

Studentský informační kanál pro OS Android

Student Information Channel for the OS Android Application

Bc. Jan Dubina

Diplomová práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan DUBINA**
Osobní číslo: **A10409**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Studentský informační kanál pro OS Android**

Zásady pro vypracování:

1. Vytvořte rešerši na téma specifík tvorby mobilních aplikací pro OS Android.
2. Navrhněte a vytvořte klientský informační systém schopný koncentrovat aktuální informace o dění na UTB a poskytovat je mobilním zařízením.
3. Vytvořte webovou službu poskytující dané informace a klient služby pro OS Android s možností offline provozu.
4. Systém publikujte pod licencí GPL v3.
5. Vytvořte programovou a uživatelskou dokumentaci.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. MEIER, Reto. Professional Android 2 Application Development. Indianapolis: Wiley Publishing, 2010. ISBN 978-0-470-56552-0
2. STEELE, James a Nelson TO. The Android Developers Cookbook: Building Applications with the Android SDK. Upper Saddle River: Pearson Education, 2010. ISBN 978-0-321-74123-3
3. ROGERS, Rick, John LOMBARDO, Zigurd MEDNIEKS a Blake MEIKE. Android Application Development. Sebastopol: OReilly Media, 2009. ISBN 05-965-2147-2
4. COLLINS, Charlie, Michael GALPIN a Matthias KÄPPLER. Android in Practice. Shelter Island: Manning Publications, 2011. ISBN 9781935182924
5. BEGUN, Daniel. Amazing Android Apps For Dummies. Indianapolis: Wiley Publishing, 2011. ISBN 978-0-470-93629-0
6. KANISOVÁ, H. – MÜLLER, M. UML srozumitelně. 2. akt. vyd. Brno: Computer Press, 2006. 176 s. ISBN 80-251-1083-4

Vedoucí diplomové práce:

Ing. Michal Bližňák, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

24. února 2012

Termín odevzdání diplomové práce:

21. května 2012

Ve Zlíně dne 24. února 2012



prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem této práce je vytvoření informačního systému pro Univerzitu Tomáše Bati ve Zlíně. Systém by měl sbírat informace o aktuálním dění a dále je poskytovat mobilním zařízením. Navržené řešení se skládá z webové aplikace vytvořené pomocí Yii frameworku a klientské aplikace pro operační systém Android. Informace jsou primárně získávány z RSS kanálů a jejich distribuce probíhá pomocí komunikačního protokolu SOAP.

Klíčová slova: informační systém, webová aplikace, Android aplikace, Yii Framework, protokol SOAP

ABSTRACT

The purpose of this thesis is to create an information system for the Thomas Bata University in Zlín. System should collect information about current events and provide them to mobile devices. Proposed solution consists of web application made using Yii framework and client application for the Android operating system. Information is primarily obtained from RSS feeds and distribution is accomplished using SOAP communication protocol.

Keywords: information system, web application, Android application, Yii Framework, SOAP protocol

Tímto bych rád poděkoval panu Ing. Michalu Bližňákovi, Ph.D. za cenné rady, připomínky a veškerý čas, který mi věnoval při odborném vedení této práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 YII FRAMEWORK	11
1.1 MVC ARCHITEKTURA.....	12
1.1.1 Model v Yii	12
1.1.2 Ovladač v Yii	13
1.2 POSTUP VYTVOŘENÍ APLIKACE POMOCÍ Yii	13
1.2.1 Struktura Yii frameworku a ověření požadavků na systém.....	13
1.2.2 Vytvoření webové aplikace	14
1.2.3 Adresářová struktura vytvořené aplikace	15
1.2.4 Generování MVC architektury pomocí Gii.....	16
2 OS ANDROID	18
2.1 VÝVOJ APLIKACÍ PRO OS ANDROID.....	20
2.1.1 Požadavky pro vývoj	20
2.1.2 Komponenty aplikace.....	20
2.1.2.1 Aktivity (Activities)	20
2.1.2.2 Služby (Services)	22
2.1.2.3 Poskytovatelé obsahu (Content Receiver)	23
2.1.2.4 Přijímače systémových hlášení (Broadcast receivers)	23
2.1.3 Android manifest.....	23
2.1.4 Aplikační zdroje	24
3 DALŠÍ POUŽITÉ TECHNOLOGIE	25
3.1 SOAP.....	25
3.2 RSS KANÁLY.....	25
II PRAKTICKÁ ČÁST	27
4 TECHNICKÁ DOKUMENTACE	28
4.1 WEBOVÁ APLIKACE	28
4.1.1 Požadavky	28
4.1.2 Dostupné metody webové služby.....	29
4.1.3 Komunikace s webovou službou.....	30
4.1.4 Use-case diagram webové aplikace.....	31
4.1.5 Class-diagram webové služby	32
4.1.6 Databáze	32
4.2 ANDROID APLIKACE	33
4.2.1 Požadavky	34
4.2.2 Use-case diagram	34
4.2.3 Class diagram	35
4.2.4 Databáze	37
4.2.5 Způsob komunikace s webovou službou.....	38
5 UŽIVATELSKÁ DOKUMENTACE	39

5.1	WEBOVÁ APLIKACE	39
5.2	ANDROID APLIKACE	42
ZÁVĚR	45
ZÁVĚR V ANGLIČTINĚ	46
SEZNAM POUŽITÉ LITERATURY	47
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	49
SEZNAM OBRÁZKŮ	51
SEZNAM TABULEK	52
SEZNAM PŘÍLOH	53

ÚVOD

Informační systémy jsou každodenní součástí našeho života. Slouží pro sběr, zpracování a uchování dat a jejich následnou prezentaci uživateli. V dnešní době se vyskytují v téměř všech odvětvích lidské činnosti a stále častěji využívají výhody mobilních zařízení a internetu. Jejich uživatelé tak mohou prohlížet data poskytovaná těmito systémy kdekoliv na světě.

Práce se skládá z teoretické a praktické části. První kapitola teoretické části se zabývá PHP frameworkem Yii, je zde vysvětlen princip MVC architektury a stručně popsány základy vytváření webových aplikací v tomto frameworku. Další kapitola se věnuje operačnímu systému Android a vývoji aplikací pro tento systém. V poslední kapitole této části je vysvětlen princip komunikace pomocí protokolu SOAP a popsány specifika RSS kanálů. Praktická část je tvořena technickou a uživatelskou dokumentací pro vytvořenou webovou aplikaci a Android aplikaci.

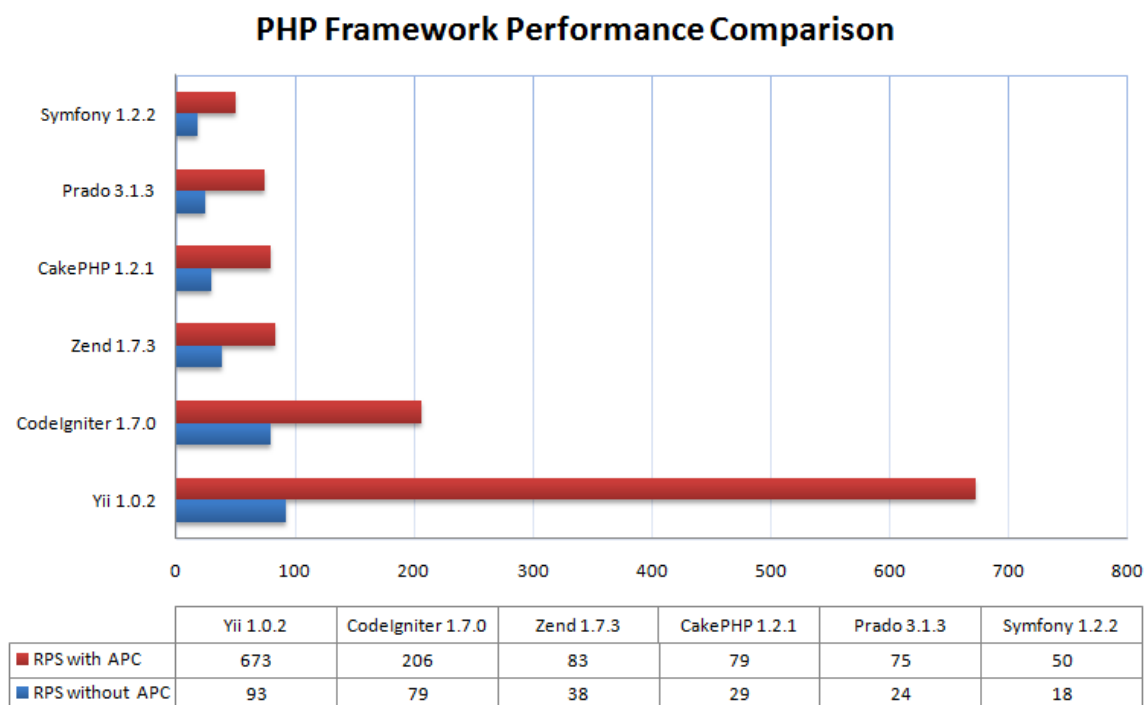
Od čtenáře této práce se očekává základní znalost webových technologií a programovacích jazyků PHP a Java.

I. TEORETICKÁ ČÁST

1 Yii FRAMEWORK

Yii je PHP framework určený pro vytváření webových aplikací, který propaguje rychlý vývoj a čistý, DRY (Don't Repeat Yourself) přístup k návrhu. Jedná se o open-source projekt, který je možné zdarma používat při dodržení podmínek BSD licence uvedených na webových stránkách frameworku (www.yiiframework.com). Yii je zkratkou prvních písmen slov „Yes It Is!“, což je podle tvůrců tohoto frameworku nejstručnější odpověď na otázku, zda je Yii dostatečně rychlý, bezpečný, profesionální atd. [7]

Yii začal vytvářet Qiang Xue, který se předtím podílel na vývoji a údržbě frameworku Prado. Projekt začal 1. ledna 2008 s cílem vytvořit framework, který by odpovídal požadavkům na vývoj webových aplikací pro Web 2.0. První verze byla vydána po necelém roce v prosinci 2008. Tým, který se stará o vývoj a údržbu frameworku, má aktuálně 7 členů.[7]



Obr. č. 1 – Srovnání výkonu PHP frameworků (Zdroj: [7])

Vysoký výkon Yii je způsoben použitím techniky tzv. líného načítání. Principem je vytváření zdrojů až když je k nim poprvé přistupováno, např. začlenění souboru třídy až při jejím prvním použití nebo vytvoření instance třídy až při prvním pokusu o přístup k objektu. Rychlost Yii lze dále zvýšit použitím Alternative PHP Cache – open-source framework používaný pro optimalizaci PHP kódu.

Podmínky, při kterých byl prováděn test uvedený výše, lze nalézt na www.yiiframework.com/performance/. Hodnoty v grafu jsou uvedeny v požadavcích za sekundu (RPS).

1.1 MVC architektura

MVC (Model-View-Controller) je typ architektury používané při vývoji aplikací. Hlavním principem je striktní oddělení datové části, části zpracovávající uživatelský vstup a části zajišťující vizuální prezentaci. MVC architektura se tedy skládá ze tří částí:

- **Model** – model slouží jako rozhraní pro komunikaci s databází, definuje vztahy mezi daty v databázi, definuje pravidla pro ukládané hodnoty, zajišťuje čtení a zápis hodnot do databáze.
- **View** – pohled zajišťuje pouze prezentaci dat uživateli, obdržená data zobrazí pomocí technologií HTML, CSS, ...
- **Controller** – ovladač zpracovává uživatelské vstupy a data získaná z databáze, probíhá zde veškeré rozhodování, výpočty, ...



Obr. č. 2 – Schéma architektury MVC (Zdroj: [7])

1.1.1 Model v Yii

Modely v Yii jsou velmi jednoduché, lze je automaticky vygenerovat včetně všech CRUD (Create, Read, Update and Delete) operací například pomocí nástroje Gii. Vygenerovaný kód zajišťuje objektově orientovaný přístup k databázi, k ukládání a čtení dat se tak používají funkce modelu místo SQL dotazů. Modely také umožňují vytvářet

vazby mezi tabulkami, které fungují stejně jako klíčové slovo Join v jazyce SQL. Ukázka čtení dat z databáze pomocí Yii modelu:

```
$result = ModelName::model()->findAll();
```

1.1.2 Ovladač v Yii

Ovladač se spouští, kdykoliv je do prohlížeče zadána adresa webové aplikace. Ovladačů může být více, každý z nich poté zajišťuje navigaci po určité části webové aplikace. Typickým znakem Yii a obecně aplikací využívajících MVC architekturu je specifický formát adres:

```
www.mydomain.com/index.php?r=myController/myAction
```

Adresa vždy obsahuje index.php a o navigaci se stará ovladač, jehož název je uvedený za parametrem 'r'. Každý ovladač má dále jednu nebo více akcí, které se v adrese uvádí za jeho názvem. Pokud zde není akce uvedena, volá se implicitní akce ovladače. Například adresa při mazání komentáře pak může vypadat takto:

```
www.mydomain.com/index.php?r=comments/delete
```

1.2 Postup vytváření aplikace pomocí Yii

1.2.1 Struktura Yii frameworku a ověření požadavků na systém

Po stažení a rozbalení má framework tuto adresářovou strukturu:

- *demos* – obsahuje ukázky aplikací – „Hello world!“, blog, jednoduchou hru a adresář
- *framework* – obsahuje vše potřebné pro běh webové aplikace vytvořené v Yii
- *requirements* – ověření požadavků Yii

K ověření požadavků Yii poté stačí umístit adresář s frameworkem do kořenového adresáře webového serveru a do prohlížeče zadat adresu:

```
www.mydomain.com/yii/requirements
```

Prvky vyžadované Yii jsou potřebné pro správný běh webových aplikací. Přítomnost dalších rozšíření je volitelná, jsou vyžadovány pouze pro použití určitých tříd případně pro komunikaci s určitými databázovými systémy.

Name	Result	Required By	Memo
PHP version	Passed	Yii Framework	PHP 5.1.0 or higher is required.
\$_SERVER variable	Passed	Yii Framework	
Reflection extension	Passed	Yii Framework	
PCRE extension	Passed	Yii Framework	
SPL extension	Passed	Yii Framework	
DOM extension	Passed	CHtmlPurifier , CWsdlGenerator	
PDO extension	Passed	All DB-related classes	
PDO SQLite extension	Warning	All DB-related classes	This is required if you are using SQLite database.
PDO MySQL extension	Passed	All DB-related classes	This is required if you are using MySQL database.
PDO PostgreSQL extension	Warning	All DB-related classes	This is required if you are using PostgreSQL database.
Memcache extension	Warning	CMemCache	
APC extension	Warning	CApcCache	
Mcrypt extension	Passed	CSecurityManager	This is required by encrypt and decrypt methods.
SOAP extension	Passed	CWebService , CWebServiceAction	
GD extension with FreeType support	Warning	CCaptchaAction	GD not installed
Ctype extension	Passed	CDateFormatter , CDateTimeParser , CTextHighlighter , CHtmlPurifier	

■ passed
 ■ failed
 ■ warning

Apache/2.2.21 (Win32) PHP/5.2.17 [Yii Framework](#)/1.1.10 2012-05-13 15:25

Obr. č. 3 – Ověření požadavků Yii frameworku

1.2.2 Vytvoření webové aplikace

Vytvoření základní webové aplikace probíhá z prostředí příkazové řádky daného systému spuštěním souboru yiic, který se nachází v adresáři framework. Jedná se o spustitelný soubor, který dále volá PHP skript yiic.php ve stejné složce. Správná syntaxe pro spuštění je:

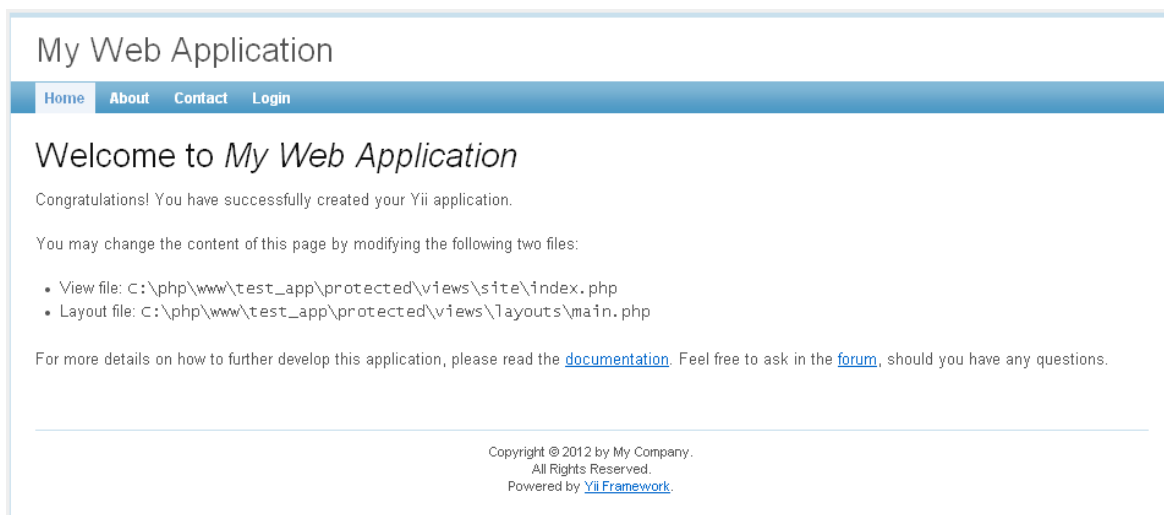
```
yiic webapp path
```

nebo

```
php yiic webapp path
```

při přímém spouštění PHP skriptu. Klíčové slovo webapp je příkazem pro vytvoření nové webové aplikace v daném adresáři. Takto vytvořená webová aplikace již obsahuje některé základní funkce:

- Kontaktní formulář s CAPTCHA
- Přihlašovací formulář
- Jednoduchou přihlašovací a odhlašovací funkcionalitu



Obr. č. 4 – Základní aplikace vygenerovaná pomocí Yii

1.2.3 Adresářová struktura vytvořené aplikace

Adresářová struktura vytvořené aplikace:

- *assets* – používáný pro integraci jQuery (JavaScript framework)
- *css* – obsahuje soubory CSS stylů
- *images* – obsahuje grafické data
- *protected* – obsahuje veškeré PHP skripty
- *themes* – obsahuje šablony vzhledu

Nejdůležitějším adresářem je *protected*, kde jsou uloženy konfigurační soubory, modely, ovladače atd. Některé důležité adresáře:

- *components* – obsahuje třídy používané aplikací
- *config* – obsahuje konfigurační soubory
- *controllers* – obsahuje ovladače
- *extensions* – zde se nacházejí knihovny třetích stran používané aplikací

- *models* – obsahuje modely
- *runtime* – slouží pro ukládání dynamicky generovaných souborů
- *views* – obsahuje soubory popisující grafické rozvržení stránek (layouts) a pohledy

1.2.4 Generování MVC architektury pomocí Gii

Generování lze provést dvěma způsoby: z prostředí příkazové řádky pomocí spustitelného souboru *yii* nebo z webového prohlížeče pomocí nástroje Gii. Před samotným generováním je nutné navrhnout strukturu databáze a vytvořit v ní dané tabulky. Také je třeba v konfiguračním souboru *main.php* nastavit všechny parametry potřebné pro připojení k databázi.

Povolení nástroje Gii se provádí v konfiguračním souboru *main.php* pomocí následujícího kódu:

```
'gii'=>array(  
    'class'=>'system.gii.GiiModule',  
    'password'=>'pass',  
    'ipFilters'=>array('127.0.0.1',':::1'))
```

Nastavuje se zde heslo pro přihlášení k webovému rozhraní a pro lepší zabezpečení také omezení IP adres, ze kterých je možné s nástrojem pracovat. Gii by nemělo být povoleno u webových aplikací v běžném provozu, po ukončení vývoje aplikace je vhodné tento nástroj zakázat.

Gii spustíme zadáním adresy

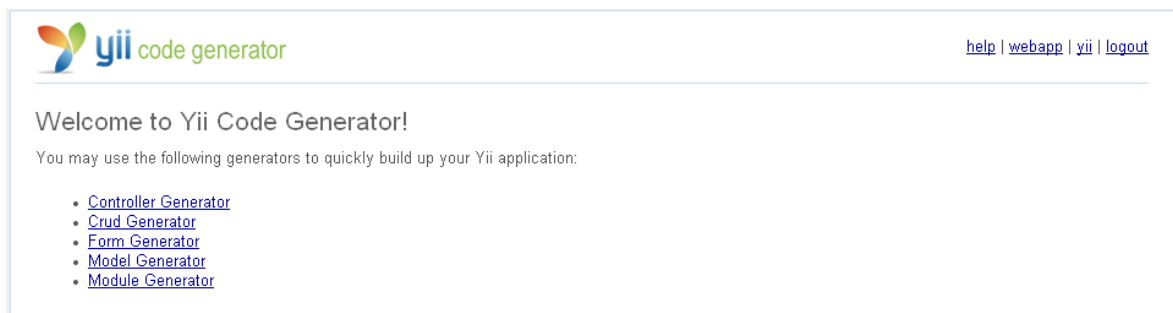
```
www.mydomain.com/index.php?r=gii
```

do webového prohlížeče. Po zadání zvoleného hesla se zpřístupní generátory zdrojových souborů. Nejprve je třeba vygenerovat modely pro tabulky. Zde stačí pouze zadat název tabulky (případně symbol ‘*’ pro všechny tabulky v databázi) a potvrdit generování souboru modelu. Vytvořený model lze poté najít v adresáři *protected/models*. Dále musíme vytvořit ovladač a pohledy pomocí CRUD generátoru. Generování kódu zde vychází z třídy modelu, vytvořený ovladač je uložený v adresáři *protected/controllers* a pohledy v *protected/views/nazevModelu*. U některých modelů není potřeba implementovat veškerou

funkcionalitu, proto je vhodné v náhledu generovaných souborů vybrat pouze ty, které opravdu potřebujeme.

Přehled vytvářených pohledů a jejich funkce:

- *_form.php* – formulář pro vytváření a editování záznamů
- *_search.php* – formulář pro vyhledávání záznamů
- *_view.php* – šablona pro zobrazení záznamu
- *admin.php* – zobrazí seznam záznamů s odkazy na CRUD funkce
- *create.php* – zobrazí formulář pro přidání záznamu
- *index.php* – zobrazí seznam záznamů
- *update.php* – zobrazí formulář pro editaci záznamu
- *view.php* – zobrazí detail záznamu



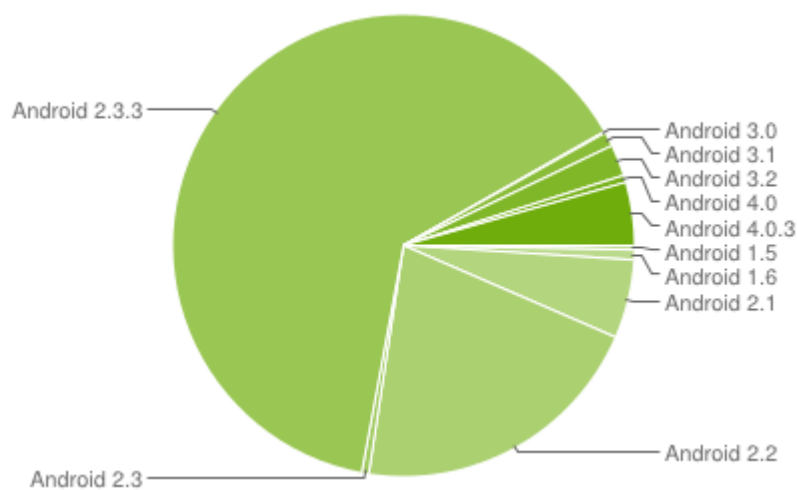
Obr. č. 5 – Nástroj Gii

Informace uvedené v této části práce byly čerpány ze zdrojů [7] a [8].

2 OS ANDROID

Android je operační systém postavený na OS Linux, primárně vytvořený pro použití na mobilních zařízeních jako jsou smartphony a tablety. Android je vyvíjený konsorciem společností Open Handset Alliance, které sdružuje telekomunikační společnosti, společnosti zabývající se výrobou polovodičových součástek nebo vývojem hardware a software a je vedené společností Google. Členy jsou například firmy HTC, Intel, LG, Motorola, Nvidia, Samsung Electronics atd. Cílem tohoto sdružení je vytvoření otevřených standardů v oblasti mobilních zařízení. [10]

OS Android má aktuálně více než padesát procentní podíl na trhu mobilních platforem. Počet zařízení využívajících tento operační systém překročil 300 milionů a každý den narůstá přibližně o 850 tisíc zařízení. Internetový obchod Google Play (původně Android Market) nabízí přes 450 tisíc aplikací pro OS Android a proběhlo z něj 10 miliard stažení.[12]



Obr. č. 6 – Rozložení verzí OS Android (Zdroj: [9])

Společnost Android založili Andy Rubin, Rich Miner, Nick Sears a Chris White v říjnu roku 2003 v USA. V srpnu roku 2005 byla zakoupena společností Google jako součást plánu na vstup na trh s mobilními zařízeními.

Android je postavený na jádru Linux verze 2.6. Knihovny a API jsou napsané v programovacích jazycích C a C++. Aplikace využívají k běhu aplikační framework používající Java knihovny vycházející z projektu Apache Harmony (open-source Java implementace). Využívá se zde sandbox přístupu – snaha o izolaci běžících aplikací. Toho

je dosaženo virtualizací v softwaru Dalvik, který je součástí jádra systému. Aplikace jsou psány v programovacím jazyku Java, před instalací na zařízení jsou převáděny do formátu kompatibilního s Dalvik (Dalvik Executable - .dex). První verze OS Android a Android SDK byla vydána v říjnu roku 2007. Verze jsou pojmenovány podle sladkostí v abecedním pořadí.

Tab. č. 1 – Verze OS Android (Zdroj: [9])

Verze OS	Název	Úroveň API	Podíl na trhu
Android 1.5	Cupcake	3	0.3 %
Android 1.6	Donut	4	0.7 %
Android 2.1	Eclair	7	5.5 %
Android 2.2	Froyo	8	20.9 %
Android 2.3 – Android 2.3.2	Gingerbread	9	0.5 %
Android 2.3.3 – Android 2.3.7		10	63.9 %
Android 3.0	Honeycomb	11	0.1 %
Android 3.1		12	1.0 %
Android 3.2		13	2.2 %
Android 4.0 – Android 4.0.2	Ice Cream Sandwich	14	0.5 %
Android 4.0.3 – Android 4.0.4		15	4.4 %

OS Android je open-source projektem. Většina zdrojových kódů je publikována pod Apache licenci verze 2.0 (kromě síťové části systému), k jejich vydání dochází krátce po vydání nové verze OS Android. Úpravy provedené na jádru Linux jsou chráněny GNU General Public License verze 2 a jejich zdrojový kód je stále dostupný i během vývoje.

Ačkoliv byl OS Android původně určen pro použití na mobilních zařízeních, dnes se hlavně díky snadné přizpůsobitelnosti rozšířil i do jiných oblastí elektroniky. Používají ho například laptopy, netbooky, televize, čtečky elektronických knih atd.

Pozn.: Hodnoty uvedené v tabulce a grafu vychází z počtu zařízení, které se během 14 denního období končícího 1. května 2012 připojily ke službě Google Play.

Informace uvedené v této části práce byly čerpány ze zdrojů [9], [10], [11], a [12].

2.1 Vývoj aplikací pro OS Android

2.1.1 Požadavky pro vývoj

Požadavky, které musí být splněny před začátkem vývoje Android aplikací:

- Použití některého z podporovaných operačních systémů
- Java Development Kit 6 (samotné JRE je nedostačující)
- Android SDK
- Vývojové prostředí – jediné doporučené prostředí je Eclipse (verze 3.6.2 (Helios) nebo vyšší)
- Doporučuje se instalace ADT rozšíření pro Eclipse

2.1.2 Komponenty aplikace

Komponenty jsou základními stavebními kameny Android aplikace, každá slouží k jinému účelu a má svůj specifický životní cyklus. Existují 4 typy komponent: aktivita, služba, poskytovatel obsahu a přijímače systémových hlášení.

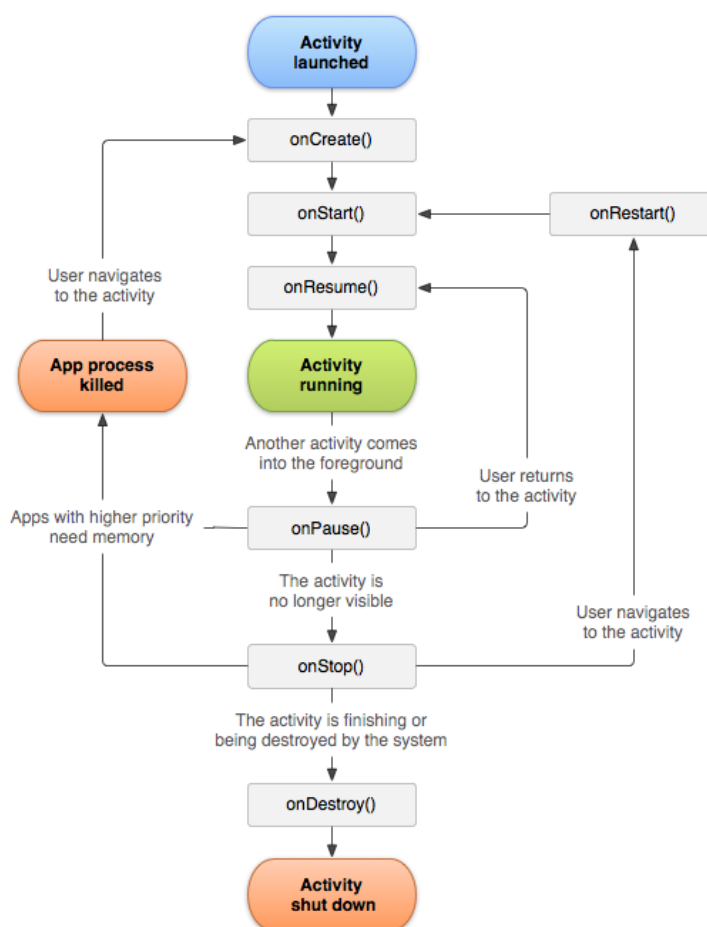
2.1.2.1 *Aktivita (Activities)*

Aktivita reprezentuje jednu obrazovku uživatelského rozhraní, se kterou lze pracovat. Okno aktivity nemusí vyplňovat celou obrazovku, může být menší nebo

transparentní. Většina aplikací obsahuje více aktivit, které jsou na sobě nezávislé, jedna z nich je obvykle označena jako hlavní a slouží tak jako vstupní bod aplikace. Každá aktivita může dále spouštět další aktivity. Přepínání mezi aktivitami způsobuje změnu jejich stavů. Existují 3 stavy, ve kterých se aktivity mohou nacházet:

- **Resumed (Running)** – běžící aktivita, která je v popředí obrazovky.
- **Paused** – část obrazovky aktivity je stále viditelná, i když je jiná aktivita v popředí. Aktivita v tomto stavu je uložena v paměti, je zachován její stav a všechny informace, stále zůstává připojena ke správci oken, ale při extrémním nedostatku paměti může být ukončena.
- **Stopped** – aktivita je v pozadí, není viditelná. Také zůstává v paměti, ale není připojena ke správci oken. Může být kdykoliv ukončena, pokud systém potřebuje uvolnit paměť.

Při změnách stavu aktivity jsou volány její metody zpětné vazby, díky kterým je možné korektně inicializovat nebo ukončit její činnost.

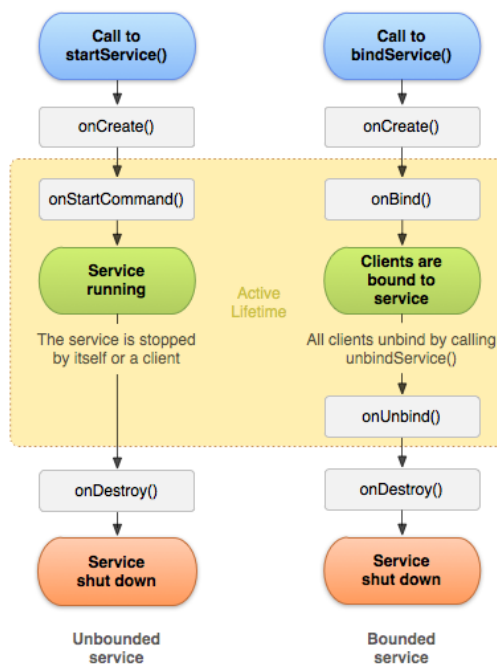


Obr. č. 7 – Životní cyklus aktivity (Zdroj: [9])

2.1.2.2 Služby (Services)

Služby jsou komponenty bez uživatelského rozhraní, které slouží k provádění dlouhotrvajících operací na pozadí. Služby zůstávají běžet i při přepínání mezi aplikacemi. Lze je spouštět z jiných komponent a s komponentou je lze také spojit tak, že je umožněna komunikace mezi komponentou a službou. Existují 2 typy služeb:

- **Started** – po spuštění běží služba v pozadí, i když je komponenta, která ji spustila, ukončena. Obvykle pouze vykoná určitou operaci a sama se ukončí (např. stažení souboru).
- **Bound** – služba je spojená s jednou nebo více komponentami. Toto spojení je typu klient-server, umožňuje komponentě komunikovat se službou pomocí zasílání požadavků a přijímání odpovědí. Tento typ služby zaniká spolu s poslední komponentou, která je se službou svázaná.



Obr. č. 8 – Životní cyklus služby (Zdroj: [9])

2.1.2.3 Poskytovatelé obsahu (Content Receiver)

Poskytovatelé obsahu řídí přístup ke strukturovaným datům. Zajišťují zapouzdření dat a poskytují mechanismy pro přístup k datům a jejich zabezpečení. Primárně slouží jako rozhraní pro přístup k datům pro externí aplikace.

2.1.2.4 Přijímače systémových hlášení (Broadcast receivers)

Slouží k přijímání systémových hlášení. Jejich zdrojem je buď samotný operační systém (vybitá baterie telefonu) nebo aplikace (stažení dat, které by mohly využít ostatní aplikace). Tato komponenta by měla sloužit pouze pro přijetí hlášení, jeho zpracování by mělo probíhat jinde.

2.1.3 Android manifest

Jedná se o XML soubor umístěný v kořenovém adresáři aplikace, který poskytuje operačnímu systému informace o aplikaci. Jsou zde definovány například:

- Informace o aplikaci (popisek, ikona atd.)
- Existující komponenty a jejich vlastnosti
- Uživatelské povolení, potřebné pro běh aplikace (např. přístup k internetu, přístup k uživatelským kontaktům atd.)
- Minimální úroveň API potřebná ke spuštění aplikace
- Požadavky na hardware a software (např. bluetooth, fotoaparát atd.)
- Vyžadované knihovny API (např. Google Maps Library)

2.1.4 Aplikační zdroje

Android aplikace nejsou tvořeny pouze zdrojovými kódy, ale také dalšími soubory, které se nazývají zdroje. Existují 2 typy zdrojů:

- **XML soubory** – používají se k definici stylů a rozložení prvků uživatelského rozhraní nebo pro aplikační konstanty (text tlačítek, barvy atd.)
- **Ostatní soubory** – zvukové soubory, obrázky, ...

Jednou z výhod použití XML souborů je to, že mohou být upravovány bez nutnosti zasahovat do zdrojového kódu aplikace. Další velkou výhodou je možnost mít více XML souborů definujících stejné prvky. Díky této vlastnosti je možné mít různé definice rozložení prvků uživatelského rozhraní podle orientace displaye nebo jednoduše měnit popisky v aplikaci pro různé jazyky.

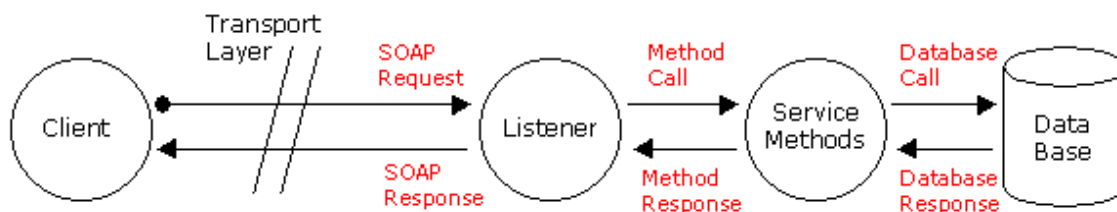
Informace uvedené v této části práce byly čerpány ze zdroje [9].

3 DALŠÍ POUŽITÉ TECHNOLOGIE

3.1 SOAP

SOAP (Simple Object Access Protocol) je jednoduchý protokol pro výměnu informací založený na XML. Je možné ho používat pro obyčejné zasílání zpráv, ale obvykle se používá pro RPC (Remote Procedure Call) komunikaci založenou na dotazech a odpovědích. Tento protokol není závislý na specifickém transportním protokolu, ani na operačním systému nebo programovacím jazyku. Klienti a servery tak mohou běžet na libovolné platformě, důležitá je pouze schopnost SOAP zprávy formulovat a porozumět jim. SOAP protokol se skládá ze 3 částí:

- SOAP obálka, která definuje formát zprávy
- Pravidla pro formátování dat vkládaných do zprávy
- Způsob reprezentace volání vzdálených procedur a jejich odpovědí



Obr. č. 9 – Schéma RCP komunikace (Zdroj: [14])

SOAP obálka se dále skládá z hlavičky a těla, povinné je pouze tělo. Obálka musí být vždy prvním elementem v XML souboru reprezentujícím SOAP zprávu. Hlavička slouží pro definici vlastností zprávy, což umožňuje komunikaci bez předchozí domluvy. Tělo zprávy obsahuje posílaná data a tzv. Fault element, který slouží pro hlášení chyb.

Informace uvedené v této části práce byly čerpány ze zdrojů [13] a [14].

3.2 RSS kanály

RSS (Really Simple Syndication) ke sdružování obsahu webových stránek. Bez použití RSS by museli uživatelé denně kontrolovat webovou stránku kvůli novým informacím, nebo se přihlásit k odběru novinek e-mailem. Díky použití RSS mohou

odebírat novinky ze všech sledovaných stránek pomocí RSS agregátoru (webová stránka nebo software, který sbírá a třídí novinky z RSS kanálů). RSS je vytvářené jako XML dokument a jeho hlavní výhodou je jednoduchost a rychlé načítání způsobené minimálním objemem přenášených dat.

XML dokument se skládá z informací o RSS kanálu a jednotlivých novinek. Informace, které jsou povinné, jsou uvedeny ve specifikaci formátu RSS, pro údaje o kanálu RSS jsou to tyto prvky:

- Název kanálu
- Odkaz na webovou stránku korespondující s kanálem
- Krátký popis kanálu

Nepovinnými prvky jsou například jazyk, poslední aktualizace, e-mail editora obsahu a webmastera atd. U novinek jsou vyžadovány tyto údaje:

- Název novinky
- Odkaz na novinku
- Text novinky

Mezi nepovinné prvky patří datum vydání, autor, kategorie, odkaz na komentáře atd.

Informace uvedené v této části práce byly čerpány ze zdrojů [15] a [16].

II. PRAKTICKÁ ČÁST

4 TECHNICKÁ DOKUMENTACE

Informační systém se skládá z webové aplikace, kterou tvoří webová služba a jednoduché rozhraní pro její správu, a Android aplikace. Webová služba sbírá novinky z RSS kanálů a na požádání je dále distribuuje klientským zařízením. Tímto klientem může být například libovolné mobilní zařízení s vytvořenou Android aplikací, která pomocí webové služby získává aktuální informace a poté je prezentuje uživateli.

4.1 Webová aplikace

Webová část aplikace byla vyvíjena ve freeware editoru PSPad s použitím následujících technologií:

- PHP verze 5.2.17
- Yii framework verze 1.1.10.r3566
- MySQL verze 5.1

Dále byly použity tyto externí knihovny:

- LastRSS verze 0.9.1
- Portable PHP password hashing framework verze 0.3

4.1.1 Požadavky

Nejdříve je třeba stanovit požadavky, které musí vytvářená webová aplikace splňovat:

- Webová služba koncentrující aktuální informace o dění na UTB ve Zlíně a poskytující je dále klientským zařízením
- RSS kanály jako hlavní zdroj informací
- Struktura aplikace navržena tak, aby bylo možné snadné rozšíření zdrojů informací o např. HTML parsery
- Vývoj aplikace s využitím libovolného PHP frameworku

- Použití protokolu SOAP pro komunikaci s webovou službou
- Jednoduché webové rozhraní pro správu webové služby umožňující například přidávání nových zdrojů informací apod.
- Autentizace uživatele
- Rozdělení rolí uživatelů kvůli možnosti budoucího rozšíření tohoto informačního systému

4.1.2 Dostupné metody webové služby

Pomocí dotazů komunikačního protokolu SOAP lze volat tyto dvě metody webové služby:

- *getSources* – tato metoda vrací názvy všech dostupných zdrojů informací jako pole prvků typu řetězec
- *getNews* – vstupním parametrem této funkce je pole zdrojů, kde je každý zdroj reprezentován asociativním polem s prvky *name* (řetězec, název zdroje) a *lastUpdate* (integer, timestamp poslední aktualizace novinek z tohoto zdroje na klientském zařízení uvedený v časovém pásmu UTC/GMT). Pro reprezentaci zdroje lze také použít objekt se stejnými datovými proměnnými jako u pole. Tyto objekty musí být v XML dokumentu ohraničeny tagy s názvem *SoapRequestObject*. Metoda vrací pole, kde jednotlivé prvky přísluší zdrojům v zaslaném požadavku. Na těchto pozicích se poté nacházejí pole novinek pro příslušný zdroj. Struktura pole novinky:
 - [0] – název novinky
 - [1] – odkaz na novinku
 - [2] – text novinky
 - [3] – datum novinky

Všechny položky jsou datového typu řetězec.

4.1.3 Komunikace s webovou službou

Webová služba se nachází na adrese

`www.mydomain.com/index.php/service/news?ws=1`

případně na

`www.mydomain.com/index.php/service/news.`

Liší se podle programovacího jazyka, ve kterém je vytvořena klientská aplikace.

WSDL soubor lze zobrazit zadáním druhé z uvedených adres do prohlížeče.

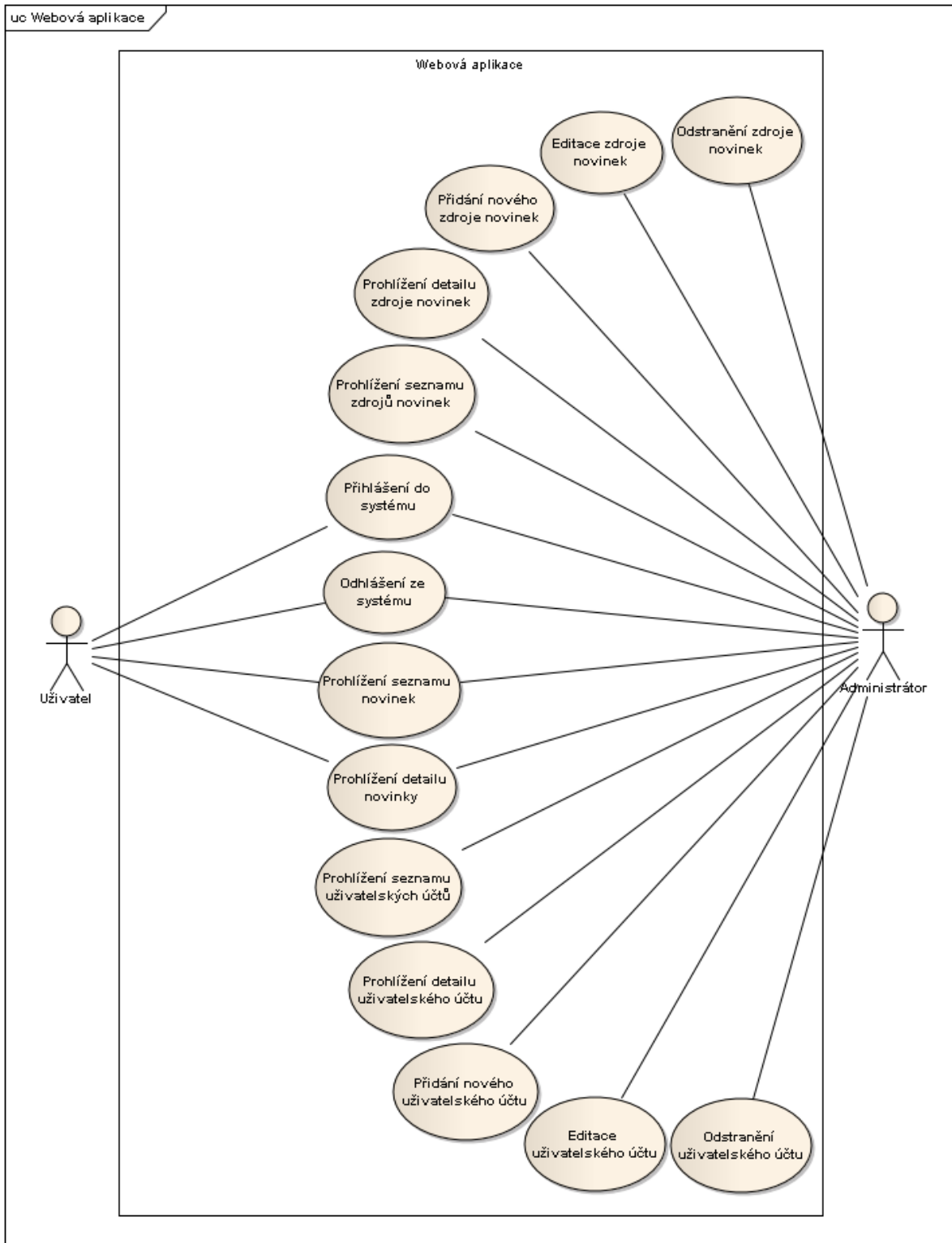
Další údaje potřebné pro komunikaci s webovou službou:

- **Namespace:** *urn:ServiceControllerwsdl*
- **Soap akce:** *urn:ServiceControllerwsdl#getNews* resp.
urn:ServiceControllerwsdl#getSources
- **Název metody:** *getSources* resp. *GetNews*

Ukázka připojení k webové službě v jazyce PHP:

```
<?php
$client = new
SoapClient('http://www.mydomain.com/index.php/service/news');
$sources = $client->getSources();
$news = $client->getNews(array(
array('name' => 'UTB', 'lastUpdate' => 0),
array('name' => 'FAI UTB', 'lastUpdate' => 0)));
?>
```

4.1.4 Use-case diagram webové aplikace

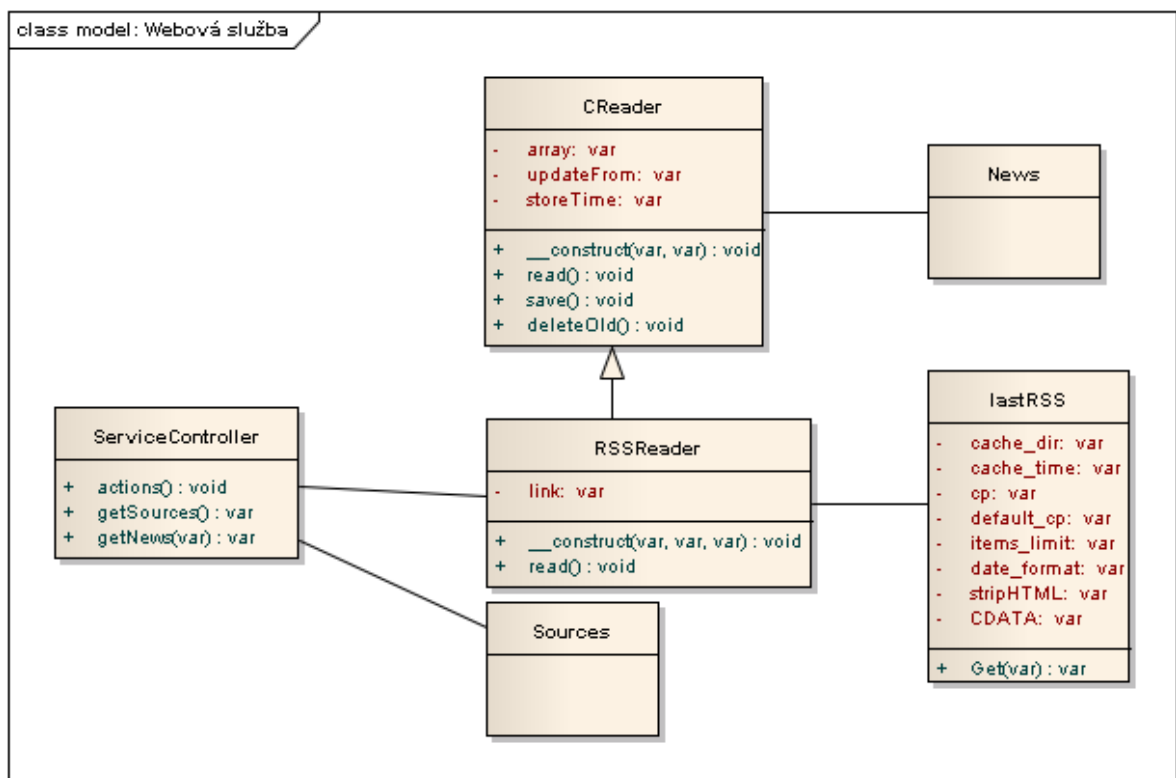


Obr. č. 10 – Use-case diagram webové aplikace

4.1.5 Class-diagram webové služby

Popis tříd v diagramu:

- *ServiceController* – třída webové služby (ovladač)
- *CReader* – obecná třída pro zpracování novinek
- *RSSReader* – dědí z třídy *CReader*, přepisuje metodu *read* tak, aby načítala novinky z RSS kanálů
- *lastRSS* – externí knihovna pro parsování RSS kanálů
- *News*, *Sources* - modely sloužící jako rozhraní pro práci s databází, tyto třídy byly vygenerovány pomocí nástrojů Yii frameworku



Obr. č. 11 – Class-diagram webové služby

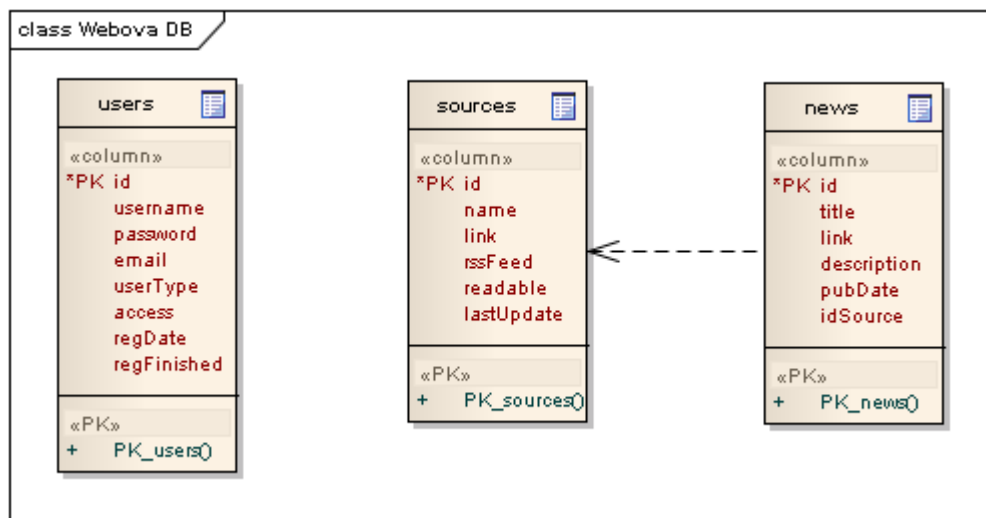
4.1.6 Databáze

Pro správnou funkčnost webové aplikace je třeba doplnit údaje potřebné pro připojení k databázi do konfiguračního souboru Yii frameworku *main.php*.

```
'db' => array (
```

```
'connectionString' => 'mysql:host=DBurl;dbname=DBname',
'emulatePrepare' => true,
'username' => 'user',
'password' => 'pass',
'charset' => 'utf8')
```

Dále je potřeba vytvořit tyto 3 tabulky: *users* pro uživatelské účty, *sources* pro zdroje novinek a *news* pro samotné novinky. Soubor s MySQL dotazy pro vytvoření těchto tabulek se nachází v adresáři *protected/data*.



Obr. č. 12 – Diagram databáze webové aplikace

4.2 Android aplikace

Android aplikace byla vytvářena v open-source vývojovém prostředí Eclipse Indigo (verze 3.7.2) pro verzi OS Android 2.2 (Froyo, úroveň API 8). Podle informací získaných z [9] by tato aplikace měla bezproblémově fungovat na přibližně 96% zařízeních používajících tento operační systém.

Dále byly použity tyto externí knihovny:

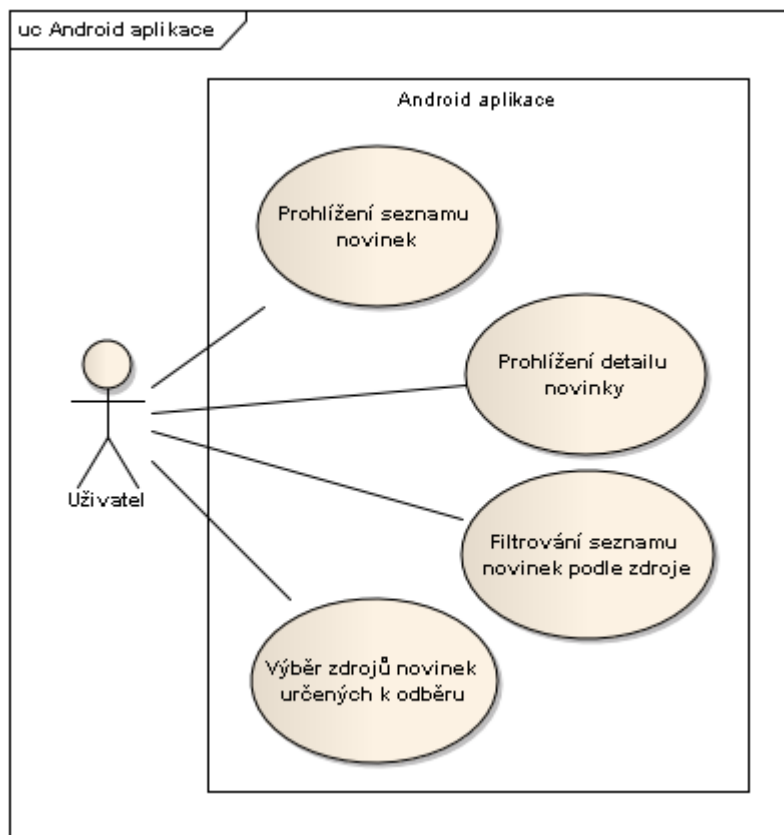
- kSOAP 2 verze 2.6.3
- Apache Commons Lang verze 3.1

4.2.1 Požadavky

Požadavky na Android aplikaci:

- Zobrazení novinek o dění na UTB ve Zlíně
- Schopnost komunikovat s webovou službou
- Správné zpracování dat získaných z webové služby
- Možnost offline provozu aplikace
- Filtrování novinek
- Možnost odebírat novinky pouze z některých zdrojů
- Zobrazení úvodní obrazovky při spuštění aplikace

4.2.2 Use-case diagram



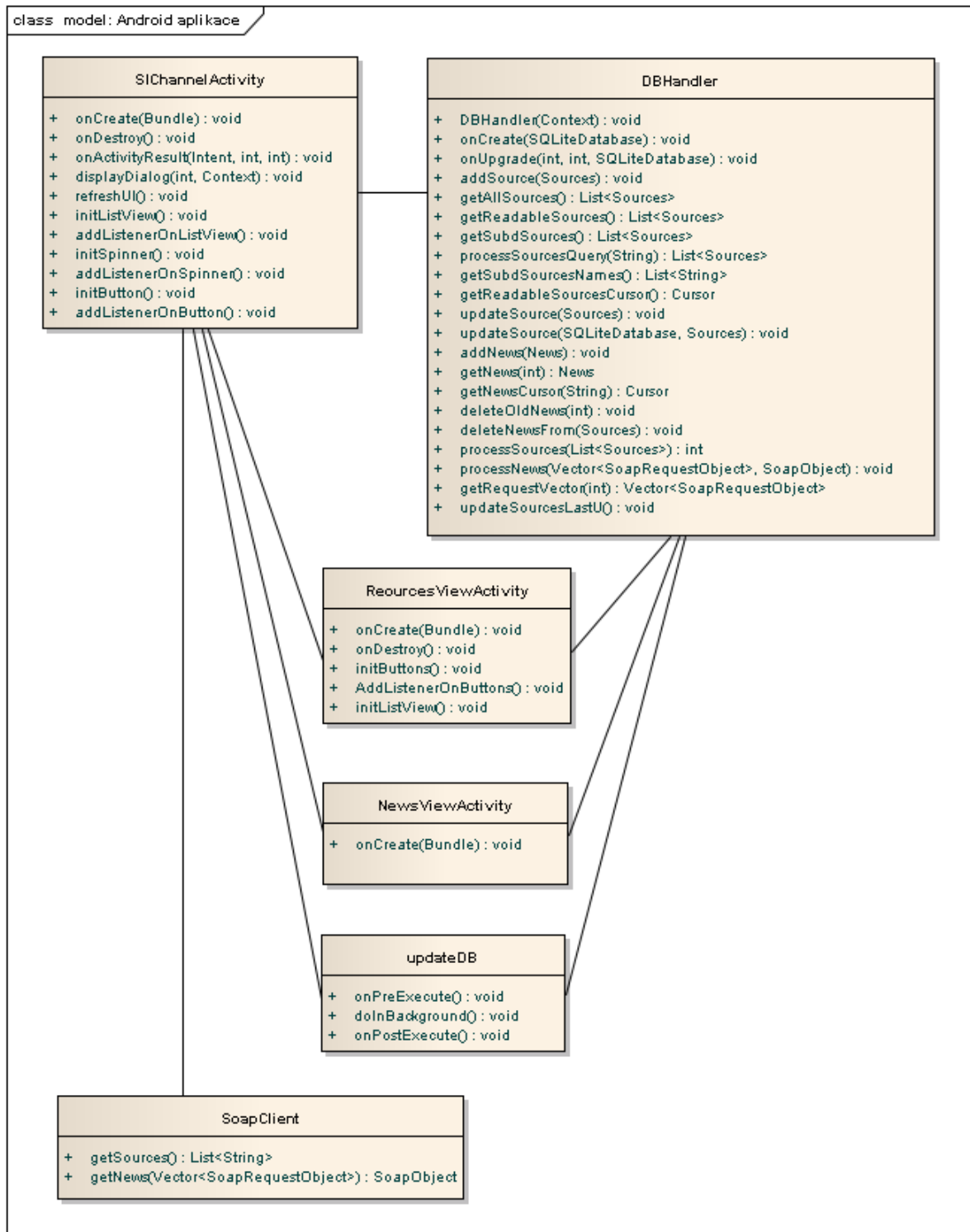
Obr. č. 13 – Use-case diagram Android aplikace

4.2.3 Class diagram

Popis tříd v diagramu:

- *SICchannelActivity* – hlavní aktivita aplikace, zajišťuje zobrazení seznamu novinek
- *ResourcesViewActivity* – aktivita, která zobrazuje seznam zdrojů dostupných k odběru
- *NewsViewActivity* – aktivita, která zobrazuje detail novinky
- *DBHandler* – třída sloužící jako rozhraní pro přístup k databázi
- *updateDB* – služba, která zajišťuje zobrazení úvodní obrazovky a zároveň na pozadí aktualizuje databázi
- *SoapClient* – třída sloužící pro komunikaci s webovou službou

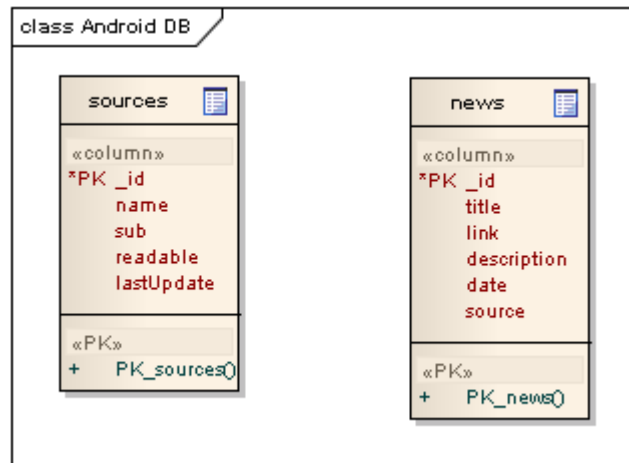
Dále je použito několik tříd sloužících pro zapouzdření dat – *News*, *Sources* a *SoapRequestObject*.



Obr. č. 14 – Class-diagram Android aplikace

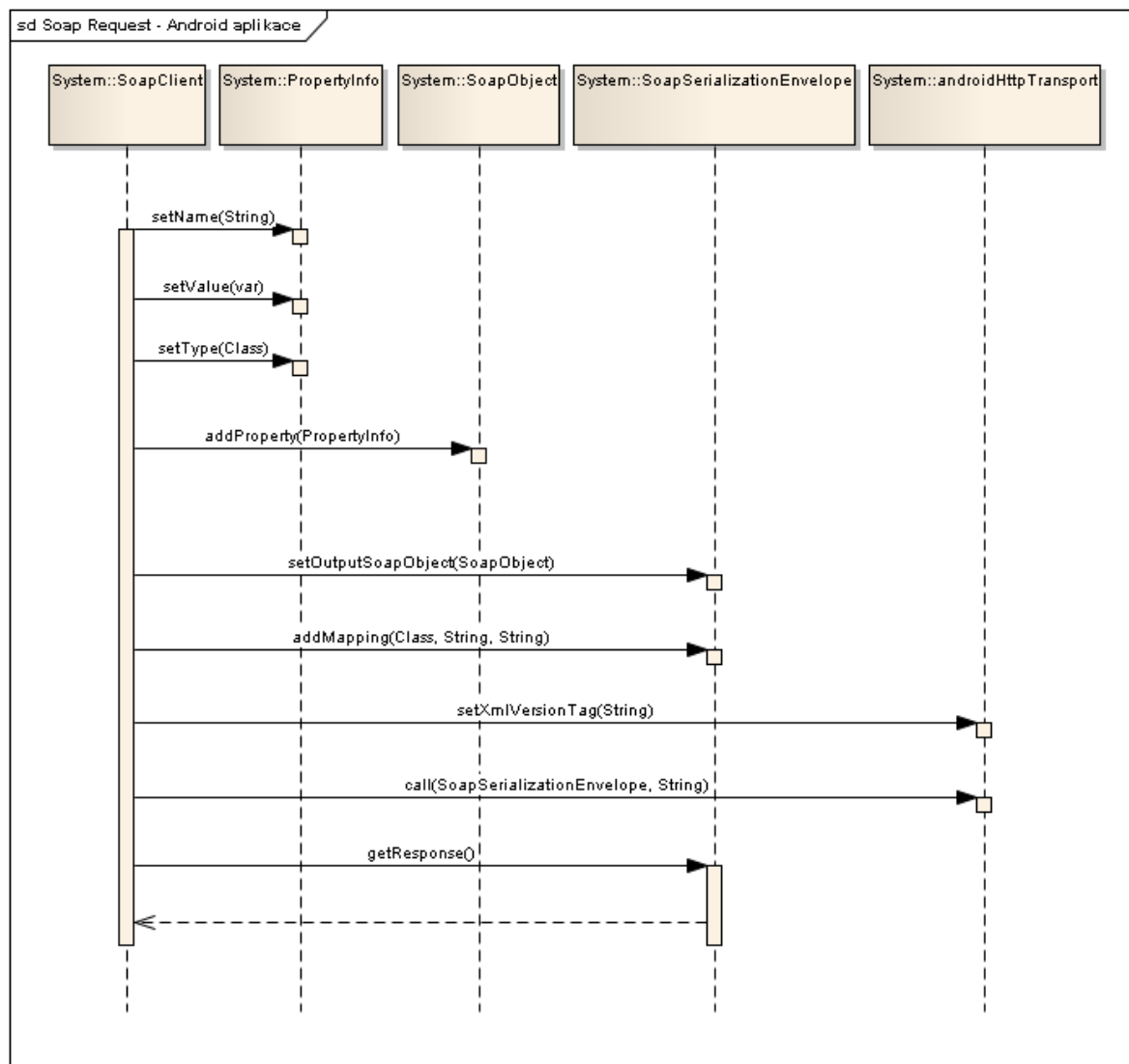
4.2.4 Databáze

Pro správnou funkci aplikace je třeba vytvořit v databázi 2 tabulky – *news* pro novinky a *sources* pro zdroje novinek. Jejich vytvoření probíhá v třídě *DBHandler* v metodě *onCreate*.



Obr. č. 15 – Diagram databáze Android aplikace

4.2.5 Způsob komunikace s webovou službou



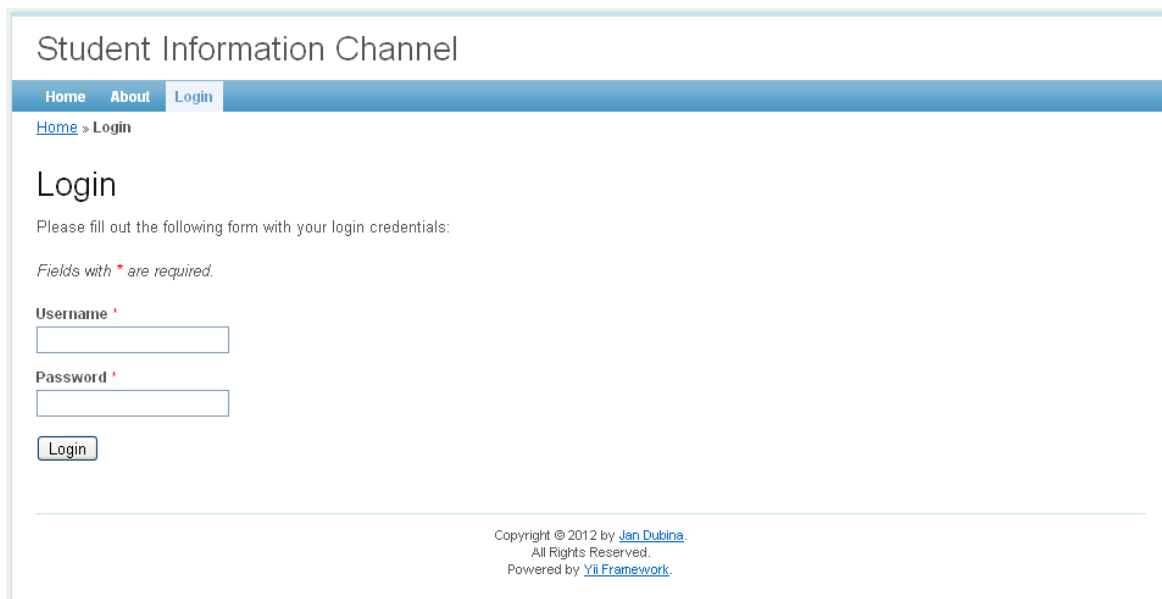
Obr. č. 16 – Sequence-diagram vytváření SOAP žádosti

Nejprve je třeba vytvořit objekt třídy *PropertyInfo*, která slouží k uchování dat určených pro SOAP komunikaci, včetně informací o těchto datech. Po nastavení názvu, hodnoty a datového typu přenášených dat je třeba vložit tento objekt pomocí funkce *addProperty* do objektu třídy *SoapObject*. Tento objekt reprezentuje tělo SOAP obálky zasílané žádosti, do vytvářené obálky ji vložíme voláním metody *setOutputObject* nad instancí třídy *SoapSerializationEnvelope*. Funkce *addMapping* slouží pro přiřazení správných XML tagů pro datové objekty přenášené v SOAP žádosti. Obálku poté zašleme pomocí objektu třídy *androidHttpTransport*, funkcí *setXmlVersionTag* přidáme k obálce XML hlavičku a funkcí *call* celou zprávu odešleme. Tělo SOAP odpovědi poté získáme voláním funkce *getResponse* nad objektem obálky.

5 UŽIVATELSKÁ DOKUMENTACE

5.1 Webová aplikace

Po zadání adresy webové aplikace do prohlížeče se zobrazí úvodní stránka. Pro přístup k funkcím webové aplikace je třeba se po kliknutí na položku *Login* v menu přihlásit do systému.



Obr. č. 17 – Webová aplikace – přihlašovací stránka

Po úspěšném přihlášení dojde k přidání položek do menu. Pro běžné uživatele přibude položka *News*, pro administrátory ještě navíc položky *Users* a *Sources*.



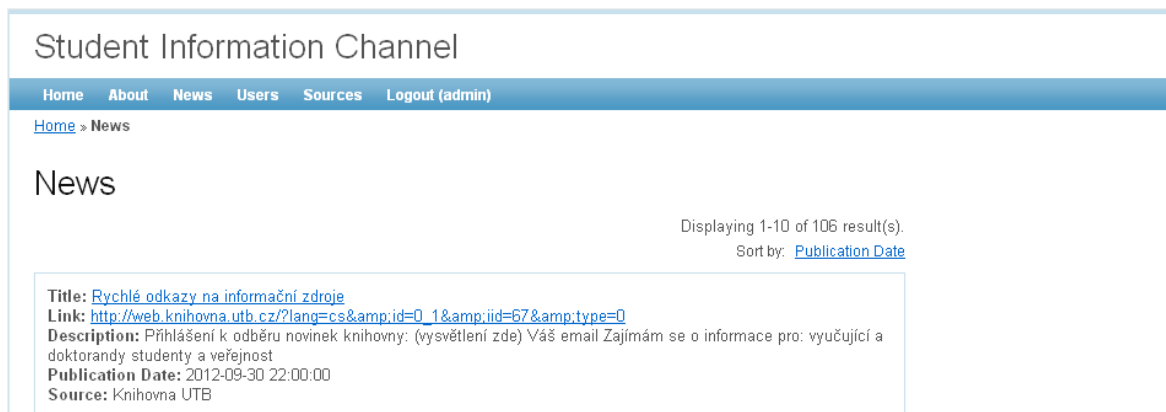
Obr. č. 18 – Webová aplikace – menu pro běžné uživatele



Obr. č. 19 - Webová aplikace – menu pro administrátory

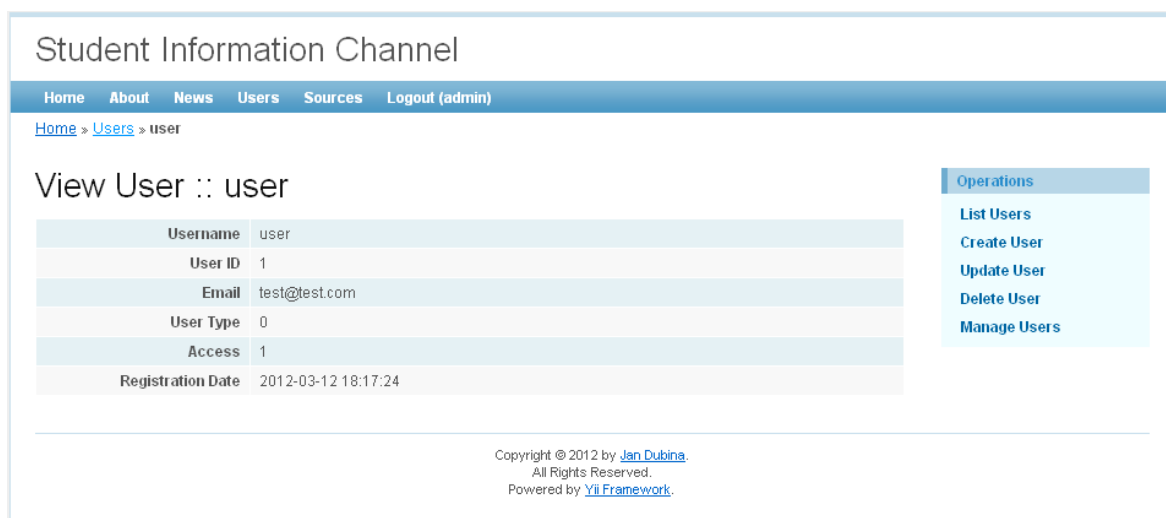
- *News* – prohlížení novinek
- *Sources* – správa informačních zdrojů
- *Users* – správa uživatelských účtů

Novinky lze pouze prohlížet, nezáleží, zda je přihlášený běžný uživatel nebo administrátor. Kliknutím na název novinky se otevře detail novinky, po kliknutí na odkaz je uživatel přesměrován na detail dané novinky na webové stránce příslušného zdroje.



Obr. č. 20 – Webová aplikace – seznam novinek

Rozhraní pro správu uživatelských účtů a zdrojů jsou velmi podobná. Při prohlížení seznamu jsou dostupné pouze operace pro vytvoření a správu, ale po vybrání určitého uživatele nebo zdroje se zobrazí nabídka všech dostupných operací.



Obr. č. 21 – Webová aplikace – detail uživatele

Dostupné operace:

- *List* – prohlížení seznamu prvků
- *Create* – vytvoření nového prvku
- *Update* – editace právě zobrazeného prvku
- *Delete* – odstranění právě zobrazeného prvku

- *Manage* – zobrazí seznam prvků, včetně vyhledávání a rychlých odkazů na operace s danými prvky

Student Information Channel

Home
About
News
Users
Sources
Logout (admin)

[Home](#) » [Users](#) » [Manage](#)

Manage Users

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

Username

Email

User Type

Access

Registration Date

Registration Finished

Operations

List Users

Create Users

Displaying 1-2 of 2 result(s).

Username	Email	User Type	Access	Registration Date	Registration Finished	
user	test@test.com	0	1	2012-03-12 18:17:24	1	
admin	test@testovaci.cz	1	1	2012-05-17 17:17:20	1	

Copyright © 2012 by [Jan Dubina](#).
 All Rights Reserved.
 Powered by [Yii Framework](#).

Obr. č. 22 – Webová aplikace – správa uživatelských účtů

Po kliknutí na *Manage Users/Sources* se otevře stránka s jednoduchým administrátorským rozhraním pro správu uživatelských účtů/zdrojů novinek. V seznamu lze vyhledávat pomocí formuláře, který se zobrazí po kliknutí na *Advanced Search*, nebo pomocí polí zobrazených nad každým sloupcem seznamu. Vedle každého prvku seznamu se dále nachází odkazy na prohlížení, editaci a smazání daného záznamu.

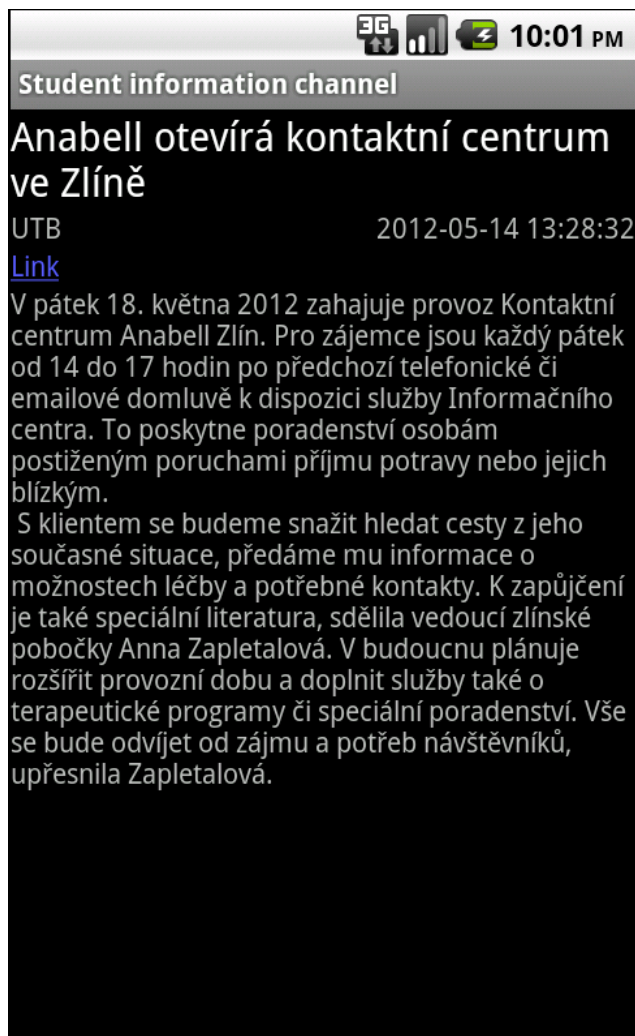
5.2 Android aplikace

Po spuštění aplikace se zobrazí úvodní obrazovka a proběhne stažení aktuálních novinek z webového serveru. Poté se zobrazí hlavní okno aplikace.



Obr. č. 23 – Android aplikace – hlavní obrazovka

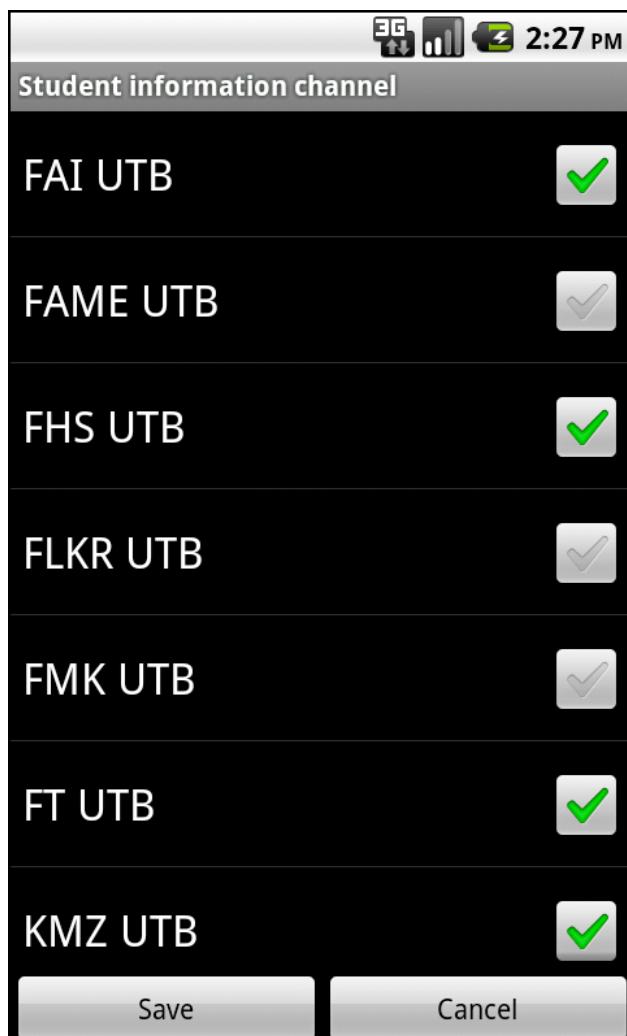
V horní části obrazovky je umístěno rozbalovací menu, pomocí kterého lze zobrazit všechny novinky nebo jen novinky z určitého zdroje. Pod ním je umístěn seznam novinek, u každé je uveden název novinky, její zdroj a datum a čas vydání novinky (datum a čas je přizpůsobený časovému pásmu nastaveném na daném zařízení). Při kliknutí na některou z novinek se otevře okno s detailem novinky.



Obr. č. 24 – Android aplikace – detail novinky

Zde je na rozdíl od hlavní obrazovky zobrazen i odkaz na novinku a její celý text. Při kliknutí na odkaz se otevře implicitní webový prohlížeč a v něm detail dané novinky na webové stránce příslušného zdroje.

Správa odebíraných zdrojů se provádí v okně, které se otevře po kliknutí na tlačítko *Sources* na hlavní obrazovce.



Obr. č. 25 – Android aplikace – seznam dostupných zdrojů

V tomto okně lze ze seznamu všech dostupných zdrojů vybrat ty, které chceme odebírat. Zdroje, které nejsou odebírány, se nezobrazí v hlavním okně aplikace a při startu aplikace nejsou aktualizovány.

ZÁVĚR

Cílem této práce bylo vytvořit informační systém koncentrující informace o aktuálním dění na UTB ve Zlíně. Součástí tohoto systému je také klientská aplikace určená pro mobilní zařízení. Webová část je vytvořená v programovacím jazyku PHP s využitím frameworku Yii. Jedná se o relativně nový framework, který vyniká svou rychlostí a snahou o změnu přístupu k využívání PHP frameworků. Nevýhodou je hlavně nedostatek informací o vytváření aplikací pomocí tohoto frameworku a také jeho složitost, díky které je naučit se pracovat s Yii poměrně časově náročné. Aplikace pro mobilní zařízení byla vyvíjena pro operační systém Android, který je aktuálně nejrozšířenější platformou pro tato zařízení. Výhodou tohoto systému je open-source přístup k jeho vývoji, veškeré nástroje potřebné pro vytváření aplikací jsou dostupné zdarma. To na druhou stranu způsobuje také potíže, protože propojení prostředí Eclipse s Android emulátorem může být problematické.

Aplikace jsou napsány tak, aby je bylo možné v budoucnu rozšířit. Vhodné by bylo například upravit aktuální webové rozhraní, které slouží pro správu webové služby, na informační webový portál nebo alespoň rozšířit službu tak, aby zpracovávala novinky ze stránek bez RSS kanálů a umožňovala vkládání novinek pomocí webového rozhraní. U Android aplikace se nabízí možnost využití fragmentů, pomocí kterých by bylo možné uzpůsobit aplikaci i pro zařízení s větším displayem.

ZÁVĚR V ANGLIČTINĚ

The purpose of this thesis was to create an information system concentrating information about current events at TBU in Zlín. Part of this system is also a client application for mobile devices. The web part is created in PHP programming language using Yii framework. This is a relatively new framework that excels in its speed and attempts to change the way of PHP frameworks usage. Main disadvantage is the lack of information about creating applications using this framework and its complexity that makes learning Yii quite time consuming. Application for mobile devices was developed for the Android operating system that is currently the most widely used platform for these devices. The advantage of this system is an open-source approach to its development, all the tools needed for creating applications are available free of charge. On the other hand this also causes problems because connecting Eclipse to Android emulator can be problematic.

Applications are written with emphasis for future extension. For example, it would be convenient to convert current web interface to a proper information web portal, or at least extend the web service to process web pages without RSS feeds and add news creation via web interface. The Android application could be customized for larger screen sizes using fragments.

SEZNAM POUŽITÉ LITERATURY

- [1] MEIER, Reto. Professional Android 2 Application Development. Indianapolis: Wiley Publishing, 2010. ISBN 978-0-470-56552-0
- [2] STEELE, James a Nelson TO. The Android Developer's Cookbook: Building Applications with the Android SDK. Upper Saddle River: Pearson Education, 2010. ISBN 978-0-321-74123-3
- [3] ROGERS, Rick, John LOMBARDO, Zigurd MEDNIEKS a Blake MEIKE. Android Application Development. Sebastopol: O'Reilly Media, 2009. ISBN 05-965-2147-2
- [4] COLLINS, Charlie, Michael GALPIN a Matthias KÄPPLER. Android in Practice. Shelter Island: Manning Publications, 2011. ISBN 9781935182924
- [5] BEGUN, Daniel. Amazing Android Apps For Dummies. Indianapolis: Wiley Publishing, 2011. ISBN 978-0-470-93629-0
- [6] KANISOVÁ, H. - MÜLLER, M. UML srozumitelně. 2. akt. vyd. Brno: Computer Press, 2006. 176 s. ISBN 80-251-1083-4
- [7] YII FRAMEWORK [online]. Dostupné online na: <http://www.yiiframework.com/>
- [8] LARRY ULLMAN [online]. Dostupné online na: <http://www.larryullman.com/>
- [9] ANDROID [online]. Dostupné online na: <http://www.android.com/>
- [10] OPEN HANDSET ALLIANCE [online]. Dostupné online na: <http://www.openhandsetalliance.com/>
- [11] ANDROID DEVELOPERS CHANNEL [video, online]. Dostupné online na: <http://www.youtube.com/user/androiddevelopers>
- [12] MATT BURNS, TECHCRUNCH.COM [online]. Dostupné online na: <http://techcrunch.com/2012/02/27/850k-android-activations-daily-300m-total-devices-says-andy-rubin/>
- [13] WORLD WIDE WEB CONSORTIUM (W3C) [online]. Dostupné online na: <http://www.w3.org/>
- [14] SOAPUSER [online]. Dostupné online na: <http://www.soapuser.com/>

[15] RSSBOARD [online]. Dostupné online na: <http://www.rssboard.org/>

[16] W3SCHOOLS [online]. Dostupné online na: <http://www.w3schools.com/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

Yii	Yes It Is!
PHP	PHP: Hypertext Preprocessor
BSD	Berkeley Software Distribution
DRY	Don't repeat yourself
RPS	Requests per second
APC	Alternative PHP Cache
MVC	Model-View-Controller
CRUD	Create, Read, Update and Delete
SQL	Structured Query Language
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CSS	Cascading Style Sheets
OS	Operační systém
API	Application programming interface
GNU	GNU's Not Unix
SDK	Software developer's kit
JDK	Java Development Kit
JRE	Java Runtime Environment
ADT	Android Development Tools
XML	Extensible Markup Language
SOAP	Simple Object Access Protocol
RPC	Remote Procedure Call
RSS	Really Simple Syndication
HTML	HyperText Markup Language

WSDL Web Services Description Language

SEZNAM OBRÁZKŮ

<i>Obr. č. 1 – Srovnání výkonu PHP frameworků (Zdroj: [7])</i>	11
<i>Obr. č. 2 – Schéma architektury MVC (Zdroj: [7]).....</i>	12
<i>Obr. č. 3 – Ověření požadavků Yii frameworku</i>	14
<i>Obr. č. 4 – Základní aplikace vygenerovaná pomocí Yii.....</i>	15
<i>Obr. č. 5 – Nástroj Gii</i>	17
<i>Obr. č. 6 – Rozložení verzí OS Android (Zdroj: [9]).....</i>	18
<i>Obr. č. 7 – Životní cyklus aktivity (Zdroj: [9])</i>	22
<i>Obr. č. 8 – Životní cyklus služby (Zdroj: [9]).....</i>	23
<i>Obr. č. 9 – Schéma RCP komunikace (Zdroj: [14])</i>	25
<i>Obr. č. 10 – Use-case diagram webové aplikace.....</i>	31
<i>Obr. č. 11 – Class-diagram webové služby</i>	32
<i>Obr. č. 12 – Diagram databáze webové aplikace.....</i>	33
<i>Obr. č. 13 – Use-case diagram Android aplikace</i>	34
<i>Obr. č. 14 – Class-diagram Android aplikace.....</i>	36
<i>Obr. č. 15 – Diagram databáze Android aplikace.....</i>	37
<i>Obr. č. 16 – Sequence-diagram vytváření SOAP žádosti</i>	38
<i>Obr. č. 17 – Webová aplikace – přihlašovací stránka</i>	39
<i>Obr. č. 18 – Webová aplikace – menu pro běžné uživatele</i>	39
<i>Obr. č. 19 - Webová aplikace – menu pro administrátory.....</i>	39
<i>Obr. č. 20 – Webová aplikace – seznam novinek.....</i>	40
<i>Obr. č. 21 – Webová aplikace – detail uživatele.....</i>	40
<i>Obr. č. 22 – Webová aplikace – správa uživatelských účtů</i>	41
<i>Obr. č. 23 – Android aplikace – hlavní obrazovka</i>	42
<i>Obr. č. 24 – Android aplikace – detail novinky</i>	43
<i>Obr. č. 25 – Android aplikace – seznam dostupných zdrojů</i>	44

SEZNAM TABULEK

<i>Tab. č. 1 - – Verze OS Android (Zdroj: [9])</i>	19
--	----

SEZNAM PŘÍLOH

Zdrojový kód webové aplikace a Android aplikace na přiloženém CD.