

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



Jakub Kinšt

### Systém pro podporu výuky

Ústav formální a aplikované lingvistiky, MFF UK

Vedoucí bakalářské práce: Mgr. Miroslav Týnovský

Studijní program: Informatika

Studijní obor: Obecná Informatika

Praha 2012

TODO: Zde bude poděkování

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne

podpis

Název práce: Systém pro podporu výuky

Autor: Jakub Kinšt

Katedra (ústav): Ústav formální a aplikované lingvistiky

Vedoucí bakalářské práce: Mgr. Miroslav Týnovský

e-mail vedoucího: tynovsky@ufal.mff.cuni.cz

Abstrakt: V předložené práci studujeme ... Uvede se abstrakt v rozsahu 80 až 200 slov. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut sit amet sem. Mauris nec turpis ac sem mollis pretium. Suspendisse neque massa, suscipit id, dictum in, porta at, quam. Nunc suscipit, pede vel elementum pretium, nisl urna sodales velit, sit amet auctor elit quam id tellus. Nullam sollicitudin. Donec hendrerit. Aliquam ac nibh. Vivamus mi. Sed felis. Proin pretium elit in neque. Pellentesque at turpis. Maecenas convallis. Vestibulum id lectus. Fusce dictum augue ut nibh. Etiam non urna nec mi mattis volutpat. Curabitur in tortor at magna nonummy gravida. Mauris turpis quam, volutpat quis, porttitor ut, condimentum sit amet, felis.

Klíčová slova:

Title: Learning management system

Author: Jakub Kinšt

Department: Institute of Formal and Applied Linguistics

Supervisor: Mgr. Miroslav Týnovský

Supervisor's e-mail address: tynovsky@ufal.mff.cuni.cz

Abstract: In the present work we study ... Uvede se anglický abstrakt v rozsahu 80 až 200 slov. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut sit amet sem. Mauris nec turpis ac sem mollis pretium. Suspendisse neque massa, suscipit id, dictum in, porta at, quam. Nunc suscipit, pede vel elementum pretium, nisl urna sodales velit, sit amet auctor elit quam id tellus. Nullam sollicitudin. Donec hendrerit. Aliquam ac nibh. Vivamus mi. Sed felis. Proin pretium elit in neque. Pellentesque at turpis. Maecenas convallis. Vestibulum id lectus. Fusce dictum augue ut nibh. Etiam non urna nec mi mattis volutpat. Curabitur in tortor at magna nonummy gravida. Mauris turpis quam, volutpat quis, porttitor ut, condimentum sit amet, felis.

Keywords:

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Motivace . . . . .	3
1.2	Struktura práce . . . . .	3
<b>2</b>	<b>Specifikace</b>	<b>4</b>
2.1	Specifikace webové aplikace . . . . .	4
2.2	Specifikace mobilní aplikace . . . . .	4
<b>3</b>	<b>Problematika propojení s mobilní aplikací</b>	<b>5</b>
3.1	Motivace . . . . .	5
3.2	Otevření API aplikace . . . . .	6
3.2.1	Architektura SOAP . . . . .	6
3.2.2	Architektura REST . . . . .	7
3.2.3	Formát přenosu dat . . . . .	7
3.2.4	Odesílání dat na server . . . . .	10
3.3	Přenos Cookies . . . . .	10
3.4	Zabezpečení API . . . . .	11
<b>4</b>	<b>Architektura implementace</b>	<b>12</b>
<b>5</b>	<b>Použité technologie a frameworky</b>	<b>13</b>
<b>6</b>	<b>Existující implementace</b>	<b>14</b>
<b>7</b>	<b>Možné rozšíření</b>	<b>15</b>
<b>8</b>	<b>Uživatelská dokumentace</b>	<b>16</b>
8.1	Webová aplikace . . . . .	16
8.1.1	Minimální požadavky . . . . .	16
8.1.1.1	Provoz aplikace . . . . .	16
8.1.1.2	Použití aplikace . . . . .	16
8.1.2	Instalace aplikace . . . . .	16

8.1.3	Spuštění aplikace . . . . .	17
8.1.4	Výchozí stránka, registrace uživatele a přihlášení do aplikace . . . . .	18
8.1.5	Zakládání nového kurzu a pozvání studentů . . . . .	18
8.1.6	Seznam lekcí a detail lekce . . . . .	18
8.1.7	Výsledky (Results) . . . . .	19
8.1.8	Úkoly/Testy (Assignments) . . . . .	19
8.1.9	Zdroje/Dokumenty . . . . .	21
8.1.10	Fórum . . . . .	21
8.1.11	Události . . . . .	21
8.1.12	Zprávy . . . . .	21
8.1.13	Uživatelský profil . . . . .	22
8.1.14	Oznámení e-mailem . . . . .	22
8.1.15	Nastavení . . . . .	22
8.2	Mobilní aplikace . . . . .	22
8.2.1	Minimální požadavky . . . . .	22
8.2.2	Instalace a spuštění aplikace . . . . .	23
8.2.3	Výchozí obrazovka a přihlášení do aplikace . . . . .	23
8.2.4	Detail kurzu . . . . .	23
8.2.5	Detail lekce . . . . .	23
8.2.6	Diskuzní fórum . . . . .	24
8.2.7	Události . . . . .	24
8.2.8	Výsledky . . . . .	24
8.2.9	Úkoly/Testy (Assignments) . . . . .	24
8.2.10	Zdroje (Resources) . . . . .	24
8.2.11	Zprávy . . . . .	25
<b>9</b>	<b>Závěr</b>	<b>26</b>
<b>10</b>	<b>Seznam příloh</b>	<b>27</b>
	<b>Literatura</b>	<b>28</b>

# Kapitola 1

## Úvod

### 1.1 Motivace

TODO

### 1.2 Struktura práce

TODO

## Kapitola 2

### Specifikace

2.1 Specifikace webové aplikace

2.2 Specifikace mobilní aplikace



# Kapitola 3

## Problematika propojení s mobilní aplikací

### 3.1 Motivace

V posledních letech je viditelný významný přesun práce s internetem z domácích počítačů na mobilní zařízení. Mobilními zařízeními jsou myšleny především mobilní telefony a tablety. Provozovatelé webových služeb jsou tak stále častěji nuceni umožnit klientovi užívat aplikaci právě z těchto zařízení.

Někteří řeší tento trend cestou nejmenšího odporu - optimalizací webových stránek aplikace pro menší obrazovky přenosných zařízení. Toto řešení spolu ale nese mnoho nevýhod. Aplikace především nemůže naplno využít potenciál zařízení. Neposkytují takový komfort jako aplikace integrované do systému, které mohou využívat kompletní sadu frameworků dodávaných výrobcem operačního systému. Nemohou například přistupovat k datům z jiných aplikací, jako je seznam kontaktů, kalendář a další. Nemohou ani naplno využívat některé hardwarové technologie zabudované v moderních zařízeních, jako jsou senzory polohy, světla, zrychlení, fotoaparát a podobně.

Toto řešení také kompletně znepřístupní obsah v případě nedostupnosti internetového připojení. Uživatel tedy může aplikaci používat pouze pokud je v tu chvíli připojen k internetu.

Jistou nadějí pro tento přístup může být technologie webových stránek v HTML5[10]. Ta přidává do tradičních HTML stránek možnost využít některé z těchto pokročilých funkcí. Je zde možnost obsah ke klientovi uložit a pak ho zobrazit offline - tedy krátkodobě bez připojení k internetu. Pomocí HTML5 elementů je také možné zjistit polohu uživatele pomocí zabudovaných senzorů. Další novinkou je element *canvas*, který umožňuje do okna prohlížeče vykreslovat libovolnou grafiku. HTML v této verzi již také nativně podporuje

přehrávání videa nebo audia - už není nutné vytvářet proprietární přehrávače například pomocí technologie Flash. Vylepšeny byly i standardní formuláře a práce s nimi.

Všechna tato vylepšení se zatím ukazují jako nedostatečná. Integrace s operačním systémem zařízení je minimální a v porovnání s nativními aplikacemi ty webové (HTML5) stále hodně zaostávají. Navíc je nutná podpora ze strany internetového prohlížeče, což se v praxi ukazuje jako velký problém. Různé prohlížeče interpretují jen různé podmnožiny pravidel HTML5.

Programování aplikace pro mobilní telefony pomocí technologie HTML5 ale přináší jednu velkou nespornou výhodu. HTML5 je totiž technologie multiplatformní. Provozovatelé služeb nemusejí platit vývoj pro všechny rozšířené mobilní platformy - Android, iOS a Windows Phone 7. Otázkou je, zda není tato výhoda zastíněna zmiňovanými nevýhodami.

Budoucnost pravděpodobně spěje k vývoji univerzálních multiplatformních aplikací, ale současná nabídka nástrojů k jeho docílení se zatím ukazuje jako nedostatečná.

### 3.2 Otevření API aplikace

Otevřít API<sup>1</sup> webové aplikace znamená zpřístupnit obsah aplikace jinému programu, tedy především navrhnout rozhraní, pomocí kterého bude možné jednak strojově přecházet data a také aplikaci poslat vstup - ovládat ji.

Pro řešení takového problému existuje mnoho odladěných a hojně používaných implementací, ze kterých lze vybírat. Výběr se však zužuje pokud se vezme v úvahu implementace webové aplikace v PHP a také to, že aplikace je již naprogramovaná a vývojář chce co nejvíce využít již implementované služby.

#### 3.2.1 Architektura SOAP

Jedna z možností je využít architekturu pro webové služby s názvem SOAP[6]. Tato architektura využívá jako formát přenášených dat standard XML a k vlastnímu přenosu zpravidla HTTP nebo SMTP. Implementace je ale náročnější a v podstatě při ní nelze použít nic z implementace samotné webové služby.

---

<sup>1</sup>Application Programming Interface

### 3.2.2 Architektura REST

Druhou možností je využití architektury REST[11]. Tato architektura je na rozdíl od SOAP orientovaná datově a ne procedurálně. REST nedefinuje vzdálené volání procedur, ale rovnou říká jak přistupovat ke vzdáleným datům.

K přenosu je využit protokol HTTP (autor REST je spoluautorem protokolu HTTP). Každý datový zdroj je identifikovaný unikátním identifikátorem URI<sup>2</sup>. K získání a ke vkládání dat se používají standardní metody HTTP GET a POST. REST definuje i konkrétnější metody jako DELETE, PUT, ale jelikož většina implementací HTTP klientů podporuje pouze metody GET a POST, využívá se v praxi k mazání a aktualizaci dat metoda POST. Jako formát přenášených dat je možné použít prakticky cokoli - XML, text oddělený oddělovačem, nebo formát JSON.

V podstatě stejně ale funguje typická aplikace naprogramovaná v jazyce PHP. Data se získávají GET požadavkem ve formátu HTML, vstupní data se posílají přes formulář pomocí požadavku POST. Díky tomu se RESTová architektura nabízí jako nejvhodnější varianta při požadavku využití stávajícího kódu v co největší míře.

K otevření API tedy stačí aby server poznal, že klientem není člověk, ale program (konkrétně mobilní aplikace), a na každé podstránce aplikace místo obsahu zformátovaného do HTML zaslal obsah zformátovaný tak, aby byl strojově čitelný.

### 3.2.3 Formát přenosu dat

Zbývá tedy zvolit standard, pomocí kterého budou data ze serveru do mobilní aplikace přenášena a zároveň bude jejich obsah snadno strojově čitelný. Nabízejí se dva nejrozšířenější a nejvíce podporované standardy - XML a JSON. Výběr záleží na použitých technologiích, konkrétně na tom, na jaké úrovni je existující implementace pro práci s těmito formáty. Dále může rozhodovat to, zda je nějaká potřeba u přenášených dat definovat strukturu. Tento požadavek lépe splňuje formát XML (XML Schema, DTD a další).

Ukázka formátu XML:

```
<auto id="auto-1">
  <vin>128988129823</vin>
  <spz>2A29288</spz>
  <znacka>Skoda</znacka>
  <typ>Superb</typ>
  <rok-vyroby>2009</rok-vyroby>
```

---

<sup>2</sup>Uniform Resource Identifier

### KAPITOLA 3. PROBLEMATIKA PROPOJENÍ S MOBILNÍ APLIKACÍ 8

```
<parametry>
  <motor>
    <spotreba>9.2</spotreba>
    <objem>3,0</objem>
    <maxrychlost>210kmh</maxrychlost>
  </motor>
</parametry>
<porizovaci-cena>750000</porizovaci-cena>
<pujcovne>1230</pujcovne>
<!-- Poznamky k vozidlu + zname poruchy -->
<poznamky>
  Potreba vymenit olej pri 20000km
  Technicke kontroly:
</poznamky>
<nahrada autoID="auto-2" />
</auto>
```

Ukázka stejných dat ve formátu JSON:

```
{
  "auto": {
    "id": "auto-1",
    "vin": "128988129823",
    "spz": "2A29288",
    "znacka": "Skoda",
    "typ": "Superb",
    "rok-vyroby": "2009",
    "parametry": {
      "motor": {
        "spotreba": "9.2",
        "objem": "3,0",
        "maxrychlost": "210kmh"
      }
    },
    "porizovaci-cena": "750000",
    "pujcovne": "1230",
    "poznamky": "
      Potreba vymenit olej pri 20000km
      Technicke kontroly:
    ",
    "nahrada": {
      "autoID": "auto-2"
```



```
// zjistíme spotřebu auta (typ double)
double spotreba = motor.getDouble("spotreba");

...
```

### 3.2.4 Odesílání dat na server

Odesílání dat na server nebude muset v případě použití RESTové architektury projít prakticky žádnou změnou na straně serveru. Pokud vstup probíhá pomocí GET požadavku, nejedná se o nic zvláštního - pokud jsou parametry součástí URL, server si s nimi poradí stejně jako v případě normálního užívání aplikace v prohlížeči.

Většina vstupu ve webové aplikaci probíhá přes formuláře za pomoci POST požadavku. Úkolem klienta v mobilní aplikaci bude tedy nasimulovat odeslání formuláře. To ale znamená provést jednoduchý POST požadavek na URL, které je nastaveno jako parametr *action* konkrétního formuláře, a k požadavku přidat jednotlivé hodnoty položek formuláře jako POST argumenty. Může se jednat o primitivní datové typy, ale musí být možné přes POST požadavek poslat i libovolný soubor, tak jako to lze udělat v HTML formuláři.

Pokud se vše správně implementuje, server zpracuje data stejně jako kdyby přicházela ze standardního formuláře z HTML.

## 3.3 Přenos Cookies

Problém, který nastane například při zachovávání možnosti přihlášení uživatelů do aplikace, je nutnost přenášet mezi požadavky hodnoty HTTP hlaviček Cookies[8].

*Cookie* a *Set-Cookie* jsou standardní hlavičky HTTP, které mají za úkol udržovat stavové informace mezi serverem a konkrétním klientem. Server může v HTTP odpovědi zaslat klientovi hlavičku *Set-Cookie* s nějakou hodnotou a do své paměti si uloží libovolnou stavovou informaci. Pokud v budoucnosti (během platnosti *Cookie*) klient ke svému HTTP požadavku přidá hlavičku *Cookie* s přijatou hodnotou, server klienta podle hodnoty identifikuje a z paměti zjistí potřebnou stavovou informaci.

V praxi se tato technika používá například právě udržení informace o přihlášení uživatele. Když se uživatel přihlásí, server si o něm vytvoří záznam, a aby se uživatel nemusel přihlašovat v každém následujícím požadavku, odešle klientovi (prohlížeči) hlavičku *Set-Cookie* s (tajným) identifikátorem a s určitou dobou platnosti. Pokud klient potom (před vypršením platnosti) pošle

požadavek kde hodnotou jedné z hlaviček *Cookie* bude i zmíněný identifikátor, pozná server o jakého klienta se jedná a už po něm nebude požadovat opětovné přihlášení - uživatel je přihlášen po celou dobu platnosti *Cookie*.

Pokud tedy požadujeme maximální využití stávajícího kódu standardní webové aplikace, je potřeba v tomto ohledu simulovat chování webového prohlížeče a s *Cookies* nakládat stejně. Z odpovědi serveru je tedy potřeba sbírat hlavičky *Set-Cookie*, kontrolovat jejich platnost a v následných dotazech na server posílat hlavičky *Cookie* se správným obsahem.

### 3.4 Zabezpečení API

Otevřené API aplikace může představovat určité riziko. Může se stát, že je potřeba do mobilní aplikace posílat data, která by jindy byla uživateli nepřístupná, nebo prostě nechceme, aby bylo API aplikace veřejně přístupné. V tuto chvíli přichází otázka zabezpečení API.

Aby si byl server jist tím, že komunikuje s oprávněným klientem, je třeba, aby se prokázal tzv. API klíčem. API klíč je tajný řetězec dostatečné délky a složitosti, kterým se klient při požadavcích na server prokáže, že je k získání dat oprávněn.

API klíč typicky nemůže být nahrazen přihlášením uživatele, jelikož je často potřeba přístup k API omezit i v případě veřejných částí aplikace s povoleným anonymním přístupem.

Klíč může být přenášen v každém HTTP požadavku, nebo jen v prvním a pak využít možnosti *Cookies* stejně jako při přihlašování. Může být v požadavcích předáván jako GET parametr, POST parametr nebo může být uložen v hlavičkách. Všechny varianty mají své výhody i nevýhody.

Samotné použití API klíče nezaručuje naprostou bezpečnost API. Při sledování síťového provozu je stejně jako při přihlašování uživatelů možné API klíč zjistit. Proto je doporučeno provozovat webovou aplikaci na zabezpečeném protokolu *HTTPS*, kde je přenos šifrován pomocí technologií *SSL*<sup>3</sup> nebo *TLS*<sup>4</sup>.

---

<sup>3</sup>Secure Sockets Layer

<sup>4</sup>Transport Layer Security

## Kapitola 4

### Architektura implementace



## Kapitola 5

### Použíte technologie a frameworky

## Kapitola 6

### Existující implementace

## Kapitola 7

### Možné rozšíření

# Kapitola 8

## Uživatelská dokumentace

### 8.1 Webová aplikace

#### 8.1.1 Minimální požadavky

##### 8.1.1.1 Provoz aplikace

Aplikace není vázaná na žádnou konkrétní platformu, jelikož potřebný software je dostupný pro většinu využívaných operačních systémů včetně *OS Linux*, *Microsoft Windows* a *Mac OS*.

Pro provoz aplikace je potřeba mít nainstalovaný následující software:

- webový server *Apache*[2] s podporou *OpenSSL*
- *PHP*[5](verze alespoň 5.2)
- *MySQL*[3](verze alespoň 5.5)
- libovolný plánovač procesů (například program *cron*)

##### 8.1.1.2 Použití aplikace

K použití uživateli stačí běžný webový prohlížeč. Aplikace je optimalizována pro prohlížeč *Google Chrome*, *Mozilla Firefox* a *Microsoft Internet Explorer* 7 a vyšší.

#### 8.1.2 Instalace aplikace

K nainstalování aplikace stačí následovat následující pokyny:

1. Zkopírovat obsah adresáře *CourseManeger* do výchozí složky webového serveru (typicky složka *htdocs*)

2. Pokud jde o UNIXový operační systém, je potřeba nastavit oprávnění pro následující složky na hodnotu `777`
  - `/temp`
  - `/document_root/webtemp`
  - `/log`
  - `/uploads`
3. Vytvořit na serveru databázi s názvem *course-manager* určenou pro aplikaci
4. Nastavit přístupové údaje k databázi v souboru `/app/config.neon`
5. Nastavit přístupové údaje k libovolnému SMTP serveru k odesílání oznámení e-mailem
6. Vytvořit základní databázové schéma v MySQL databázi spuštěním skriptu `/db/CourseManager.sql`
7. Nakonfigurovat plánovač úloh aby spouštěl HTTP GET požadavky na následující URL
  - `URL/document_root/cron/sendmailsew5q3n825ml6bm2btz81` (každou minutu)
  - `URL/document_root/cron/deleteuncheckedusersew5q3n825ml6bm2btz81` (každý den)
  - `URL/document_root/cron/sendassignmentnotificationew5q3n825ml6bm2btz81` (každý den)
  - `URL/document_root/cron/deleteoldtempfilesew5q3n825ml6bm2btz81` (každý den)

Řetězec *URL* je potřeba nahradit adresou URL provozované aplikace

### 8.1.3 Spuštění aplikace

Webovou aplikaci CourseManager lze spustit ve webovém prohlížeči přechodem na adresu aplikace. Po instalaci na lokální webový server se jedná typicky o adresu `http://localhost/slozka_aplikace/document_root`. Webová aplikace je ale také veřejně dostupná na adrese `http://rp-jakub-cz.dc.starver.net`.

### 8.1.4 Výchozí stránka, registrace uživatele a přihlášení do aplikace

Výchozí stránka nabízí základní informace o aplikaci a vyzývá uživatele k registraci, popřípadě k přihlášení. Anonymní uživatel má právo zobrazit pouze tuto stránku a stránku s uživatelským manuálem.

Kliknutím na tlačítko Přihlášení (Login) se rozbalí nabídka pro přihlášení společně s odkazem na stránku registrace uživatele.

Aplikace při registraci požaduje po uživateli jméno, příjmení, uživatelské jméno ve formě e-mailové adresy a heslo o minimální délce 5 znaků, které musí uživatel z bezpečnostních důvodů zadat dvakrát.

Po odeslání registračního formuláře je do několika minut uživateli odeslán potvrzující e-mail s odkazem, po jehož navštívení potvrdí vlastnictví zadané e-mailové adresy.

Od chvíle potvrzení adresy je uživateli umožněno přihlášení do aplikace prostřednictvím přihlašovacího formuláře. Po přihlášení je uživatel přesměrován na stránku zobrazující kurzy ve kterých je v roli učitele a kurzy, ve kterých je v roli studenta.

### 8.1.5 Zakládání nového kurzu a pozvání studentů

Nový kurz může přihlášený uživatel vytvořit kliknutím na tlačítko Přidat kurz (Add Course). Kurzu je potřeba přiřadit název a krátký popis.

Uživatel, který kurz založil, automaticky získává v kurzu roli učitele. Po vytvoření je přesměrován na domácí stránku kurzu, která obsahuje seznam učitelů, seznam studentů, nadcházející události a především seznam lekcí kurzu.

Po založení kurzu na řadu přichází rozeslání pozvánek studentům. Kliknutím na tlačítko Pozvat studenty (Invite students) se učiteli zobrazí stránka, kde může postupně zadávat e-mailové adresy, které získal od studentů. Každý takto pozvaný student obdrží e-mail s pozvánkou do kurzu. Pokud je již v systému zaregistrován, stačí na domovské obrazovce akceptovat pozvánku. V opačném případě musí nejdříve projít procesem registrace (musí jako uživatelské jméno použít e-mailovou adresu, na kterou přišla pozvánka) a až poté je mu rovněž na domovské stránce zobrazena pozvánka.

### 8.1.6 Seznam lekcí a detail lekce

Domovská stránka kurzu mimo jiné obsahuje především seznam všech lekcí seřazený tak, aby nejnovější lekce byly vypsány jako první. Při otevření stránky jsou všechny lekce kromě nejnovější zobrazeny v úsporné podobě, lekci lze

“rozbalit” kliknutím na její titulek. Když je rozbalena, jsou zobrazeny základní informace jako titulek lekce, obsah lekce a odkaz na zobrazení detailu lekce.

Detail lekce zobrazuje kompletní data o lekci včetně přiložených souborů, komentářů studentů i učitelů a formuláře pro přidání nového komentáře.

Uživatel v roli učitele může navíc upravit atributy kurzu, přidávat a měnit obsah lekcí. Při přidávání lekce může učitel formátovat obsah pomocí systému Texy[7], který umožňuje přidávat nadpisy několika úrovní, obrázky, hypertextové odkazy, číslované i nečíslované seznamy, formátované zdrojové kódy nebo emotikony a další. Více informací o syntaxi viz. odkaz výše. Kdykoliv během vytváření obsahu lze kliknout na tlačítko Náhled (Preview) a nechat si zobrazit obsah zformátovaný tak, jak bude ve výsledku vypadat.

K jednotlivým lekcím může učitel také nahrávat soubory, které chce studentům poskytnout ke stažení, především studijní materiály. Pomocí tlačítek + a - nejdříve zvolí počet souborů a poté je vybere ze svého počítače. Po odeslání formuláře jsou soubory nahrány.

Poslední výsadou role učitele je možnost učinit kohokoliv ze studentů učitelem. Pokud je tedy v praxi potřeba více učitelů na jeden kurz, musí jeden z nich kurz vytvořit a ostatní nejdříve pozvat jako studenty a následně je povýšit do role učitele.

### 8.1.7 Výsledky (Results)

Stránka s výsledky zobrazuje studentům jejich průběžné výsledky. Sjednocuje výsledky klasických (offline) testů/úkolů, které učitel zadá ručně, dále online úkolů, které učitel vyvěsí v systému a nechá uživatele odevzdat svá řešení, která následně opraví a nakonec také online úkoly, které se dokáží podle zadaných správných odpovědí opravit samy.

Výsledky jsou rozděleny podle toho, zda je jejich ohodnocením počet bodů nebo známka na stupnici 1-5. Podle toho je napravo tabulky také vypsán buď součet bodů nebo aritmetický průměr známek.

Učitel zde může přidávat klasický (offline) úkol, kterému přiřadí jméno, typ (body/znamky) a dále rovnou ohodnotí všechny studenty v kurzu.

### 8.1.8 Úkoly/Testy (Assignments)

Jednou ze stěžejních částí aplikace je možnost zadávat studentům online domácí úkoly nebo testy. Učitel může jednoduše vytvořit úkol, nastavit mu datum, kdy se úkol stane přístupným pro řešení, datum, do kdy bude přístupný pro řešení a volitelně může také nastavit, jak dlouho může student strávit vyplňováním úkolu (tím se z úkolu v podstatě stává test).

Po vyplnění údajů o úkolu nastává fáze tvoření formuláře pro zadávání odpovědí. Učitel má k dispozici několik možností, jak od studentů získat odpověď.

- Krátký text (Text input) - umožňuje zadat krátký text jako odpověď na otázku
- Dlouhý text (Text area) - umožňuje zadat komplexnější odpověď na otázku
- Jednoduchý výběr z možností (Radio list) - umožňuje vybrat jednu z předem nabídnutých odpovědí
- Výběr podmnožiny z možností (Multiselect list) - lze vybrat libovolnou podmnožinu z předem nabídnutých odpovědí
- Nahrání souboru (File upload) - lze nahrát jakýkoliv soubor s řešením

Úkol může představovat libovolnou kombinaci těchto prvků. Každému prvku je potřeba nastavit popis - nejčastěji otázky, kterou má student zodpovědět. Učitel může strukturu úkolu kdykoliv změnit.

Pokud je již úkol „otevřený“, mohou ho studenti začít kdykoliv řešit až do uzávěrky. V případě, že učitel nastavil maximální dobu řešení, je student s touto skutečností obeznámen a jakmile stiskne tlačítko Řešit (Solve), začíná tato lhůta běžet. Aplikace si zapamatuje, kdy s řešením začal (pro případ zavření okna prohlížeče, odhlášení atd. ) a po daném časovém úseku již není možné úkol řešit. Pokud se stane, že uživatel řeší úkol v těsné blízkosti uzávěrky, zobrazí se mu na stránce odpočet času. Po uplynutí lhůty bude formulář automaticky odeslán v aktuálním stavu.

Učitel může kdykoliv začít opravovat řešení studentů, kteří ho již odevzdali. Při opravování se vytvoří tabulka, kde sloupce reprezentují otázky v úkolu a řádky odpovědi jednotlivých studentů. Na konci každého řádku je místo na vložení bodového ohodnocení. Výsledky se automaticky zapíší do tabulky Výsledky.

Pokud to úkol umožňuje, může ho učitel vytvořit jako samoopravný (Auto-correct). To znamená, že při vytváření struktury nastaví pro každou otázku i správnou odpověď, nastaví pro úkol maximální možný počet bodů a aplikace podle odpovědi studenta sama spočítá kolika procent, potažmo bodů získává.

Kvůli automatickému vyhodnocování úkolů není možné nabídnout při vytváření úkolu všechny komponenty jako v klasickém online úkolu. Do samoopravného úkolu nelze přidat komponentu pro dlouhý text a nahrávání souborů.



### 8.1.9 Zdroje/Dokumenty

Tato sekce sdružuje materiály poskytnuté studentům ke stažení. Nejčastěji se jedná o různé studijní materiály. Tyto dokumenty však nejsou vázány na konkrétní lekci, nýbrž pouze na kurz. Učitel soubory nahrává stejným způsobem jako při nahrávání souborů k lekci.

### 8.1.10 Fórum

Každý kurz má k dispozici jednoduché diskuzní fórum, které má za cíl usnadnit komunikaci mezi studenty, popřípadě lektory. Na rozdíl od soukromé komunikace mezi studenty je komunikace v diskuzním fóru dostupná všem ostatním studentům. Tudíž může například řešení nějakého problému posloužit i ostatním. Fórum se skládá z Témat (Topics), která jsou vypsána na výchozí stránce a uspořádána dle data poslední reakce na téma. Uživatel vytváří téma, které symbolizuje vlákno, ve kterém probíhá diskuze. Po kliknutí na téma se zobrazí odpovědi (replies) na téma seřazené dle data přidání. Zde mohou uživatelé kurzu prostřednictvím jednoduchého formuláře na přidávání odpovědí na dané téma diskutovat.

### 8.1.11 Události

Učitelé mohou ke kurzu přidat libovolnou událost, která je definovaná názvem, krátkým popisem a datem. Události naplánované na nadcházejících 10 dní jsou zobrazeny na hlavní stránce kurzu, ostatní události lze nalézt na stránce události. Zde jsou zobrazeny v kalendáři, který má několik možností zobrazení. Lze ho zobrazit jako denní kalendář, týdenní nebo měsíční. Kliknutí na událost v kalendáři vyvolá zobrazení detailu události.

Učitel může události jednoduše pomocí formuláře přidávat, popřípadě existující v jejich detailu měnit.

### 8.1.12 Zprávy

Aby bylo možné komunikovat s ostatními účastníky kurzu také soukromě, poskytuje aplikace systém zasílání soukromých zpráv. Hlavní obrazovky zpráv může uživatel zobrazit kliknutím na ikonu zprávy v horním fixním menu. Zobrazí se seznam příchozích zpráv.

Kliknutím na tlačítko Nová zpráva (New message) lze vytvořit a následně odeslat novou zprávu.

Každá zpráva je identifikována odesílatelem, příjemcem (oba ve formě uživatelského jména - e-mailové adresy), předmětem a obsahem zprávy - podobně

jako u e-mailové komunikace.

Na zprávy lze jednoduše odpovídat tak, že je původní zpráva v odpovědi citována.

### 8.1.13 Uživatelský profil

Uživatelský profil je osobní profilová stránka, která shrnuje data o uživateli - jeho jméno, příjmení, e-mailovou adresu nebo adresu webové stránky. Uživatel má právo svůj profil kdykoliv upravit.

### 8.1.14 Oznámení e-mailem

Aplikace umožňuje automatické rozesílání e-mailových oznámení v předem daný čas. Například po přidání online úkolu v modulu Úkoly (Assignments) jsou automaticky obesláni všichni studenti kurzu a jsou obeznámeni s novým úkolem. Dále si každý uživatel může v Nastavení zvolit jak dlouho před uzávěrkou úkolu má být na uzávěrku upozorněn.

### 8.1.15 Nastavení

Tato stránka umožňuje uživateli měnit své preference. Lze nastavit preferovaný jazyk aplikace, který bude aplikován po každém uživatelském přihlášení, dále to, zda bude pro ostatní uživatele viditelná e-mailová adresa a také lze nastavit kolik dní předem bude uživatel upozorněn na blížící se uzávěrku online úkolu.

## 8.2 Mobilní aplikace

### 8.2.1 Minimální požadavky

Aplikace je určena a vyvíjena pro mobilní zařízení se systémem Android[1] ve verzi 2.1 (Eclair), avšak pravděpodobně bude z velké části fungovat i na starších zařízeních se starší verzí systému. Testování proběhlo na mobilních telefonech se systémem verze 2.1, 2.2, 2.3 a 4.0. Aplikace vyžaduje připojení zařízení k internetu.

### 8.2.2 Instalace a spuštění aplikace

Aplikaci lze nainstalovat na zařízení standardním způsobem pomocí instalačního balíčku APK<sup>1</sup>. Soubor s příponou .apk je potřeba nakopírovat do zařízení a tam spustit. Proběhne standardní instalace aplikace a zástupce s titulkem *CourseManager* se objeví v seznamu nainstalovaných aplikací. Kliknutím na zástupce aplikace se aplikace spustí a zobrazí se výchozí obrazovka.

### 8.2.3 Výchozí obrazovka a přihlášení do aplikace

Po prvním spuštění je pro správné fungování potřeba aplikaci nakonfigurovat. Stisknutím standardního tlačítka MENU se zobrazí nabídka, ve které lze vyvolat obrazovku Nastavení. V nastavení je nutné nejdříve nastavit adresu běžící instance webové aplikace CourseManager - tedy například `http://rp-jakub-cz.dc.starver.net`. Dále musí uživatel vyplnit přihlašovací údaje - e-mailovou adresu a heslo.

Po návratu na výchozí obrazovku by už aplikace měla být úspěšně připojena na webový protějšek a zobrazovat seznam uživatelských kurzů - nejdříve kurzy, které učí a poté kurzy, které studuje. Po kliknutí na jeden z kurzů se zobrazí jeho detailní stránka s dalšími možnostmi navigace. V nabídce je v tuto chvíli kromě nastavení ještě přístup k soukromým zprávám uživatele.

### 8.2.4 Detail kurzu

Stránka kurzu nabízí stejně jako webová verze seznam jednotlivých lekcí, které jsou seřazeny podle jejich data. Po rozbalení lekce a kliknutí na tlačítko *Více* bude zobrazen detail této konkrétní lekce.

Důležitá je ale v tuto chvíli nabídka, která obsahuje odkazy na všechny moduly týkající se tohoto kurzu. Uživatel si může nechat zobrazit Diskuzní fórum kurzu, Události kurzu, Výsledky studentů, Úkoly (Assignments) a Zdroje (Resources).

### 8.2.5 Detail lekce

Detailní stránka konkrétní lekce je rozdělena pomocí záložek na dvě části. V první záložce se nacházejí komentáře k lekci. Uživatelé zde mohou lekci libovolně komentovat. Nový komentář se přidá pomocí položky v nabídce menu. Ve druhé záložce jsou vypsány soubory, které učitel k lekci přidal. Po kliknutí na položku je soubor stažen do zařízení a pokud je to možné, je otevřen v příslušné aplikaci. Stažené soubory se ukládají do složky `CourseManager_downloads`.

---

<sup>1</sup>APK, application package file

### 8.2.6 Diskuzní fórum

Mobilní aplikace nabízí kompletní možnost diskuze ve fóru - tedy zakládání nových témat diskuze, výpis odpovědí a přidávání nových odpovědí.

### 8.2.7 Události

Obrazovka událostí nabízí prostý výpis nejbližších událostí kurzu. Při dlouhém kliknutí na položku události se zobrazí kontextová nabídka s možností vložení události do kalendáře v zařízení. Otevře se dialogové okno s předvyplněnými detaily události a po potvrzení se událost uloží do výchozího kalendáře ve výchozí aplikaci Kalendář systému Android.

### 8.2.8 Výsledky

Na obrazovce Výsledky je k dispozici přehledná tabulka, kde sloupce představují jednotlivé úkoly/testy a řádky odpovídají konkrétním studentům kurzu. Tabulka je rozdělena na dvě části - úkoly/testy ohodnocené body a úkoly/testy ohodnocené známkou. Na konci tabulky je příslušnému studentovi vypočítán součet bodů, respektive aritmetický průměr známek.

### 8.2.9 Úkoly/Testy (Assignments)

Výchozí obrazovka Úkoly vypisuje všechny úkoly/testy kurzu pod sebou. Při kliknutí na položku se zobrazí detail, kde jsou možné dvě akce v závislosti na roli uživatele.

Pokud je uživatel v roli studenta, může úkol/test vyřešit přímo na mobilním zařízení. Aplikace podporuje klasické i samoopravné testy, s časovým omezením i bez něho. Řešení probíhá stejně jako ve webové verzi - pomocí standardních formulářových prvků zadává řešitel své odpovědi a následně řešení odešle příslušným tlačítkem. V horní části obrazovky je zobrazen zbývajících čas pro řešení testu.

Pokud je uživatel v roli učitele, nemůže úkol/test řešit, ale může opravovat již odevzdaná řešení. Pomocí tabulky jsou mu zobrazeny jednotlivé odpovědi studentů a vzápětí pole pro vložení počtu získaných bodů.

### 8.2.10 Zdroje (Resources)

Tento modul nabízí výpis souborů, které nejsou přiřazeny ke konkrétní lekci, ale k celému kurzu. Stejně jako u souborů lekce stačí kliknout na položku a soubor se stáhne/otevře.

### 8.2.11 Zprávy

Mobilní verze, stejně jako ta webová, umožňuje kompletní komunikaci mezi uživateli aplikace pomocí systému soukromých zpráv. Modul Zprávy je rozdělen do dvou záložek - Doručené zprávy a Odeslané zprávy. Pomocí menu je možné vytvořit a následně odeslat zprávu novou. Samozřejmě lze i na příchozí zprávy jednoduše odpovídat.

## Kapitola 9

### Závěr

## Kapitola 10

## Seznam příloh

# Literatura

- [1] Android <sup>TM</sup>. <http://android.com>.
- [2] Apache. <http://apache.org>.
- [3] Mysql. <http://mysql.com>.
- [4] Nette framework. <http://nette.org>.
- [5] Php. <http://php.net>.
- [6] Soap. <http://www.w3schools.com/soap/>.
- [7] Taxy. <http://taxy.info/cs>.
- [8] Rfc 2109, http state management mechanism, 1997. <http://www.ietf.org/rfc/rfc2109.txt>.
- [9] Rfc 2818, http over tls, 2000. <http://tools.ietf.org/html/rfc2818>.
- [10] Gail Rahn Frederick. Html5 and mobile web, 2010. <http://learnthemobileweb.com/2010/06/html5-and-mobile-web/>.
- [11] Martin Malá". Rest: architektura pro webové API, 2009. <http://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/>.