.1.5 Symbian

Historie Symbianu se začala psát již v 80. letech minulého století, kdy vznikla firma PSION, vyrábějící kapesní organizéry. V roce 1998 vznikl projekt Symbian Ltd., který sdružoval firmy Nokia, Ericsson a softwarovou divizi firmy PSION [5], která jako základ pro Symbian poskytla systém EPOC. Na konci roku 2008 koupila společnost Nokia většinový podíl v Symbianu. Tento systém byl donedávna nejvíce rozšířeným systémem v mobilních telefonech, ovšem v poslední době jeho oblíbenost velmi klesla. Nyní jej používá téměř výhradně pouze samotná firma Nokia ve svých telefonech. Na začátku roku 2011 Nokia oznámila partnerství s Microsoftem a bude se podílet na vývoji Windows Phone, který chce nasazovat do svých telefonů. Pro distribuci aplikací představila Nokia v roce 2009 Ovi store, což je on-line obchod zaměřený na prodej softwaru pro zařízení Nokia.

1.1.6 Ostatní platformy

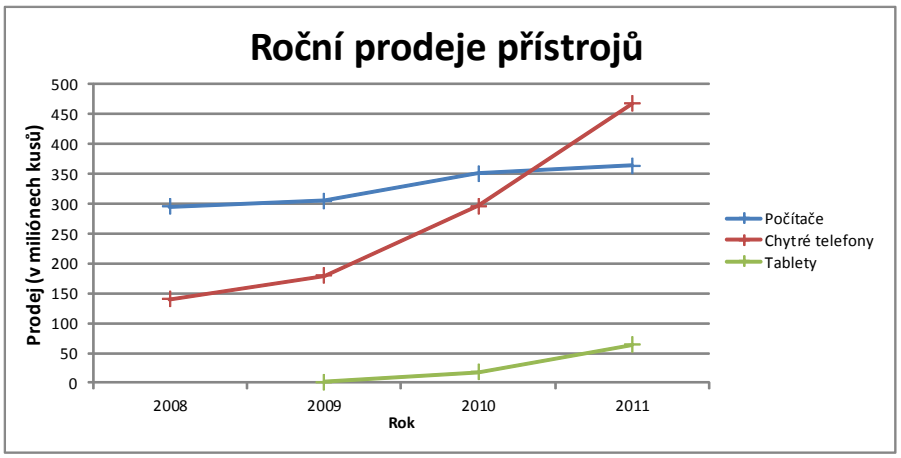
Z dalších platforem, které již ale nemají takové zastoupení na trhu, je možno jmenovat systém Bada společnosti Samsung [6], který je postaven na jádru Linuxu, zatím se objevil jen v několika přístrojích samotné firmy Samsung. Systém firmy Hewlett-Packard WebOS je taktéž založený na jádře Linuxu, firma jej získala akvizicí společnosti Palm [7]. Posledním ze známých systémů je MeeGo [8], na kterém se podílely firmy Intel a Nokia. MeeGo je open source systém založený na Linuxu. Tento systém se zatím objevil pouze v jednom zařízení. Po ohlášení toho, že Nokia bude využívat pro své telefony systém Windows Phone, ukončila spolupráci na tomto systému. V září roku 2011 oznámil Intel spojení systému MeeGo se systémem Limo, ze kterých má vzniknout systém Tizen.

1.2 Vývoj mobilních aplikací

Zpravidla jsou dvě možnosti jak vyvíjet mobilní aplikace. Tou první je vyvíjet aplikace pro každou platformu zvlášť, použít k tomu nativní jazyk, který je pro danou platformu určen a nástroje, které nabízí výrobci konkrétních operačních systémů. Tou druhou je použít univerzální jazyk, který je schopný běžet na všech platformách.

1.2.1 Motivace pro vývoj mobilních aplikací

Na světě je přes 5 miliard všech mobilních telefonů. Již na konci roku 2010 dokázal prodej smartphonů předstihnout prodej klasických počítačů, což je vidět na obr. 1.1 a podle agentury Gartner se za třetí kvartál roku 2011 prodalo 115,1 milionů smartphonů [10], jedná se o nejrychleji rostoucí segment trhu s elektronikou, jehož význam se má v dalších letech ještě zvyšovat.



Obr. 1.1: Roční prodeje počítačů, smartphonů a tabletů (převzato z [9]).

1.2.2 Nativní vývoj

Android

Pro platformu Android se programuje pomocí SDK [1], které obsahuje knihovny, emulátor a debugger. Dále je možné využít ADT (Android Development Tools – vývojářské nástroje pro Android) plugin pro vývojové prostředí Eclipse, který umožní vývojářům ovládat SDK, ladit aplikace, a to buď v emulovaném telefonu nebo přímo v zařízení. Emulovaných telefonů si vývojář může zvolit více, a odladit tak aplikaci pro různé rozlišení displejů nebo pro různý hardware.

Primárně se pro Android vyvíjí v Javě, v některých případech, kdy se vyžaduje vysoký výkon je možno psát aplikaci v C/C++ a poté pomocí NDK (Native Development Kit – nativní vývojářský balík) ji přeložit přímo do nativního kódu. Po napsání aplikace v jazyce Java je kód přeložen do bytekódu pro běh na Dalvik VM (Virtual Machine – virtuální stroj). Protože je na mobilních zařízeních důležitá výdrž baterie jsou instrukce Dalvik VM oproti Javě a jejího bytekódu optimalizovány přímo pro běh na mobilních zařízeních.

Systém Android je velmi otevřený a všechny aplikace od jádra až po aplikace třetích stran mají stejné možnosti v používání systémových knihoven a frameworků. Základní části, ze kterých se skládají aplikace pro Android:

- **Activity** – část odpovídající jedné obrazovce, stará se o interakci mezi aplikací a uživatelem;
- **Service** – služba, která slouží k přístupu ke vzdáleným zdrojům;
- **Content Provider** – část sloužící ke sdílení dat mezi aplikacemi;
- **Broadcast receiver** – komponenta sloužící k naslouchání oznámení z aplikace i ze systému.

iOS

Jazyk pro platformu iOS se jmenuje Objective-C [2]. Vznikl v 80. letech minulého století jako nástavba nad jazykem C. Jedná se o objektově orientovaný jazyk, jehož objektovou syntaxi převzali vývojáři z jazyka Smalltalk. Protože je kompilátor Objective-C zpětně kompatibilní lze pro iOS vyvíjet také v jazyce C. Pro vývoj aplikací je určené vývojové prostředí Xcode, které lze spustit pouze na počítačích se systémem MacOS. Vrstvy, ze kterých se skládá systém:

- **Cocoa Touch**,
- **Media**,
- **Core Services**,
- **Core OS**.

Tato platforma je velmi uzavřená, a tak vývojáři mohou používat pouze API (Application Programing Interface – aplikační programové rozhraní), které poskytují jednotlivé vrstvy.

Windows Phone

Pro vývoj Windows Phone aplikací je určena technologie Silverlight [3], vhodná pro vytváření obdobných aplikací jako pomocí technologie Flash firmy Adobe. Druhá možnost jak tvořit aplikace, je pomocí frameworku XNA, který je určen pro tvorbu 2D a 3D her, využívající DirectX. Obě technologie využívají Framweork .NET a je možné programovat v jazycích C#, F#, VisualBasic .NET. Jako vývojové nástroje Microsoft nabízí zdarma Visual Studio s emulátorem telefonů a Microsoft Expression Blend pro tvorbu grafiky.

Blackberry OS

Blackberry OS umožňuje vývoj aplikace hned několika způsoby [4], programátoři pro svou aplikaci mohou využít Java BlackBerry SDK a plugin pro vývojové prostředí Eclipse. Dále je možno využít technologie BlackBerry Widgets, což hybridní technologie kombinující prvky HTML (Hypertext Markup Language) pro uživatelské rozhraní a kód v jazyce Java pro aplikační logiku.

Symbian

O uživatelské rozhraní se stará Qt framework, vývoj aplikací probíhá nejčastěji v jazyce C++ [5], dále je možné použít Python, Adobe Flash nebo Java ME. Doporučovaným vývojovým prostředím je Qt creator.

Ostatní platformy

Samsung nabízí pro vývoj Bada OS aplikací SDK a plugin pro vývojové prostředí Eclipse, jako programovací jazyk se používá C++ [6]. Programy pro systém systém MeeGo se píší v C++ [8] a pro WebOS se aplikace vyvíjí pomocí JavaScriptu a HTML nebo pomocí C++ [7].

1.2.3 Multiplatformní vývoj

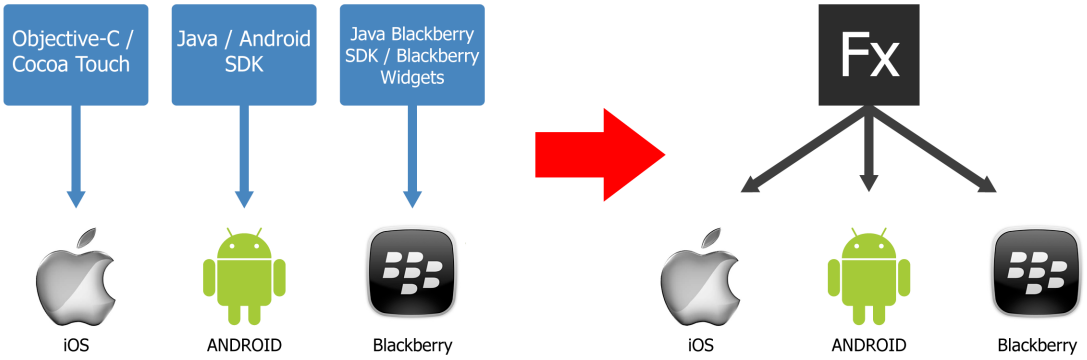
Pod pojmem multiplatformní vývoj se často myslí vyvíjení pomocí webových technologií, dále je možné vyvíjet multiplatformně např. pomocí technologie Adobe Flex, která je určena pro tvorbu RIA (Rich Internet Application) aplikací.

Webové technologie

Mobilní platformy jsou různé, mají rozdílnou strukturu, podporují různé jazyky a nástroje, jedna věc je ale společná pro téměř všechny moderní mobliní platformy a tou je webový prohlížeč podporující HTML5. Proto lze pomocí této technologie psát aplikace, které budou fungovat na různých platformách. Tato technologie je popsána v kapitole 2.

Adobe Flex

Flex je techologie určená k vyvoji aplikací pro různé platformy [11], je to technologie obdobná HTML a JavaScriptu, kdy se značkovací jazyk používá pro uživatelské rozhraní a skript pro funkčnost aplikace. Zatím podporuje platformy Android, iOS a Blackberry.



Obr. 1.2: Princip vývoje aplikace v Adobe Flex.

Značkovací jazyk Flexu se jmenuje MXML, má vyšší sémantiku pro aplikační vývoj oproti HTML a každá značka představuje komponentu, která má své vlastnosti, události a styly. K vytváření designu uživatelského rozhraní lze použít CSS (Cascading Style Sheets) jako v HTML. Zápis je striktně XML (eXtensible Markup Language) a pro odlišení různých sekcí a komponent se používají jmenné prostory. Výkonný kód, se zapisuje v ActionScriptu, což je jazyk podobný JavaScriptu obohacený o datové typy a několik dalších vlastností.

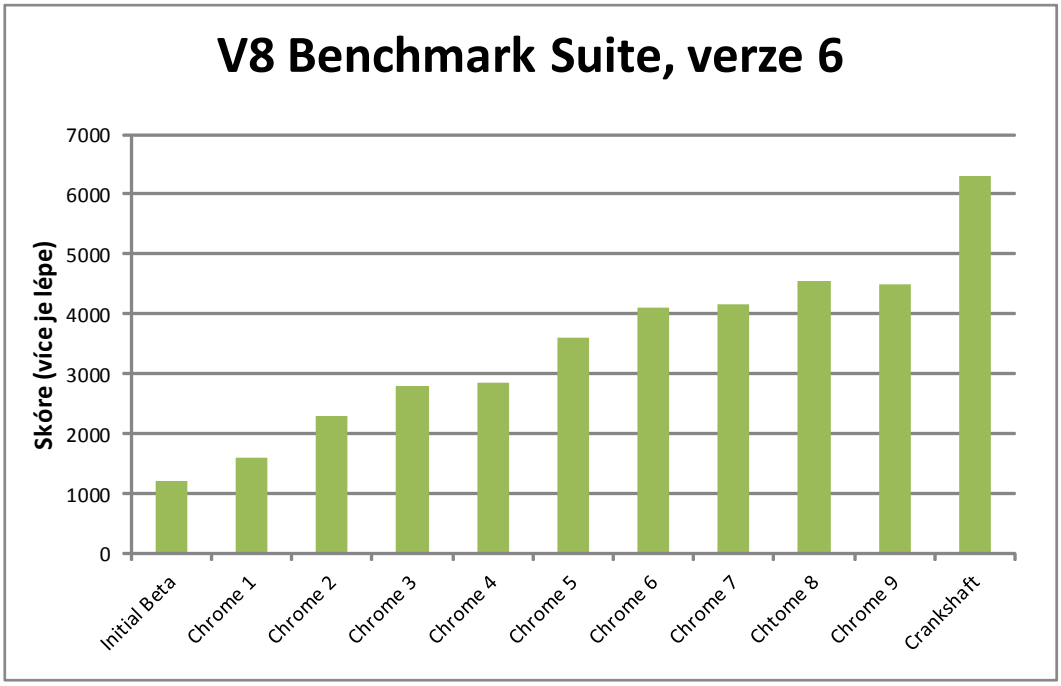
Adobe nabízí pro vývoj aplikací prostředí Flash Builder, které umí napovídat názvy funkcí, parametrů, debugovat běžící kód a navrhovat uživatelské rozhraní ve WYSIWYG (What You See Is What You Get) editoru. Flex lze spouštět v prohlížeči pomocí Flash Playeru, na mobilních platformách běží Flex v prostředí Adobe Air, které je při kompilaci přibaleno k aplikaci.

1.2.4 Vlastnosti multiplatformního webového vývoje

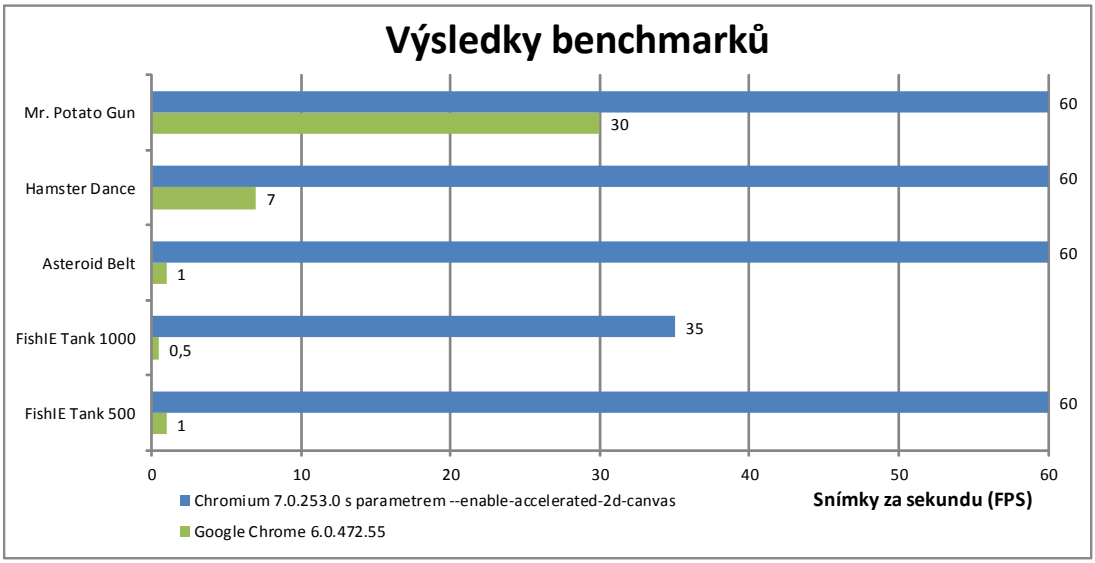
Z předchozích částí vidíme, že existuje mnoho druhů operačních systémů a pokud chceme poskytnout aplikace pro co největší okruh uživatelů, museli bychom aplikaci napsat pomocí mnoha různých jazyků. Tento způsob realizace je hodně náročný a to jak časově, tak i finančně. Proto je pro určitý druh aplikací, např. ty, které jsou informačně založené, vhodné psát pomocí multiplatformní technologie.

Pokud je aplikace napsaná pouze jednou pro všechny platformy je snadnější ji udržovat stále aktuální. Zastánci nativního vývoje argumentují tím, že pomocí webových technologií není možné přistupovat k funkcím systému a hardwaru jako jsou notifikace, kamera, GPS apod. S postupem času se situace mění, a již nyní W3C (World Wide Web Consortium) pracuje na specifikaci Device API, díky kterému lze přistupovat k hardwaru pomocí webových technologií. Pokud vytvoříme hybridní (nativně-webovou) aplikaci pomocí frameworků jako jsou např. Phonegap nebo Titanium, lze funkce a vlastnosti systému využívat téměř jako v nativní aplikaci [12].

V poslední době velmi zrychlily grafické renderovací enginy, což je vidět na obrázku 1.3, kde je znázorněn výkon JavaScriptového enginu V8 v prohlížeči Chrome. Nastupuje hardwarová akcelerace, která zrychlí vykreslování aplikace, výsledky testů na obrázku 1.4 ukazují značný nárůst výkonu, a tak nelze říct, že webové aplikace oproti nativním budou pomalé. HTML5 navíc nabízí Web Workers API, které umožňuje využívat multithreading. Webovou aplikaci lze snadno stylovat, takže je možné vytvořit aplikaci se stejným vzhledem jako mají nativní aplikace. Webová aplikace je snadno objevitelná, uživatelé je mohou šířit pouhým zveřejněním odkazu na sociální síti, posláním odkazu SMSkou nebo emailem. Nevýhoda je, že takovou aplikaci nelze umístit do distribučních systémů jednotlivých platforem, jako je AppStore, AndroidMarket apod. Toto omezení neplatí pokud vytvoříme jednoduchou nativní „wrapper aplikaci“, která obsahuje pouze komponentu webového prohlížeče a v ní zobrazujeme naši webovou aplikaci. Poté je již možné umístit aplikaci do distribučních systémů.



Obr. 1.3: Výkon V8 JavaScriptového enginu vyvinutého pro Chrome (převzato z [12]).



Obr. 1.4: Výsledky hardwarově akcelerovaného canvasu (převzato z [12]).

2 WEBOVÉ TECHNOLOGIE

Systém WWW (World Wide Web) vznikl na konci 80. let 20. století. První verzi HTML vytvořil v roce 1991 Tim Berners-Lee jako součást projektu WWW [13], který umožňoval vědcům komunikaci a sdílení výsledků. Pomocí této verze se dali tvořit pouze jednoduché dokumenty. Postupem času bylo potřeba více funkcí, a tak vznikaly nové verze HTML.

Po vydání specifikace HTML 4.0 zveřejnilo W3C jazyk XML, který byl přijat jako formát pro výměnu a ukládání dat. V roce 2000 byla vydána specifikace jazyka XHTML 1.0 (Extensible Hypertext Markup Language), která měla svoji syntaxi odvozenou od XML. W3C začalo poté vyvíjet další verzi XHTML, ta ale neměla být zpětně kompatibilní a pro výrobce prohlížečů to byl příliš revoluční krok. Byli velmi nespokojení, a proto někteří z nich v roce 2004 založili společnou pracovní skupinu WHATWG (The Web Hypertext Application Technology Working Group) a začali připravovat specifikaci platformy pro webové aplikace běžící v prohlížeči. Tato specifikace obsahuje rozšíření jazyka HTML a také důležité rozhraní pro využití pomocí JavaScriptu. V roce 2007 došlo ke spojení W3C a WHATWG, pracovní skupina HTML vzala jako základ specifikace vytvořené ve WHATWG a na něm začalo vznikat HTML5.

2.1 HTML5

HTML5 navazuje na HTML 4.01, ale nezahrnuje pouze sémantiku značkovacího jazyka [14], nýbrž se zabývá i JavaScriptovými API, offline fungováním aplikací, kreslením v prohlížeči a dalšími věcmi, které jsou potřeba při tvoření webových aplikací. Specifikace HTML5 však ještě není hotová. Některé části tak nemusí fungovat ve všech prohlížečích anebo nemusí být vůbec ve finálové verzi zahrnuty, jiné části jsou již poměrně stabilní a podporuje je většina moderních prohlížečů.

Sémantika

HTML5 upravuje sémantický význam některých starších HTML tagů a zavádí také několik nových značek. V předchozích verzích HTML se logické celky vkládali do elementu `div`, ten nemá žádný sémantický význam, a tak se s výsledným dokumentem nedalo jednoduše strojově pracovat. HTML5 definuje elementy, které mají sémanticky popsat různé časti dokumentu, jsou to např `<article>`, `<header>`, `<footer>` a další.

U elementu `<script>` se již nemusí uvádět `type="text/javascript"` a do tohoto elementu přibyl atribut `async`, který má zajišťovat asynchronní zpracování

skriptu a atribut `defer`, který naopak čeká až je dokument zparsován. Element `<link rel="stylesheet">` má nyní defaultní hodnotu `type="text/css"`, takže ji není třeba zadávat. HTML5 ruší používání Doctype specifikace na začátku dokumentu a zavádí se pouze `<!DOCTYPE html>`. Dále zavadí nový tag `<meta charset>`, kterým se určuje kódování dokumentu.

Microdata

Sémantika nových HTML5 prvků nestačí na popsání významu všech webů a aplikací, HTML5 přichází s technologií microdat, která má umožnit tvůrcům webových stránek označovat obsah tak, aby byl strojově zpracovatelný, informace byly dostupné prohlížečům či vyhledávačům a také, aby byly stránky více přístupné např. slepcům.

Microdata používají slovníky, které definují jednotlivá jména vlastností. V tvůrce webu pak v HTML definuje slovník, který používá, a poté už jen popisuje jednotlivé prvky webu pomocí vlastností ve slovníku.

Audio a video

Pro vložení videa a audia na web, nenabízelo HTML žádný standardizovaný způsob. Tento obsah se zobrazoval pomocí přehrávačů, které se do prohlížeče instalovaly jako pluginy. HTML5 nabízí element `<video>` a `<audio>`, které umožňují jednoduše vložit video nebo zvuk do stránky, poskytuje také API k vytvoření vlastních ovládacích prvků, které je možné pomocí CSS stylovat. Elementy jsou párové a uvnitř lze nastavil fallback, tedy chování pro případ, že prohlížeč tyto tagy nepodporuje. Umožňuje také nalinkovat video a audio ve více formátech a pokrýt situace, kdy prohlížeč neumí přehrát některý formát videa. V současné době jsou používány následující kodeky [15]:

Video kodeky

- **kodek H.264** byl vytvořen a standardizován v roce 2003 skupinou MPEG. Mnoho zařízení obsahuje hardware, který kódování a dekódování H.264 urychluje. H.264 je patentově chráněný;
- **kodek Theora** byl vyvinut v roce 2004 z kodeku VP3. Je to otevřený kodek pro kódování videa, vytvořila jej společnost Xiph.org Foundation;
- **kodek VP8** je vyvinutý firmou On2. V roce 2010 On2 koupil Google a kodek otevřel jako open-source a patenty licencoval jako royalty-free;

Audio kodeky

- **MP3** je dneska již průmyslový standard. Umí maximálně 2 kanály a zvuk lze kódovat v různém bitrate od 32k̇bps do 320k̇bps;
- **AAC** umí až 48 kanálů s horní hranicí bitrate 320k̇bps. AAC bylo standardizováno v roce 1997 a je patentově chráněno;
- **Vorbis** podporuje neomezené množství kanálů a není patentově chráněn.

Problém s používáním videa a audia je ten, že není standardizován jeden základní kodek, který by podporovaly všechny prohlížeče. Proto musí vývojáři použít video či audio ve více formátech, aby jej bylo možno přehrát ve všech moderních prohlížečích.

Formuláře

Pokud v předchozí verzi HTML chtěli vývojáři použít ve formulářích specifické prvky jako např. slider nebo vstupní pole pro datum, museli použít JavaScript, který klasické textové pole přeformátoval na požadovaný typ. HTML5 přidává několik užitečných typů formulářových prvků a několik atributů, a tak se zde již JavaScript nemusí používat. Pokud starší prohlížeče neznají tyto prvky, automaticky zobrazí klasický prvek `<input type="text">`. Nové prvky jsou velmi užitečné na dotykových zařízeních, jenž mají virtuální klávesnici, která tak může být upravena speciálně pro konkrétní vstupní pole. Formuláře mají u některých typů prvků zabudovanou automatickou validaci a pokud validace selže, prohlížeč zobrazí chybovou hlášku.

Offline aplikace

Ještě nedávno neexistoval snadný způsob, jak dosáhnout toho, aby aplikace fungovaly i bez připojení k internetu. HTML5 nyní nabízí Application Cache, a tak lze jednoduše psát aplikace, které půjdou spouštět offline stejně jako online. Situace s nedostupným připojením se řeší určením způsobu, jak v takovém případě uložit potřebná data do lokální cache tak, aby je měl prohlížeč stále k dispozici. Soubory, které se mají uložit, se určí pomocí souboru `manifest`.

AppCache se od normální cache prohlížeče liší především tím, že je určena speciálně pro webové aplikace (ukládá obsah, který je určený v manifestu), cache prohlížeče naopak ukládá obecně webové stránky a jejich obsah. Programátor má možnost ovlivnit chování AppCache. Má přístup k událostem, které informují o stavu AppCache, a k funkcím pro asynchronní aktualizaci.

Data Storage

Většina aplikací potřebuje nejen ukládat stránky, skripty a další soubory, ale i nějakým způsobem ukládat vytvořená data. HTML5 nabízí hned několik řešení.

Web Storage definuje dvě lokální úložiště, které pracují jako jednoduchá key-value databáze a jejich fungování lze připodobnit k „asociativním polím“ – tedy polím dat, které jsou indexovány nikoli celočíselnou hodnotou, ale obecným klíčem, nejčastěji řetězcem. Obě úložiště jsou identické až na jednu věc, a tou je jejich perzistence:

- **LocalStorage** ukládá data v prohlížeči, dokud nejsou smazány skriptem;
- **SessionStorage** ukládá data jen po dobu trvání sezení (session).

Ukládání v prohlížeči podle klíčů je podobné cookies, ty jsou ovšem primárně určeny pro data, která si chce server uložit do klientského prohlížeče pro své účely (i když jsou přístupné ze skriptu běžícího v prohlížeči). Posílají se v hlavičkách HTTP dotazů, počet odpovědí a jejich velikost je omezená. Data uložená ve WebStorage naproti tomu zůstávají v prohlížeči, nikam se neposílají a o případné uložení na server se musí programátor postarat sám.

Databáze

- **Web SQL Database** nabízí API rozhraní k databázi SQLite. Tvorba specifikace byla zastavena, protože specifikace nedefinovala vlastní SQL, pouze se odkazovala na SQLite, tudíž nebyla vhodná pro webový standard. Tvůrci prohlížečů Mozilla Firefox a Internet Explorer ani neimplementovali toto rozhraní do svých prohlížečů.
- **IndexedDB** je úložiště pro strukturovaná data s možností indexace. Místo jazyka pro vyhledávání dat v indexovaném úložišti nabízí přímý přístup k tomuto úložišti.

Canvas a SVG

Jsou technologie, které umožňují vývojářům vkládat pokročilou grafiku do webů a webových aplikací. Obě technologie jsou rozdílné v tom, že jedna používá bitmapový a druhá vektorový formát.

- **SVG** (Scalable Vector Graphics – škálovatelná vektorová grafika) je vektorový grafický formát založený na XML. SVG lze stylovat pomocí CSS a lze s ním pracovat dynamicky pomocí DOM (Document Object Model – objektový model dokumentu). SVG je flexibilní – jednotlivé objekty mohou být statické, dynamické, interaktivní či animované.
- **Canvas** je bitmapový formát, HTML5 specifikace poskytuje JavaScriptové API pro programové kreslení do webové aplikace. Canvas definuje objekt canvas (zapsaný jako element `<canvas>` v HTML stránce), do kterého se umisťuje grafika. Kreslení probíhá pomocí tzv. kontextů:
  - **2D kontext** neposkytuje žádné funkce pro animace, umožňuje pouze operace s jednotlivými pixely, jako je např. používání obrazových filtrů.

Lze do něj vložit bitmapové obrázky, výsledný obrázek lze uložit ve formátu `png` nebo `jpg`;
- **3D kontext** slouží pro kreslení trojrozměrné grafiky, textur a stínování. Podporuje i animace.

WebSockets

Protokol HTTP, pomocí kterého komunikuje klient (nejčastěji prohlížeč uživatele) se serverem, je bezstavový na principu výzva-odpověď. Klient vyšle požadavek a server odpoví. S postupem času a s růstem požadavků na webové aplikace přestává být tento protokol dostačující, protože server nemůže poslat klientovi data, aniž by si o to klient požádal (protokol je pouze jednosměrný). Nebylo tedy možné jednoduše vytvářet pomocí webových technologií aplikace např. pro komunikaci uživatelů v reálném čase. Tento problém řeší např. technologie Comet [18].

HTML5 nyní přináší technolgii Web Sockets, pomocí níž klient může navázat obousměrné spojení se serverem přes HTTP protokol a poté si můžou vyměňovat data v reálném čase.

Web Workers

Většina moderních jazyků (a moderních systémů) je vícevláknová (multi-threaded), což znamená, že podporují vykonávání několika procesů zároveň. V JavaScriptu toto nebylo donedávna možné, nyní lze s technologií Web Workers spouštět více operací najednou. Web Workers jsou vhodné pro jednodušší úkoly a mají určitá omezení, nemohou například pracovat s DOM.

File API

Webové aplikace byli limitovány tím, že nemohli přistupovat k datům v uživatelově souborovém systému, FILE API nyní umožňuje vytvářet, číst, procházet soubory ve vymezené části uživatelova systému.

Drag and Drop

HTML5 zavádí nativní podporu metody Drag and Drop, pokud chtěli vývojáři dříve používat tuto techniku, museli ji implementovat např. pomocí některých JavaScriptových knihoven.

2.2 CSS3

Kaskádové styly slouží k nastavení vzhledu a chování prvků v dokumentu. Vznikly proto, aby se oddělila definice vzhledu dokumentu od obsahu. CSS3 je již třetí verzí specifikace a má mnoho nových možností. Je to například vytvoření kulatých rohů prvků, barevných přechodů, stínů nebo animací [16]. V minulosti se tyto efekty museli tvořit pomocí JavaScriptu.

CSS3 zavádí moduly, což je část specifikace řešící jednu konkrétní věc, jako například animace nebo transformace prvků. Kompletní specifikace CSS3 ještě není hotová, ale některé moduly jsou již dokončeny a jsou také implementovány v prohlížečích. Většina modulů však není dokončena a vývojáři prohlížečů je nemohou implementovat, protože by mohlo dojít k tomu, že by bylo více implementací jedné konkrétní věci. Tvůrci prohlížečů tak používají tzv. vendor prefixy. Jedná se o jednoduchý prefix určité CSS vlastnosti, který říká vývojáři, že tato věc je ve vývoji a její implementace se může změnit. Příklad prefixu u prohlížečů s jádrem Gecko: `-moz-` nebo s jádrem Webkit: `-webkit-`.

Jednou z novinek, které zavádí CSS3 jsou Media Queries. Umožňují zjistit jaké schopnosti zobrazování má zařízení, na kterém se stránka zobrazuje a podle toho může vývojář zvolit vhodný styl pro konkrétní zařízení. Již v předchozích verzích CSS existovaly Media typy, které umožňovali přiřadit styly, pro stránku, která byla zobrazená na monitoru anebo pro stránku, která se tiskla na papír. Tento jednoduchý princip Media Queries výrazně rozšiřují, a tak je možnost zjistit jaké má zařízení rozlišení, úhlopříčku, zda je položeno na šířku (landscape) či na výšku (portrait).

2.3 JavaScript

Jedná se o multiplatformní, objektově orientovaný jazyk sloužící pro tvorbu interaktivních webových aplikací. Je to jazyk interpretovaný, běží přímo v prohlížeči na straně klienta. Byl vyvinut společností Netscape v 90. letech minulého století. Poté jej převzala společnost ECMA zabývající se standardy a standardizovala jej pod jménem ECMAScript [17].

2.3.1 Frameworky

Pro vytváření aplikací pomocí webových technologií je výhodné používat frameworky a pluginy, které nabízí témata vzhledu, uživatelské prvky, animace, známé z nativních aplikací, dále slouží pro snadné zpracování událostí a interakcí uživatele. Nejznámější dostupné frameworky jsou například:

- **Sencha Touch** [22] framework vytvořený společností Sencha. Pomocí tohoto frameworku je možné psát kompletní aplikace pracující s daty, dále nabízí funkce pro práci s grafy. Zatím podporuje pouze 3 platformy a to Android, iOS a Blackberry;
- **jQuery Mobile** [23] je open source projekt, který používá knihovnu jQuery. Pomocí tohoto frameworku lze snadno vytvořit uživatelské rozhraní aplikace;
- **Dojo Mobile** [25] je JavaScriptový framework pro tvorbu mobilních webových aplikací, který umožňuje tvorbu uživatelského rozhraní, a také podporuje práci s daty a tvorbu grafů;
- **jQTouch** [24] je plugin pro JavaScriptový framework jQuery. Slouží k snadné tvorbě uživatelského rozhraní aplikací.

2.4 Webové frameworky

Webové aplikace běží v sandboxu (jsou odděleny od systému), tudíž nemohou přistupovat k funkcím a službám systému. Dále je není možné distribuovat pomocí kanálů pro distribuci standardních nativních aplikací. Tyto nedostatky je možné překonat tím, že se aplikace umístí do nativní „wrapper aplikace“, která obsahuje webový prohlížeč. Webová aplikace je v něm zobrazena. Pokud je potřeba vytvořit aplikaci pro mnoho platforem je nutné vytvořit mnoho „wrapper aplikací“ nebo je možnost použít frameworky jako jsou PhoneGap či Titanium, které již takové „wrapper aplikace“ mají vytvořené, a tím ulehčí nasazení aplikace do telefonů.

2.4.1 PhoneGap

PhoneGap je framework, který umožňuje tvorbu nativních aplikací pomocí webových technologií [20]. Jedná se open source software šířený pod Apache Licencí 2.0. Podporuje následující platformy iOS, Android, Windows Phone, Blackberry, WebOs, Symbian a Bada. Aplikace běží v komponentě webového prohlížeče, kde se vykresluje uživatelské rozhraní, dále PhoneGap poskytuje JavaScriptový objekt `navigator`, pomocí kterého lze v Javascriptu přistupovat ke zdrojům a službám systému jako jsou např.:

- akcelerometr,
- gps,
- kompas,
- kamera,
- kontakty,
- notifikace.

PhoneGap nabízí službu PhoneGap Build, pomocí které lze vytvořit z webové aplikace aplikaci nativní bez instalace SDK, používání IDE (Integrated development environment – integrované vývojové prostředí) a bez vlastní kompilace. Aplikace se pouze nahraje do služby PhoneGap Build, ta vygeneruje nativní aplikace pro zvolené platformy a poté je nabídne ke stažení.

2.4.2 Titanium

Titanium je framework, který pracuje odlišně oproti PhoneGapu. Aplikace se tvoří pomocí HTML a JavaScriptu a používají se funkce z Titanium SDK. Výsledná aplikace se přeloží v Titanium studio do nativního kódu pro mobilní platformy. Zatím podporuje pouze platformy Android a iOS [21].

3 GEOLOKACE

Geolokace je proces zjištění aktuální polohy zařízení na základě dostupných informací o síťovém zařízení nebo na základě měření přenosu signálu [26].

3.1 Metody geolokace

Metody geolokce lze rozdělit na dva druhy, aktivní a pasivní. Pasivní metody spočívají v principu získávání informací důležitých k určení lokace ze síťových zařízení. Tyto informace jsou porovnány s veřejnými nebo soukromými databázemi a podle výsledku srovnání je určena pozice zařízení. Aktivní metody spočívají v určení polohy zařízení pomocí měření signálu v přenosové síti.

3.1.1 Gelokace podle IP adresy

Velmi málo přesná metoda zjištění pozice, za to je dostupná vždy, protože každé zařízení se připojuje do internetu pomocí veřejné IP adresy (buď své vlastní nebo adresy providera). Při požadavku na zjištění aktuální pozice dojde k porovnání IP adresy s databází IP adres a k vrácení polohy, která přísluší požadované IP adrese. Přesnost této metody je od několika metrů až po stovky km (uživatel přistupuje k internetu ze vzdálené vesnice a provider má internetovou bránu s veřejnou IP adresou ve 100 km vzdáleném městě), tudíž na tuto metodu není spolehnutí.

3.1.2 Gelokace podle GSM signálu

Metoda využívající se pouze u zařízení, která mají GSM modul. Princip je takový, že zařízení přijímá signál z BTS (Base Transceiver Station), informace o dostupných BTS se porovnají s databází. Na základě výsledků, které vrátí polohu jednotlivých BTS, a síly signálů od jednotlivých stanic se vypočítá pozice zařízení. Tato metoda vrací pozici s přesností od stovek metrů do několika kilometrů.

3.1.3 Gelokace podle Wi-Fi

Pokud má zařízení Wi-Fi (Wireless Fidelity) kartu, může být poloha uživatele zjištěna pomocí aktuálně dostupných Wi-Fi sítí. Při požadavku na zjištění polohy si zařízení zjistí aktuální dostupné sítě, jejich SSID (Service Set Identifier) a MAC (Media Access Control) adresu, poté pošle dotaz do databáze, kde se hodnoty porovnají, zpracují a zpět uživateli je poslána přibližná pozice. Přesnost této metody je okolo několika desítek metrů. Databází, podle kterých je možné lokalizovat uživatele,

existuje několik, např. od společnosti Skyhook, Inc. nebo Google, ta vlastní největší a pravděpodobně nejpřesnější databázi. Tato metoda nemusí být vždy úspěšná, protože v databázích nejsou zapsány všechny existující Wi-Fi sítě.

3.1.4 Geolokace podle satelitních navigačních systémů

Družicový systém je služba umožňující pomocí signálů z družic určit polohu. Jedná se o nejpřesnější metodu určení polohy a poskytuje lokaci s přesností kolem 5-15 m, hlavní nevýhodou je, že nefunguje uvnitř budov. Družice vysílají signál v přesně definovaný okamžik. Přijímač umístěný na zemi, přijímá signál od několika družic a na základě rychlosti a zpoždění signálů, vypočítá časový rozdíl mezi jednotlivými signály, tím určí svou polohu vzhledem k jednotlivým družicím [28].

Systém NAVSTAR GPS

Jedná se o systém Ministerstva obrany Spojených států Amerických. Do plného provozu byl zaveden v roce 1994, kdy bylo na oběžné dráze všech 24 družic. Systém GPS se také používá jako přesný referenční nástroj při určování času.

GPS se dělí na tři hlavní segmenty. Jsou to:

- **Kosmický segment** se skládá z GPS družic, kterých je při plném provozu 24. Jsou rozděleny do 6 oběžných drah a obíhají přibližně ve výšce 20 000 km;
- **Řídící segment** monitoruje dráhy letu GPS družic. Tato data jsou zpracovávána a posílána každé GPS družici pro aktualizaci navigačních dat;
- **Uživatelský segment** tvoří přijímače uživatelů. Přijímače se skládají z antény, procesoru a vysoce stabilních hodin.

Sytém GALILEO

Je plánovaný evropský Globální družicový polohový systém. Při plném provozu má být na oběžné dráze 30 satelitů, z nichž 27 je operačních a 3 záložní. Satelity mají obíhat ve výšce 23 222 km. Systém Galileo má nabízet určení polohy s přesností lepší než 1 metr. Největší potenciál využití má být ve všech druzích dopravy.

Systém GLONASS

Jedná se o navigační systém provozovaný Ruskem. V této době je již systém v provozu a je možnost jej využívat i pro civilní účely. Kompletní konstelace se skládá z 24 družic, z nichž 21 je v provozu a 3 jsou záložní. Družice obíhají Zemi ve výšce 19 100 km. Používají se dva typy signálů, jeden o standardní přesnosti a druhý o vysoké přesnosti. Přenos dat z více družic k jednotlivým uživatelům je řešen pomocí přístupu FDMA (Frequency Division Multiple Access).

3.1.5 Geolokace na základě měření signálu v přenosové síti.

Patří mezi aktivní geolokační metody. Poloha je odhadována na základě informací získaných měřením datového přenosu stanice v internetové síti. Při zjišťování pozice se měří zpoždění signálu mezi uzly přenosové cesty. Poloha je určena pomocí referenčních uzlů, u kterých jsou známy jejich geografické souřadnice [27].

3.2 Geolokace pomocí webových technologií

Zjišťovaní pozice pomocí webových technologií zpravidla využívá 4 metody geolokace, a to podle IP adresy, GSM nebo Wi-Fi signálu, nebo podle GPS. W3C standardizovalo specifikaci Geolocation API, která umožňuje snadné zjištění polohy uživatele přímo z webového prohlížeče. V JavaScriptu existuje objekt `navigator`, který slouží pro získávání údajů o zeměpisné šířce a délce, o nadmořské výšce a dalších užitečných informacích. Při zpracování se vybere nepřesnější dostupná metoda geolokace, tzn. že pokud zařízení nemá GPS, GSM ani Wi-Fi modul použije se lokace pomocí IP adres, na druhou stranu pokud má zařízení GPS modul a programátor určí, že se má použít nejpřesnější metoda geolokace, použije se GPS lokace. Při požadavku na zjištění pozice musí uživatel z bezpečnostního hlediska povolit sdílení své pozice.

3.3 Přehled mapových podkladů a API

Pro tvorbu geolokační aplikace mohou vývojáři využít některé z veřejně dostupných mapových podkladů a API. V následující části je přehled těch nejběžnějších.

Google Maps

Společnost Google vydala první verzi svých map v roce 2005 od té doby je neustále vyvíjí a přidává nové možnosti. V současné době je Google Maps API již ve třetí verzi [29]. Nabízí základní mapové podklady, satelitní snímky, terénní mapu a také hybridní zobrazení mapy, což je kombinace základní mapy a satelitních snímků. Pomocí API lze snadno vykreslit trasu mezi dvěma body, vypsat instrukce pro navigaci, přidávat vlastní body, zjišťovat jejich nadmořskou výšku a mnoho dalších informací. Dále je možné využít několik knihoven např. pro kreslení přímo do mapy nebo pro přidání fotografií ze služby Panoramio. Google dále poskytuje službu Street View, která umožňuje zobrazit různá místa světa prostřednictvím panoramatických snímků [30].

Google Maps jsou celosvětové, tudíž nenabízejí některé specifické vlastnosti pro konkrétní státy. Například v České republice neposkytuje Google zobrazení turistických a cyklistických tras. Od konce roku 2011, přestalo být Google Maps API zcela zdarma. Při stálém překračování 25 000 zobrazení mapy za jeden den je nutné zaplatit licenční poplatek, poté je již možno využívat mapy téměř neomezeně.

Mapy.cz

Mapy.cz vznikly jako první ryze český projekt v oblasti online map. API je ve verzi 4.7. Nenabízí tolik možností jako Google Maps API, ale obsahuje pouze základní funkce jako zobrazení vlastních značek, vizitek, práce s vektorovou grafikou a geokódování [1].

Jsou cíleny přímo pro Českou republiku, a tak zde nabízí jedny z nejkvalitnějších podkladů. Na rozdíl od Google Maps poskytují mapu s turistickými a cyklistickými trasami a také historickou mapu [31].

Ovi Maps

Stejně jako Google Maps poskytují základní mapu, satelitní snímky, terénní a hybridní mapu. API rozhraní umožňuje vkládání bodů, vizitek a vektorů do mapy, dále umožňuje snadno vytvořit kontextové menu, podporuje tvorbu tras a příkazů pro navigaci [32].

Bing Maps

Jsou mapy společnosti Microsoft byly představeny v roce 2009, zatím jsou určeny primárně pro Severní Ameriku, a tak v České republice nejsou satelitní mapové podklady příliš kvalitní. API nabízí možnosti srovnatelné s ostatními mapami. Bing Maps nabízí Streetside View což je obdoba Street View od Googlu, zatím obsahuje snímky pouze ze Severní Ameriky a Anglie [33].

Open street map

Otevřený projekt OpenStreetMap [34] nabízí volně dostupná geografická data, která jsou získávána integrací dat z různých zdrojů, především individuálním sběrem dat pomocí GPS zařízení. V České republice není kvalita pokladů zatím příliš velká.

[1] Proces zjištění zeměpisných souřadnic ze zadané adresy

4 VÝVOJ MULTIPLATFORMNÍ APLIKACE

4.1 Cíle aplikace

Cílem práce bylo prozkoumat možnosti moderních webových technologií, jako jsou HTML5, CSS3 a JavaScript. A pomocí nich vytvořit geolokační aplikaci, která bude schopna fungovat na různých mobilních a webových platformách.

Zaměření aplikace

Jelikož nebylo v zadání uvedeno konkrétní téma aplikace, bylo dohodnuto s vedoucím práce, že bude vytvořena aplikace, která bude sloužit studentům Fakulty elektrotechniky a komunikačních technologií VUT v Brně. Zejména novým studentům na této fakultě. Protože není snadné se zorientovat v novém prostředí a aplikace jim poskytne přehledné zobrazení důležitých bodů, jako jsou budovy fakult, knihovny, menzy nebo sportoviště CESA. Tyto body budou zobrazeny na mapě a uživatelé si k nim budou moci zobrazit trasu. Aplikace byla nazvána **xStudent**. Aplikace umožní zobrazit doplňující informace k jednotlivým budovám, jako jsou např. otevírací doby u knihoven a menz. U menz bude také možné zobrazit aktuální jídelní lístek. Uživatelé budou moci do aplikace vložit svůj rozvrh hodin. Poté se bude přehledně zobrazovat na úvodní straně s dalšími informacemi. Jako je budova, kde probíhá vyučování, vzdálenost a čas za jaký tam dorazí.

Shrnutí požadavků na aplikaci:

- zobrazení důležitých míst na mapě;
- zobrazení trasy k jednotlivým bodům;
- výpis informací k budovám;
- zobrazení jídelníčku u menz;
- možnost vložení a správy rozvrhu.

4.2 Výběr technologií a frameworků

Důvodů proč využít framework je několik. Frameworky jsou knihovny, které ulehčují práci při programování. Řeší typické problémy v určitých oblastech programovacího jazyka a vývojář se tak může lépe soustředit na tvorbu aplikace. Dále nabízí například ladící a testovací nástroje nebo již předpřipravené komponenty pro tvorbu uživatelského rozhraní. Výhodou je, že mají většinou vhodně zvolenou architekturu a vedou prgramátora k psaní správně strukturovaného a snadno udržovatelného kódu.

JavaScriptový framework

Vzhledem ke zkušenostem se standardní knihovnou jQuery, byl na začátku vývoje zvolen framework jQuery Mobile pro vytvoření uživatelského rozhraní. Tento framework byl v té době ve stádiu beta verze, a ještě nefungovali všechny části správně. Aplikace se začala čím dál více rozšiřovat, bylo potřeba rozdělit aplikaci na jednotlivé celky a jelikož jQuery Mobile nepodporuje návrhové vzory např. MVC (Model-View-Controller), nepoužívá ani jmenné prostory bylo rozhodnuto pro změnu frameworku.

Následně byl vybrán framework Sencha Touch, který podporuje MVC strukturu, umožňuje práci s daty a obsahuje pokročilejší možnosti pro tvorbu mobilních aplikací.

Mapové podklady

Jelikož je aplikace zaměřena na Českou republiku, konkrétně na Brno, bylo zvažováno využití mapových podkladů Mapy.cz. Nakonec bylo rozhodnuto pro využití Google Maps API, jelikož nabízí propracovanější API, a také nabízí více možností pro další rozšiřování, jako je například využití vrstvy Panoramio pro zobrazování fotografií nebo zobrazení snímků ze služby Street View, které se začíná velmi rozšiřovat i v České republice.

Tab. 4.1: Porovnání možností Google Maps API a Mapy.cz API

	Google Maps	Mapy.cz
Street view	+	-
Turistické trasy	-	+
Cyklististické trasy	-	+
Vrstva s fotografiemy	+	-

Serverová část

Z důvodu potřeby centrální správy a ukládání dat o jednotlivých budovách, je nutné vybrat vhodnou serverovou technologii. Pro tyto účely byla zvolena technologie PHP. Informace o budovách se budou ukládat do databáze MySQL, z důvodu, že PHP a MySQL spolu velmi snadno komunikuje. Jazyk PHP ve své verzi 5.3 je již velmi vyzrálý a obsahuje pokročilé možnosti programování. Jeho výhodou je, že běží na webovém serveru Apache, který je dostupný pro více operačních systémů.

Jelikož i při vývoji serverové aplikace je vhodné použít některý z frameworků, byly prozkoumány funkce a vlastnosti frameworků Nette, Zend a Symphony. Z těchto tři byl vybrán Nette framework [35].

Nette Framework – na rozdíl od Zend frameworku má téměř dokonalé zabezpečení proti XSS (Cross-site scripting), CSRF (Cross-site request forgery), SQL Injection a dalším útokům. Dále obsahuje kontextové escapovaní výstupů, což znamená, že framework pozná, v jakém místě se řetezec vypisuje (HTML stránka, JavaScript, CSS) a podle toho použije vhodnou escapovací funkci. Na rozdíl od jiných frameworků obsahuje pokročilý ladicí nástroj, díky kterému, se velmi snadno odhalují chyby v programu. Pro českého programátora má velkou výhodu v tom, že jej vytvořili čeští vývojáři a v Česku má největší aktivní komunitu PHP vývojářů, tudíž je možné řešit jakékoliv problémy téměř okamžitě.

4.3 Implementace pomocí vybraných technologií

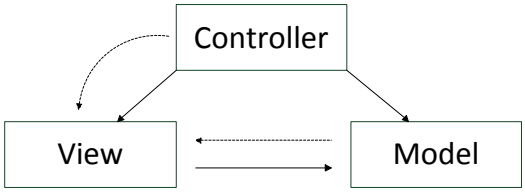
4.3.1 Návrh uživatelského rozhraní

Jelikož se aplikace bude primárně využívat na mobilních zařízeních s dotykovou obrazovkou je nutné přizpůsobit ovládací prvky, tak aby je bylo možno používat dotykem prstu. Framework Sencha Touch nabízí ovládací prvky a komponenty uživatelského rozhraní přímo přizpůsobené pro ovládání dotykem. Rozvržení rozhraní aplikace bylo zvoleno tak, že každá část aplikace má vlastní kartu a tyto karty se přepínají pomocí záložek umístěných ve spodní části displeje (obr. 4.1).

Obr. 4.1: Návrh uživatelského rozhraní aplikace.

4.3.2 Implementace webové aplikace

Jako první byla vytvořena webová aplikace pomocí Sencha Touch. Sencha Touch používá MVC (Model-View-Controller) architekturu, aplikace je rozdělena do tří částí Model, View a Controller.



Obr. 4.2: MVC Architektura aplikace.

- modelová vrstva představuje základ celé aplikace, obsahuje práci s daty;
- vrstva view má na starosti zobrazení a prezentaci dat uživatelům;
- controller je část aplikace, která řídí a zpracovává požadavky uživatele, na základě těchto požadavků volá funkce z modelové vrstvy a výsledky předává do vrstvy view.

Adresářová struktura

Aplikace používá následující adresářovou strukturu.

```
xstudent/
    app/                # adresář s aplikací
        controller/     # adresář pro umístění controllerů
        model/          # modelová část aplikace
        store/          # třídy pro práci s modely a pro
                          ukládání dat do persitentního uložiště
        view/           # vrstva view aplikace

    build/              # adresář, do kterého je uložena
                          produkční verze aplikace
    resources/          # adresář pro obrázky, ikony a css soubory
    sdk/                # knihovna Sencha Touch

    app.js              # základní soubor pro aplikaci
    app.json            # konfigurační soubor aplikace, používá
                          se při vytváření produkční verze
    index.html          # vstupní soubor celé aplikace, stará se
                          o načtení css a js souborů
```

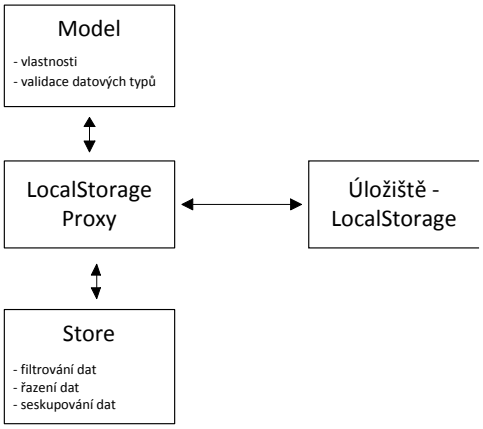
Struktura jmenných prostorů

Jmenné prostory oddělují aplikaci do logických celků, díky nim se názvy tříd aplikace zpřehlední. V aplikaci jsou používány následující jmenné prostory:

- Jmenné prostory modelové vrstvy mají tvar `xStudent.model` a za ním následuje název třídy;
- Jmenné prostory controllerů používají tvar `xStudent.controller` a název třídy controlleru;
- View vrstva používá jmenné prostory `xStudent.view`.

Modelová část aplikace

Modelovou část aplikace tvoří třídy, které dědí od `Ext.data.Model` a `Ext.data.Store`. Potomci třídy `Ext.data.Model` vytváří jednotlivé entity. Což jsou třídy, které slouží jako reprezentace dat, obsahují vlastnosti a metody pro validaci datových typů vlastností. V aplikaci jsou takto reprezentovány jednotlivé budovy a předměty v rozvrhu hodin. Třídy, které dědí od `Ext.data.Store`, slouží jako kolekce instancí jednotlivých entit a obsahují funkce pro práci s nimi jako je filtrování, řazení apod.



Obr. 4.3: Struktura modelové vrstvy.

Ukládaní dat

Aby nedocházelo při každém spuštění aplikace k načítání informací o budovách ze serveru, a tím ke zbytečnému přenosu dat. Je nutné ukládat informace o budovách v nějakém persistentním úložišti. Dále je potřeba ukládat rozvrh hodin. Pro ukládání těchto dat v aplikaci je použita technologie z HTML5 LocalStorage. Ukládání a načítání dat z tohoto persistentního úložiště zajišťuje třída

`Ext.data.proxy.LocalStorage` z frameworku Sencha Touch. Po načtení dat z LocalStorage jsou vytvořeny instance entit modelu a předány konkrétním potomkům třídy `Ext.data.Store`.

View vrstva aplikace

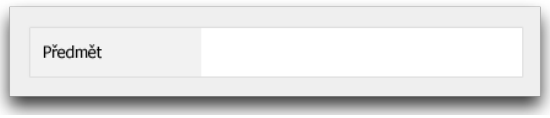
Při vytváření aplikace pomocí Sencha Touch Frameworku se uživatelské rozhraní vytváří odlišně oproti ostatním knihovnám jako je např. jQurey Mobile. V těchto knihovnách se uživatelské rozhraní definuje pomocí html tagů a datových atributů, ve kterých je určeno jaký uživatelský prvek se má zobrazit. Poté dochází pomocí JavaScriptu k překonvertování těchto tagů na požadované uživatelské prvky.

V Sencha Touch frameworku je uživatelské rozhraní tvořeno pomocí JavaScriptových tříd, ve kterých se pomocí objektů a vlastností definují jednotlivé ovládací a vizuální prvky rozhraní. Z těchto tříd se vytváří DOM výsledné aplikace.

```
Ext.create('Ext.form.Panel', {
  fullscreen: true,

  items: [{
    xtype: 'fieldset',
      items: [
        {
          xtype: 'textfield',
          name : 'subject',
          label: 'Předmět'
        }
      ]
  }]
});
```

Program 1: Definice jednoduchého formuláře pomocí Sencha Touch.



Obr. 4.4: Vygenerovaný formulář.

4.3.3 Implementace nativní aplikace

Pro tvorbu nativní aplikace z již vytvořené webové byl použit framework PhoneGap. Jelikož je pro tvorbu aplikací na platformu iOS potřeba vývojářský účet, který je zpoplatněn, byla vytvořena nativní aplikace pouze pro Android. Nativní aplikace

pro ostatní platformy nebyly vytvořeny, protože v době tvorby nebyly k dispozici zařízení s jinými systémy. Pro platformu iOS byla aplikace vytvořena a vyzkoušena v iOSSimulátoru. Zde byla aplikace plně funkční a měla totožné chování jako na platformně Android.

Platforma Android

Při tvorbě Android aplikace bylo využíváno prostředí Eclipse a Android SDK. Po založení základní Android aplikace, je nutné vytvořit v adresářové struktuře složky pro knihovnu PhoneGap a pro webovou aplikaci (konkrétně se jedná o složky `libs` a `assets/www`). Nativní aplikace se skládá pouze z jedné aktivity, kterou je potřeba upravit tak, aby tato třída dědila od třídy DroidGap z knihovny PhoneGap. Poté tato aktivita obsahuje komponentu `Webview`. V této komponentě se zobrazuje již vytvořená webová aplikace. Aby bylo možné využívat funkcí a vlastností zařízení je nutné ve webové aplikaci načíst JavaScriptový soubor `cordova.js`, který zpřístupní jednotlivé funkce systému webové aplikaci. Na závěr je potřeba upravit soubor AndroidManifest.xml, ve kterém jsou uvedeny informace o aplikaci.

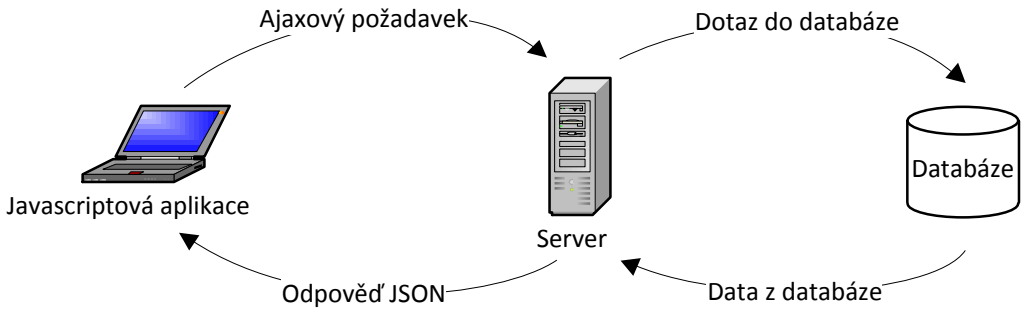
Soubor AndroidManifest.xml - v tomto souboru jsou základní informace o aplikaci určené pro systém. Je v něm uvedeno jméno aplikace, jméno balíku, dále popisuje komponenty aplikace jako jsou aktivity, služby apod. Obsahuje minimální požadovanou verzi Android API, a jsou zde vyjmenovány všechny části a funkce systému, které aplikace vyžaduje pro svůj běh (např. přístup k GPS, přistup ke kameře atd.).

4.3.4 Serverová část

Serverová část není přímo spojená s mobilní aplikací, slouží pro uchování a centralizované spravování informací o budovách. Mobilní aplikace získává informace pomocí AJAX (Asynchronous JavaScript and XML) požadavků ze serveru. Na serveru se informace vyberou z MySQL databáze, zpracují se do formy JSON (JavaScript Object Notation) objektů a následně jsou odeslány do mobilní aplikace (obr. 4.5).

Na serveru se také zpracovává jídelní lístek z jednotlivých menz. Při zobrazení jídelního lístku, mobilní aplikace pošle požadavek na server, zde se načtou data z webových stránek konkrétní menzy zpracují se opět do formy JSON objektů a jsou poslány mobilní aplikaci, která výsledek přehledně zobrazí.

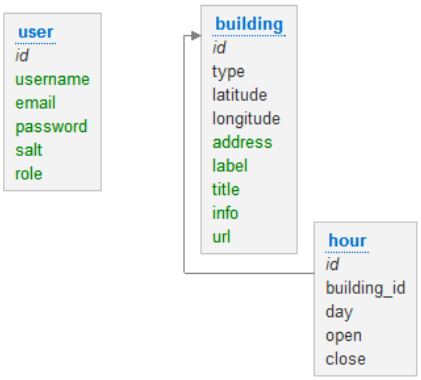
V serverové části bylo vytvořeno rozhraní pro snadnou správu informací o budovách. Screenshoty do tohoto rozhraní a přihlašovací údaje jsou v příloze A.2.



Obr. 4.5: Průběh komunikace mezi aplikací a serverem.

Databáze na serveru

Na serveru jsou informace uloženy v databázi MySQL. Jelikož je potřeba zobrazovat pouze základní informace o budovách bylo vytvořeno jednoduché schéma databáze (obr. 4.6), které obsahuje tři databázové tabulky. V tabulce `building`, jsou uloženy základní informace o budově jako je název, adresa, zeměpisná šířka, délka apod. U některých typů budov je důležité zobrazit otevírací dobu. Tyto informace jsou uloženy v tabulce `hour`. Mezi těmito tabulkami je vztah 1:N. V tabulce `hour` je u každého záznamu uložen identifikátor budovy, ke které se konkrétní záznam vztahuje. Dále je uloženo číslo dne a otevírací doba. Otevírací a zavírací doba je uložena ve formě celého čísla, vyjadřujícího počet sekund v aktuálním dni, tento způsob uložení času byl vybrán z důvodu snadné manipulace a porovnávání jednotlivých časů.



Obr. 4.6: Struktura MySQL databáze.

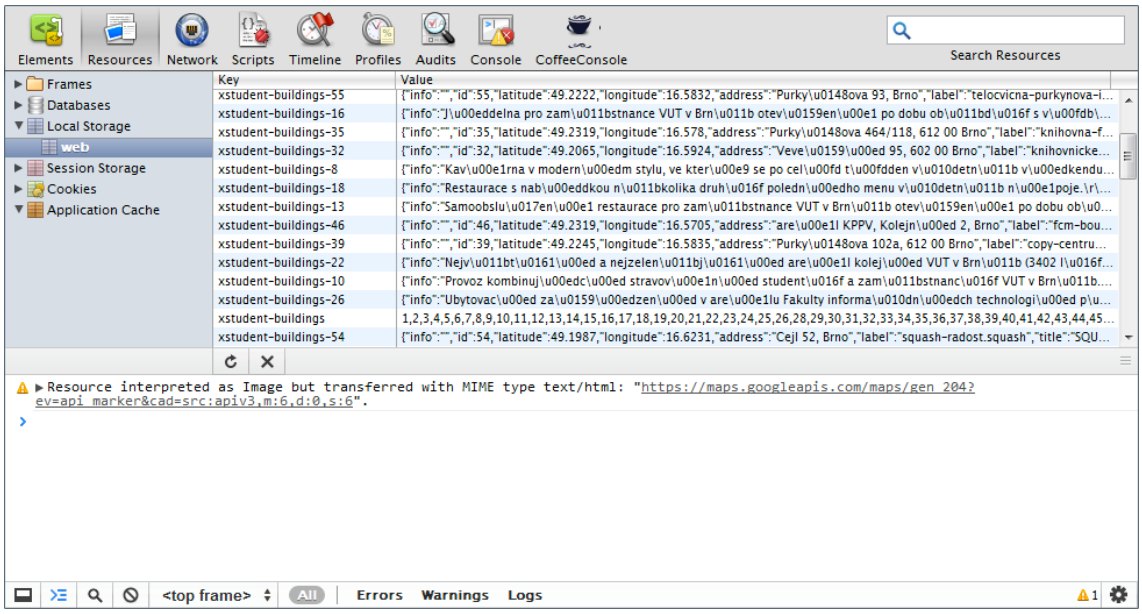
Aby neměl kdokoli přístup do serverového rozhraní je využívána autentizace uživatelů. K tomu slouží třetí tabulka `user`, ve které jsou uchovány informace o uživatelích, jenž mají do tohoto rozhraní přístup.

5 TESTOVÁNÍ APLIKACE A ZHODNOCENÍ VÝVOJE

Testování a ladění aplikace probíhalo ve dvou krocích. V prvním kroku byla aplikace testována a laděna ihned při vývoji, zde bylo testování zaměřeno na funkčnost a spolehlivost celé aplikace. V dalším kroku bylo provedeno uživatelské testování [36], které odhalilo nedokonalosti v uživatelském rozhraní aplikace. Na závěr kapitoly jsou zhodnoceny vlastnosti aplikace na různých platformách a zařízeních.

5.1 Testování při vývoji

JavaScript je interpretovaný jazyk, to znamená, že příkazy jsou vykonávány přímo v prohlížeči při spuštění programu. Z toho důvodu se program nekompiluje ani nepřekládá do strojového kódu. Není tedy snadné zjistit některé chyby ihned při programování ve vývojovém prostředí. Z toho důvodu je vhodné používat některý z programátorských nástrojů nabízených přímo v prohlížečích. Při vývoji aplikace byl použit prohlížeč Google Chrome, který nabízí ladící a testovací nástroj **Web Inspektor** (obr. 5.1). Tento program obsahuje chybovou konzoli, profilovací nástroj, možnost zobrazení HTTP požadavků, procházení DOM a další užitečné funkce, díky kterým se dají snadněji odhalit chyby v aplikaci.



Obr. 5.1: Webinspektor v prohlížeči Google Chrome se zobrazeným výpisem LocalStorage a chybovou konzolí.

5.2 Uživatelské testování

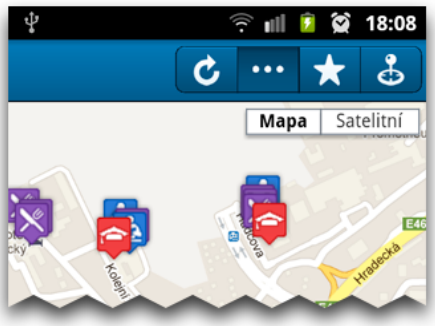
Uživatelské testování bylo provedeno jak s uživateli, kteří jsou již studenti VUT v Brně, tak i s uživateli, kteří VUT v Brně nenavštěvují. Uživatelé měli za úkol splnit několik jednoduchých úkonů, které odhalily problémy v uživatelském rozhraní.

Scénáře uživatelského testování:

- vytvořit rozvrh hodin a poté jej upravit;
- nalézt otevřené knihovny v určitém dosahu;
- zobrazit jídelní lístek v nejbližší otevřené menze.

Výsledky uživatelského testování

Základní struktura rozhraní, která byla zvolena, se dle uživatelského testování zdá být vhodnou. Většina uživatelů neměla při plnění úkolů problémy. Komplikace nastaly pouze v některých částech aplikace, kdy uživatelé nevěděli, který ovládací prvek k čemu slouží, viz obr. 5.2. Bylo to z důvodu, že nebyli vhodně zvolené ikony reprezentující konkrétní ovládací prvky. Řešení tohoto problému spočívá ve vybrání a použití vhodnějších ikon.



Obr. 5.2: Nevhodně zvolené ikony v horní liště způsobily problémy při rozhodování, který ovládací prvek zvolit.

5.3 Zhodnocení vlastností aplikace na různých platformách

Aplikace vytvořené pomocí Sencha Touch frameworku, zatím podporují platformy Android, iOS, Blackberry a webové prohlížeče s jádrem Webkit. Aplikace byla otestována na všech platformách dostupných při vývoji. Aplikace byla otestována na platformě PC a MAC v prohlížečích Google Chrome a Safari. Dále bylo provedeno testování aplikace v prohlížečích na mobilních platformách Android a iOS. Nakonec

byla vytvořena nativní aplikace pro platformu Android a iOS. Android aplikace byla otestována jak na mobilním telefonu tak i na tabletu. iOS aplikace byla otestována v iOS simulátoru.

Na všech těchto platformách běží aplikace téměř stejně, rozdíl je pouze v plynulosti a rychlosti aplikace. Jak je patrné z výsledků testů v příloze C, prohlížeče v mobilních zařízeních zatím neposkytují takový výkon jako v desktopových počítačích. To se projevuje i na plynulosti vytvořené aplikace. Hlavně u platformy Android, kde je nejvíce patrné trhání např. při posunování listu s výpisem budov nebo při práci s mapou.

6 ZÁVĚR

Cílem bakalářské práce bylo vytvořit geolokační aplikaci, která je schopna běžet na různých mobilních a webových platformách. Výsledkem bakalářské práce je aplikace **xStudent**, která slouží studentům FEKT VUT v Brně. Aplikace jim nabízí přehledné zobrazení důležitých míst na mapě a trasu k nim. Dále si mohou uživatelé vypsat doplňující informace k jednotlivým budovám, jako jsou adresy, vzdálenosti nebo čas, kdy k nim dorazí. Aplikaci je možné spustit v prohlížečích na platformách Android, iOS, MAC a PC. Dále byla vytvořena nativní aplikace pro Android.

V úvodu práce jsou sepsány vlastnosti dostupných mobilních platforem a možnosti vývoje aplikací. Poté se práce věnuje webovým technologiím vhodným k multiplatformnímu vývoji. Třetí kapitola se zabývá geolokačními metodami a možnostmi využití mapových podkladů při tvorbě aplikací. V předposlední kapitole je popsán výběr technologií použitých pro tvorbu aplikace **xStudent**, následně je popsána struktura celé aplikace. Pro tvorbu aplikace byl využit framework Sencha Touch a aplikace používá MVC architekturu. Pro vývoj serverové části aplikace, kde se uchovávají informace o budovách, a kde je možno tyto informace spravovat, je použita technologie PHP a MySQL. Na závěr práce je popsáno testování vytvořené aplikace. Jsou zde zhodnoceny problémy, které testování aplikace odhalilo.

V průběhu práce bylo zjištěno, že současné době mobilní platformy neposkytují takový výkon, pro běh aplikací v prohlížeči, jako desktopové platformy, to je také patrné z přílohy C. Proto technologie HTML5 zatím není vhodná pro vytváření velmi komplexních mobilních aplikací. Pomocí této technologie není ani možné zatím vytvořit všechny druhy aplikací. Například u systému Android je při přepnutí aplikace na jinou, takto vytvořená aplikace pozastavena. Proto je nemožné vytvářet aplikace, které by pracovaly na pozadí operačního systému. U systému Android 2.3 také nelze v JavaScriptu zachytit události pro multidotyková gesta. To je patrné i u vytvořené aplikace, kde nelze přibližovat nebo oddalovat mapu pomocí gesta `pinch`.

Jelikož se většina studentů dopravuje do školy pomocí MHD bylo by vhodné implementovat do aplikace zobrazování MHD spojů, aby měli uživatelé přehled, kterým spojem a za jak dlouho se dostanou do školy. Tato funkce nebyla implementována z důvodu nedostupnosti API, pomocí kterých se dají vyhledávat spoje a objem práce na vlastní implementaci této funkce by značně přesahoval zadání bakalářské práce.

Protože jsou budovy VUT celkem velké a dále vzniká na Technické 12 nový vzdělávací komplex bylo by užitečné, aby aplikace uměla zobrazit vnitřní strukturu a plán jednotlivých budov. Díky tomu by se uživatelé mohli jednoduše orientovat i uvnitř budov.

LITERATURA

[1] GARGENTA, M. *Learning Android*. Sebastopol: O'Reilly, 2011. 268 s. ISBN 978-1-449-39050-1.

[2] BRANNAN, J. – WARD, B. *iOS SDK Programing*. McGraw-Hill Osborne Media, 2011. 528 s. ISBN 978-0-07-175909-0.

[3] CAMERON, R. *Pro Windows Phone 7 Development*. New York: Apress, 2011. 481 s. ISBN 978-1-4302-3219-3.

[4] *Blackberry Developer Zone* [online]. 2012 [cit. 2012-3-15]. Dostupné z WWW: <http://us.blackberry.com/developers/>.

[5] MORRIS, B. *The Symbian OS Architecture Sourcebook*. Chichester: Wiley, 2007. 630 s. ISBN 978-0-4700-1846-0.

[6] MORRIS, B. *Introducing to bada: A Developer's Guide*. Chichester: Wiley, 2010. 504 s. ISBN 978-0-4709-7401-8.

[7] *Introducing HP webOS 2.1* [online]. 2012 [cit. 2012-3-15]. Dostupné z: <https://developer.palm.com/content/resources/intro_to_hp_webos_2_1.html>.

[8] *About MeeGo* [online]. 2012 [cit. 2012-3-15]. Dostupné z: <https://meego.com/about>.

[9] MAIER, D. *Sales of Smartphones and Tablets to Exceed PCs* [online]. 2011-10-6 [cit. 2012-3-18]. Dostupné z: <http://www.practicalecommerce.com/articles/3069-Sales-of-Smartphones-and-Tablets-to-Exceed-PCs->.

[10] *Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent* [online]. 2011-11-15 [cit. 2012-3-18]. Dostupné z: <http://www.gartner.com/it/page.jsp?id=1848514>.

[11] BERNARD, B. *Adobe Flex*. Brno: Computer Press, 2011. 400 s. EAN 978-8-0251-2765-0.

[12] MAHEMOFF, M. *HTML5 vs Native: The Mobile App Debate* [online]. 2011-6-3 [cit. 2012-3-22]. Dostupné z <http://www.html5rocks.com/en/mobile/nativedebate.html>.

[13] KOSEK, J. *Historie a vývoj HTML* [online]. 2012 [cit. 2012-3-25]. Dostupné z <http://htmlguru.cz/uvod-historie.html>.

[14] LAWSON, B. – SHARP, R. *Introducing HTML5*. 2. vyd. Berkeley: New Riders Press, 2011. 312 s. ISBN 978-0-132-79297-4.

[15] PFEIFFER, S. *HTML5 - audio a video, kompletní průvodce*. Brno: Zoner Press, 2011. 352 s. ISBN 978-80-7413-147-9.

[16] GASSTON, P. *The Book of CSS3*. San Francisco: No Starch Press, 2011. 278 s. ISBN 978-1-59327-286-9.

[17] FLANAGAN, D. *JavaScript: The Definitive Guide*. Sebastopol: O'Reilly, 2011. 1098 s. ISBN 978-0-596-80552-4.

[18] RESIG, J. *JavaScript a Ajax – Modermí programování webových aplikací*. Brno: Computer Press, 2007. 360 s. ISBN 978-80-251-1824-5.

[19] LUNNY, A. *PhoneGap Beginner's Guide*. PACKT PUBLISHING, 2011. 328 s. ISBN 978-1-8495-1536-8.

[20] *About PhoneGap* [online]. 2012 [cit. 2012-4-5]. Dostupné z <http://phonegap.com/about>.

[21] *Getting Started with Titanium Studio* [online]. 2012 [cit. 2012-4-5]. Dostupné z <http://wiki.appcelerator.org/display/tis/Getting+Started+with-+Titanium+Studio#GettingStartedwithTitaniumStudio-Runningyour-application>.

[22] MATTHEW, D. *Using Sencha Touch to Build a Mobile Website*. Waltham: Focal Press, 2011. 40 s. ISBN 978-0-240-81909-9.

[23] REID, J. *jQuery Mobile*. Sebastopol: O'REILLY, 2011. 130 s. ISBN 978-1-4493-0668-7.

[24] MATTHEW, D. *Working with jqTouch o Build Websites on Top of jQuery*. Waltham: Focal Press, 2011. 40 s. ISBN 978-0-240-81908-2.

[25] *Dojo Mobile – The Dojo Toolkit* [online]. 2012 [cit. 2012-4-10]. Dostupné z <http://dojotoolkit.org/features/mobile/>.

[26] LUBBERS, P. – SALIM, F. – ALBERS, B. *Pro Html5 Programming: Powerful APIs for Richer Internet Application Development*. New York: Apress, 2010. Kapitola 4, Using the HTML5 Geolocation API, s. 87-114. ISBN 978-1-4302-2790-8.

[27] KOMOSNÝ, Dan – VERNER Lukáš. Geolokace síťových zařízení v internetových sítích. *Elektrorevue* [online]. 2011-6-17 [cit. 2012-4-10]. Dostupné z <http://www.elektrorevue.cz/cz/clanky/komunikacni-technologie/0/geolokace-sitovych-zarizeni-v-internetovych-sitich/>.

[28] *GNSS systémy – Obor kosmických technologií a družicových systémů* [online]. 2012 [cit. 2012-4-12]. Dostupné z <http://www.spacedepartment.cz/3-sekce/gnss-systemy/>.

[29] SVENNERBERG, G. *Beginning Google Maps API 3*. New York: APRESS, 2010. 310 s. ISBN 978-1-4302-2802-8.

[30] *Google Maps JavaScript API V3* [online]. 2012 [cit. 2011-12-4]. Dostupné z <https://developers.google.com/maps/documentation/javascript/>.

[31] *API Mapy.cz* [online]. 2011 [cit. 2012-4-23]. Dostupné z <http://api4.mapy.cz/>.

[32] *Ovi Maps API* [online]. 2011 [cit. 2012-4-25]. Dostupné z <http://api.maps.ovi.com/>.

[33] *Bing Maps Developer Resources* [online]. 2012 [cit. 2012-4-25]. Dostupné z <http://www.microsoft.com/maps/developers/web.aspx>.

[34] *OpenStreetMap Wiki* [online]. 2012 [cit. 2012-4-25]. Dostupné z <http://wiki.openstreetmap.org/wiki/Main_Page>.

[35] *Dokumentace | Nette Framework* [online]. 2012 [cit. 2012-4-28]. Dostupné z <http://doc.nette.org/cs/>.

[36] KRUG, S. *Nenuťte uživatele přemýšlet! Praktický průvodce testováním a opravou chyb použitelnosti webu*. Brno: Computer Press, 2010. 168 s. EAN 9788025129234.

[37] *Spaceport PerfMarks Report II* [online]. 2012 [cit. 2012-5-15]. Dostupné z <http://spaceport.io/spaceport_perfmarks_2_report_2012_5.pdf>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ADT Android Development Tools – vývojářské nástroje pro Android

AJAX Asynchronous JavaScript and XML

API Application Programing Interface – aplikační programové rozhraní

BTS Base Transceiver Station

CSRF Cross-site request forgery

CSS Cascading Style Sheets

DOM Document Object Model – objektový model dokumentu

FDMA Frequency Division Multiple Access

GPS Global Positioning System – globální polohový systém

HTML Hypertext Markup Language

IDE Integrated development environment – integrované vývojové prostředí

JSON JavaScript Object Notation

kbps Kilobite per second – kilobit za sekundu

MAC Media Access Control

MVC Model-View-Controller

MVP Model-View-Presenter

NDK Native Development Kit – nativní vývojářský balík

OHA Open Handset Alliance

PHP Hypertext preprocessor

RIA Rich Internet Application

SDK Software Development Kit – softwarový vývojářský balík

SMS Short Message Service

SSID Service Set Identifier

SVG Scalable Vector Graphics – škálovatelná vektorová grafika

VM Virtual Machine – virtuální stroj

WHATWG The Web Hypertext Application Technology Working Group

Wi-Fi Wireless Fidelity

WWW World Wide Web

WYSIWYG What You See Is What You Get

W3C World Wide Web Consortium

XHTML Extensible Hypertext Markup Language

XML eXtensible Markup Language

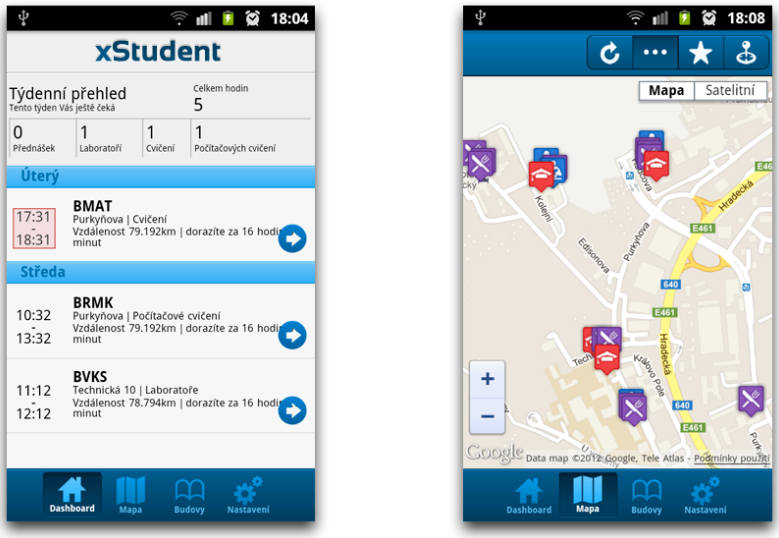
XSS Cross-site scripting

SEZNAM PŘÍLOH

A POPIS A SCREENSHOTY APLIKACE

A.1 Mobilní aplikace

Na úvodní kartě aplikace (obr. A.1 vlevo), je zobrazen aktuální týden rozvrhu uživatele. V horní části je přehled, ve kterém je uvedeno kolik hodin a jaké typy vyučování uživatele ještě čeká. Pod tímto přehledem jsou zobrazeny jednotlivé předměty, u nichž je také uvedena vzdálenost od budovy, ve které má konkrétní předmět. Po kliknutí na předmět si může uživatel zobrazit budovu na mapě a trasu k ní. Aplikace také upozorní, že je již čas se vydat na předmět, viz červeně podbarvený čas u předmětu **BMAT** na obr. A.1 vlevo.

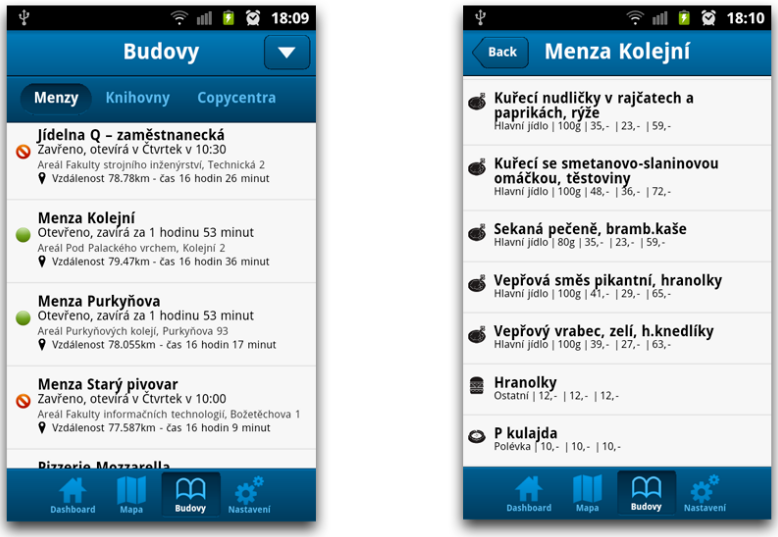


Obr. A.1: Vlevo úvodní karta aplikace - zobrazení aktuálního týdne rozvrhu. Vpravo karta s mapou a zobrazenými budovami.

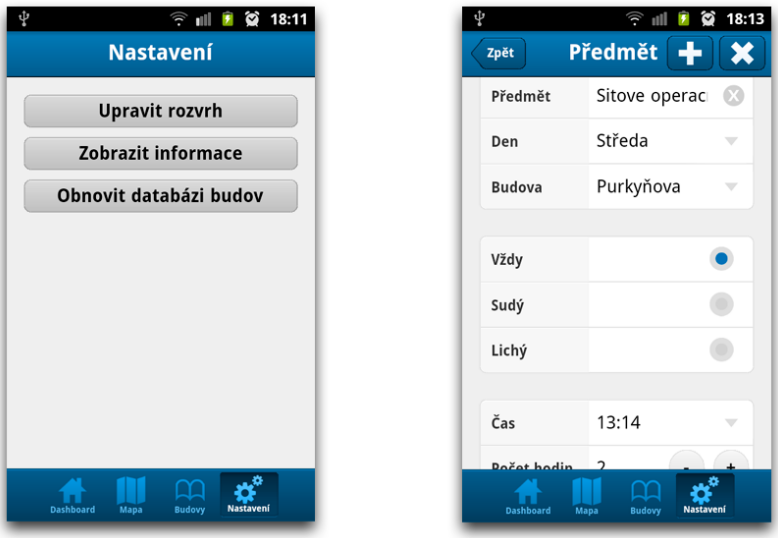
Na kartě s mapou (obr. A.1 vpravo), jsou zobrazeny jednotlivé budovy. Pomocí ovládacích prvků v horním panelu si může uživatel vyfiltrovat požadované budovy, nebo nalézt např. otevřené menzy v určitém dosahu. Dále si může zobrazit trasu a příkazy pro navigaci.

Na třetí kartě (obr. A.2 vlevo) je zobrazen přehled budov s dalšími užitečnými informacemi, jako je adresa, vzdálenost nebo to jestli je aktuálně otevřeno popř. za jak dlouho otevírá. Po kliknutí na budovu si může uživatel zobrazit informace nebo nechat zobrazit budovu na mapě. U menz je možnost zobrazení aktuálního jídelního lístku (obr. A.2 vpravo).

Na poslední kartě (obr. A.3 vlevo) se nachází jednoduché nastavení. Uživatelé mohou obnovit databázi budov ze serveru, nebo si mohou spravovat svůj rozvrh. Správa předmětů je řešena přes formulář (obr. A.3 vpravo).



Obr. A.2: Vlevo karta s přehledem budov. Vpravo karta se zobrazeným jídelním lístkem.

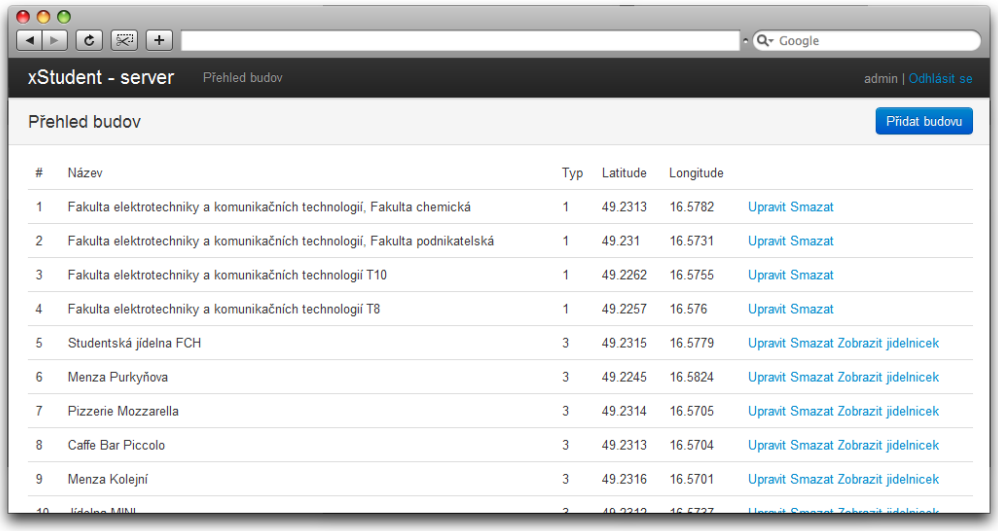


Obr. A.3: Vlevo karta s nastavením. Vpravo karta s formulářem pro vložení nového předmětu do rozvrhu.

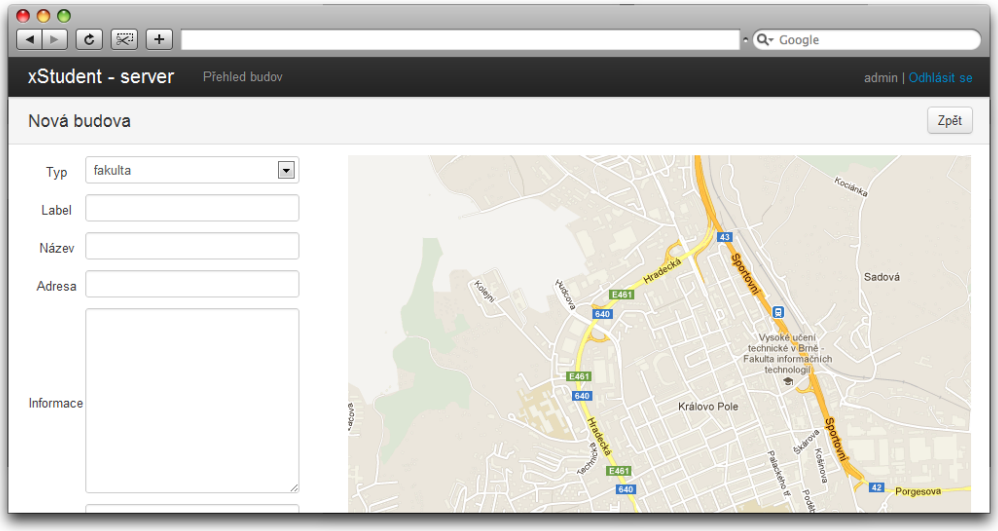
A.2 Serverová část

Rozhraní pro správu informací:

- adresa: http://marekpinkava.co.cc/xstudent-server;
- uživatelské jméno: admin;
- heslo: admin_xstudent.



Obr. A.4: Uživatelské rozhraní na serveru - přehled budov.



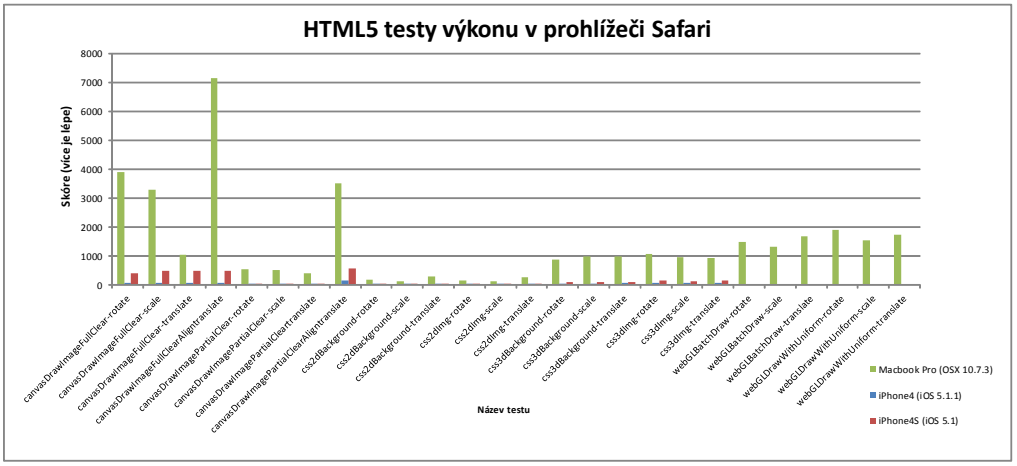
Obr. A.5: Uživatelské rozhraní na serveru - přidání nové budovy.

B OBSAH PŘILOŽENÉHO CD

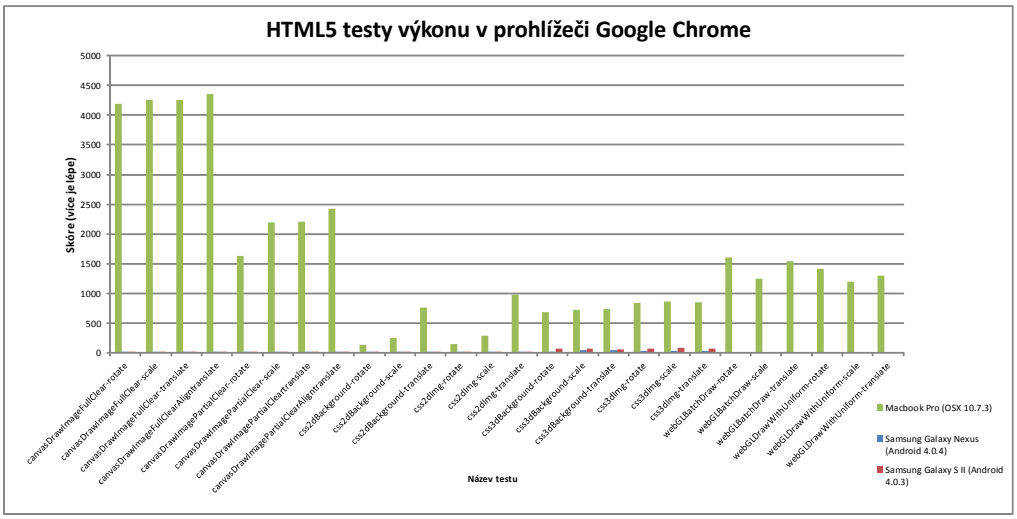
Na přiloženém CD se nachází:

- elektronická verze bakalářské práce ve formátu `pdf`;
- zdrojové kódy webové aplikace;
- zdrojové kódy android aplikace;
- zdrojové kódy serverové části aplikace.

C POROVNÁNÍ VÝKONU MOBILNÍCH A DESKTOPOVÝCH PROHLÍŽEČŮ



Obr. C.1: Výsledky HTML5 testů v prohlížeči Safari na různých zařízeních. (převzato z [37]).



Obr. C.2: Výsledky HTML5 testů v prohlížeči Google Chrome na různých zařízeních. (převzato z [37]).