

Virtuálna učebnica

Tímový projekt

Vedúci tímového projektu: Ing. Jamal Hasan

Členovia tímu:

Bc. Matej Mayer

Bc. Juraj Hrubša

Bc. Andrej Letkovský

Bc. Peter Kalanin

Bc. Miroslav Mika

Bc. Matej Fajnor

OBSAH

0	Úvod	1
0.1	Zadanie.....	1
0.2	Ciele projektu.....	2
0.3	Skratky a pojmy.....	2
0.4	Zoznam obrázkov.....	3
1	Analýza	4
1.1	E-learning.....	4
1.1.1	Porovnanie e-learningu s klasickým spôsobom vzdelávania.....	4
1.2	Frameworky.....	5
1.2.1	Wikipédia.....	5
1.2.2	Moodle.....	7
1.3	Vlastné riešenie.....	8
1.3.1	HTML.....	9
1.3.2	CSS.....	9
1.4	Riešenie postavené na gridovom systéme vs. riešenie postavené na jednom serveri, prípadne rozloženie záťaže na viacero serverov.....	10
1.4.1	Čo je to Grid?.....	10
1.4.2	Stavebné bloky Grid Computingu.....	11
	Zjednotené infraštruktúry (Common infrastructure): štandardy.....	12
1.4.3	Riešenie postavené na gridovom systéme.....	12
1.4.4	Riešenie s rozložením záťaže.....	14
2	Špecifikácia riešenia	16
2.1	Požiadavky.....	16
2.1.1	Požiadavky na projekt.....	16
2.1.2	Požiadavky na systém.....	16
2.1.3	Funkcionálne požiadavky.....	16
3	Hrubý návrh	27
4	Riadenie projektu	29
4.1	Ponuka.....	30
4.1.1	Predstavenie členov tímu.....	30
4.1.2	Motivácia.....	33
4.1.3	Návrh riešenia.....	34
4.1.4	Vlastné riešenie.....	35
4.1.5	Predpokladané zdroje.....	37
4.1.6	Zoradenie ponúkaných tém podľa priority.....	38
4.1.7	Rozvrh.....	39
4.2	Rozdelenie úloh.....	40
4.2.1	Dlhodobé úlohy.....	40
4.2.2	Krátkodobé úlohy.....	41
4.2.3	Časový harmonogram.....	42
4.3	Zápisnice zo stretnutí.....	44
5	Prototyp	60
5.1	Ciele prototypu.....	60
5.2	Softvérový prototyp.....	60
5.3	Použitie implementačné prostredie.....	61
5.4	Funkcie prototypu.....	61

5.5	Testovanie prototypu.....	62
5.6	Zhodnotenie.....	64
6	Použitá literatúra.....	65
	Príloha A: Používateľská príručka prototypu.....	66
	Novinky.....	69
	Obsah.....	69
	<i>Dokumenty</i>	69
	<i>Multimediálny obsah</i>	70
	<i>Odkazy</i>	70
	Bannery.....	70
	Fóra.....	70
	Ankety.....	71
	Formuláre.....	71

o Úvod

V súčasnosti sa počítač stáva samozrejmosťou pre každého jedinca a je súčasťou väčšiny domácností, firiem a organizácií. Oblasť IT technológií je rýchlo vyvíjajúcou sa oblasťou, ktorá oplýva množstvom technológií a denne so sebou prináša množstvo noviniek, ktoré viac či menej uľahčujú náš život, spriemňujú voľné chvíle, prinášajú nám možnosti vyriešenia našich pracovných problémov rýchlejšie a efektívnejšie.

o.1 Zadanie

Analyzujte požiadavky virtuálnej vzdelávania pre potreby pedagogiky. Zamerajte sa najmä na organizáciu predmetov a odborných tematických oblastí v zmysle multimedialného poskytovania informácií, študijných materiálov a mechanizmov testovania nadobudnutých znalostí.

Využitie multimedialných technológií pri tvorbe učebných pomôcok môže významným spôsobom zlepšiť ich obsahovú aktuálnosť a cenovú dostupnosť, a čo je najpodstatnejšie, zredukujú sa časové nároky na ich inováciu a výrobu. Vzhľadom na zodpovedajúcu obsahovú náplň aktuálnej témy je možné kedykoľvek vymeniť náplň (texty, obrázky, fotografie, animácie, videosekvencie, akustické efekty, testy, a pod.) jednotlivých častí, upraviť ich rozsah a tak vytvárať virtuálnu učebnú pomôcku, ktorá bude zodpovedať aktuálnemu stavu poznania.

Cieľom projektu je podpora uvedených činností pri príprave multimedialnej internetovej učebnice z príslušnej oblasti. V rámci riešenia bude potrebné:

- Analyzovať a posúdiť dostupné voľne šíriteľné nástroje pre poskytovanie vedomostí, výučbu a vzdelávacích programov elektronickou formou, oboznámiť sa s hlavnými zásadami tvorby učebníc pre dištančné vzdelávanie.
- Naštudovať základy teórie príslušnej tematickej oblasti.
- Navrhnuť architektúru informačného systému, ktorý bude prostredníctvom webového rozhrania poskytovať potrebné informácie pre študentov, pedagógov, ako aj správcu systému, vrátane kontrolných príkladov, kontrolných otázok a testov.
- Návrh overiť implementáciou vybraných funkcií (kapitol virtuálnej učebnice) tohto systému.

0.2 Ciele projektu

Cieľom projektu je podpora uvedených činností pri príprave multimedialnej internetovej učebnice z príslušnej oblasti.

V rámci riešenia bude potrebné:

- Analyzovať a posúdiť dostupné voľne šíriteľné nástroje pre poskytovanie vedomostí, výučbu a vzdelávacích programov elektronickou formou, oboznámiť sa s hlavnými zásadami tvorby učebníc pre dištančné vzdelávanie.
- Naštudovať základy teórie príslušnej tematickej oblasti.
- Navrhnuť architektúru informačného systému, ktorý bude prostredníctvom webového rozhrania poskytovať potrebné informácie pre študentov, pedagógov, ako aj správcu systému, vrátane kontrolných príkladov, kontrolných otázok a testov.
- Návrh overiť implementáciou vybraných funkcií (kapitol virtuálnej učebnice) tohto systému.

Tento projekt má pomáhať študentom na z kvalitnenie a zlepšenie e-learningového štúdia. Pevne dúfame, že tento projekt bude osožný pre veľké množstvo študentov, ako aj pedagógov, a že nezapadne do zabudnutia.

0.3 Skratky a pojmy

HTML – „hypertext markup language“ jazyk na tvorbu internetových stránok

CSS – „cascade style sheeting“ jazyk na popis zobrazenia internetových stránok napísaných v HTML, XHTML alebo XML.

E-learning – učenie pomocou počítača, využitia počítačovej siete alebo iných interaktívnych metód

Webová stránka – internetová stránka, podporovaná webovými prehliadačmi

CMS systém – Content management systém

URL - Uniform Resource Locator

WSIL - Web Services in Learning

DNS - Domain Name system

IP - Internet Protocol

0.4 Zoznam obrázkov

Obr. 1 Printscreen užívateľského rozhrania Wiki

Obr. 2 Printscreen užívateľského rozhrania Moodle

Obr. 3 Architektúra gridového systému

Obr. 4 Architektúra systému s rozložením záťaže

Obr. 5 Model prípadov použitia k správe systému

Obr. 6 Model prípadov použitia k vytváraniu obsahu.

Obr. 7 Model prípadov použitia k prezeraniu obsahu

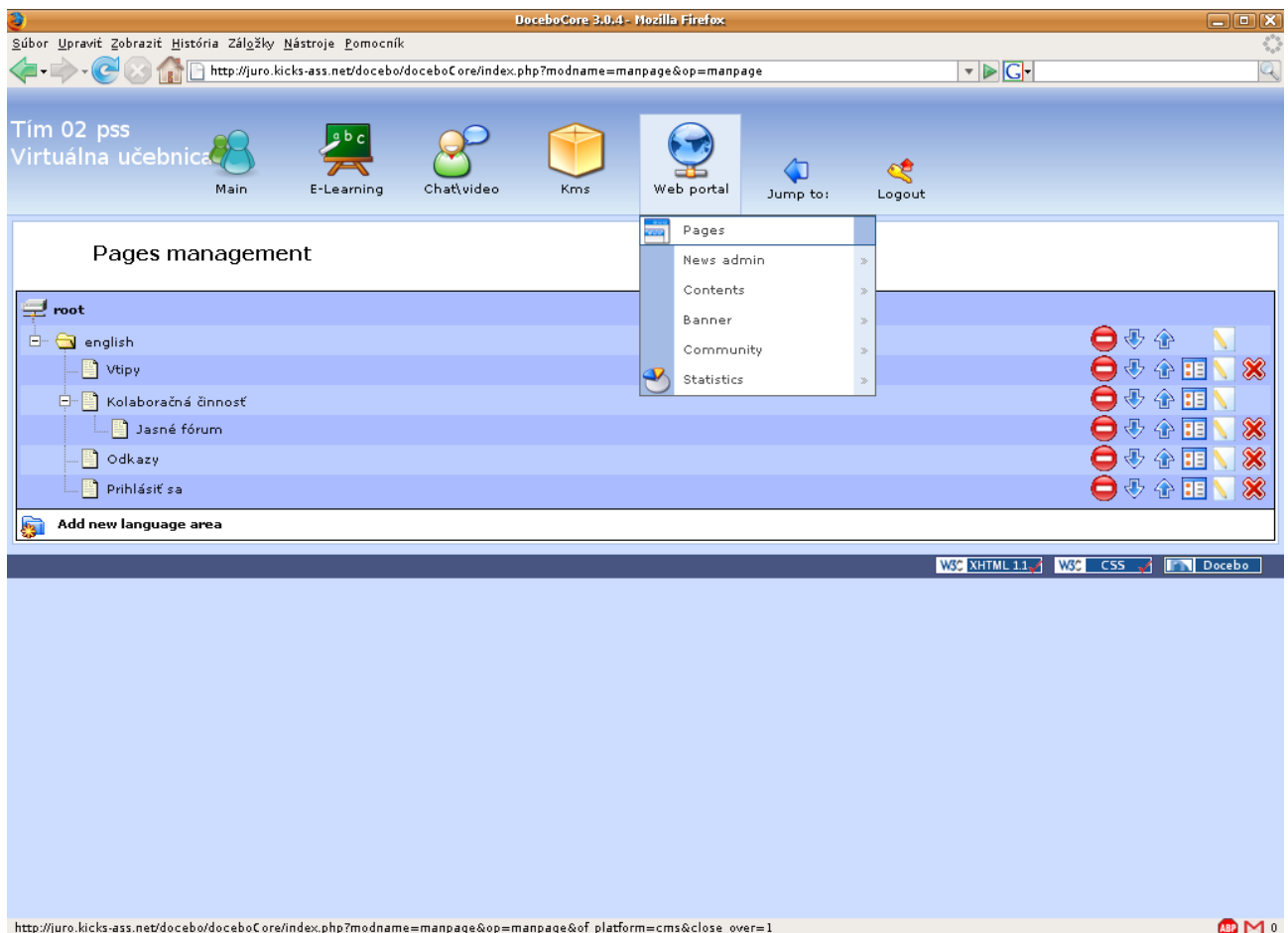
Obr. 8 Model prípadov použitia ku kalendáru

Obr. 9 Model prípadov použitia k fóru

Obr. 10 Model prípadov použitia k slovníku

Obr. 11 Model prípadov použitia k vyhľadávaniu

Obr. 12 Model prípadov použitia k testom



Obr. 13 Rozhranie pre správu štruktúry CMS

Obr. 14 Rozhranie pre import/export jazykov

1 Analýza

1.1 E-learning

V dnešnej dobe je veľmi dôležité vzdelávanie. Nároky na mladých aj starších ľudí sú stále vyššie a preto sa aj vzdelávanie stáva stále dôležitejšie. Keďže ľudia sú od prírody stvorení, hľadajúce si najjednoduchšiu cestu k cieľu, ktorý chcú dosiahnuť, kladie sa dôraz aj na to, aby vzdelávanie bolo čo najefektívnejšie, aby bolo dostupné a rýchle. Keďže je doba internetu, do popredia sa dostáva vzdelávanie práve prostredníctvom internetu, ako rýchle, pohodlné, efektívne a s ešte mnohými inými kladmi, ktoré jeho popularitu len zvyšujú. Vzdelávanie prostredníctvom internetu, alebo rôznych iných interaktívnych systémov, založených na použití počítača alebo zariadení počítača podobných môžeme združiť do pojmu „e-learning“.

E-learning má mnoho definícií, niekoľko z nich som vybral, ktoré tento dôležitý pojem vystihujú najlepšie.

E-learning je :

- Učenie založené na elektronickom formáte
- Proces, ktorý uľahčuje vzdelávanie za použitia internetu
- Vzdelávanie uskutočňované prostredníctvom internetu, počítačovej siete alebo CD-ROMu
- Technologicky sprostredkované učenie za použitia počítača
- Softvér vytvorený na učenie používateľa novým znalostiam, odovzdaný za použitia web technológie
- Poskytovanie učenia, tréningu alebo výučby elektronickou formou. Vyžaduje si použitie počítača, alebo iného elektronického média za účelom poskytnutia tréningu, vzdelania alebo výučbového materiálu.

1.1.1 Porovnanie e-learningu s klasickým spôsobom vzdelávania

Keď porovnáme e-learning s klasickou formou vzdelávania, zistíme, že e-learning má veľa výhod oproti klasickej forme

- V rôznych oblastiach sveta, kde urbanizácia a infraštruktúra ešte nie je na najvyššej úrovni sa ľudia denno-denne stretávajú s problémom vzdialenosti. Denne musia prekonávať veľké vzdialenosti, aby sa dostali na čas do školy alebo iného vzdelávacieho zariadenia. S e-learningom tento problém odpadá. Študent šetrí svoj čas aj finančné prostriedky tým, že nemusí za vzdelaním cestovať. Samozrejme sa vynára nový problém a to dostupnosť internetu, ktorá sa ale v dnešnej dobe stále len zvyšuje.
- Ďalším rozdielom je to, že od študenta sa vyžaduje vyššia samostatnosť v porovnaní s klasickou formou, lebo v podstate nie je nútený sa učiť. Ide len o jeho záujem. Učiteľ je nahradený počítačom a všetkými prostriedkami, ktoré počítač poskytuje. V prvom momente si môžeme myslieť, že to je skôr nevýhoda ako výhoda, ale ten, kto sa učiť chce, tak sa učiť bude a pritom sa naučí aj samostatnosti.
- Klasické knihy sú nahradené nespočetným množstvom elektronických informácií, ktoré si užívateľ musí vedieť efektívne vyhľadať (napr. prostredníctvom vyhľadávačov Google atď.). Informácií je podstatne viac a dajú sa aj rýchlejšie vyhľadať a spracovať, čiže e-learning šetrí aj čas.

1.2 Frameworky

Existuje mnoho systémov, ktoré sú zamerané na e-learning, teda poskytujú určitú formu vzdelávania a interakcie. Niektoré z nich analyzujeme podrobnejšie. Bude sa jednať skôr o známe systémy, ktoré si už vyslúžili úspech tým, že sú využívané miliónmi ľudí na celom svete.

1.2.1 Wikipédia

Wikipédia je na webe založená encyklopédia s otvoreným obsahom, ktorú možno voľne upravovať a slobodne čítať. Je sponzorovaná neziskovou organizáciou Wikimedia Foundation. Má 195 nezávislých jazykových vydaní, z ktorých najrozsiahlejšie je anglické. Wikipédia obsahu články encyklopedického typu zo všetkých oblastí života, či už je to veda, šport, história, IT technológie ale aj stránky

¹ [3] www.google.sk, define: "e-learning"

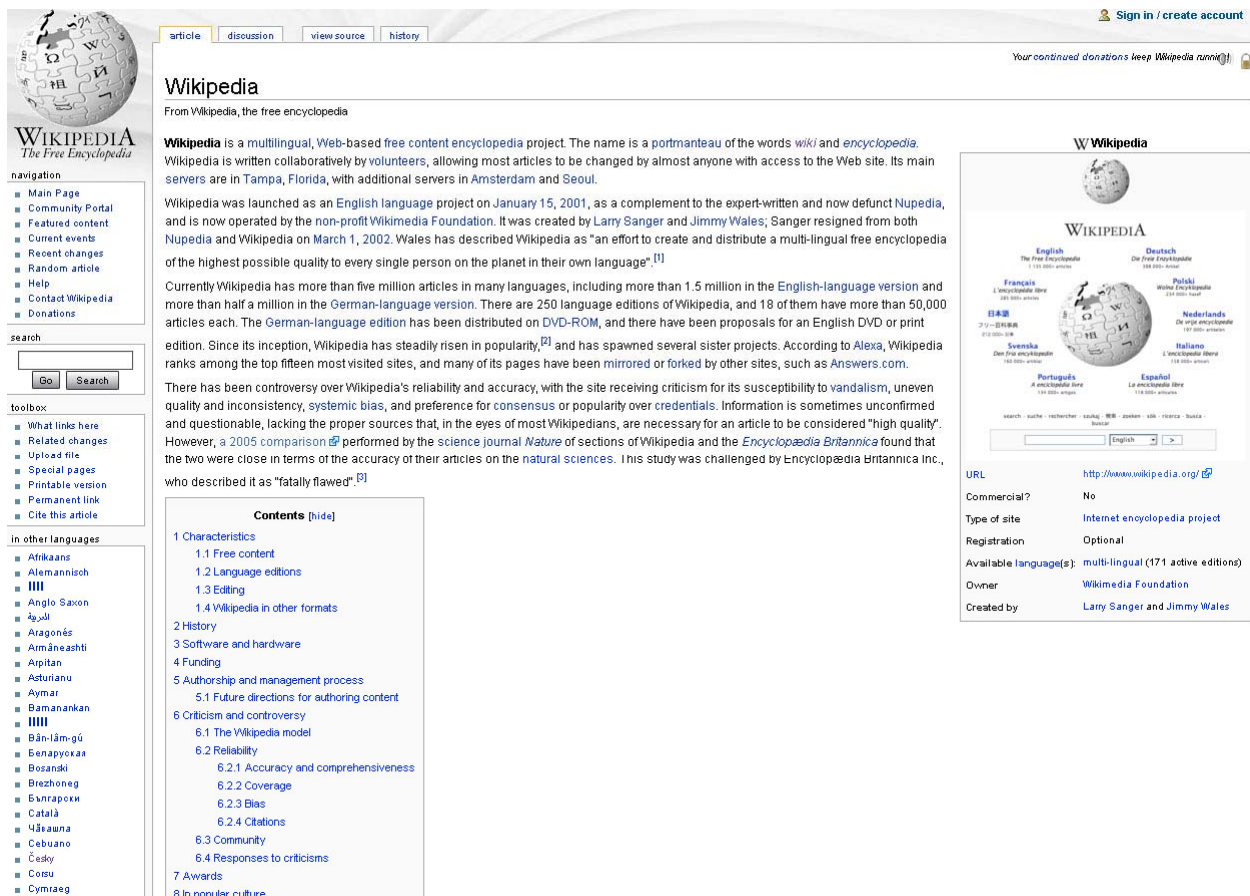
aktuálnych udalostí, či rôzne atlasy. Wikipédia je jedným z najpopulárnejších zdrojov informácií na webe s približne 80 miliónmi zobrazeniami denne.

Wikipédia obsahuje približne 1,5 milióna článkov, z ktorých viac ako 1 000 000 je v angličtine, viac ako 380 000 v nemčine a viac ako 200 000 v japončine a francúzštine. Začínala 15. januára 2001 ako doplnok k expertmi písanej Nupedii. So stále rastúcou popularitou sa Wikipédia stala základom pre sesterské projekty ako Wiktionary (Wikislovník), Wikibooks (Wikiknihy) a Wikinews (Wikisprávy). Jej články sú upravované dobrovoľníkmi vo wiki štýle, čo znamená, že články môže meniť v podstate každý užívateľ. Základnou politikou wikipédie je, že presadzuje politiku „neustranný uhol pohľadu“. Vzhľadom na to, že Wikipédia presadzuje otvorenú filozofiu, jej najväčším problémom je vandalizmus a nepresnosť.

Cieľom Wikipédie je vytvoriť voľne dostupnú, spoľahlivú a najväčšiu encyklopédiu, pokiaľ ide o rozsah a dôkladnosť. Wikipédia je svojím zakladateľom Jimmym Walesom charakterizovaná ako „snaha o vytvorenie a distribuovanie voľne dostupnej encyklopédie každej osobe na planéte v jej rodnom jazyku a to v čo možno najlepšej kvalite.“ Je vytvorená na stránke wikipedia.org použitím softvéru zvaným „wiki“ (z havajského slova wiki wiki – rýchlo).

Wikipédia beží na programe zvanom MediaWiki, čo je program s otvoreným zdrojom. Server je situovaný na Floride v USA. MediaWiki je tretia fáza programového softvéru Wikipédie. Pôvodne Wikipédia bežala na programe zvanom UseModWiki od Clifforda Adamsa (fáza 1). V januári 2002 Wikipédia začala bežať na PHP wiki s MySQL databázou. Tento softvér (fáza 2) bol zostavený Magnusom Manskem špecificky pre Wikipédiu. Bolo vykonaných niekoľko úprav tohto programu, aby sa vylepšil výkon. Nakoniec bol program prepísaný nanovo, tento krát sa o to postaral Lee Daniel Crocker. Jeho činnosť začala v júli 2002. Táto tretia fáza softvéru sa nazvala MediaWiki. Dostala sa pod licenciu GNU General Public License a je používaná všetkými projektmi Wikipédie.²

² [4] http://en.wikipedia.org/wiki/Main_Page



Obr. 1 Printsreen užívateľského rozhrania Wiki

Užívateľské rozhranie wikipédie je prehľadné, má efektívne vyhľadávanie výrazov zadaných užívateľom a v príspevkoch sú užitočne zlinkované výrazy, ktoré sú v nejakom inom článku spracované podrobnejšie.

1.2.2 Moodle

Moodle je open-source e-learningový portál. Po celom svete je má viac ako 18000 registrovaných stránok, okolo 7 270 000 užívateľov v 712 531 kurzoch.

Je vytvorený na to, aby pomáhal vyučovateľom vytvárať on-line kurzy so širokou interakciou. Tento systém je open-source, to znamená, že ľudia môžu vytvárať prídavné funkcie systému. Moodle má veľa funkcií, ktoré by užívateľ od e-learningového systému očakával. Medzi základné patria fóra, kvízy s rôznymi otázkami, správa obsahu, blogy. Samozrejmosťou systému je multijazyková podpora.

Moodle je multiplatformový systém, teda beží bez problémov na platformách Unix, Linux, FreeBSD, Windows, MAC OS a všetkých ostatných, ktoré podporujú

PHP, na ktorom je Moodle postavený. Dáta sú uložené prostredníctvom MySQL databázy.³

The screenshot displays the Moodle user interface for the 'EdNA Groups Game Maker' course. At the top, the user is logged in as 'Roland Gesthuizen'. The page is divided into several sections:

- People:** Includes links for 'Participants', 'Sub-Groups', and 'Edit profile'.
- Menu:** Lists various course features like 'Chats', 'Choices', 'Forums', 'Glossaries', 'Resources', and 'Wikis'.
- Online Users:** Shows 'Roland Gesthuizen' as the only user online in the last 10 minutes.
- Administration:** Provides links for 'Turn editing on', 'Settings...', 'Owners...', 'Members...', and 'Backun...'.
- Sections:** The main content area contains:
 - Welcome:** A message for teachers interested in teaching computer game design, accompanied by a cartoon character.
 - 1 Background Survey:** A survey titled 'Background Survey' asking for user information, with a question mark icon.
 - 2 Get started:** A section titled 'Get started' with a 'GAME DEVELOPER' badge.
- Calendar:** A calendar for April 2005 showing dates and events.
- Upcoming Events:** Lists an event 'Game Maker for Beginners' on Monday, 18 April, from 04:30 PM to 07:30 PM.
- Latest News Posts:** Shows recent posts by Roland Gesthuizen, including one about 'Gaps in thinking about games'.

Obz. 2 Printsreen užívateľského rozhrania Moodle

1.3 Vlastné riešenie

Analyzované frameworky nám ponúkajú jednu možnosť vytvorenia systému na podporu výučby. Ďalšou možnosťou pre nás by bolo vytvorenie vlastného systému na výučbu, ktorý by poskytoval asi menej možností využitia ako poskytujú vyššie spomenuté frameworky, ale zato by toto riešenie bolo originálnejšie.

Prvým krokom vytvorenia vlastného systému by bolo vytvorenie samotného systému, ktorý by sa neskôr vyplňal dátami. Na vytvorenie takéhoto systému by boli použité jazyky XHTML a CSS, ktoré by zabezpečili vzhľad systému. Jadrom systému by bolo PHP. To by zabezpečilo celú funkcionálnosť systému, s tým, že dáta by boli uložené v databáze, či už MySQL alebo PostgreSQL.

System by bol samozrejme funkčný ako CMS systém

³ [4] <http://en.wikipedia.org/wiki/Moodle> , [6] <http://moodle.org/>

V stručnosti by som ešte popísal vyššie spomenuté nástroje tvorby systému.

1.3.1 HTML

HyperText Markup Language je značkový jazyk určený na vytváranie webových stránok a iných informácií zobraziteľných vo webovom prehliadači. HTML kladie dôraz skôr na prezentáciu informácií (odseky, fonty, váha písma, tabuľky atď.) ako na sémantiku.

Jazyk HTML umožňuje vytvárať dokumenty obsahujúce text, hypertextové odkazy, multimediálny a iný obsah, formuláre, skripty a metainformácie prehliadateľné vo webovom prehliadači.

Jazyk HTML je textový, umožňuje čítanie a upravovanie priamo v textovom editore. Existujú aj špecializované programy na grafické editovanie stránok HTML, napríklad Macromedia Dreamweaver alebo Microsoft FrontPage. Ďalšou možnosťou ako vytvárať stránky v jazyku HTML je dynamická (alebo aktívna) tvorba obsahu programom bežiacim na strane servera, napríklad PHP alebo ASP, ktorý na požiadanie vygeneruje stránku a pošle používateľovi.

V značkách jazyka HTML existujú štyri druhy prvkov:

- Štruktúrové prvky. Označujú zmysel textu. Napríklad

```
<h1>Golf</h1>
```

označuje, že slovo "Golf" je nadpisom prvej úrovne.

- Prezentačné prvky. Popisujú výzor textu, bez ohľadu na jeho zmysel. Napríklad

```
<b>tučné</b>
```

zobrazí slovo "tučné" tučným písmom. Výzor textu by mal byť popísaný pomocou kaskádových štýlov.

- Hyperlinky. Linky do iných dokumentov. Napríklad

```
<a href="http://sk.wikipedia.org/">Wikipédia</a>
```

urobí zo slova Wikipédia linku na dané URL.

- Ovládacie prvky. Vytvárajú tlačidlá, zaškrtačacie políčka, zoznamy, atď. ⁴

1.3.2 CSS

Kaskádové štýly alebo CSS (skratka z angl. Cascading Style Sheets) je všeobecné rozšírenie HTML o možnosti opisu vzhľadu textu. Štýly umožnili oddeliť štruktúru HTML alebo XHTML od vzhľadu.

Kaskádové štýly môžu byť vnesené do HTML tromi spôsobmi:

- parameter každej značky (tagu):

```
<tag style="...">
```

- definícia v záhlaví dokumentu:

```
<style type="text/css">
```

```
</style>
```

- externý súbor s príponou *.css:

```
<link rel="stylesheet" type="text/css" href="..."> 5
```

1.4 Riešenie postavené na gridovom systéme vs. riešenie postavené na jednom serveri, prípadne rozloženie záťaže na viacero serverov

Na podnet vedúceho tohto tímového projektu sme sa snažili navrhnúť riešenie, ktoré by využívalo Gridové prostredie vytvorené na Fakulte Informatiky a Informačných Technológií. Pri takomto riešení je v prvom rade potrebné pochopiť ako Gridové prostredie funguje a aké možnosti poskytuje.

1.4.1 Čo je to Grid?

Technológia Grid Computing môže byť definovaná ako využívanie prostriedkov z mnohých počítačov v sieti - v rovnakom čase - na jeden problém. Zvyčajne je to problém, ktorý vyžaduje veľké množstvo cyklov spracovania alebo prístup k veľkým objemom dát.

V roku 1998, Foster a Kesselman definovali Grid Computing:

Výpočtový Grid je hardwarová a softwarová infraštruktúra, ktorá umožňuje spoľahlivý, konzistentný, zovšadiaľ prístupný a lacný prístup k výkonným výpočtovým možnostiam

Grid musí spĺňať nasledujúce tri body:

- koordinuje zdroje nepodliehajúce centralizovanej správe
- používa štandardné, otvorené, obecné protokoly a rozhrania
- poskytuje netriviálnu kvalitu služieb (viac než jednotlivé časti každá zvlášť)

⁴ [4] http://sk.wikipedia.org/wiki/Hypertext_markup_language

Poznáme viacero druhov Gridov:

- Computational Grid – výpočtový Grid. Využíva sa pre výpočtov náročné aplikácie ako interaktívne simulácie (modelovanie podnebia), rozsiahle simulácie a analýzy (formácie galaxií, simulácia bojových operácií) a v technike.
- Data Grid – dátový Grid. Slúži na skladovanie a prístupňovanie informácií medzi viacerými organizáciami.
- Scavenging Grid – Typicky sa využíva pri veľkom počte užívateľských staníc. Umožňuje užívateľom využívať výpočtové prostriedky (napríklad cykly CPU) iných staníc a poskytovať vlastné nevyužité prostriedky ostatným.

1.4.2 Stavebné bloky Grid Computingu

Gridy možno logicky rozdeliť na komponenty, ktoré sú spoločné pre všetky druhy Gridov. V nasledujúcich riadkoch sa budeme stručne venovať opisu týchto logických častí.

Siete

Srdcom akéhokoľvek Gridu je sieť – sieť spája geograficky distribuované prostriedky a umožňuje tak ich kolektívne použitie pri výkone aplikácie. Ak sieť poskytuje „veľké rúry“ tak vydarené aplikácie môžu využiť distribuované prostriedky v jednotnejšom a dátovo intenzívnejšom štýle. Ak siete ponúkajú „malé rúry“ tak úspešné aplikácie by mali využívať minimálnu komunikáciu a dátový prenos medzi programovými komponentmi by mal byť schopný tolerovať vysokú latenciu.

Vysokokapacitné sieťovanie zvyšuje schopnosti Gridu podporovať súčasne paralelné a distribuované aplikácie. V budúcnosti budú káblové siete rozšírené o stále sa rozvíjajúce bezdrôtové spojenia, ktoré umožnia integrovať menšie zariadenia do Gridov.

Výpočtové uzly

Siete prepájajú prostriedky v Gridoch, najbežnejšími sú počítače a ich dátové úložisko. I keď výpočtový prostriedok môže mať rôznu úroveň výkonu a schopností, niektoré zaujímavé Gridy pre vedecký výskum zahrňujú uzly, ktoré sú samy o sebe vysoko-výkonné paralelné stroje alebo clustery. Takéto vysoko-výkonné Grid uzly

⁵ [4] http://sk.wikipedia.org/wiki/Kaskádové_štýly

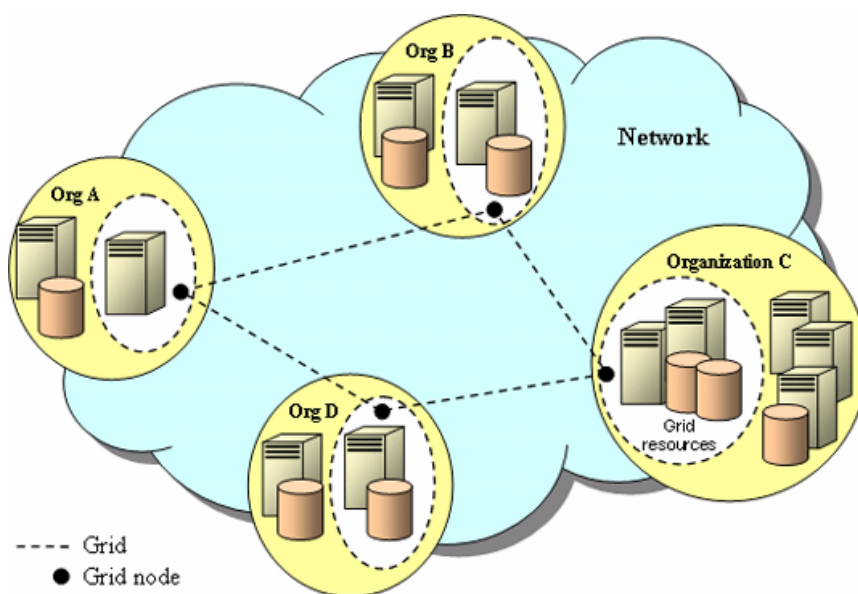
poskytujú hlavné prostriedky pre simulácie, analýzy, data mining a iné počítačovo intenzívne aktivity.

Zjednotené infraštruktúry (Common infrastructure): štandardy

Do budúcnosti budú technológie zlepšovať svoj potenciál a kapacitu a bude ich potrebné integrovať do Grid technológií. Pre správnu organizáciu stále meniacich sa technológií Gridy využívajú zjednotenú infraštruktúru (common infrastructure) pre dosiahnutie virtuálnej reprezentácie pre softwarových vývojárov a používateľov spoločne s možnosťou včlenenia nových technológií. Tvorba kľúčových štandardov, ktoré umožňujú vytváranie zložitých Grid je dôležitá pre dosiahnutie úspechu Gridov.

1.4.3 Riešenie postavené na gridovom systéme

Aby sme zistili aké sú možnosti využitia gridového prostredia pre náš projekt, konzultovali sme túto tému s Ing. Michalom Zimenom, ktorý sa zaoberá problematikou gridových aplikácií. Ing. Zimen navrhol riešenie s nasledovnou architektúrou.



Obr. 3 Architektúra gridového systému

V tomto prípade by celý systém tvorilo viacero uzlov s rôznym obsahom, ktoré by svoje údaje poskytovali užívateľom. Uzly by boli tvorené servermi, ktoré by mali každý vlastnú databázu článkov a iných informácií. Vyhľadávanie informácie požadovanej užívateľom a jej sprístupnenie danému používateľovi by mala na starosti samotná gridová aplikácia. Na vytvorenie takejto aplikácie je potrebné využitie technológie Web Services.

Web Services je technológia pre vzdialené volanie funkcií v distribuovaných systémoch pomocou prenosu správ v jazyku XML protokolom HTTP. Hlavnou myšlienkou Web Services je, aby sa www zo súčasného súboru HTML stránok, zrozumiteľných iba ľuďom, zmenilo na súbor XML stránok, ktoré sú čitateľné pre programy. Programy na rôznych platformách (JavaScript, Java, C, MS .NET) by tak mohli navzájom jednoducho komunikovať. Samotná technológia sa skladá z troch častí:

- protokol pre vzdialené volanie procedúr, zvaný SOAP, prenášajúci údaje zapísané ako XML
- jazyk pre popis poskytovaných služieb, zvaný WSDL
- mechanizmy pre nájdenie služieb, spolu nazývané UDDI a WSIL

Systém, vytvorený s pomocou technológie Web Services by potom fungoval nasledovne. Na každom uzle by fungovala rovnaká aplikácia, pozostávajúca z troch častí:

- používateľské rozhranie
- webová služba
- databáza

Používateľ si do svojho internetového prehliadača zadá adresu našej aplikácie a dostane stránku, ktorá bude identická pre každého používateľa kdekoľvek na svete. V skutočnosti by však mohol túto stránku získať z ktoréhokoľvek z viacerých web serverov, na ktorých by aplikácia bežala. Tento výber by sa prenechal na systém DNS, ktorý by odkazoval na IP adresy na základe geografického regiónu, z ktorého sa používateľ pripája. Týmto prístupom by sa výrazne znížila celková potrebná kapacita dátových liniek lokálneho internetu v prípade, že by bol systém nasadený na relevantné množstvo používateľov.

Aplikácia by získala obsah stránky, ktorú si používateľ vyžiadal najmä z databázy. V prvom rade by hľadala, či sa hľadané informácie nenachádzajú v lokálnej databáze, a až v prípade neúspechu by webová služba vytvorila požiadavku vo forme XML súboru a rozoslala ju na ostatné uzly. Webová služba na prijímajúcej strane by túto požiadavku spracovala a požadovaný článok by hľadala na svojom uzle. V prípade, že by uzol mohla požiadavke vyhovieť, webová služba by z informácií vytvorila XML súbor a odoslala uzlu, ktorý požiadavku vytvoril. Webová služba

pracujúca na tomto uzle by prijaté informácie spracovala a poskytla používateľskému rozhraniu v požadovanej forme, teda tak, aby ich bolo možné zobrazíť užívateľovi.

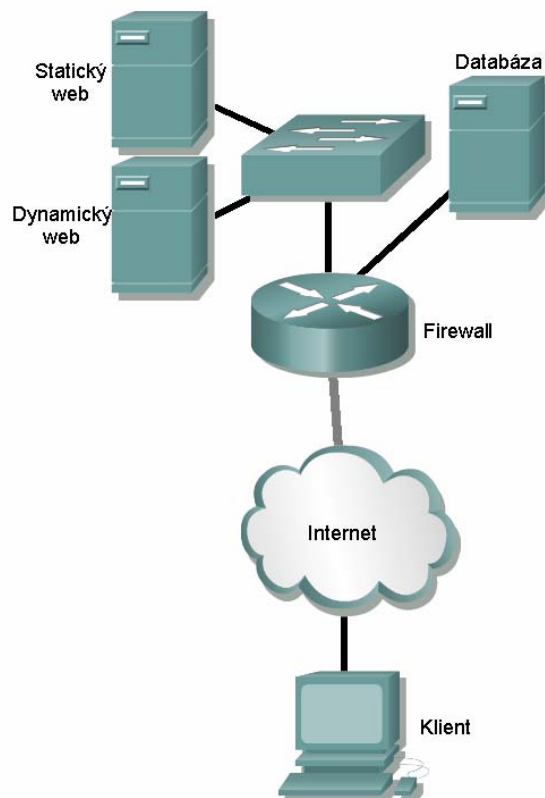
Takéto riešenie prináša viacero nevýhod.

- Impulzom pre hľadanie gridového riešenia bolo využitie dostupných výpočtových prostriedkov, na urýchlenie a zefektívnenie práce systému. Navrhnuté riešenie však využíva poskytnuté prostriedky predovšetkým na skladovanie údajov. Ide teda skôr o dátový grid a nie výpočtový, ako bolo pôvodne zamýšľané.
- Hoci je webovú službu možné vytvoriť v ľubovoľnom vyššom programovacom jazyku, skutočnosť je taká, že potrebné technické predpoklady nutné na jej vytvorenie sú v dostatočnej kvalite a za reálnych finančných podmienok dostupné iba pre jazyk J2EE. Aplikácia vytvorená v jazyku Java by však systém neurýchlila, naopak, vzhľadom na svoje nároky by systém spomaľovala. Navyše skúsenosti členov tímu s jazykom J2EE nedosahujú úroveň potrebnú na riešenie tak zložitého problému, aký synchronizácia databáz viacerých počítačov v gride nesporne je.
- Aplikácia by musela vytvárať a spracovávať požiadavky vo forme XML súborov. S týmto jazykom však náš tím nemá žiadne skúsenosti.
- Navrhovaný systém je veľmi rozsiahly a vhodný je skôr na vytvorenie virtuálnej knižnice. Je tiež otázne, či bolo možné takýto systém vytvoriť počas jedného semestra.
- Efektívnejšie riešenie sme schopní vytvoriť podľa nasledujúceho návrhu.

1.4.4 Riešenie s rozložením záťaže

Tento návrh, rovnako ako predošlý, predpokladá rozdelenie celého systému na viacero častí. Je založený na riešení, ktoré sa reálne používa pre rozsiahle systémy, alebo pre systémy, ktoré vyžadujú vyššiu mieru zabezpečenia údajov. Podľa činností, ktoré budú jednotlivé časti vykonávať, sme systém dekomponovali nasledovne:

- Databáza
- Dynamický web server
- Statický web server



Obr. 4 Architektúra systému s rozložením záťaže

Toto rozdelenie systému na viacero častí však pred samotným užívateľom zostáva skryté. Ten bude so systémom komunikovať prostredníctvom internetového prehliadača, v ktorom sa mu zobrazí používateľské rozhranie, využívajúce služby dynamického a statického web servera. Databáza nebude priamo prístupná koncovému používateľovi.

Rozdelenie systému na spomenuté komponenty so sebou prináša niekoľko výhod:

- Zvýšená bezpečnosť osobných údajov používateľov
- Zvýšená bezpečnosť vložených informácií
- Nižšie nároky na hardware každého servera
- Jednoduchšia správa
- Lepšie podmienky škálovateľnosti

2 Špecifikácia riešenia

2.1 Požiadavky

2.1.1 Požiadavky na projekt

Výstupom nášho tímového projektu bude produkt, na ktorý boli kladené požiadavky. Z nich nám vyplývajú určité náležitosti, ktoré musí výsledok našej práce splniť.

Produkt by mal byť hlavne:

- univerzálny a všeobecne použiteľný – z toho vyplýva, že musí byť prehliadateľný vo všetkých dostupných prehliadačoch (keďže sa jedná o webovú prezentáciu).

2.1.2 Požiadavky na systém

Na to, aby výsledný produkt mohol byť plnohodnotne využitý, musí užívateľ, ktorý ho chce použiť splniť požiadavky, ktoré sa týkajú najmä systémového vybavenia a hardvéru.

Týmito sú:

- Počítač s webovým prehliadačom (ľubovoľným, napr. Internet Explorer, Mozilla, Opera atď.)
- Monitor
- Webový server, ktorý podporuje PHP (min. verzie4), SQL (MySQL) – táto podmienka musí splnená kvôli balíku médiaWiki, ktorý využíva PHP a dáta ukladá do databázy.

2.1.3 Funkcionálne požiadavky

Počas analýzy sme dospeli k tomu, že vytváraný systém by mal splniť nasledovné funkcionálne požiadavky:

- Správa užívateľov
- Vytváranie obsahu učebnice
- Prezeranie obsahu učebnice
- Fórum

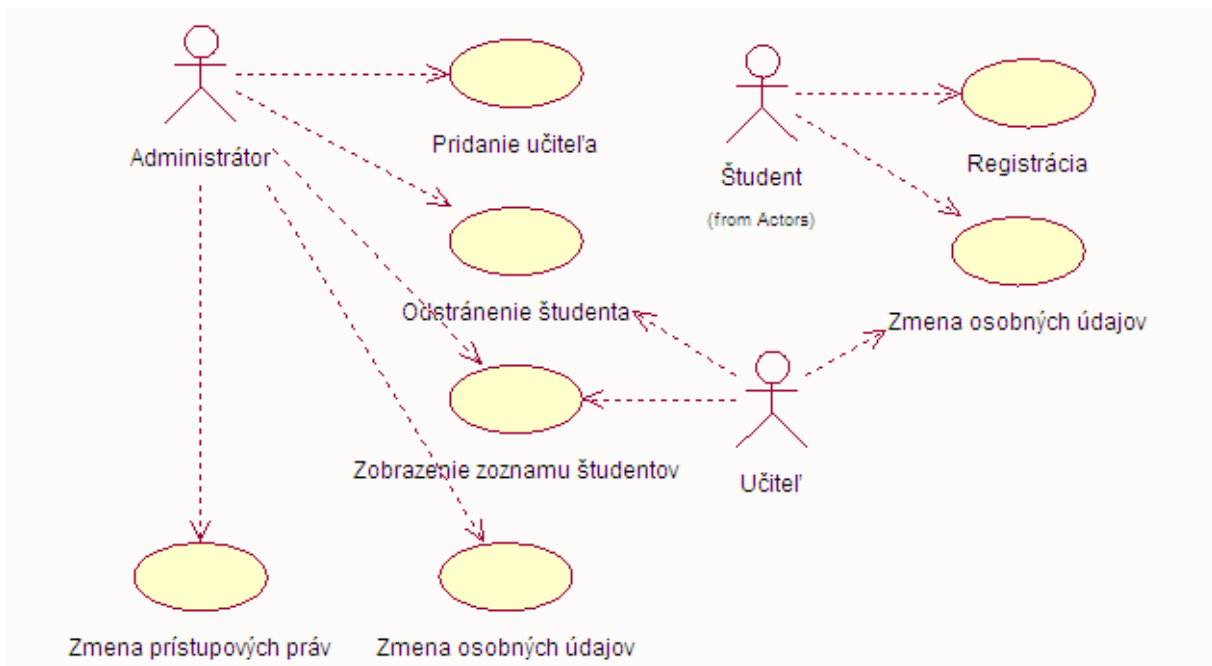
- Kalendár
- Slovník
- Testy

V nasledujúcej časti podrobne popíšeme jednotlivé funkcionálne požiadavky na vytváraný systém. Ich popis sme doplnili modelmi prípadov použitia. V jednotlivých modeloch vystupujú nasledovní účastníci:

- Administrátor
- Študent
- Učiteľ
- Systém

Správa užívateľov

Užívateľom virtuálnej učebnice sa študent môže stať jednoduchou registráciou, pri ktorej udá svoje osobné údaje a prihlasovacie meno a heslo. Rovnakým spôsobom sa užívateľom systému môže stať aj učiteľ. Aby mohol v systéme vykonáva potrebné činnosti, musí mu administrátor prideliť prístupové práva. Inou možnosťou je pridanie učiteľa do systému priamo administrátorom. Na obr. 1.1 je znázornený model prípadov použitia k správe užívateľov.



Obr. 5 Model prípadov použitia k správe systému

UC1 – Pridanie učiteľa

Administrátor vytvorí v systéme konto pre učiteľa s príslušnými prístupovými právami.

UC2 – Zmena prístupových práv

Administrátor podľa potreby zmení prístupové práva užívateľa.

UC3 – Odstránenie študenta

Administrátor alebo učiteľ môžu študenta zo systému odstrániť v prípade, že vážne porušil pravidlá používania virtuálnej učebnice.

UC4 – Zobrazenie zoznamu študentov

Administrátor a učiteľ si môžu nechať zobraziť zoznam študentov, používajúcich virtuálnu učebnicu.

UC5 – Zobrazenie zoznamu učiteľov

Administrátor si nechá zobraziť zoznam užívateľov s prístupovými právami učiteľa.

UC6 – Registrácia

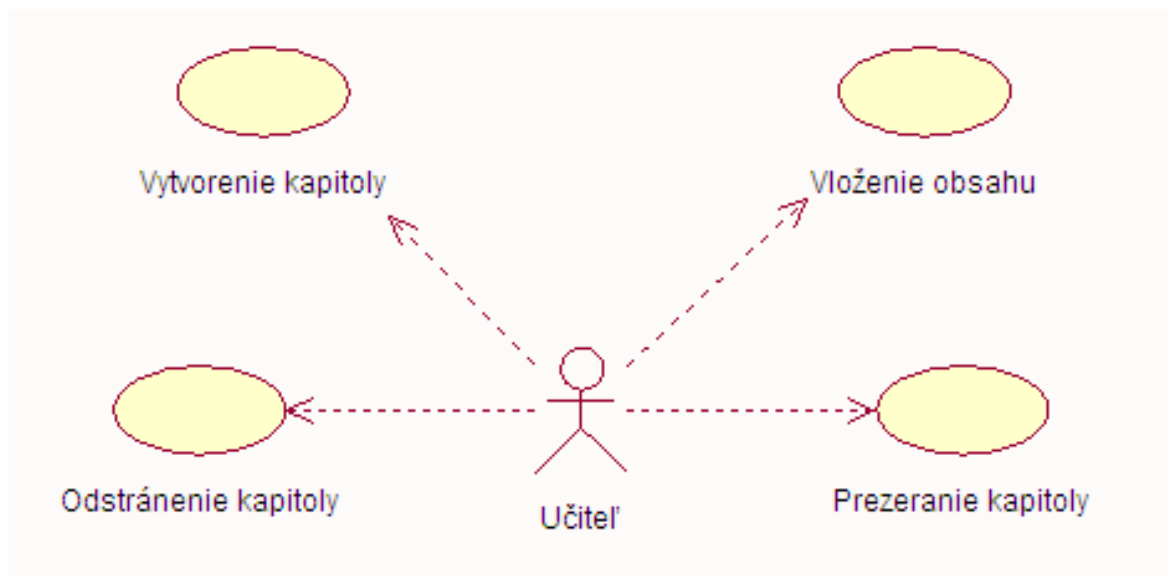
Študent zadá do systému svoje osobné údaje, prihlasovacie meno a heslo, aby mohol využívať virtuálnu učebnicu.

UC1 – Zmena osobných údajov

Študent a učiteľ si môžu svoje osobné údaje v prípade potreby zmeniť.

Vytváranie obsahu

Vyváranie obsahu učebnice je ponechané na samotného učiteľa. Z tohto dôvodu by mu mal systém tvorbu jednotlivých kapitol čo najviac uľahčiť. Máme tým na mysli jednoduché importovanie textu i multimediálnych prvkov. V prípade potreby by mal systém učiteľovi umožniť aj upravenie alebo odstránenie celej kapitoly. Na obr. 1.2 sa nachádza model prípadov použitia k vytváraniu obsahu učebnice.



Obr. 6 Model prípadov použitia k vytváraniu obsahu.

UC1 – Vytvorenie kapitoly

Učiteľ vytvorí vo virtuálnej učebnici novú kapitolu.

UC2 – Vloženie obsahu

Učiteľ do kapitoly vloží potrebné texty a multimediamiálne prvky.

UC3 – Editovanie obsahu

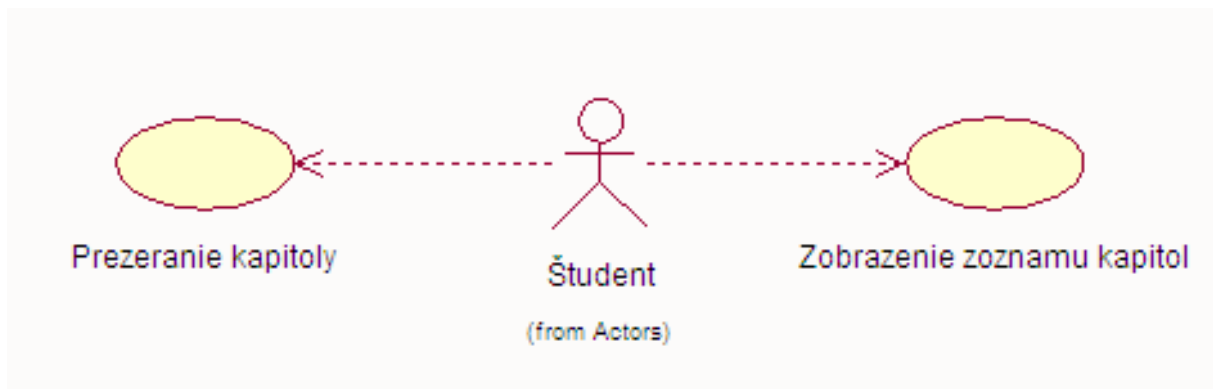
Učiteľ upraví obsah kapitoly.

UC1 – Odstránenie kapitoly

Učiteľ odstráni z virtuálnej učebnice celú kapitolu.

Prezeranie obsahu

Prezeranie obsahu virtuálnej učebnice je základným prípadom použitia systému. Preto musí systém na túto činnosť poskytovať študentovi príjemné a intuitívne užívateľské rozhranie. Študent musí mať možnosť prezrieť si obsah učebnice a vybrať si potrebnú kapitolu. Okrem toho musí systém umožniť študentovi využívanie multimediamiálnych prvkov, ktoré by mali dopĺňať písaný obsah učebnice. Model prípadov použitia vzťahujúci sa k prezeraniu obsahu virtuálnej učebnice je na obr. 1.3.



Obr. 7 Model prípadov použitia k prezeraniu obsahu

UC1 – Zobrazenie zoznamu kapitol

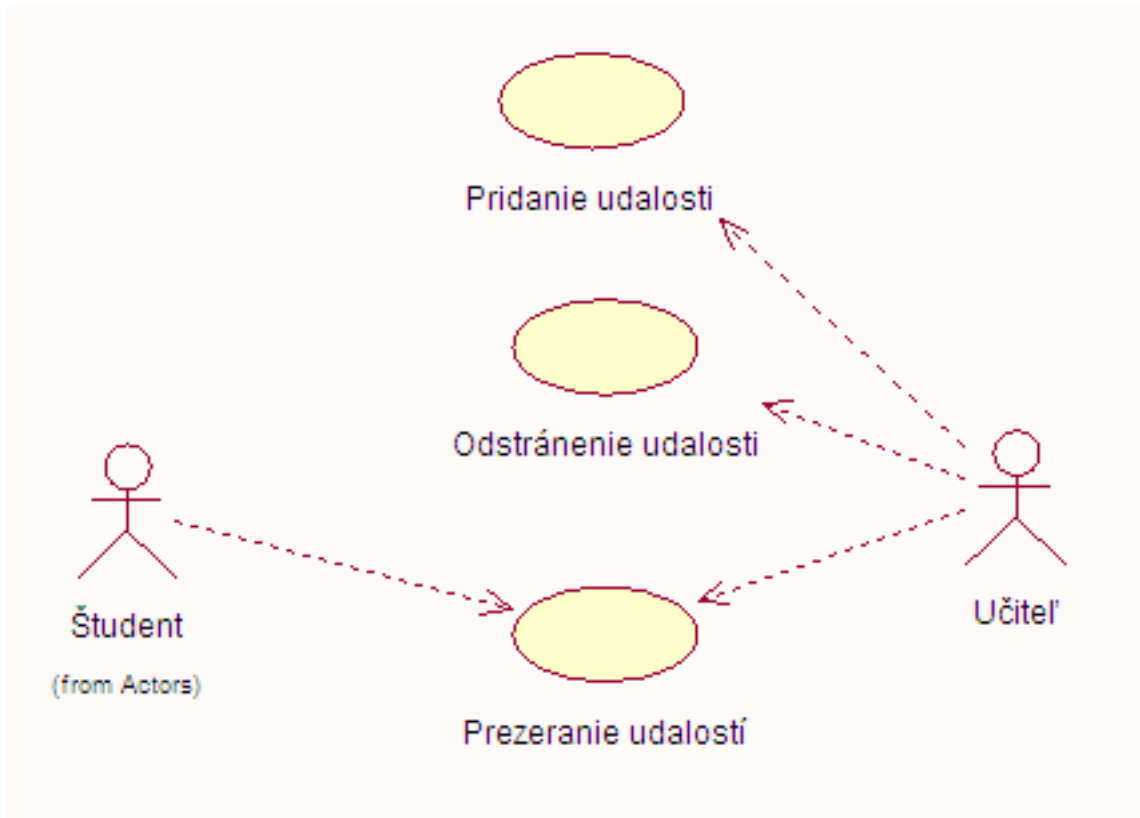
Študent si nechá zobrazíť zoznam kapitol virtuálnej učebnice

UC2 – Prezeranie kapitoly

Študent si vyberie kapitolu a následne si prezerá jej obsah.

Kalendár

Pre zlepšenie komunikácie medzi učiteľom a študentmi by mal systém poskytovať kalendár, ktorý by umožňoval oznámenie niektorých udalostí ako napríklad zapisovanie zápočtu, odpadnutie prednášky, konanie sa písomky. Tieto udalosti by boli v kalendári zvýraznené. Pri kliknutí na takéto miesto v kalendári by sa študentovi zobrazil bližší popis danej udalosti. Na obr. 1.4 sa nachádza model prípadov použitia ku kalendáru.



Obr. 8 Model prípadov použitia ku kalendáru

UC1 – Pridanie udalosti

Učiteľ pridá do kalendára novú udalosť.

UC2 – Odstránenie udalosti

Učiteľ z kalendára existujúcu udalosť odstráni.

UC3– Prezeranie udalostí

Učiteľ aj študent si môžu prezerať udalosti vložené do kalendára.

Fórum

Fórum je ďalšou možnosťou komunikácie učiteľa so študentmi, ale aj medzi študentmi navzájom. Systém by mal umožňovať vytvárať témy diskusií, ku ktorým by študenti mohli vyjadriť svoj názor, alebo položiť otázky ktoré by potrebovali zodpovedať. Systém musí poskytovať možnosť zobrazenia existujúcich tém. Na obr. 1.5 je znázornený model prípadov použitia vzťahujúci sa k fóru.

UC1 – Pridanie témy

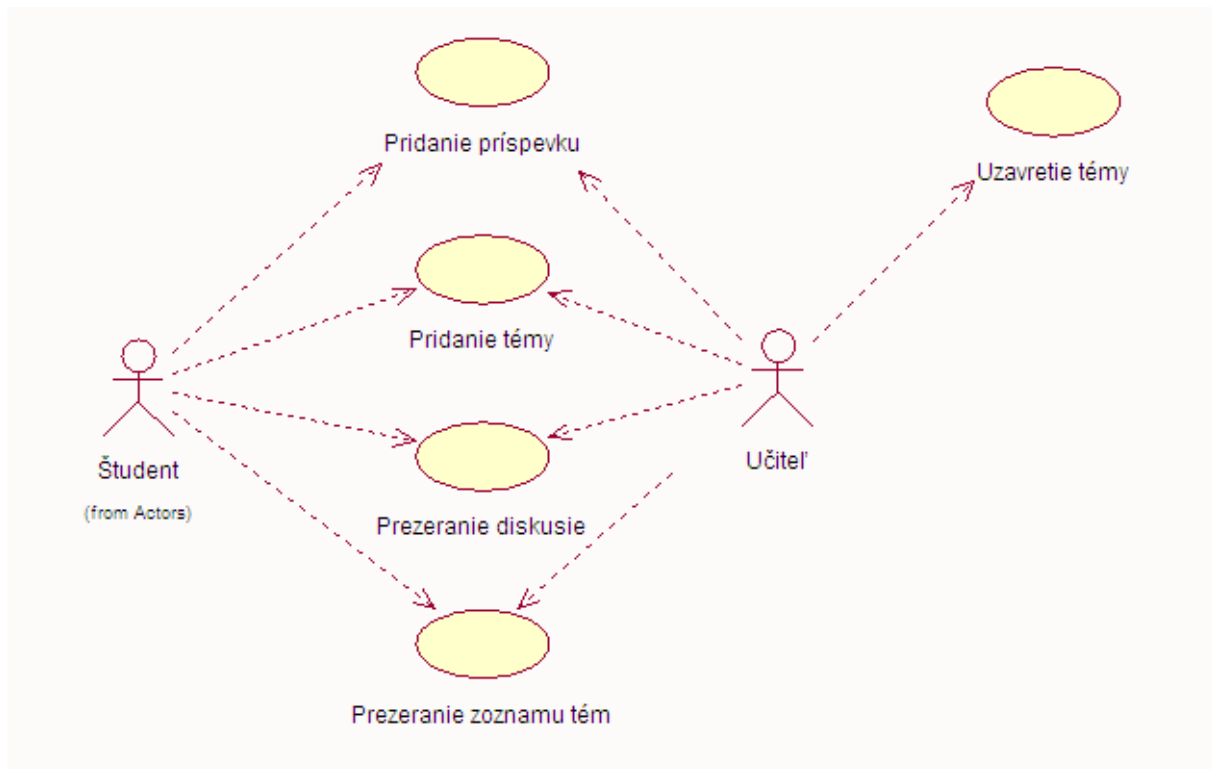
Učiteľ alebo študent vytvorí vo fóre diskusiu s novou témou.

UC2 – Pridanie príspevku

Učiteľ alebo študent pridá do diskusie svoj príspevok.

UC3 – Prezeranie diskusie

Učiteľ alebo študent si prezerá diskusné príspevky k danej téme.



Obr. 9 Model prípadov použitia k fóru

UC4 – Prezeranie zoznamu tém

Učiteľ alebo študent si nechá zobrazíť zoznam tém ku ktorým na fóre existujú diskusie.

UC5 – Uzavretie témy

Učiteľ v prípade potreby môže uzavrieť diskusiu na danú tému.

Slovník

Pretože v obsahu virtuálnej knižnice sa môžu nachádzať cudzie výrazy a rôzne termíny používané v danej problematike, je potrebné aby systém poskytoval miesto kde by takéto výrazy boli vysvetlené. Takýmto miestom by mal byť terminologický slovník. Navyše by systém mal poskytovať možnosť vyhľadávania v tomto slovníku. Obr 1.6 znázorňuje model prípadov použitia k slovníku.

UC1 – Pridanie slova

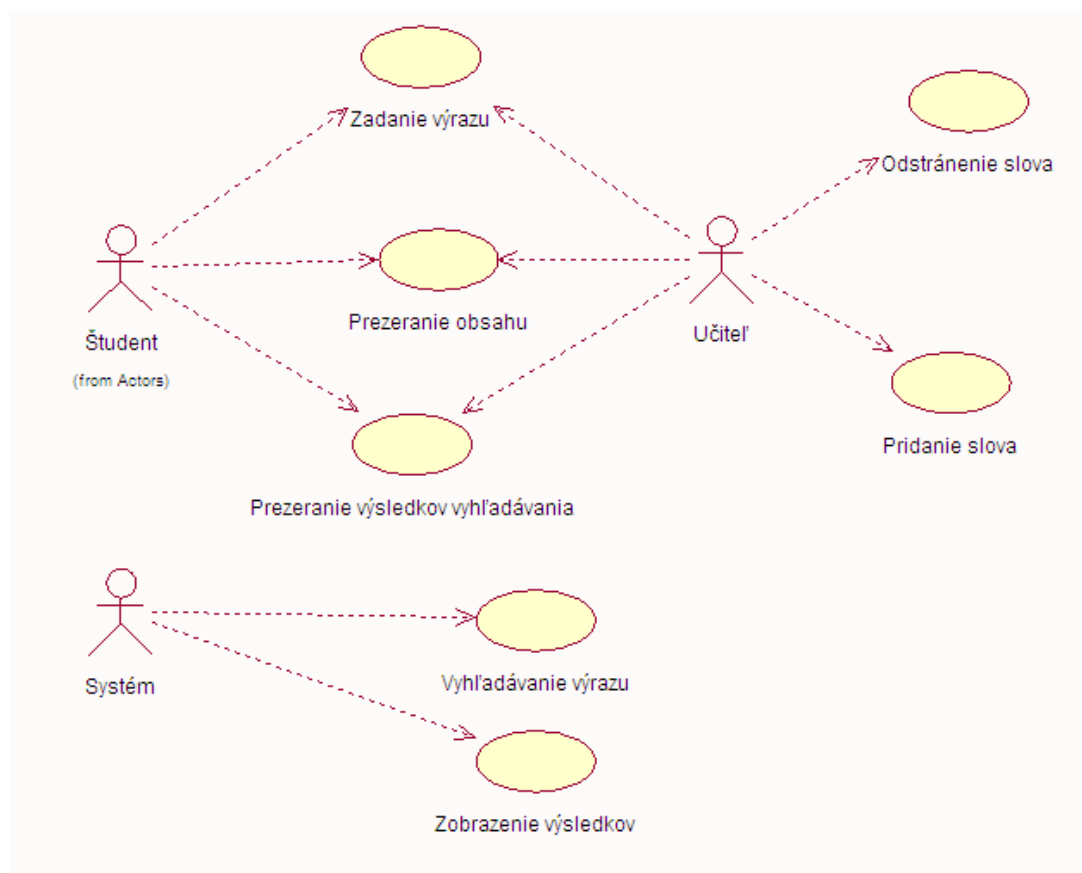
Učiteľ pridá do slovníka nové slovo.

UC2 – Odstránenie slova

Učiteľ zo slovníka existujúce slovo odstráni.

UC3 – Prezeranie slovníka

Učiteľ a študent majú možnosť prezerať si obsah slovníka.



Obr. 10 Model prípadov použitia k slovníku

UC4 – Zadanie výrazu pre vyhľadavanie

Učiteľ alebo študent zadá výraz, ktorý chce nechať v slovníku vyhľadať.

UC5 – Prezeranie výsledkov vyhľadávania

Učiteľ alebo študent si prezrie, výsledky vyhľadávania zadaného výrazu v slovníku, ktoré mu poskytol systém.

UC6 – Vyhľadanie výrazu

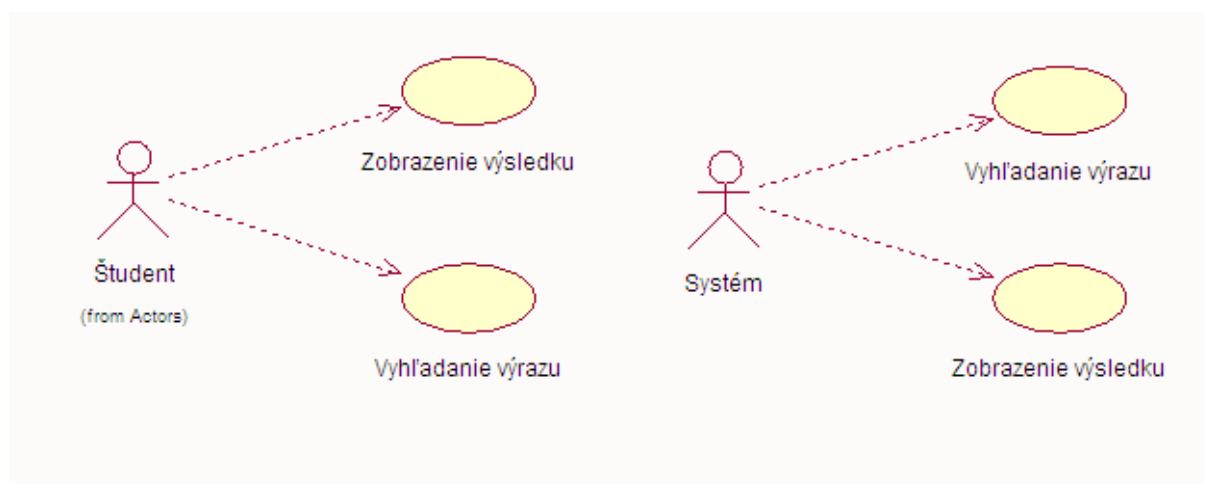
Systém vyhľadá v slovníku zadaný výraz.

UC7 – Zobrazenie výsledkov vyhľadávania

Systém zobrazí užívateľovi výsledky vyhľadávania zadaného výrazu v slovníku.

Vyhľadavanie

Pretože obsah učebnice nemusí byť pre študenta postačujúci, musí mať možnosť vyhľadať ďalšie materiály prostredníctvom Internetu. Po zadaní výrazu systém vyhľadá na Internete stránky s príbuznými témami. Model prípadov použitia vyhľadávania je na obr. 1.7.



Obr. 11 Model prípadov použitia k vyhľadávaniu

UC1 – Zadanie výrazu

Študent zadá výraz, ktorý chce nechať na Internete vyhľadať.

UC2 – Prezeranie výsledkov vyhľadávania

Študent si prezerá výsledky vyhľadávania, ktoré mu systém k danej výrazu poskytol.

UC1 – Vyhľadanie zadaného výrazu

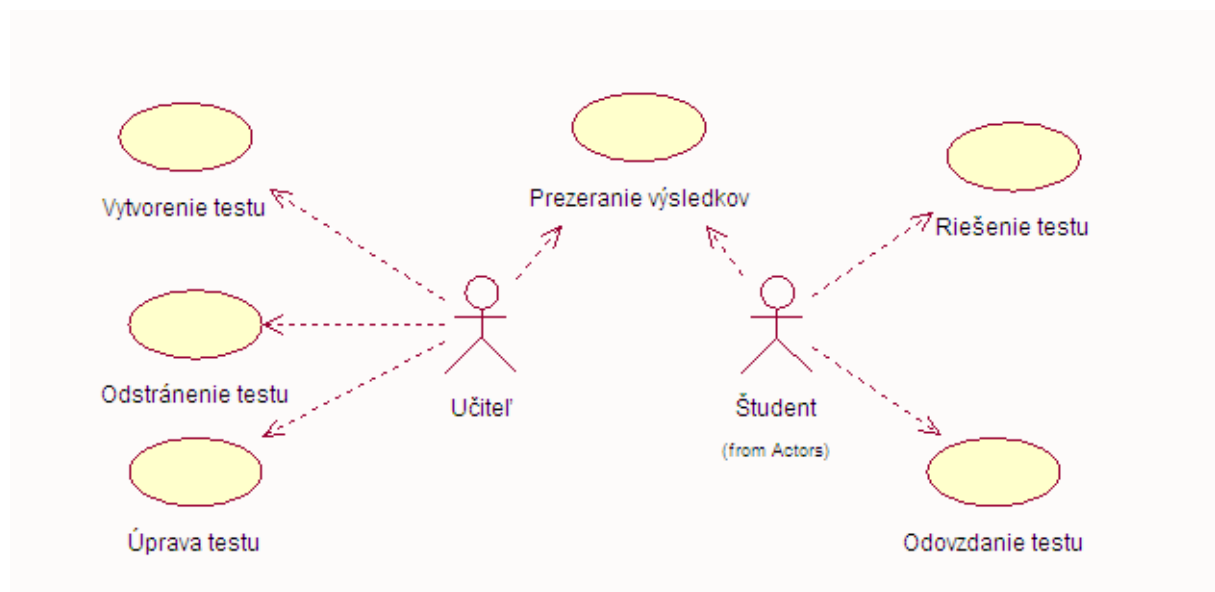
Systém vyhľadá v prostredí Internetu stránky s daným výrazom.

UC1 – Zobrazenie výsledkov

Systém študentovi zobrazí výsledky vyhľadávania daného výrazu.

Testy

Súčasťou virtuálnej učebnice musí byť aj možnosť overenia nadobudnutých poznatkov. Preto musí systém poskytovať učiteľovi možnosť vytvorenia testov, ktoré by mali študenti riešiť. Po vyriešení testu by mal systém vyhodnotiť počet správnych odpovedí a zobraziť výsledky študentovi. Výsledky všetkých študentov musí systém zobraziť aj učiteľovi, ktorý môže na ich základe študentov ohodnotiť. Obr. 1.8 znázorňuje model prípadov použitia k testom.



Obr. 12 Model prípadov použitia k testom

UC1 – Vytvorenie testu

Učiteľ vytvorí test a naplní ho otázkami.

UC2 – Odstránenie testu

Učiteľ odstráni existujúci test.

UC3 – Úprava testu

Učiteľ môže test upraviť pridaním , či odstránením otázok.

UC4 – Riešenie testu

Študent rieši test odpovedaním na jednotlivé otázky.

UC5 – Odovzdanie testu

Študent po zodpovedaní otázok test odovzdá aby ho systém vyhodnotil

UC6 – Prezeranie výsledkov

Študent si môže prezrieť výsledok svojho testu. Učiteľ má možnosť prezerať si výsledky všetkých študentov.

3 Hrubý návrh

Ako sme spomínali v predchádzajúcej kapitole, systém bude pozostávať z 3 častí - základných kameňov, ktorými sú databáza, HTTP server pre statický obsah a HTTP server pre dynamický obsah.

Statický web server bude uspokojovať používateľove požiadavky na obrázky, zvuky, CSS súbory a iný, s časom nemenný materiál, ktorý potrebuje používateľov klient na správne fungovanie používateľského rozhrania. Zároveň bude poskytovať aj obsah, ktorý bude do systému pridaný ako učebný materiál a bude veľmi veľký na to, aby bol uložený v databáze - napr. obrázky vo vysokom rozlíšení, high-definition videá, alebo bližšie nešpecifikovaný binárny obsah.

Statický web server bude použitá serverová aplikácia LigHTTPD. Je to open-source riešenie, ktoré má oproti konkurencii obrovskú výhodu v nízkych hardwarových nárokoch. To znamená, že pri porovnateľnom použítom hardware poskytuje vyšší výkon. Podstata zrýchlenia spočíva v prepísaní a optimalizácii kódu, v použití asynchrónnych vstupov a výstupov a v použití FastCGI, čo znovu predstavuje zrýchlenie oproti štandardnému CGI.

Dynamický webserver bude riešiť hlavnú komunikáciu aplikácie s klientom. Server bude ak osvoj backendový generátor HTTP odpovedí využívať dynamický aplikačný server, s najvyššou pravdepodobnosťou PHP preprocessor. PHP bude riešiť všetky výpočtové záležitosti, bude udržiavať informácie o otvorených spojeniach s klientami, bude vyberať informácie z databázy na základe požiadaviek klienta a generovať HTML kód, ktorý odošle klientovi prostredníctvom dynamického web servera ako výsledok jeho požiadavky. PHP bude tvoriť gro celej aplikácie. Bude rozhodovať, kam sa uloží nová informácia od klienta a pri jej neskoršom vyžiadaní klientom bude PHP rozhodovať, odkiaľ ju získa. Bude riešiť konfliktné situácie (napr. ak budú chcieť dvaja klienti naraz modifikovať jednu informáciu v databáze).

Ako dynamický web server plánujeme použiť Apache HTTPD, ktorý je veľmi flexibilný a bez problémov sa do neho dajú zakomponovať vlastnosti, ktoré budú riešiť problémy, ktoré môžu vyvstať počas implementácie. Zároveň je táto serverová aplikácia veľmi bezpečná, stabilná a zároveň dostatočne výkonná. Tento server patrí k najpoužívanejším a voči najväčšej konkurencii, ktorú predstavuje Microsoft IIS má tú

výhodu, že dodržiava štandardy a preto sa netreba obávať nekompatibility s niektorými klientskými prehliadačmi.

Databáza bude zodpovedať štandardu SQL. S najväčšou pravdepodobnosťou použijeme databázu MySQL, alebo PostgreSQL, ktoré sú poskytované pod licenciou GPL a sú teda k dispozícii zadarmo, no zároveň poskytujú dostatočne široké možnosti na vysokej kvalitatívnej úrovni. Databáza bude kvôli zabezpečeniu vyššej rezervy výkonu web serverov a kvôli zabezpečeniu ostatných kvalít, ktoré toto riešenie umožňuje, umiestnená na vlastnom serveri. Okrem výkonu toto riešenie prináša výhodu aj v tom, že zabezpečuje oddelenie informácií. Na web server má prístup aj klient a preto je náchylnejší na penetráciu nevyžiadanou osobou. Databáza bude chránená firewallom na úrovni servera a pracovať s ňou bude môcť len dynamická PHP aplikácia bežiaci na dynamickom web serveri. Databázový server bude v ideálnom prípade umiestnený vo vlastnej IP podsieti s pridelenými privátnymi IP adresami podľa RFC 1918, ktoré sa nedajú routovať von do internetu. Toto riešenie poskytuje ďalšiu úroveň ochrany pred neželanými prístupmi.

4 Riadenie projektu

Predkladaný dokument má za úlohu sumarizovať všetky dôležité dokumenty súvisajúce s vypracovávaním tímového projektu.

Obsahuje súčasť:

Ponuku – zaoberá sa stručnou analýzou problému, popisuje náš subjektívny pohľad na problematiku virtuálneho vzdelávania.

Plán projektu – obsahuje predovšetkým podrobnejšie rozpracovaný harmonogram prác na projekte počas roka.

Úlohy členov tímu – definuje zodpovednosť jednotlivých členov tímu v súvislosti s prácou na projekte a zároveň hodnotí dielče úkony na práci.

Zápisy zo stretnutí – popis priebehu stretnutí členov tímu

4.1 Ponuka

4.1.1 Predstavenie členov tímu

Bc. Matej Mayer

Bakalársky stupeň štúdia ukončil na Fakulte informatiky a informačných technológií STU (FIIT STU) v Bratislave, kde momentálne pokračuje v inžinierskom štúdiu. Počas tohto štúdia nadobudol znalosti v oblasti programovacích jazykov C/C++, assembler a iné.

Vo voľných chvíľach strávených v práci sa venuje programovaniu dynamických CMS webov a webových aplikácií v jazyku PHP. K tomu je bezpodmienečne nutná znalosť jazykov XHTML, SQL, Javascript. Inak sa rád venuje grafickým editorom, akými sú Adobe Photoshop alebo Adobe Flash.

V záverečnej práci bakalárskeho štúdia sa venoval problematike pamäťového podsystému počítača, ktorý spracoval v jednom z existujúcich e-learningových systémov – Mediawiki.

Aktívne ovláda anglický a nemecký jazyk.

Bc. Andrej Letkovský

Bakalársky stupeň štúdia ukončil na Fakulte informatiky a informačných technológií STU (FIIT STU) v Bratislave, kde momentálne pokračuje v inžinierskom štúdiu v odbore počítačové systémy a siete. Počas tohto štúdia nadobudol znalosti v oblasti programovacích jazykov C, C++, C#, VHDL a assembler. Má skúsenosti aj s prácou prostredí Adobe Flash. Ako záverečnú prácu bakalárskeho štúdia vytvoril výučbový systém vo forme HTML aplikácie na podporu výučby predmetu „Asemblery a systémové programovanie“. Vo voľnom čase sa zaujíma o štúdium programovacích jazykov PHP a SQL. Aktívne ovláda anglický a čiastočne aj nemecký jazyk.

Bc. Peter Kalanin

Bakalársky stupeň štúdia ukončil na Fakulte informatiky a informačných technológií STU (FIIT STU) v Bratislave, kde momentálne pokračuje v inžinierskom štúdiu. Počas tohto štúdia nadobudol znalosti v oblasti programovacích jazykov C/C++, Java, VHDL, Assembler. Okrem toho sa vo voľných chvíľach samoštúdiom zaujíma aj o programovacie jazyky PHP, SQL, HTML. Zúčastnil sa na niekoľkých príležitostných

malých projektoch tvorby dynamických web stránok, kde využil všetky svoje nadobudnuté vedomosti v tejto oblasti. Pri záverečnej práci bakalárskeho štúdia sa venoval problematike spanning-tree protokolu v počítačových sieťach a útokom prostredníctvom tohto protokolu. Vo voľnom čase sa venuje tiež počítačovým sieťam a problematike s tým spojenej.

Aktívne ovláda slovenský a anglický jazyk. Pasívne zvláda jazyk nemecký.

Bc. Juraj Hrubša

Bakalársky titul získal na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave. Má znalosti viacerých programovacích jazykov nadobudnuté nielen v škole, ale aj v praxi systémového administrátora. Ovláda jazyky od nízkoúrovňových, ako x86 assembler, cez C, C++, Javu, až po PHP, HTML, Javascript a Flash. Má širokú základňu znalostí s návrhom, inštaláciou a prevádzkou technických prostriedkov ako web servery a aplikačné servery. Aktívne programuje aplikácie v dynamickom HTML s multimediálnymi prvkami a zaoberá sa backend systémami, ktoré sú potrebné pre správny chod web stránok. Pri práci na svojej bakalárskej práci s témou Prostriedky pre podporu výučby predmetu ASP získal skúsenosti s elektronickým vzdelávaním, ktoré má záujem ďalej prehľbovať. Ovláda slovom i písmom anglický jazyk a písmom francúzsky jazyk.

Bc. Miroslav Míka

Absolvent bakalárskeho štúdia na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave. Počas štúdia nadobudol základné znalosti v programovaní v mnohých nižších a vyšších programovacích jazykoch, medzi inými aj PHP a MySQL, ktoré si popri štúdiu na vysokej škole zdokonaľoval samoštúdiom. Nadobudnuté vedomosti zúžitkoval v bakalárskej práci, zaoberajúcou sa kompletným návrhom a vývojom internetového obchodného portálu v prostredí MySQL a PHP.

Popri štúdiu sa aktívne venoval učeniu základov programovania ako učiteľ na 8-ročnom gymnáziu v Bratislave, kde získal bohaté skúsenosti v metodológii práce so študentami a v súčasnosti pracuje ako sieťový špecialista pre firmu zaoberajúcu sa mimo iné aj podporou pre webové servery. Aktívne ovláda anglický a pasívne aj nemecký jazyk.

Bc. Matej Fajnor

Bakalárske štúdium úspešne ukončil na Fakulte informatiky a informačných technológií STU v Bratislave, kde momentálne pokračuje v inžinierskom štúdiu. Počas svojho štúdia nadobudol znalosti od nízkoúrovňových programovacích jazykov (assembler pre procesory x86) až po programovacie jazyky vyššej úrovne (C/C++, Java, VHDL). Vo svojej záverečnej práci bakalárskeho štúdia sa venoval podpore pre výučbu predmetu assemblerov a systémového programovania. Vo svojom voľnom čase sa venuje samoštúdiu jazykov PHP, HTML, SQL, ktorých znalosti plánuje ďalej zdokonaľovať.

Aktívne ovláda anglický jazyk a pasívne nemecký jazyk.

4.1.2 Motivácia

Téma Virtuálna učebnica je pre náš tím zaujímavá z viacerých dôvodov. Ako študenti Fakulty informatiky a informačných technológií máme skúsenosti s používaním viacerých systémov elektronického vzdelávania. Tieto skúsenosti by sme preto chceli využiť pri riešení tímového projektu. Chceli by sme vytvoriť virtuálnu učebnicu, ktorá by bola pre užívateľa zrozumiteľná a nenáročná na používanie. Výsledkom by mal byť systém, ktorý odbremení užívateľa od práce so systémom samotným a umožní mu v plnej miere sústrediť sa na obsahovú náplň učebnice.

Viacerí členovia nášho tímu vytvárali v rámci bakalárskeho projektu prostriedky pre podporu výučby predmetov, ktoré sú vyučované na Fakulte informatiky a informačných technológií. Majú teda praktické skúsenosti nielen s tvorbou samotných systémov elektronického vzdelávania, ale aj s tvorbou výučbových textov, kde zároveň využijeme ročné skúsenosti člena nášho tímu pracujúceho v tejto oblasti. Navyše majú skúsenosti s používaním softvérových prostriedkov na tvorbu multimediálnych prvkov vhodných pre použitie v elektronickom výučbovom systéme. Niektorí členovia nášho tímu majú tiež profesionálne skúsenosti s využívaním technológií používaných na tvorbu webových aplikácií. Ich znalosti by teda mali zaručiť, že výsledný projekt bude spĺňať stanovené podmienky a bude konkurencie schopný medzi množstvom kvalitných tímových projektov. Pre ostatných členov tímu je to zároveň možnosť a motivácia rozšíriť si svoje znalosti a získať nové schopnosti a vedomosti v tejto oblasti.

4.1.3 Návrh riešenia

Implementáciu učebnice plánujeme umiestniť do prostredia modifikovaného internetového redakčného systému Claroline.

Pri riešení projektu navrhujeme postupovať postupne v 3 fázach

1. Ujasnenie tematického obsahu vyvíjanej aplikácie po konzultácii s vedúcim projektu
2. Modifikovanie redakčného e-learningového systému tak, aby spĺňal požiadavky zadania a poskytoval nám dostatočnú flexibilitu pri dopĺňaní nových funkcií a modulov
3. Umiestnenie obsahovej zložky do redakčného systému

Kľúčovú úlohu v celom procese riešenia projektu zohráva druhý krok, v ktorom budeme modifikovať systém na základe požiadaviek, ktoré musí spĺňať virtuálna učebnica :

- Multimediálny obsah, ktorého základom sú interaktívne grafické prvky Flash
- Neobmedzený navigačný strom
- Intuitívne ovládanie, ktoré umožní používateľovi efektívne získavať informácie zo systému
- Zachovanie prehľadnosti rozložených prvkov

4.1.4 Vlastné riešenie

Jednou z možností riešenia danej tematiky by bol vlastný návrh „virtuálnej učebnice“, teda by sme nevyužili žiadne hotové riešenie, do ktorého by sme doimplementovali ďalšiu funkcionálnosť, ale by sme spravili nový informačno-vzdelávací portál. Tento portál by mal bežať na linuxovom serveri, ktorý by mal nainštalované Apache, PHP, databázu MySQL s podporou funkcie SMTP servera, Cron a iných.

Ako už vyplýva zo špecifikácie nárokov na server, portál bude implementovaný v jazyku PHP a ako úložisko dát bude využitá MySQL databáza, ktorá zabezpečuje kompaktné riešenie a zároveň pomerne vysoké zabezpečenie uložených dát. Aplikácia bude prístupná cez webový server Apache.

Portál bude rozoznávať 3 základných užívateľov systému: časť pre užívateľov, časť pre učiteľov a časť pre administrátora systému.

Funkcionalita pre administrátora:

- vytváranie kurzov administrátorom
- vytváranie používateľských účtov a určovanie ich právomocí
- editácia existujúcich článkov

Funkcionalita pre učiteľa:

- pridávanie článkov do jednotlivých kurzov
- vytváranie testových otázok
- priradovanie zadaní do jednotlivých kurzov
- hodnotenie zadaní
- pridávanie užívateľov do jednotlivých kurzov učiteľom

Funkcionalita pre užívateľa:

- registrácia do systému
- prihlásenie sa do systému
- štúdium materiálov k danému kurzu
- možnosť odovzdania zadaní prostredníctvom systému
- prístup k mailing-list o jednotlivých kurzoch

Funkcionalita systému:

- testovacie moduly pre jednotlivé kurzy (náhodné generovanie otázok, vyhodnocovanie odpovedí, postupové testy na základe vykonaných testov)
- notifikácie akcií podstatných pre užívateľa prostredníctvom mailu
- automatické periodické zálohovanie databázy (prostredníctvom plánovača Cron)

4.1.5 Predpokladané zdroje

V tejto časti sa chceme zamerať na odhadnutie nákladov a prostriedkov potrebných pri vypracúvaní projektu virtuálnej učebnice.

Implementačné prostredie

Riešenie bude implementované prostredníctvom jazykov HTML, PHP, SQL, Javascript. Backend software s podporou aplikácií napísaných v týchto jazykoch je dostupný aj vo forme opensource, z čoho je možné usúdiť, že finančné nároky budú minimálne. Rovnako na tvorbu takýchto aplikácií existuje nespočetné množstvo nástrojov, z ktorých spomenieme napr. CofeeCup HTML Editor, PSPad a Context Programmer's Editor. Okrem toho bude nutné v rámci nášho projektu použiť nasledujúce softvérové produkty:

- Pre tvorbu a úpravu obrázkov Adobe Photoshop
- Pre tvorbu dynamických prvkov Adobe Flash

Hardvérové nároky

Na vývoj predkladaného systému postačujú bežné technické prostriedky a to Intel 386 kompatibilný osobný počítač s LCD monitorom a myšou. LCD monitor je vhodný z hľadiska relatívne dlhej doby potrebnej na vývoj, ktorá je odôvodnená v nasledujúcej kapitole.

Potrebný je operačný systém s podporou väčšiny majoritne používaných internetových prehliadačov, pretože nutnou súčasťou vývoja web aplikácie je aj intenzívne testovanie funkcionality systému a správneho zobrazovania priamo v prehliadači.

4.1.6 Zoradenie ponúkaných tém podľa priority

1. Virtuálna učebnica
2. Podpora vzdelávania v predmete Špecifikačné a opisné jazyky
3. Modelovanie a riadenie systému automaticky navádzaných vozidiel pre dopravu vo výrobných procesoch
4. Návrh a realizácia experimentálnych mikropočítačov
5. Simulátor komunikácie v počítačovej sieti

4.1.7 Rozvrh

Deň	7.00-7.50	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50	20.00-20.50	
Po	de150 (BA-MD-FEI D-E) Architektúra počítačových systémov <i>L. Hudec</i>			C 102 (BA-MD-FEI C) Vnorené systémy <i>T. Krajčovič</i>		Priorita 2				E-701 (BA-MD-FEI E) Tímový projekt I ⁽¹⁾ <i>J. Hudec</i>					
	D 105 (BA-MD-FEI D) Komunikačné služby a siete <i>B. Dado</i>														
Út	bc150 (BA-MD-FEI B-C) Bezpečnosť počítačových systémov <i>L. Hudec</i>			Priorita 1				cd300 (BA-MD-FEI C-D) Kódovanie <i>K. Œpková</i>		cpu (BA-MD-FEI D) Bezpečnosť počítačových systémov <i>A. Bagala</i>		cpu (BA-MD-FEI D) Bezpečnosť počítačových systémov <i>A. Bagala</i>			
St								de150 (BA-MD-FEI D-E) Komunikačné služby a siete <i>M. Kotočová</i>		E-702 (BA-MD-FEI E) Vnorené systémy <i>T. Krajčovič</i>					
Št	de150 (BA-MD-FEI D-E) Kódovanie <i>K. Œpková</i>			D 105 (BA-MD-FEI D) Komunikačné služby a siete <i>B. Dado</i>				D 105 (BA-MD-FEI D) Komunikačné služby a siete <i>B. Dado</i>							

4.2 Rozdelenie úloh

Rozdelením úloh sme sa snažili docieľiť zefektívnenie práce celého tímu rozdelením práce na menšie celky. Zodpovednosť za jednotlivé úlohy bola v tíme rozdelená počas 4. stretnutia, po nabudnutí hrubej predstavy o riešení projektu.

4.2.1 Dlhodobé úlohy

Dlhodobé úlohy sledovali základný cieľ nášho projektu, ktorým je vývoj virtuálnej učebnice. Snažili sme sa rozdeliť podľa odbornosti jednotlivých členov tímu v danej problematike a na základe skúsenosti nadobudnutých prevažne behom mimoškých aktivít.

Meno	Pozícia	Úlohy
Bc. Miroslav Mika	vedúci tímu, programátor, analytik	Zodpovednosť za riadenie úloh v tíme, komunikácia s vedúcim projektu a implementácia produktu
Bc. Matej Mayer	programátor, analytik, tester	Návrh realizovaných zmien, implementácia produktu, finálne testovanie produktu
Bc. Juraj Hrubša	programátor, správca softvérovej platformy	Implementácia zálohovacej časti databázy, správa prostredia pre vývoj aplikácie
Bc. Matej Fajnor	tvorca obsahovej časti, tester, dokumentarista	Zodpovednosť za obsahovú náplň systému, spracovanie dokumentačnej stránky projektu a manuál používateľa
Bc. Peter Kalanin	administrátor webstránky, tester, grafik	Zodpovednosť za grafické prevedenie aplikácie a jej rozhranie, správne fungovanie systému po implementácii
Bc. Andrej Letkovský	programátor, analytik	Návrh realizovaných zmien, implementácia testovacieho modulu aplikácie

4.2.2 Krátkodobé úlohy

Krátkodobé úlohy vyplývajú z charakteru práce v jednotlivých etapách vývoja a snažili sa kopírovať zameranie jednotlivých členov tímu. Tabuľka krátkodobých úloh nie je finalizovaná a jej obsah budeme pravidelne aktualizovať po každotýždenných stretnutiach.

Činnosť	M.Mi	M.Ma	P.K.	A.L.	J.H.	M.F.
Prehľad výučbových systémov	X	X		X		
Spracovanie ponuky		X	X			
Prezentácia ponuky	X					
Vytvorenie tímovej stránky					X	X
Štúdium gridového riešenia				X		
Dokumentácia – Analýza		X			X	
Dokumentácia – Špecifikácia	X		X			
Dokumentácia – Hrubý návrh						
Dokumentácia – Riadenie pr.	X					X
Spracovanie zápisníc			X			
Posudok				X		X
Doplnenie dokumentácie	X			X		
Dokumentácia - Prototyp			X			
Používateľská príručka		X				
Implementácia prototypu		X			X	

4.2.3 Časový harmonogram

Zimný semester	
1. týždeň	Zostavenie tímu, voľba a spracovanie ponuky
2. týždeň	Prezentácia ponuky
3. týždeň	Prvé konzultácie s vedúcim projektu, vytvorenie predběžného plánu a špecifikácia dostupného riešenia
4. týždeň	Rozdelenie úloh medzi členov tímu, analýza problematiky
5. týždeň	Analýza problematiky, práca na návrhu riešenia
6. týždeň	Analýza problematiky, práca na návrhu riešenia
7. týždeň	Finalizácia prác na analýze problému, špecifikácii a návrhu riešenia
8. týždeň	Odovzdanie dokumentácie a štúdium obdržanej dokumentácie
9. týždeň	Vypracovanie a odovzdanie posudku na súperiáci projekt
10. týždeň	Dopracovanie nedostatkov na projekte, práca na prototypu
11. týždeň	Implementácia prototypu vybraných častí
12. týždeň	Odovzdanie dopracovaného prototypu s dokumentáciou a prezentácia prototypu.

Letný semester	
1. týždeň	Dopracovanie zistených nedostatkov, podrobný návrh riešenia
2. týždeň	Dokončenie prác na podrobnom návrhu riešenia
3. týždeň	Implementácia podľa vypracovaného návrhu – správa užívateľov
4. týždeň	Implementácia podľa vypracovaného návrhu – správa obsahu
5. týždeň	Implementácia podľa vypracovaného návrhu – testovací modul
6. týždeň	Implementácia podľa vypracovaného návrhu – administrátorské rozhranie
7. týždeň	Implementácia podľa vypracovaného návrhu – finálne zosúladenie
8. týždeň	Integrácia produktu na testovaciu stanicu, tvorba dokumentácie
9. týždeň	Dokončenie dokumentácie, prípadné doladenie implementácie
10. týždeň	Odovzdanie produktu a dokumentácie
11. týždeň	Testovanie produktu a následné zapracovanie výsledkov do dokumentácie
12. týždeň	Odovzdanie výsledku projektu

4.3 Zázpisnice zo stretnutí

Zázpisnica č.	Dátum	Miesto	Čas
1	15.10.2007	Softvérové štúdio	13:00

Zúčastnení členovia: Matej Fajnor, Bc.
Juraj Hrubša, Bc.
Peter Kalanin, Bc.
Andrej Letkovský, Bc.
Matej Mayer, Bc.
Miroslav Mika, Bc.

Zúčastnený pedagóg: Hasan Jamal, Ing. PhD.

Zapisovateľ: Peter Kalanin, Bc.

Program stretnutia:

1. Úvod do tímového projektu
2. Zhodnotenie ponuky
3. Web stránka tímu
4. Riešenie problému „Griding Computing“
5. Stanovenie úloh na budúce stretnutie
6. Dohodnutie termínu budúceho stretnutia

Priebeh stretnutia:

1. Úvod do tímového projektu

- zoznámenie sa, dohodnutie vzájomnej komunikácie
- oboznámenie sa s priebehom stretnutí
- stanovenie vzájomných pravidiel, ktoré je potrebné dodržiavať v rámci stretnutí, ako aj počas samotnej práce

2. Zhodnotenie ponuky

- pán Ing. Hasan Jamal prehodnotil našu ponuku, vyjadril spokojnosť s nami podanou ponukou a samotnou prezentáciou

3. Web stránka tímu

- zodpovedný: Matej Fajnor, Bc.
- obsah webstránky:
 - informácie o projekte, členoch
 - odkazy na zaujímavé tímy a linky
 - softvér, ktorý budeme používať
 - zázpisnice
 - časti dôležitého kódu
 - tabuľka úloh

4. Riešenie problému „Griding Computing“

- pán Ing. Hasan Jamal navrhol spôsob riešenia virtuálnej učebnice na technológii „grid computing“
- oboznámil nás s funkciou a využitím tejto technológie
- ešte pred samotným stretnutím nám poskytol v digitálnej podobe informácie o vyššie spomínanom probléme v rámci pochopenia danej problematiky
- voľná diskusia a návrhy riešenia tohto problému
- bolo navrhnuté stretnutie s odborníkom v tejto oblasti na našej fakulte, s pánom Ing. Michalom Zimenom

5. Stanovenie úloh na budúce stretnutie

- kontaktovanie pána Ing. Michala Zimena a diskusia s ním o tom, či je možné realizovať projekt virtuálnej učebnice v prostredí „GRID“ na našej fakulte
- určenie úloh a zodpovedností jednotlivých členov tímu
- hlbšie naštudovanie problematiky „GRIDU“ a definitívne vyjadrenie rozhodnutia o realizácii, alebo nerealizácii nášho projektu v tomto prostredí

6. Dohodnutie termínu budúceho stretnutia

- termín budúceho stretnutia bol stanovený na 25.10.2007 o 09:00 v softvérovom štúdiu

Vypracoval Bc. Peter Kalanin

Zápisnica č.	Dátum	Miesto	Čas
2	29.10.2007	D-112	13:00

Zúčastnení členovia: Matej Fajnor, Bc.
Juraj Hrubša, Bc.
Peter Kalanin, Bc.
Andrej Letkovský, Bc.
Matej Mayer, Bc.
Miroslav Mika, Bc.

Zúčastnený pedagóg: Hasan Jamal, Ing. PhD.

Zapisovateľ: Peter Kalanin, Bc.

Program stretnutia:

1. Web stránka tímu
2. Diskusia o probléme „Grid Computing“
3. Výber riešenia implementácie
4. Rozdelenie úloh
5. Dohodnutie termínu budúceho stretnutia

Priebeh stretnutia:

1. Web stránka tímu

- za tvorbu web stránky nášho tímu (*VU tím*) je zodpovedný Bc. Juraj Hrubša
- stránka bola už v čase stretnutia v prevádzke a funkčná, jej obsah bude postupne dopĺňaný
- ostatní zúčastnení súhlasili s takouto formou stránky
- adresa web stránky:
<http://www2.dcs.elf.stuba.sk/TeamProject/2007/teamo2pss/>

2. Diskusia o probléme „Grid Computing“

- zúčastnení referovali svoje poznatky a zistenia získané samoštúdiom v tejto oblasti ako aj analýzou už existujúcich projektov v prostredí „Grid Computing“
- pán Ing. Hasan Jamal bol oboznámený s priebehom konzultácie s pánom Ing. Michalom Zimenom, ako aj s jeho názorom na implementovanie nášho projektu v prostredí „Grid Computing“
- viedla sa diskusia o probléme využitia a používania technológie „Grid Computing“ v našom projekte a o možnostiach realizácie nášho projektu v tomto prostredí

3. Výber riešenia implementácie

- na základe konzultácie a záverečného vyjadrenia pána Ing. Michala Zimena, ako aj na základe našich poznatkov a analýzy už existujúcich riešení sme sa rozhodli implementovať náš projekt v prostredí, ktoré je bližšie špecifikované v návrhu riešenia

4. Rozdelenie úloh

- za vypracovanie analýzy sú zodpovední:
Bc. Andrej Letkovský, Bc. Matej Fajnor a Bc. Peter Kalanin
- za vypracovanie návrhu riešenia a špecifikácie požiadaviek sú zodpovední:
Bc. Matej Mayer, Bc. Juraj Hrubša a Bc. Miroslav Mika
- za vypracovanie zápisníc zo stretnutí je zodpovedný Bc. Peter Kalanin

5. Dohodnutie termínu budúceho stretnutia

- termín budúceho stretnutia bol stanovený na 8.11.2007 o 09:00 v softvérovom štúdiu

Vypracoval Bc. Peter Kalanin

Zápisnica č.	Dátum	Miesto	Čas
3	8.11.2007	Softvérové štúdio	10:00

Zúčastnení členovia: Matej Fajnor, Bc.
Juraj Hrubša, Bc.
Peter Kalanin, Bc.
Andrej Letkovský, Bc.
Matej Mayer, Bc.
Miroslav Mika, Bc.

Zúčastnený pedagóg: Hasan Jamal, Ing. PhD. Sa nemohol zúčastniť z dôvodu pracovných povinností

Zapisovateľ: Peter Kalanin, Bc.

Program stretnutia:

1. Diskusia o návrhu riešenia
2. Diskusia a analýza už existujúcich podobných projektov
3. Zisťovanie konkrétnych skúseností a vedomostí členov tímu v oblasti Php, Mysql, HTML a CSS, využiteľných v našom projekte
4. Diskusia a spresnenie štruktúry kapitol: Analýza, Špecifikácia požiadaviek, Návrh riešenia
5. Diskusia o podieľaní sa jednotlivých členov na práci v tímovom projekte
6. Dohodnutie termínu budúceho stretnutia

Priebeh stretnutia:

1. Diskusia o návrhu riešenia

- po dôkladnom preskúmaní možností realizácie nášho projektu v prostredí „GRID“, ako aj po mnohých konzultáciách a vzájomnej dohode sme sa rozhodli implementovať náš projekt v nami navrhovanom prostredí „databáza + dynamický web server + statický server“
- spomínané prostredie je bližšie špecifikované v návrhu riešenia
- náš projekt bude realizovaný v prostredí HTML, Php, s využitím možností Mysql a CSS, voliteľne aj FLASH animácií
-

2. Diskusia a analýza už existujúcich podobných projektov

- členovia tímu viedli aktívnu diskusiu o realizácii podobných projektov v minulých rokoch tímového projektu
- poukázalo sa na niektoré hlavné časti a funkčné realizácie, ktoré je potrebné riešiť aj v našom projekte

3. Zisťovanie konkrétnych skúseností a vedomostí členov tímu v oblasti Php, Mysql, HTML a CSS, využitelných v našom projekte

- v tomto bode stretnutia členovia tímu viedli diskusiu o možnostiach využitia svojich skúseností a vedomostí, ktoré sú nevyhnutné pri realizácii nášho projektu
- po vzájomnej dohode bolo určené, že k riešeniam jednotlivých funkčných častí nášho projektu budú pridelené úlohy prioritne podľa čo najväčších skúseností a vedomostí problematiky daného problému

4. Diskusia a spresnenie štruktúry kapitol: Analýza, Špecifikácia požiadaviek, Návrh riešenia

- členovia, zodpovední za vypracovanie jednotlivých kapitol predniesli svoje návrhy o štruktúre a obsahu jednotlivých kapitol
- po vzájomnej diskusii boli presne určené štruktúry a obsahy spomínaných kapitol, za ktorých finálne vypracovanie budú zodpovední členovia, určení na minulom stretnutí

5. Diskusia o podieľaní sa jednotlivých členov na práci v tímovom projekte

- po vzájomnej dohode bolo určené, že každý člen tímu sa bude podieľať rovnakou mierou na realizácii nášho projektu

6. Dohodnutie termínu budúceho stretnutia

- termín budúceho stretnutia bol stanovený na 15.11.2007 o 09:00 v softvérovom štúdiu

Vypracoval Bc. Peter Kalanin

Zápisnica č.	Dátum	Miesto	Čas
4	15.11.2007	Softvérové štúdio	10:00

Zúčastnení členovia: Matej Fajnor, Bc.
Juraj Hrubša, Bc.
Peter Kalanin, Bc.
Andrej Letkovský, Bc.
Matej Mayer, Bc.
Miroslav Mika, Bc.

Zúčastnený pedagóg: Hasan Jamal, Ing. PhD.

Zapisovateľ: Peter Kalanin, Bc.

Program stretnutia:

1. Diskusia o vypracovanej téme Analýzy, špecifikácie a hrubého návrhu projektu
2. Zhodnotenie vypracovanej témy Analýzy, špecifikácie a hrubého návrhu projektu a posúdenie individuálnej činnosti jednotlivých členov na vypracovávaní dokumentu
3. Diskusia o vypracovaní nasledujúceho dokumentu „Posudok konkurenčného tímu“
4. Zhodnotenie plnenia povinností jednotlivých členov tímu
5. Pridelenie úloh k nasledujúcemu stretnutiu
6. Dohodnutie termínu budúceho stretnutia

Priebeh stretnutia:

1. Diskusia o vypracovanej téme Analýzy, špecifikácie a hrubého návrhu projektu

- na vypracovaní tohto dokumentu sa podieľali všetci členovia tímu rovnakou mierou
- zohľadnili sa všetky požiadavky, ktoré boli stanovené na vypracovanie tohto dokumentu podľa predstáv nášho pedagogického vedúceho
- v tomto bode nášho stretnutia sme riešili záverečné otázky a nejasnosti jednotlivých kapitol dokumentu, tak aby spĺňal všetky požadované vlastnosti

2. Zhodnotenie vypracovanej témy Analýzy, špecifikácie a hrubého návrhu projektu a posúdenie individuálnej činnosti jednotlivých členov na vypracovávaní dokumentu

- členovia tímu vyjadrili svoje zhodnotenie na dokument ako na celok a všetky záverečné nedostatky zo strany členov tímu ako aj nášho vedúceho tímu boli odstránené
- po vzájomnej dohode sme dokument označili ako vyhovujúci a spĺňajúci všetky potrebné informácie, ktoré je potrebné uviesť v tomto dokumente nášho projektu
- pán Ing. Hasan Jamal vyjadril spokojnosť so štruktúrou a samotným obsahom nami vytvoreného dokumentu Analýzy, špecifikácie a hrubého návrhu projektu

3. Diskusia o vypracovaní nasledujúceho dokumentu „Posudok konkurenčného tímu“

- v tomto bode nášho stretnutia sme sa venovali tvorbe vyššie spomínaného dokumentu
- zhodli sme sa, že na vypracovávaní tohto dokumentu sa budú podieľať všetci členovia tímu rovnakou mierou
- stanovila sa hlavná štruktúra dokumentu a obsah samotného dokumentu
- na samotné odovzdanie vypracovaného dokumentu konkurenčnému tímu ako aj vedúcemu konkurenčného tímu bol poverený Bc. Miroslav Míka

4. Zhodnotenie plnenia povinností jednotlivých členov tímu

- v tejto časti nášho stretnutia sme diskutovali o plnení povinností jednotlivých členov tímu, bola vyjadrená pochvala niektorým členom tímu za dôkladné plnenie si svojich povinností v rámci nášho projektu
- vo všeobecnosti sme sa opäť zhodli, že na budúcom vývoji nášho projektu sa budú všetci členovia tímu podieľať rovnakou mierou

5. Pridelenie úloh k nasledujúcemu stretnutiu

- k nasledujúcemu stretnutiu si majú členovia pripraviť návrh dokumentu „Prototypu vybraných častí systému spolu s dokumentáciou“ a ujasniť si spôsob, akým chcú implementovať náš projekt a ktoré vybrané časti bude vhodné spomenúť v tomto dokumente

6. Dohodnutie termínu budúceho stretnutia

- termín budúceho stretnutia bol stanovený na 22.11.2007 o 09:00 v softvérovom štúdiu

Vypracoval Bc. Peter Kalanin

Zápisnica č.	Dátum	Miesto	Čas
5	22.11.2007	Softvérové štúdio	09:00

Zúčastnení členovia: Matej Fajnor, Bc.
Juraj Hrubša, Bc.
Peter Kalanin, Bc.
Andrej Letkovský, Bc.
Matej Mayer, Bc.
Miroslav Mika, Bc.

Zúčastnený pedagóg: Hasan Jamal, Ing. PhD.

Zapisovateľ: Peter Kalanin, Bc.

Program stretnutia:

1. Diskusia a posledné úpravy vytvoreného dokumentu „posudku analýzy, špecifikácie a návrhu konkurenčného tímu“
2. Zhodnotenie vytvorenia nášho dokumentu posudku pánom Ing. Hasanom Jamalom
3. Diskusia na tému vytvorenia dokumentu: „Prototyp vybraných častí systému spolu s dokumentáciou“
4. Diskusia o návrhu publikácie článkov v oblasti „grid computing“
5. Dohodnutie termínu budúceho stretnutia

Priebeh stretnutia:

1. Diskusia a posledné úpravy vytvoreného dokumentu „posudku analýzy, špecifikácie a návrhu konkurenčného tímu“

- v tomto bode nášho stretnutia sa riešili otázky posledných úprav vyššie spomínaného dokumentu
- každý člen tímu vyjadril svoj názor na súčasný stav dokumentu a po súhlase ostatných členov tímu boli niektoré nedostatky odstránené, alebo vložené nové časti dokumentu

2. Zhodnotenie vytvorenia nášho dokumentu posudku pánom Ing. Hasanom Jamalom

- pán Ing. Hasan Jamal vyjadril spokojnosť so súčasnou štruktúrou ako aj obsahom dokumentu a spomenul len záverečné úpravy, ktoré by bolo vhodné vykonať v dokumente

3. Diskusia na tému vytvorenia dokumentu:

„ Prototyp vybraných častí systému spolu s dokumentáciou “

- v tomto bode stretnutia členovia tímu viedli diskusiu o možnostiach návrhu štruktúry dokumentu
- každý člen tímu vyjadril svoje názory, čo by mal tento dokument obsahovať a akú hrubú štruktúru by mal mať
- po vzájomnej dohode sa stanovila hrubá štruktúra tohto dokumentu a stanovili sa hlavné časti, ktoré budú neskôr uvedené a bližšie špecifikované v dokumente

4. Diskusia o návrhu publikácie článkov v oblasti „grid computing“

- pán Ing. Hasan Jamal nám v dôsledku nami nadobudnutých vedomostí navrhol možnosť publikácie článkov v oblasti „grid computing“
- viedla sa diskusia o obsahu článkov ako aj samotnej publikácie jednotlivých článkov tejto problematiky

5. Dohodnutie termínu budúceho stretnutia

- termín budúceho stretnutia bol stanovený na 29.11.2007 o 09:00 v softvérovom štúdiu

Vypracoval Bc. Peter Kalanin

Zápisnica č.	Dátum	Miesto	Čas
6	29.11.2007	Softvérové štúdio	09:00

Zúčastnení členovia: Matej Fajnor, Bc.
Juraj Hrubša, Bc.
Peter Kalanin, Bc.
Andrej Letkovský, Bc.
Matej Mayer, Bc.
Miroslav Mika, Bc.

Zúčastnený pedagóg: Hasan Jamal, Ing. PhD.

Zapisovateľ: Peter Kalanin, Bc.

Program stretnutia:

1. Zhodnotenie posudku na náš projekt zo strany konkurenčného tímu
2. Diskusia o implementácii nášho projektu
3. Diskusia na tému vytvorenia dokumentu: „Prototyp vybraných častí systému spolu s dokumentáciou“
4. Diskusia a pridelenie úloh pri vypracovaní publikácie článkov o problematike návrhu nášho riešenia
5. Návrh obsahovej časti nášho projektu virtuálnej učebnice
6. Dohodnutie termínu budúceho stretnutia

Priebeh stretnutia:

1. Zhodnotenie posudku na náš projekt zo strany konkurenčného tímu

- v tomto bode nášho stretnutia sme podrobne analyzovali všetky nedostatky, ktoré nám boli vytknuté zo strany konkurenčného tímu v podobe posudku na náš projekt
- všetky nedostatky zistené konkurenčným tímom sme sa rozhodli odstrániť a znova prepracovať už odovzdaný dokument
- stanovili sme si jasné úlohy jednotlivých členov, ktorí budú zodpovední za prepracovanie odovzdaného dokumentu

2. Diskusia o implementácii nášho projektu

- viedla sa voľná diskusia na tému implementácie nášho projektu
- každý člen vyjadril svoje nadobudnuté poznatky vo forme názoru, ako by mal byť náš projekt implementovaný
- riešili sa rôzne možnosti implementácie a hodnotili sa zápory, klady ako aj obtiažnosť realizácie samotnej implementácie

3. Diskusia na tému vytvorenia dokumentu:

„ Prototyp vybraných častí systému spolu s dokumentáciou “

- v tomto bode stretnutia členovia tímu viedli diskusiu o možnostiach návrhu štruktúry dokumentu
- páni Ing. Hasanovi sme predložili návrh interakcie nášho systému s rôznymi typmi užívateľov
- opäť sa riešili otázky štruktúry dokumentu, ako aj samotného obsahu

4. Diskusia a pridelenie úloh pri vypracovaní publikácie článkov o problematike návrhu nášho riešenia

- pán Ing. Hasan Jamal nám v dôsledku nami nadobudnutých vedomostí navrhol možnosť publikácie článkov v oblasti implementácie virtuálnej učebnice v nami navrhovanom prostredí
- viedla sa diskusia o obsahu článkov ako aj samotnej publikácie jednotlivých článkov tejto problematiky
- pán Ing. Hasan Jamal presne stanovil úlohy jednotlivým členom tímu, ktorý sa budú podieľať na tvorbe spomínaných článkov

5. Návrh obsahovej časti nášho projektu virtuálnej učebnice

- v tomto bode stretnutia sa viedla diskusia o obsahovej časti našej virtuálnej učebnice
- po vzájomnej dohode sme sa rozhodli, že náplňou našej virtuálnej učebnice bude učivo predmetu vyučovaného na FIIT STU BA v rámci bakalárskeho alebo inžinierskeho štúdia
- druhou možnosťou náplne našej virtuálnej učebnice bude prezentácia ľubovoľného periférneho zariadenia s obsahom rôznych technických informácií

6. Dohodnutie termínu budúceho stretnutia

- termín budúceho stretnutia bol stanovený na 6.12.2007 o 09:00 v softvérovom štúdiu

Vypracoval Bc. Peter Kalanin

Zápisnica č.	Dátum	Miesto	Čas
7	06.12.2007	Softvérové štúdio	09:00

Zúčastnení členovia:

Juraj Hrubša, Bc.
Peter Kalanin, Bc.
Andrej Letkovský, Bc.
Matej Mayer, Bc.

Zúčastnený pedagóg: Hasan Jamal, Ing. PhD.

Zapisovateľ: Peter Kalanin, Bc.

Program stretnutia:

1. Diskusia o problémoch implementácie nášho projektu
2. Návrh riešenia na už existujúcich CMS
3. Zhodnotenie dopracovania detailnejších funkcionálnych požiadaviek
4. Diskusia a zhodnotenie pri vypracovaní publikácie článkov o problematike návrhu nášho riešenia
5. Problémy pri implementácii na už existujúcich LMS, LCMS
6. Dohodnutie termínu budúceho stretnutia

Priebeh stretnutia:

1. Diskusia o problémoch implementácie nášho projektu:

- v tomto bode nášho stretnutia sme analyzovali všetky problémy, s ktorými sme sa stretli pri samotnej implementácii jednotlivých častí nášho projektu
- všetky nedostatky v implementácii sme spolu prekonzultovali a navrhli sme spôsob ako ich odstrániť, resp. nahradiť iným riešením
- stanovili sme si jasné úlohy jednotlivých členov, ktorí budú zodpovední za vypracovanie jednotlivých častí prototypu

2. Návrh riešenia na už existujúcich CMS:

- v rámci vylepšenia funkcionality nášho projektu sme sa rozhodli implementovať náš projekt na už existujúcom systéme
- každý člen tímu vyjadril svoje skúsenosti a znalosti s rôznymi CMS systémami
- konzultovali sme túto možnosť riešenia aj s pánom Ing. Hasanom Jamalom

3. Zhodnotenie dopracovania detailnejších funkcionálnych požiadaviek:

„ Prototyp vybraných častí systému spolu s dokumentáciou “

- znovu sme detailnejšie prepracovali funkcionálne požiadavky nášho projektu
- pánovi Ing. Hasanovi sme predložili návrh funkcionálnych požiadaviek nášho systému s rôznymi typmi užívateľov a vysvetlili vzájomnú interakciu

4. Diskusia a zhodnotenie pri vypracovaní publikácie článkov o problematike návrhu nášho riešenia

- s pán Ing. Hasanom Jamalom sme opäť prehodnotili možnosť publikácie článkov v oblasti implementácie virtuálnej učebnice v nami navrhovanom prostredí
- prebrali sme odovzdané vypracované časti článku a viedla sa diskusia o ich úprave, resp. nahradení inou témou
- v dôsledku nedostatku času sme sa dohodli na vypracovávaní článkov až po odovzdaní prototypu nášho projektu

4. Problémy pri implementácii na už existujúcich LMS, LCMS

- v tomto bode stretnutia sa viedla diskusia o možnom využití jednotlivých open source LMS alebo LCMS systémov
- po vzájomnej dohode sme sa rozhodli, že náš projekt by mohol byť implementovaný v Docebo (LMS a LCMS) e-learningovej platforme, ktorá je voľne dostupná k stiahnutiu, je „open source“ distribúciou a prináša mnoho ďalších výhod

5. Dohodnutie termínu budúceho stretnutia

- termín budúceho stretnutia bol stanovený na 13.12.2007 o 09:00 v softvérovom štúdiu

Vypracoval Bc. Peter Kalanin

Zápisnica č.	Dátum	Miesto	Čas
8	13.12.2007	Softvérové štúdio	10:00

Zúčastnení členovia: Matej Fajnor, Bc.
Juraj Hrubša, Bc.
Peter Kalanin, Bc.
Andrej Letkovský, Bc.
Matej Mayer, Bc.
Miroslav Mika, Bc.

Zúčastnený pedagóg: Hasan Jamal, Ing. PhD.

Zapisovateľ: Peter Kalanin, Bc.

Program stretnutia:

1. Sprístupnenie osobného servera nášho projektu
2. Prezentácia čiastočne funkčného prototypu nášho projektu
3. Diskusia o možnostiach vylepšenia implementácie
4. Diskusia o ďalšej implementácii nášho projektu
5. Stanovenie úloh do ďalšieho stretnutia
6. Dohodnutie si prezentácie prototypu za prítomnosti konkurenčného tímu
7. Záver, zhodnotenie práce v zimnom semestri

Priebeh stretnutia:

1. Sprístupnenie osobného servera nášho projektu:

- za účelom tzv. „online“ úpravy a implementácie nášho projektu, s možnosťou vlastných nastavení servera bol vytvorený náš súkromný server, ktorý je dostupný na verejnej ip adrese
- tento vlastný server obsahuje softvér na implementáciu nášho zadania, ktorý je bližšie špecifikovaný v návrhu a prototypu nášho projektu
- za sprostredkovanie servera, ako aj jeho správu je zodpovedný Bc. Juraj Hrubša

2. Prezentácia čiastočne funkčného prototypu nášho projektu:

- pánovi Ing. Hasanovi Jamalovi sme predstavili náš funkčný prototyp, ktorý predstavuje základné funkcie nášho celého projektu
- pán Ing. Hasan Jamal vyjadril svoje pripomienky ako aj návrhy, ktoré budú neskôr spracované a samotná implementácia projektu bude upravená tak, aby v čo najväčšej miere spĺňala všetky kritéria virtuálnej učebnice

3. Diskusia o možnostiach vylepšenia implementácie:

- v tomto bode stretnutia členovia tímu viedli diskusiu o možnostiach budúceho vylepšenia nášho projektu
- boli navrhnuté ďalšie komponenty a moduly, ktoré by bolo vhodné do nášho projektu implementovať

4. Diskusia o ďalšej implementácii nášho projektu:

- v tejto časti sme sa venovali najmä samotnej implementácii projektu, rozoberali sme všetky dôležité body, ktoré je potrebné implementovať, aby sme dosiahli komplexný systém virtuálnej učebnice s požadovanými funkciami
- opäť sa riešili problémy pri samotnej implementácii zadania, ako aj možnosti, ako sa vyhnúť problémom, resp. akým spôsobom ich riešiť

5. Stanovenie úloh do ďalšieho stretnutia:

- jednotlivým členom tímu boli stanovené úlohy do ďalšieho stretnutia
- hlavnou úlohou bolo dokončenie implementácie prototypu a príprava dokumentu prototypu ako aj užívateľskej príručky

6. Dohodnutie si prezentácie prototypu za prítomnosti konkurenčného tímu :

- po vzájomnej dohode s pánom Ing. Hasanom Jamalom sme stanovili prezentáciu nášho prototypu za účasti konkurenčného tímu na pondelok (17.12.2007) v dopoludňajších hodinách

7. Záver, zhodnotenie práce v zimnom semestri

- na záver nám vyjadril pán Ing. Hasan Jamal spokojnosť s našou prácou v tímovom projekte v zimnom semestri, no poukázal aj na menšie nedostatky v organizácii tímu, ktoré by bolo vhodné v ďalšom semestri tímového projektu odstrániť

Vypracoval Bc. Peter Kalanin

5 Prototyp

V tejto kapitole sa nachádzajú informácie, týkajúce sa prototypu nášho projektu. Sú tu uvedené postupy, ciele a výsledky samotného prototypu virtuálnej učebnice. V závere tejto časti dokumentu sú uvedené prínosy a zistenia, ktoré boli získané na základe implementácie samotného prototypu.

5.1 Ciele prototypu

Hlavným cieľom prototypu bolo navrhnuť používateľské prostredie so základnými funkcionálnymi požiadavkami. Týmto spôsobom môžeme poskytnúť budúcemu zákazníkovi (užívateľovi) lepšiu predstavu o funkcionalite nášho projektu, ako aj samotnom navrhovanom dizajne používateľského rozhrania. Vďaka realizácii prototypu má záujemca možnosť lepšie zhodnotiť poskytovaný systém, poprípade ľahšie opísať dodatočné požiadavky alebo zmeny v systéme. Samozrejme samotný prototyp má prínos aj pre nás a to v podobe overenia si základnej funkcionality nášho systému v reálnom prostredí a odhalenie niektorých chýb, ktoré sa môžu zviditeľniť až pri samotnej implementácii v reálnom prostredí.

5.2 Softvérový prototyp

Keďže náš projekt má za úlohu implementovať virtuálnu učebnicu, tak určite najdôležitejším faktorom je samotné používateľské rozhranie, s ktorým budú do styku prichádzať všetky typy užívateľov nášho systému.

V našom softvérovom prototypu sa preto zameriame najmä na implementáciu základných používateľských rozhraní. Čo sa týka funkcionality systému, budú implementované len niektoré základné funkcionálne požiadavky.

Naším prvoradým cieľom je vytvoriť čo najviac používateľsky príjemné, jednoduché a ľahko ovládateľné prostredie už v navrhovanom prototypu virtuálnej učebnice.

Tento prototyp by sme chceli samozrejme prezentovať a zistiť vzťah používateľa k navrhnutému riešeniu a neskôr vykonať na základe týchto zistení úpravy alebo vylepšenia, ktoré by ešte viac zvýšili kvalitu spomínaných faktorov užívateľského prostredia.

5.3 Použité implementačné prostredie

Pre jednoduchú a efektívnu prácu s možnosťou vlastného nastavenia jednotlivých implementačných nástrojov bol pre náš tím vytvorený vlastný linuxový server s verejnou ip adresou, na ktorý má každý náš člen prístup.

Ako implementačné prostredie pre databázu sme zvolili MySQL. S týmto prostredím sa stretol takmer každý člen nášho tímu a po analýze požiadaviek sme zistili, že toto prostrediu poskytuje rozsiahle možnosti implementácie databázy, ktoré sú postačujúce pre realizáciu nášho projektu.

Samotná aplikačná logika prototypu bola implementovaná v jazyku PHP. Vzhľadom k nezávislosti tohto jazyka od platformy je možné spustiť prototyp pod operačným systémom Windows, ako aj Linux.

Ako webový server, ktorý je potrebný na preklad php skriptov bol zvolený server Apache.

5.4 Funkcie prototypu

Ako bolo spomínané v kapitole vyššie, základným cieľom prototypu bolo predviesť zákazníkovi rôzne používateľské prostredia v závislosti od typu užívateľa (administrátor, učiteľ, študent, hosť).

Prvoradým cieľom bola teda autentifikácia používateľa, na základe ktorej sa jednotlivé typy používateľov identifikujú. Keďže sme sa rozhodli implementovať náš projekt na už existujúcom riešení LMS/LCMS systému Docebo, samotný autentifikačný mechanizmus už bol implementovaný.

Našou úlohou bolo, aby sa po procese autentifikácie zobrazilo používateľovi užívateľské rozhranie, odpovedajúce typu prihláseného užívateľa.

Náš prototyp vykonáva niektoré základné funkčné požiadavky, ktoré sú prehľadne znázornené v tabuľke nižšie (Tab. 1).

Funkcia	Používateľ	Popis
Registrácia	Hosť	Používateľ má možnosť registrácie do systému. Každú registráciu do systému musí potvrdiť Administrátor, alebo v prípade študenta vedúci pedagóg daného kurzu (predmetu)

Prihlásenie sa	Administrátor, Pedagóg, Študent	Ak je používateľ zaregistrovaný v databáze systému, má možnosť prihlásiť sa a na základe jeho práv v rámci systému využívať jeho funkcie
Odhlásenie sa	Administrátor, Pedagóg, Študent	Po skončení práce so systémom by sa mal každý používateľ zo systému odhlásiť, aby nemohlo dôjsť k zneužitiu jeho konta
Správa profilu	Pedagóg, Študent	Po prihlásení má možnosť užívateľ zmeniť svoje atribúty ako napr. heslo, emailová adresa, poprípade signatúra a pod.
Vytvorenie kurzu	Administrátor	Po prihlásení má možnosť administrátor vytvoriť nový kurz
Vytvorenie konta	Administrátor	Administrátor môže vytvoriť konto typu Študent, resp Pedagóg inému používateľovi
Úprava konta	Administrátor	Administrátor môže ľubovoľne zmeniť typ konta, resp. jeho údaje
Prezeranie obsahu	Administrátor, Pedagóg, Študent	Používateľ si môže prezeráť ľubovoľný kurz v systéme, jeho obsah ako aj jednotlivé dokumenty prislúchajúce danému kurzu

Tab.1 Základné funkčné požiadavky prototypu

5.5 Testovanie prototypu

Pri testovaní prototypu sme sa snažili overiť tieto základné funkcie:

➤ Funkcia autentifikácie

systém musí na základe prihlasovacích údajov povoliť vstup do systému a správne identifikovať typ používateľa

systém tiež musí po skončení práce korektne odhlásiť užívateľa a zabrániť tak zneužitiu jeho konta

➤ Zmena používateľského rozhrania po prihlásení

po prihlásení do systému sa musí používateľovi sprístupniť prislúchajúce užívateľské rozhranie s prislúchajúcimi nástrojmi použitia systému

to znamená, že používateľ, ktorému jeho práva v rámci systému nedovoľujú používať určitý nástroj, nemôže mať tento nástroj vo svojom používateľskom rozhraní zobrazený

- Funkcia jednotlivých nástrojov
v tejto časti sme sa snažili otestovať funkciu jednotlivých nástrojov, ako napr. vytvorenie kurzu administrátorom, alebo napr. správa profilu
po vykonaní požadovanej akcie pomocou príslušného nástroja sme verifikovali zmeny v databáze a samotnom používateľskom rozhraní

- Korektné zobrazenie v rôznych typoch prehliadačov a operačných systémov
snažili sme sa testovať zobrazenie používateľských rozhraní v rôznych typoch „web-prehliadačov“ a v rôznych operačných systémoch
na základe zvolených implementačných prostredí ako aj implementačného prostredia pre databázu sa v tomto bode testovania nevyskytli žiadne problémy

- Výmena údajov medzi databázou a aplikačnou vrstvou
najdôležitejším bodom správnej funkcie systému je korektná výmena údajov medzi aplikačnou vrstvou a databázou
v tomto bode sme sa snažili kontrolovať konzistentnosť databázy ako aj samotnú zmenu údajov v databáze

Na základe testovania môžeme zhrnúť, že sa nám podarilo vytvoriť funkčný prototyp nášho zadania s požadovanými funkcionálnymi požiadavkami. I napriek tomu, že nie sú všetky časti v prototyp implementované, samotný prototyp spĺňa účel prezentácie našej práce v zimnom semestri.

5.6 Zhodnotenie

Nami vytvorený prototyp obsahuje niektoré základné funkcie navrhovaného systému. Jeho hlavným cieľom je poskytnúť záujemcovi o náš systém lepšiu vizuálnu predstavu o našom projekte ako aj odskúšanie si jeho niektorých základných funkcií.

Všetky uvedené funkcie boli testované a ich správna funkčnosť im dáva predpoklad k ich plnému využitiu vo vytváranom systéme virtuálnej učebnice.

6 Použitá literatúra

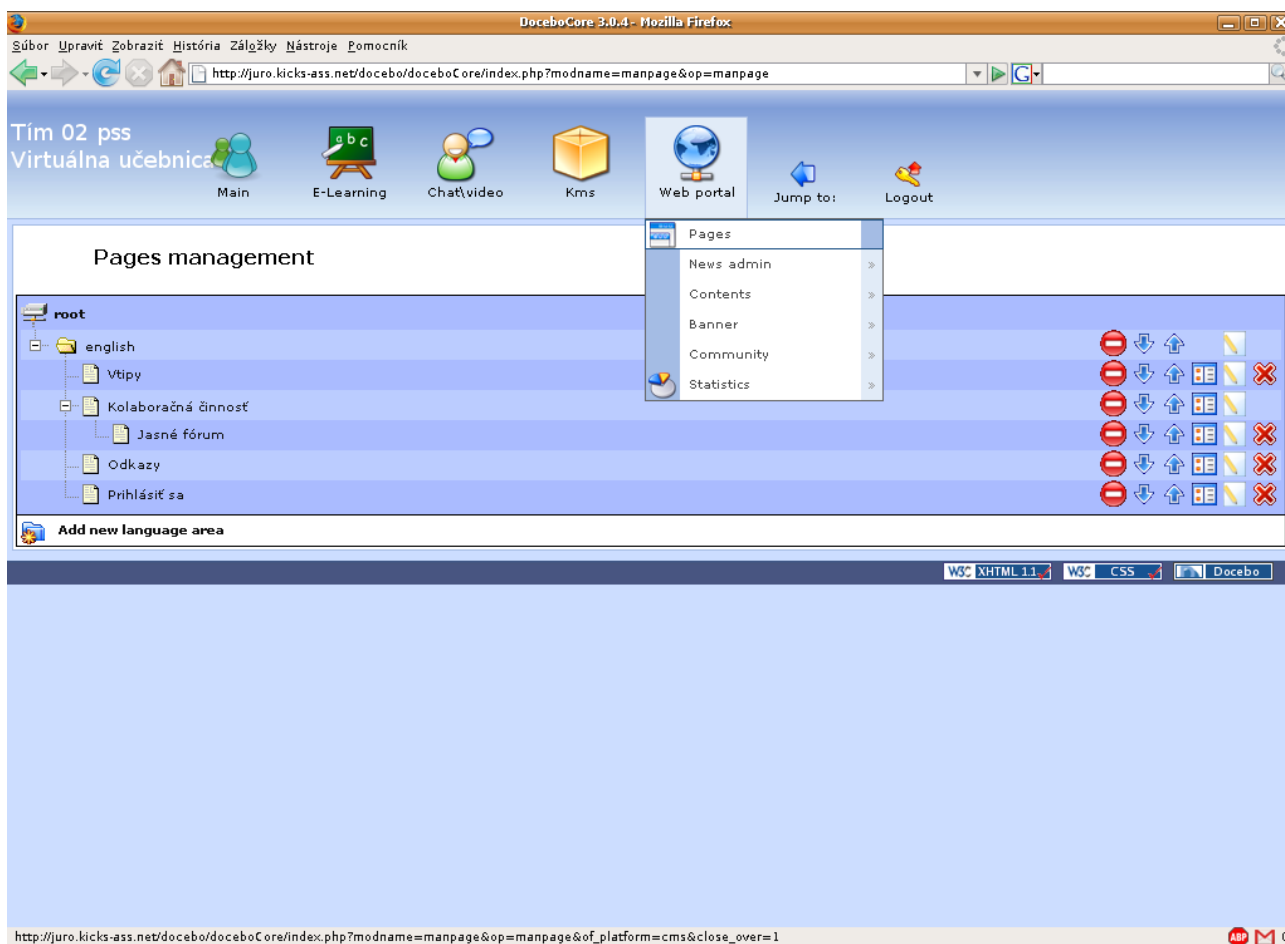
- [1] vyhľadávač Google
www.google.sk
- [2] Wikipédia
www.wikipedia.org
- [3] Moodle
<http://moodle.org/>
- [4] Grid
<http://gridgate.fiiit.stuba.sk/wiki/doku.php>
- [5] PHP maunál
<http://sk.php.net>
- [6] W3Schools
<http://www.w3schools.com/>

Príloha A: Používateľská príručka prototypu

Príručka poskytuje informácie potrebné pre pochopenie základných číť práce s prototypom implementácie Virtuálnej učebnice. Súčasné implementačné štádium umožňuje použitie najmä časti finálneho systému nazvanej CMS, ktoré umožňuje vkladanie, organizáciu a publikovanie obsahu pre zobrazenie na webovej stránke.

Vytváranie štruktúry údajov

Štruktúra údajov sa špecifikuje v položke menu *Web Portal > Pages*. Rozhranie je zobrazené na obrázku číslo 1.



Obr. 13 Rozhranie pre správu štruktúry CMS

Základnou vlastnosťou systému CMS je podpora viacerých jazykov pre prípad, že študenti pochádzajú z rôzne hovoriacich krajín. Tomuto faktoru je podriadená aj základná štruktúra vloženého materiálu, ktorá je implementovaná ako strom.

Koreňom stromu je položka nazvaná *root*. Do tejto položky je možné vložiť položky korešpondujúce s jazykmi, ktorých podpora je v systéme zapnutá. Jazyky sa vytvárajú v menu *Main*, ktorému sa budeme venovať v samostatnej kapitole. V rámci jazykov je možné vytvárať buď jedinečný obsah a samostatnú organizáciu údajov, alebo použiť rovnakú organizáciu pre viaceré jazyky. Jazyky sa pridávajú po označení koreňovej položky stlačením tlačítka myši, keď sa kurzor nachádza nad hypertextovým odkazom (ďalej len odkaz) s označením *Add new language area*. Následne si administrátor môže vybrať štandardný vzhľad pre jazykovú mutáciu z viacerých systémom ponúkaných.

Po vytvorení jazykovej mutácie môže správca vytvoriť ďalšiu úroveň organizačnej štruktúry a to oblasť nazvanú *macroarea* rovnakým spôsobom, ako vytvoril jazykovú mutáciu. Pri vytváraní oblasti je možné špecifikovať názov, ktorý sa zobrazí v hornom menu, je možné použiť iný vzhľad ako bol špecifikovaný pre celú jazykovú mutáciu, názov stránky ako ten, ktorý zobrazí návštevníkov web prehliadač, tak aj optimalizovaný pre vyhľadávacie systémy (napr. Google(r)). Navyše je možné uviesť krátky opis a kľúčové slová znovu pre lepšiu indexáciu vyhľadávačmi. Je možné vytvoriť oblasť, ktorá bude v skutočnosti len odkazovať na externú web stránku. Vtedy je nutné zvoliť možnosť *Use this page as a link to an external resource* a vpísať HTTP URI odkazovanej stránky do príslušného textového poľa.

Dôležitým bodom pri vytváraní oblasti je výber oprávnení na prezeranie oblasti. Je možnosť vybrať si z registrovaných používateľov, čo sú všetci tí, ktorí majú vytvorené konto v systéme, a anonymných, t.j. tí, ktorí konto nemajú. Overovanie používateľov prebieha pomocou jedinečného prihlasovacieho mena a hesla. Okrem týchto dvoch kategórií používateľov je možné určovať práva na základe príslušnosti k skupinám. Skupiny sa vytvárajú v položke menu *Main*.

Poslednou možnosťou voľby pri vytváraní oblasti je štruktúra stránky, ktorá bude oblasť reprezentovať. Na výber je niekoľko možností, ktoré sú intuitívne zobrazené podľa toho, ako vybraná štruktúra bude skutočne vyzeráť. Po vložení všetkých údajov je nutné nasledovať hypertextový odkaz *Save*. Tým sa správca dostane k vloženiu obsahu do štruktúry stránky. Obsah sa vkladá jednoduchým nasledovaním odkazu *Add*. Tam je možné vybrať si typ informačného poľa, ktoré

bude na danom mieste v štruktúre stránky. Na výber je mnoho typov, napr. reklamné spoty (*banner-y*), formuláre, polia so statickým textom, multimedialny obsah alebo menu, ktoré bude používateľa navigovať v rámci oblasti, prípadne aj v rámci všetkých oblastí. Po vybratí typu je nutné vybrať ho jednoduchým stlačením tlačidla myši. Správca bude následne upozornený, že požadovaná položka bola vytvorená. Po potvrdení tlačidlom *Continue* je možné upresniť detaily pridávaného obsahu. Jednotlivé polia sa líšia pre každý typ obsahu a ich názvy sú dostatočne informatívne, preto ich tu nebudeme všetky uvádzať. Za zmienku stojí snáď nastavenie časov publikácie a expirácie. Čas publikácie určí, kedy bude stránka automaticky vložená do systému a začne sa reálne zobrazovať návštevníkom CMS. Obdobne, ak nastane čas expirácie, stránka a teda aj samotná oblasť sa prestane zobrazovať návštevníkom.

V spodnej časti je možné nastavovať práva na jednotlivé pole oblasti obdobne ako pri oblasti samotnej.

Ďalšiu úroveň organizácie predstavujú web stránky. Na rozdiel od oblastí sa ale môžu vetviť na ďalšie web stránky a tak vytvárať neobmedzene hlbokú stromovú štruktúru. Postup ich vytvárania je rovnaký, ako vytváranie oblastí. Pri vytváraní web stránky je ale nutné mať myšou vybranú oblasť, do ktorej sa stránka pridá. Následne sa proces jej pridávania začne nasledovaním odkazu *Add webpage*.

Okrem automatickej publikácie je možné stránky publikovať (resp. expirovať) aj ručne. Toto sa vykoná namierením kurzora myši na zelenú šípku (resp. červený symbol jednosmernej cesty v protismere) v korešpondujúcom riadku a stlačením ľavého tlačidla.

Je možné meniť poradie všetkých častí CMS - od jazykových mutácií až po webstránky. Vykonávame to pomocou modrých šípok v posúvanom riadku vpravo. Nechcená časť CMS sa dá vymazať znakom červeného X. Vlastnosti a rozloženie (štruktúru) konkrétnej stránky sa dá meniť ikonami zo symbolmi štruktúry stránky a písacieho bloku s perom.

Vytváranie obsahu pre CMS

V tejto časti je popísaný postup pre vytváranie základných typov obsahu pre vloženie do systému CMS. Medzi tieto typy sa radia najmä:

- Novinky a ich kategórie
- Obsah
 - Obsah

- Dokumenty
- Odkazy
- Multimediálny obsah
- Bannery
- Fóra
- Ankety
- Formuláre

Postupy pre pridávanie tohto druhu obsahu sú veľmi podobné tým pri vytváraní oblastí a web stránok, preto tu ich podrobný opis nebudeme uvádzať. Ovládanie je intuitívne a používateľ si ho najlepšie osvojí priamou konfrontáciou so systémom. Uvádzame preto len organizáciu jednotlivých typov obsahu.

Novinky

Je možné vytvárať okruhy noviniek, takzvané *Topics*. Pre každý okruh je možné zvoliť si názov okruhu a umiestniť na server informatívny obrázok. Jednotlivé správy sa potom pri vytváraní priradia do niektorého z okruhov. Okrem hlavného priradenia je možné určiť novinke aj vedľajšie témy, ku ktorým bude patriť.

Pre každú novinku sa určuje autor, zdroj správy, krátky opis a dlhšie vysvetlenie a prípadný odkaz na externú informáciu týkajúcu sa novinky.

Po vytvorení novinky v systéme je možné priradiť jej pomocou tlačidla vo forme modrej spinky dokument, multimediálny obsah, alebo odkaz, ktoré sú vložené do systému. Všetky novinky sú organizované podobne ako celé CMS, teda do stromovej štruktúry. Medzi jednotlivými priečkami sa môžu presúvať pomocou tlačidiel v tvare zelených šípok ukazujúcich do kruhu.

Obsah

Obsah

Toto sú textové informácie, ktoré sa zobrazujú v štruktúre web stránok a oblastí. Vytvárajú sa a organizujú sa podobne ako oblasti a web stránky.

Dokumenty

Dokumenty sú počítačové súbory, ktoré sú neobmedzeného typu. V súčasnej implementácii je možné vkladať do systému súbory z lokálneho počítača s veľkosťou

do 2 MB. Pre každý súbor sa archivuje autor, autorova e-mailová adresa a adresa autorovej web stránky.

Multimediálny obsah

Multimediálny obsah môžeme charakterizovať rovnako ako Dokumenty, no s tým rozdielom, že tu sa vkladá výlučne multimediálny obsah a kvôli tomu je možné pre každý vkladajúci súbor vložiť aj vystihujúci súbor, napr. obrázok.

Odkazy

Do tejto sekcie sa vkladajú hypertextové odkazy na externé, ale aj interné internetové zdroje. Je možnosť voziť aj popisujúci súbor, napr. obrázok.

Bannery

Bannery sa organizujú do kategórií, ktoré správca môže vytvoriť na stránke, ku ktorej získa prístup cez položku menu *Web portal > Banner > Banner categories*. Následne pri vytváraní bannerov si môže vybrať, do ktorej kategórie sa má banner zaradiť. Na základe kategórií je neskôr možné zobrazovať ich používateľom.

Banner je možné vložiť ako jeden z troch objektov - obrázok, úsek HTML kódu a Macromedia Flash aplikácia. Banner sa priradí do kategórie a vyberie sa, kedy sa banner bude zobrazovať. Zobrazovanie sa určuje pre oblasti a pre jazykové mutácie. Pre každý typ banneru je nutné vložiť potrebné údaje - flash súbor, obrázok, alebo HTML kód.

Po vytvorení banneru sa pomocou tlačidla vo forme vizitky môžu meniť používatelia, ktorým sa banner bude zobrazovať - registrovaní a anonymní.

Fóra

Vo fóre spolu môžu komunikovať návštevníci stránky a navyše si môžu ku každej správe pripojiť nejaký súbor. Tak je zabezpečená interaktivita systému a podporuje sa tým prirodzená vlastnosť ľudí, ktorou je snaha spolupracovať so svojimi blízkymi na riešení spoločných problémov.

Pre každé vytvárané fórum sa určí popis a názov fóra. Po vytvorení je možné upravovať prístupové práva k fóru podobným spôsobom ako pri určovaní oprávnení pre oblasti a webové stránky (pomocou tlačidla s ikonou vizitky). Je možné určiť práva pre danú skupinu návštevníkov v nasledujúcich úrovniach - prezerať si obsah fóra, pridávať nové príspevky, moderovať správy a prikladať súbory. Navyše je ale

možné použiť pokročilé pridelovanie práv a vtedy sa práva pridelujú priamo konkrétnym používateľom registrovaným v systéme.

Ankety

Anketa sa vytvorí pomocou hypertextového odkazu *Add poll*. Následne sa zadá otázka, na ktorú bude návštevník CMS mať možnosť odpovedať. Po zadaní otázky je možné pridávať odpovede stlačením tlačidla so symbolom zoznamu. Na anketu používatelia odpovedajú výberom jednej z týchto odpovedí. Návštevníci si navyše môžu pozrieť výsledky hlasovania.

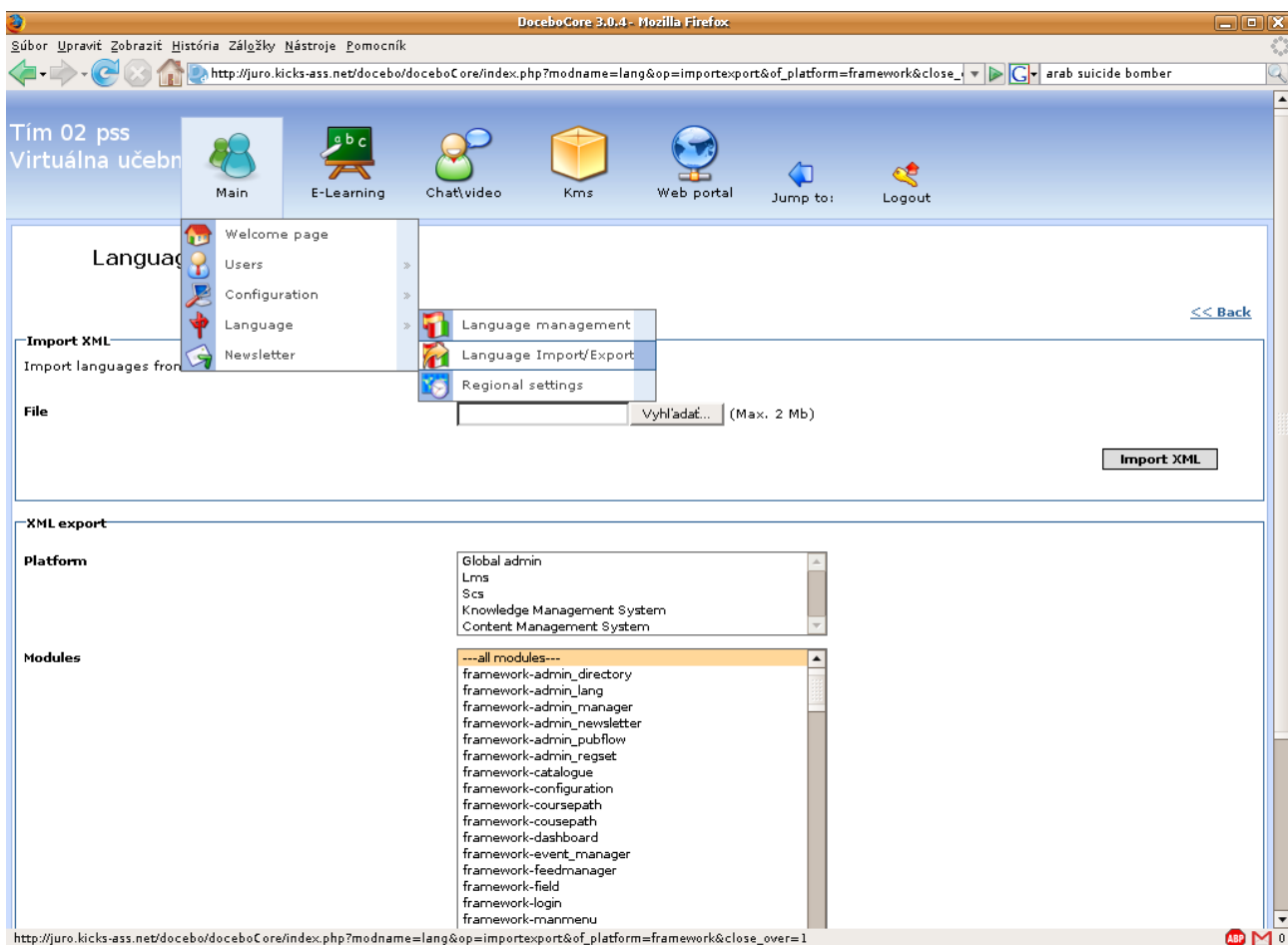
Formuláre

Formuláre nie je možné v prototyp vytvárať.

Podporované jazyky

Jazyky, ktoré budú podporované pre vytváranie jazykových mutácií CMS je možné pridávať z menu Main > Language > Language Import/Export. Jazyky sa pridávajú vo forme XML súboru, ktorý obsahuje všetky texty, s ktorými systém komunikuje s používateľom.

Ovládanie je intuitívne, vyberie sa jazykový súbor na lokálnom správcovom počítači a prenesie sa na server. Súbor je nutné vytvoriť prekladom napr. anglického vzoru, prípadne je možné nájsť takýto preložený súbor na internete. Rozhranie na import/export jazykových súborov je znázornený na Obr. 2



Obr. 14 Rozhranie pre import/export jazykov

Správa používateľov

Základom funkcionality pridelovania práv v CMS je existencia používateľských účtov. Používatelia sa do systému pridávajú pomocou rozhrania prístupného cez položku menu Main > Users > Users Management. Je možné vytvoriť nových používateľských účtov, ktorým treba prideliť unikátne prihlasovacie meno, adresu elektronickej pošty, aby mu neskôr bolo možné poslať upozornenia a novinky, heslo a úroveň používateľa. Úroveň používateľa je jedna z možností - obyčajný používateľ (*User*), správca (*Admin*) a všemocný správca (*Super Admin*). Rozsah možností poskytovaných systémom používateľovi je priamo úmerný tejto úrovni.

Už existujúce používateľské účty je možné zablokovať v prípade podozrenia zo zneužitia, prípadne z iných dôvodov - pomocou ikony zámku v príslušnom riadku tabuľky s používateľmi.