

System pre správu rozširujúcich modulov jazyka Lua

Tímový projekt

Vedúci projektu: Ing. Peter Drahoš

Autori:

Bc. Karol Marton

Bc. Jakub Marton

Bc. Miloš Cibulka

Bc. Michal Dávid

Bc. Marián Halaš

Ak. rok: 2008/2009

Úvod

Tento dokument vznikol ako výsledok práce tímu číslo 18 na predmete Tvorba informačného systému v tíme I počas zimného semestra. Zadaním projektu bolo vytvoriť systém pre správu rozširujúcich modulov jazyka Lua.

Dokument je rozdelený na dve časti, prvá z nich sa venuje samotnému systému, jeho analýze, špecifikácii, návrhu a prototypovaniu. V druhej časti s názvom Riadenie projektu obsahuje ponuku, plán projektu, zápisnice z tímových stretnutí, úlohy členov tímu a ďalšie informácie ohľadom manažmentu projektu.

História vývoja dokumentu

Dátum zmeny	Verzia dokumentu	Opis	Autor
3.11. 2008	0.1	Spracovanie kapitoly 1	Bc. Marián Halaš
5.11. 2008	0.2	Úprava kapitoly 1.2, 1.3	Bc. Marián Halaš
7.11. 2008	0.3	Pridanie kapitoly 2	Bc. Karol Marton Bc. Jakub Marton
8.11. 2008	0.4	Doplnenie Rewiki do kapitoly 2	Bc. Jakub Marton
8.11.2008	0.5	Kompletizácia kapitoly 2.	Bc. Marián Halaš
9.11.2008	0.6	Pridanie kapitoly 3.	Bc. Miloš Cibulka
9.11.2008	0.7	Pridanie kapitoly 4.	Bc. Miloš Cibulka Bc. Michal Dávid
9.11.2008	0.8	Formátovanie, úprava štýlov	Bc. Marián Halaš
11.11.2008	0.9	Zmena niektorých diagramov, obrázkov, doplnenie workflow diagramu k LuaDistu ako aj doplnenie opisu modulu .dist	Bc. Marián Halaš Bc. Miloš Cibulka Bc. Michal Dávid
12.11.2008	1.0	Doplnenie informácií o LuaDiste, formálne úpravy, doplnenie záveru, presunutie odkazov z kapitoly 1. Na koniec dokumentu.	Bc. Marián Halaš
4.12.2008	1.1	Úprava a doplnenie kapitoly 4.3	Bc. Karol Marton
6.12.2008	1.2	Kapitola 5.	Bc. Karol Marton Bc. Michal Dávid
8.12.2008	1.3	Doplnenie podkapitoly 5.3	Bc. Miloš Cibulka
8.11.2008	1.3	Pridaná podkapitola 5.4	Bc. Jakub Marton
20.4.2009	1.4	Zmena dátového modelu podľa novej funkčnej špecifikácie a implementácie	Bc. Karol Marton
21.4.2009	1.5	Pridanie kapitoly 6, testovanie	Bc. Marián Halaš
22.4.2009	1.6	Pridanie používateľskej príručky	Bc. Michal Dávid
23.4.2009	1.7	Doplnenie používateľskej príručky	Bc. Karol Marton
24.4.2009	1.8	Pridanie kapitoly Produkt	Bc. Jakub Marton
25.4.2009	1.9	Pridanie Technickej dokumentácie	Bc. Miloš Cibulka
28.4.2009	2.0	Finálne úpravy a formátovanie	Bc. Marián Halaš

Obsah

1	Úvod	3
1.1	Účel a rozsah dokumentu	3
1.2	Prehľad dokumentu	3
1.3	Slovník pojmov problémovej oblasti	3
2	Analýza	4
2.1	Lua	4
2.1.1	Hlavné výhody jazyka lua	5
2.2	LuaDist	5
2.2.1	Inštalácia modulu	6
2.2.2	Štruktúra .dist súboru	9
2.3	Wiki systémy	10
2.3.1	Nanoki	11
2.3.2	Reviki	11
2.3.3	WikkaWiki	12
3	Špecifikácia požiadaviek	14
3.1	Prípady použitia	16
3.1.1	Opis jednotlivých prípadov použitia	20
4	Návrh	23
4.1	Workflow diagramy	23
4.2	Návrh obrazoviek	35
4.3	Model údajov	43
4.3.1	Logický model údajov	43
4.3.2	Opis logického modelu údajov	44
4.3.3	Opisy atribútov entít	44
4.3.4	Fyzický model údajov	48
4.3.5	Zmeny oproti modelu WikkaWiki	49
4.3.6	Zmeny oproti zimnému semestru	49
5	Prototyp	51

Funkcionalita prototypu	51
5.1 Zmeny v dátovom modeli	51
5.2 Registrácia nového užívateľa	52
5.3 Upload súboru	52
5.4 Generovanie manifestu a získanie informácií z dist.info	52
6 Produkt	54
6.1 Zmeny oproti návrhu	54
6.2 Výber implementačného prostredia	54
6.3 Realizácia a implementácia	54
6.3.1 Realizácia UC1 – vloženie nového balíčka	55
6.3.2 Realizácia UC2 – editácia existujúceho balíčka	55
6.3.3 Realizácia UC3 – prezeranie balíčkov a UC11 zmazanie balíčka	56
6.3.4 Realizácia UC4, UC7 a UC8 – hlasovanie, komentovanie balíčka	56
6.3.5 Realizácia UC6 – prístupnenie/zverejnenie balíčka	57
6.3.6 Realizácia UC10 – registrácia nových užívateľov	57
6.3.7 Realizácia UC12 – správa kategórií	57
6.4 Možnosti rozšírenia	57
7 Testovanie	58
7.1 Testovacie scenáre	58
8 Záver	62
9 Použitá literatúra	63

1 Úvod

1.1 Účel a rozsah dokumentu

Dokument obsahuje špecifikáciu a opis systému pre správu rozširujúcich modulov jazyka Lua v rámci predmetu Tvorba informačného systému v tíme I v prvom kontrolnom bode, čiže analýzu, špecifikáciu a hrubý návrh tohto systému.

1.2 Prehľad dokumentu

Analýza problému sa nachádza v 1.kapitole. Obsahuje stručný prehľad problémovej oblasti, analýzu viacerých wiki systémov a ich zhodnotenie podľa vhodnosti k projektu. Kapitola 2 obsahuje špecifikáciu požiadaviek na vytváraný systém vo forme modelu prípadov použitia. V kapitole 3 sa nachádza hrubý návrh systému, dátové modely a workflow diagramy, takisto aj návrhy obrazoviek GUI.

1.3 Slovník pojmov problémovej oblasti

<i>Lua</i>	programovací jazyk, dizajnovaný ako skriptovací jazyk s rozširiteľnou sémantikou. Po portugalsky <i>Mesiac</i>
<i>LuaDist</i>	distribúcia jazyka Lua, ktorá poskytuje prostredie a nastavenia pre správu rozširujúcich modulov
<i>Garbage collector</i>	algoritmus, procedúra alebo vlákno zodpovedné za uvoľňovanie prostriedkov, ktoré sa už nepoužívajú.
<i>Virtual machine</i>	softvérový virtuálny počítač nezávislý od operačného systému

2 Analýza

V tejto kapitole sa venujeme analýze problematiky jazyka Lua, distribúcie LuaDist, nad ktorými má náš systém správy rozširujúcich modulov fungovať. Ďalej sa venujeme wiki systémom, pretože najlepšie spĺňajú požiadavky na tento systém, venujeme sa rôznym typom wiki systémov, ich výhodám a nevýhodám, skúmame ich vhodnosť pre použitie na tomto projekte.

2.1 Lua

Lua [7] je jednoduchý programovací jazyk navrhnutý ako skriptovací jazyk. Väčšina programovacích jazykov funguje tak, že knižnice sa nejakým spôsobom pripájajú k existujúcemu jadru jazyka. V jazyku Lua to funguje naopak. Lua je taká malá a jednoduchá, že sa jednoducho celá pripojí k existujúcim knižniciam jazyka C alebo C++. Treba len napísať a skompilovať vlastné rutiny zaisťujúce komunikáciu medzi jazykmi Lua a C. Toto je veľmi dobre zdokumentované aj z dôvodu, že je to jedna z hlavných myšlienok tohoto jazyka.

Lua je veľmi malá, celý jej zdrojový kód je aj s príkladmi a dokumentáciou v 200KB archíve. To samozrejme ovplyvňuje jej funkcionálnosť. Okrem štandardných typov v podstate len jednu špeciálnu dátovú štruktúru – asociatívnu tabuľku, pole indexované vlastne čímkol'vek. Lua umožňuje pomocou takýchto „metatabuliek“ rozširovať samú seba, takže pomocou jednoduchých príkazov možno definovať objektovosť a dedičnosť, ktoré Lua sama od seba nemá. Takým istým spôsobom možno definovať vlastné dátové štruktúry. Všetky knižnice jazyka C sú potom prístupné Lua. Lua je v značnej miere inšpirovaná funkcionálnymi jazykmi.

Lua je interpretovaný jazyk s automatickou kompiláciou do bytekódu a tým pádom je veľmi rýchla, rýchlejšia ako napríklad *Ruby* alebo *Python*. [8] Veľkou výhodou jazyka Lua je aj to, že je interpretovaný podobne ako Java na *virtual machine*, ktorá takisto má implementovanú funkcionálnosť *garbage collector* čím sa značne zjednodušuje programovanie a odladovanie programov (netreba sa starať o alokáciu a pod.). Takisto je veľkou výhodou použitie korutín, tým odpadávajú problémy s multithreadingom.

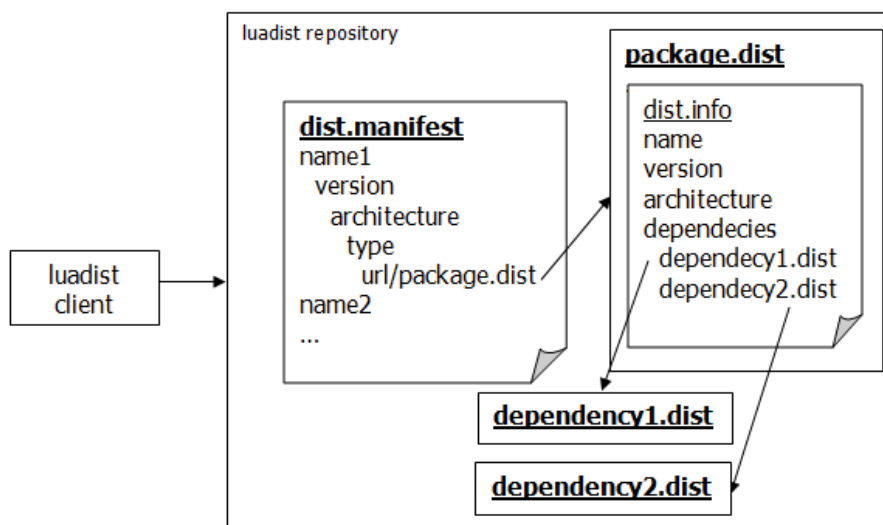
2.1.1 Hlavné výhody jazyka lua

Lua sa využíva hlavne v hernom odvetví. Napríklad vo videohrách je bežné, že multimediálne rutiny sú naprogramované v jazyku C a logiku hry riadi Lua. Dnes je rozšírený hlavne v počítačových hrách, ale bol vo veľkej miere použitý napríklad aj v programe Adobe Photoshop.

- Prenositel'nosť
- Efektívnosť a malá veľkosť
- Jednoduchosť vnárania (Lua a C môžu byť použité tam kde sú najviac vhodné)
- Jednoduchosť
- Stabilita
- Korutiny nezávislé od OS
- Rozšíriteľnosť
- Procedurálna reprezentácia dát

2.2 LuaDist

LuaDist [12] je distribúcia jazyka Lua. Implementuje správu modulov pre tento jazyk, ktoré sú potrebné pre vývoj aplikácií v tomto jazyku. Z repozitára na webe stiahne potrebné zdrojové a/alebo binárne súbory, pripojí ich do aktuálneho prostredia jazyka Lua. Repozitár je umiestnený na webe. LuaDist má cestu k tomuto repozitáru uloženú v konfiguračnom súbore.



Obr. č. 1 Architektúra repozitáru LuaDist projektu

Na strane servera je uložený manifest.dist súbor, v ktorý obsahuje cestu ku všetkým modulom, ktoré repozitár obsahuje a základné informácie o každom z nich, meno, verziu, OS a architektúru. Každý modul je uložený ako zip archív a obsahuje súbor dist.info. V tomto súbore sú znova nasledovné kľúčové informácie: meno, verzia, OS a architektúra a ešte iné potrebné informácie ako napríklad závislosti na iných moduloch.

Keď klient LuaDistu kontaktuje server s požiadavkou na nejaký modul, tento je vyhľadaný v manifeste repozitára a na základe uvedenej cesty je poslaný klientovi a s ostatnými potrebnými modulmi, ktoré sú uvedené v dist.info.

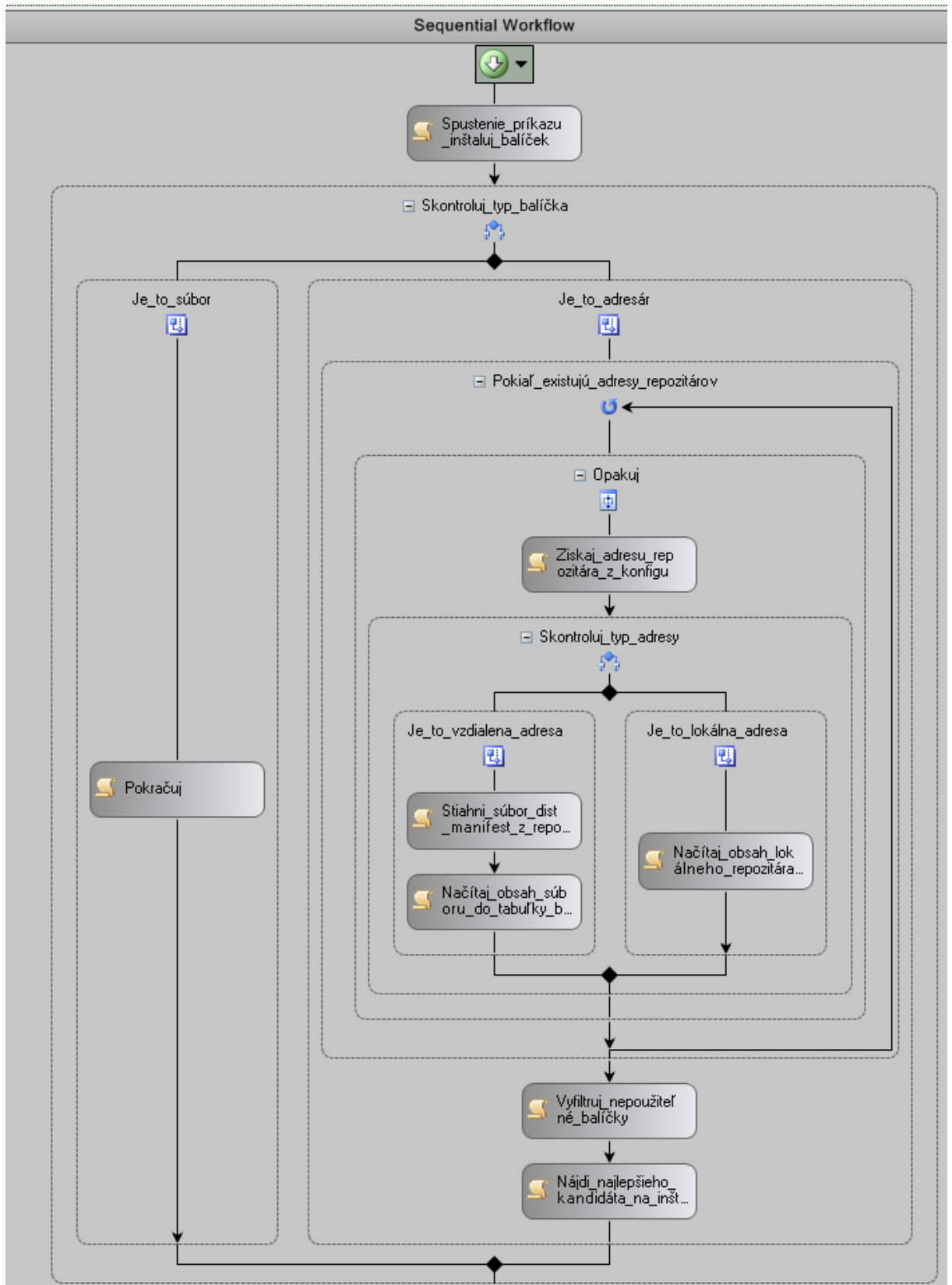
2.2.1 Inštalácia modulu

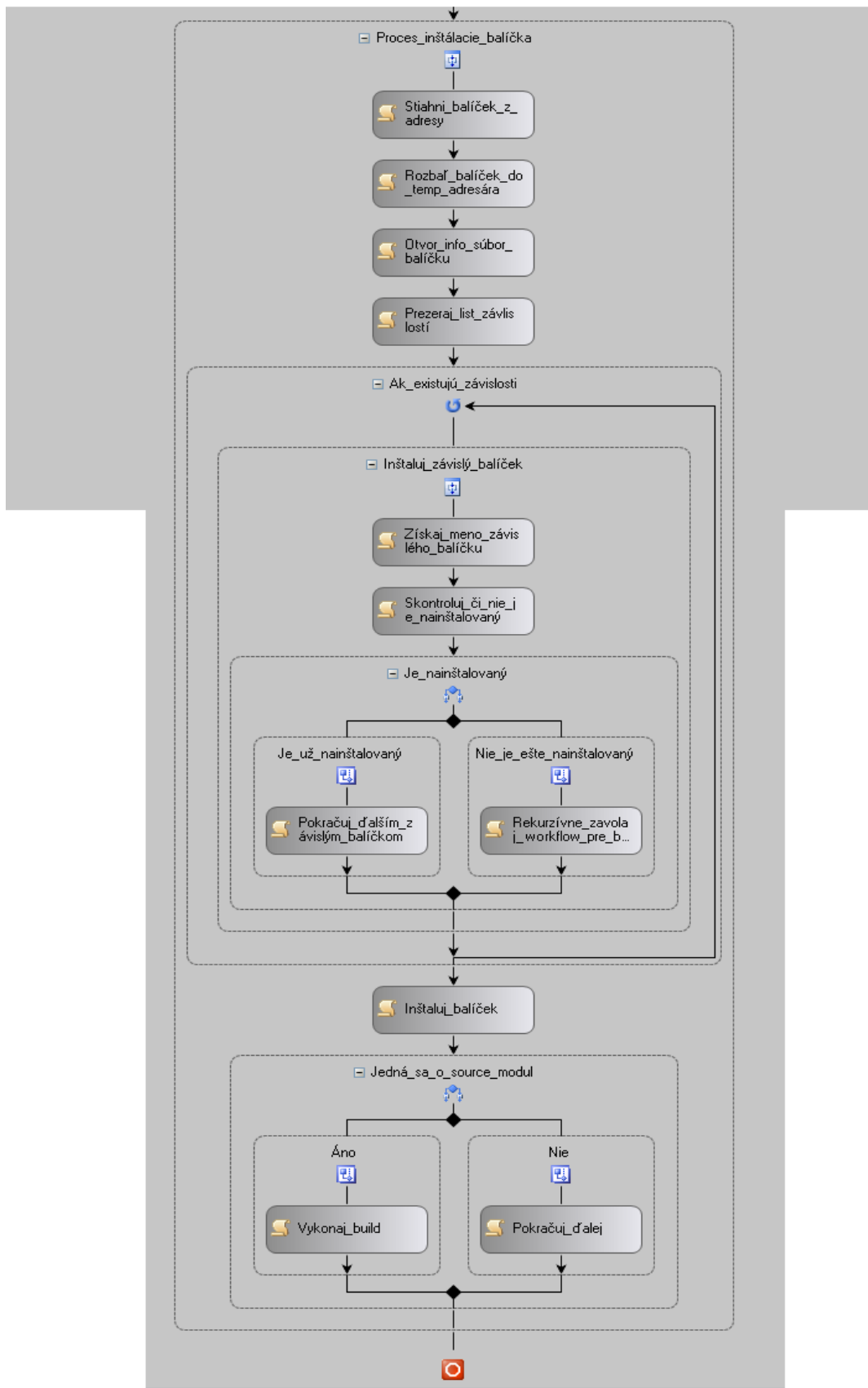
Následný diagram činností na obrázku č. 2 ukazuje, ako sa správa LuaDist počas inštalácie modulu a ako pri tomto procese využíva repozitáre. Inštaláciu ľubovoľného modulu začína užívateľ na svojom osobnom počítači jednoduchým príkazom „luadist install meno_modulu“.

Luadist najprv kontroluje, či zadané meno balíčka je súbor, alebo adresár. Ak je to súbor, vynecháva sa nasledujúca selekcia a pokračuje sa priamo inštaláciou balíčka. Ak je to adresár, prezrie všetky repozitáre, ktorých adresy má uvedené v konfiguračnom súbore. Ak je adresa vzdialená (má prefix http://), stiahne sa preň jeho dist.manifest súbor a jeho údaje sa načítajú do tabuľky. Lokálne adresy načítava taktiež do tejto tabuľky. Nasleduje filtrácia nepoužiteľných balíčkov z tabuľky a za tým pokračuje nájdením najlepšieho kandidáta na inštaláciu z vyfiltrovaných zdrojov podľa konfigurácie.

Ďalej proces pokračuje inštaláciou balíčka. Nájdený súbor sa teda stiahne a rozbalí do temporary adresára. Načíta sa dist.info súbor v ktorom sú obsiahnuté informácie o balíčku a taktiež sa tam dajú nájsť odkazy na závislosti, tzn. ktoré balíčky musia byť nainštalované pre správny beh stiahnutého balíčku.

Ak teda existujú závislé balíčky, inštalátor sa pokúsi nainštalovať najprv tie. Rekurzívne teda zavolá celý proces s každým závislým balíčkom. Po nainštalovaní všetkých závislých balíčkov nainštaluje aj samotný balíček a ak sa jedná o zdrojový balíček, balíček zkompiluje.





Obr. č. 2 Workflow diagram pre inštaláciu modulu

2.2.2 Štruktúra .dist súboru

Ako už bolo spomínané, v repozitároch sa nachádzajú moduly vo forme .dist súborov. Takýto súbor je vlastne len skomprimovaný priečinok a slúži na distribúciu zdrojového alebo binárneho modulu. Pre ilustráciu, ako taký .dist súbor vyzerá rozoberieme štruktúru .dist súboru pre modul *curl-7.18.1-Windows-x86*.

- *Include*
 - *Curl*
 - *Curl.h*
 - *Curlver.h*
 - ...
 - *Config-win32.h*
- *Lib*
 - *Libcurl.a*
 - *Libcurl.dll*
 - *Libcurl.dll.a*
- *Share*
 - *Curl*
 - *Doc*
 - *Etc*
 - *Include*
- *Dist.info*

Obsah súboru *dist.info*:

- Povinné údaje
 - name = "curl"*
 - version = "7.18.1"*
 - arch = "Windows"*
 - type = "x86"*
- Voliteľné údaje
 - dependencies = {*
 - openssl = ">=0.9.8g"*
 - }*

Tab. č. 1 - Opis jednotlivých položiek v dist.info súbore

Meno	Povinné	Opis
Name	Áno	Unikátne meno modulu. Odporúča sa používať malé písmená a vyhýbať sa znaku „-“
version	áno	Táto položka špecifikuje verziu modulu.
Arch	Áno	Určuje pre ktorú architektúru OS je modul určený. Ak ja nastavené na <i>universal</i> tak je určený pre všetky architektúry. (príklady: Cygwin, Darwin, Windows, Linux...)
type	áno	Umožňuje rozlíšiť typ architektúry (32bit/64bit)
short	nie	Krátky informatívny opis modulu
full	nie	Obširny opis modulu, môže byť neobmedzenej dĺžky.
author	nie	Reťazec autorov oddelených čiarkami.
maintainer	Nie	Reťazec správcov oddelených čiarkami.
Homepage	Nie	URL adresa modulu
Dependencies	Nie	Špecifikuje závislosť funkčnosti modulu na iných moduloch. Formát: <code>[name] = "[constraints]"</code> (napríklad <code>lua = ">=5.1.0 <=5.1.4"</code>) , povolené operátory : ">", "<", ">=", "<=" a "="
Dependencies.external	nie	To isté ako depenecies ale pri neúspešnom vyhľadání a stiahnutí takéhoto modulu inštalácia pokračuje ďalej.

2.3 Wiki systémy

V tejto kapitole opisujeme existujúce systémy, ktoré sme analyzovali z pohľadu vhodnosti pre náš projekt. Všetky sú založené na wiki prístupe kvôli rozšíriteľnosti a čo najmenším obmedzeniam. Keďže používateľ si bude stránky k svojmu balíčku vytvárať sám, je takýto prístup viac ako potrebný.

2.3.1 Nanoki

Nanoki [4] je wiki systém implementovaný v jazyku Lua s otvoreným kódom. Vyniká minimálnymi požiadavkami na server, na ktorom beží. Stačí, keď jeho obsah iba prekopírujete na server a spustíte jeden proces. Ten v sebe zahŕňa aj *http server*, ktorý obsluhuje *http requesty*. Ako databázu využíva súbory, do ktorých ukladá potrebné údaje. V tabuľke 1. sú prehľadne zobrazené ďalšie identifikované výhody a nevýhody tohto systému z pohľadu vhodnosti pre náš systém správy balíčkov.

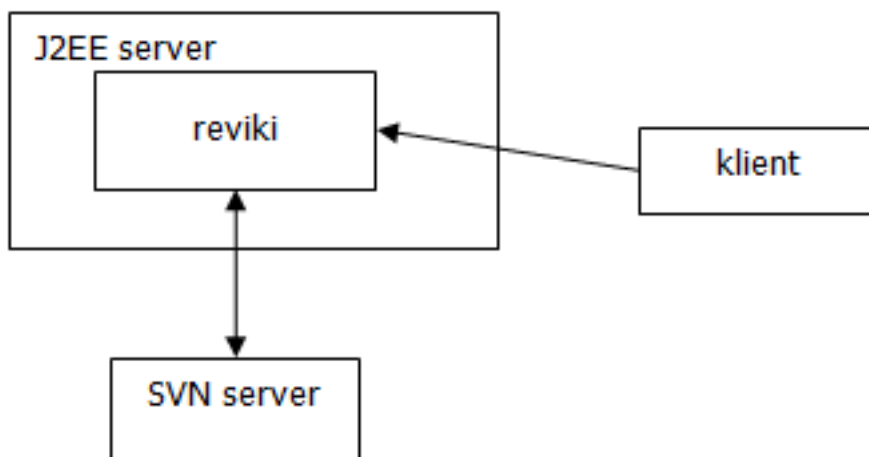
Tab. č. 2 - Výhody a nevýhody wiki systému Nanoki

Výhody	Nevýhody
<ul style="list-style-type: none">• Minimálne požiadavky (obsahuje v sebe http aj file server)• Otvorený zdrojový kód• Ľahká wiki syntax• Už implementovaná funkcia nahratia súboru na server• Prehľadná správa revízií stránok podľa dátumu• Dobrá prevencia proti spamu• Podpora rss	<ul style="list-style-type: none">• Potrebné shell konto na serveri, kvôli spusteniu interpretera jazyka lua• Nový a málo známy jazyk Lua, ťažší a nepochopiteľnejší kód• Neexistujúce práva na jednotlivé stránky, ktokoľvek môže editovať ľubovoľnú stránku• Neexistujúca dokumentácia

Na základe silných nevýhod ako je neexistujúca dokumentácia a nový jazyk Lua, sme túto možnosť odhlasovali ako nevyhovujúcu.

2.3.2 Reviki

Reviki [5] je wiki-systém implementovaný v J2EE. Na ukladanie všetkých svojich zdrojov a správu ich verzií využíva subversion. Subversion (SVN) je systém široko používaný pri vývoji projektov. Reviki realizuje SVN operácie použitím svnkit.jar knižnice.



Obr. č. 3 Architektúra wiki systému Rewiki

Výhodou tohto systému je samotné SVN. Repozitár SVN je možné zverejniť, čo prináša ďalšiu funkcionálnosť. Používatelia môžu priamo spravovať zdroje wiki, bez priameho editovania cez web rozhranie. Výhodou pre náš systém by bola možnosť vývoja balíčkov priamo u nás, v našom SVN. Pre nás je výhodou familiárne prostredie java jazyka.

Nevýhodou je potreba mať SVN, čo uberať na požiadavke minimálnych nárokov. Rovnako potrebuje J2EE server kontajner, a voľných hostingov, ktoré poskytujú túto službu je málo.

Tab. č. 3 - Výhody a nevýhody systému *reviki*.

Výhody	Nevýhody
<ul style="list-style-type: none"> • implementácia v jave, čo je prostredie nám známe, • SVN funkcionálnosť, jednoduché pridanie repozitára pre každého používateľa, • systém pre pridávanie pluginov, • podpora rss, • WikiCreole syntax pre písanie stránok, zvýrazňovanie kódu, • implementované vyhľadávanie, 	<ul style="list-style-type: none"> • závislosť na java a SVN technológií, • slabšia dokumentácia, • autentifikačný systém je riešený tiež prostredníctvom SVN

Pre veľkú závislosť na technológiách, ktoré síce poskytujú slušnú funkcionálnosť, tento systém bol zamietnutý, nakoľko nespĺňa prioritnú požiadavku jednoduchej prenosnosti.

2.3.3 WikkaWiki

Tento wiki systém [6] je implementovaný s jazyku php s použitím objektovo – orientovaného prístupu. Ako úložisko údajov využíva databázu MySQL. Má otvorený kód a voľne šíriteľnú GPL licenciu. Na jeho spustenie je potrebný server s nainštalovaným jazykom php, neplatenou databázou MySQL a bežiacim http serverom Apache. Oproti systému Nanoki je to nevýhoda, ale v dnešnej dobe je kombinácia Apache + PHP + MySQL štandardom na väčšine serverov, ktoré sú určené na hosting webových stránok.

Práve existujúci shell prístup, ktorý vyžaduje Nanoki je problémom, lebo je to alternatívna služba, za ktorú je potrebné zaplatiť príplatok. V tabuľke 3 sú prehľadne zobrazené ďalšie identifikované výhody a nevýhody tohto systému z pohľadu vhodnosti pre náš systém správy balíčkov.

Tab. č. 4 - Porovnanie výhod a nevýhod wiki systému WikkaWiki

Výhody	Nevýhody
<ul style="list-style-type: none"> • Vyžaduje APACHE + MySQL + PHP, čo je dnes poskytovaný štandard u väčšiny hostingov • Otvorený zdrojový kód • Existujúci systém používateľov a práv na jednotlivé stránky • Už existujúce funkcie ako je nahranie súboru na server, systém komentárov • Jednoduchý objektovo – orientovaný kód • Prehľadný databázový systém - iba 8 tabuliek v databáze • Existujúca dokumentácia • Údržba a podpora autorov • Mnoho dobrých referencií 	<ul style="list-style-type: none"> • Potrebné APACHE + MySQL + PHP, čo mierne narušuje požiadavku minimálnych nárokov • Ťažšia prenositeľnosť • Nahrať súbor na server môže iba administrátor

Po zvážení všetkých identifikovaných výhod a nevýhod analyzovaných wiki systémov, sme sa rozhodli pre použitie práve tohto. Hlavné dôvody sú nasledovné:

- Konsenzus medzi funkcionalitou a požiadavkami na server
- Najlepšia dokumentácia a najjednoduchší kód
- Systém (APACHE + PHP + MySQL), s ktorým má každý z nás isté skúsenosti

Po tomto rozhodnutí sme si založili nový projekt na serveri code.google.com a vytvorili si svn repozitár. Každý z nás si nainštaloval WikkaWiki lokálne pre ďalšie potreby implementácie.

3 Špecifikácia požiadaviek

Táto kapitola obsahuje požiadavky na vytváraný informačný systém. Je rozdelená na dve časti. Prvá časť obsahuje súhrn požiadaviek na systém v prehľadnej tabuľke. Druhá časť obsahuje špecifikáciu požadovaného riešenia vo forme modelu prípadov použitia. Funkcionálne požiadavky

V nasledujúcej tabuľke 4 sú opísané funkcie, ktoré systém poskytuje. V prvom stĺpci je názov, potom opis a nakoniec používateľa funkcie, ktorí budú mať k nej prístup a budú ju môcť využívať.

Tab. č. 5 - Opis funkcií systému

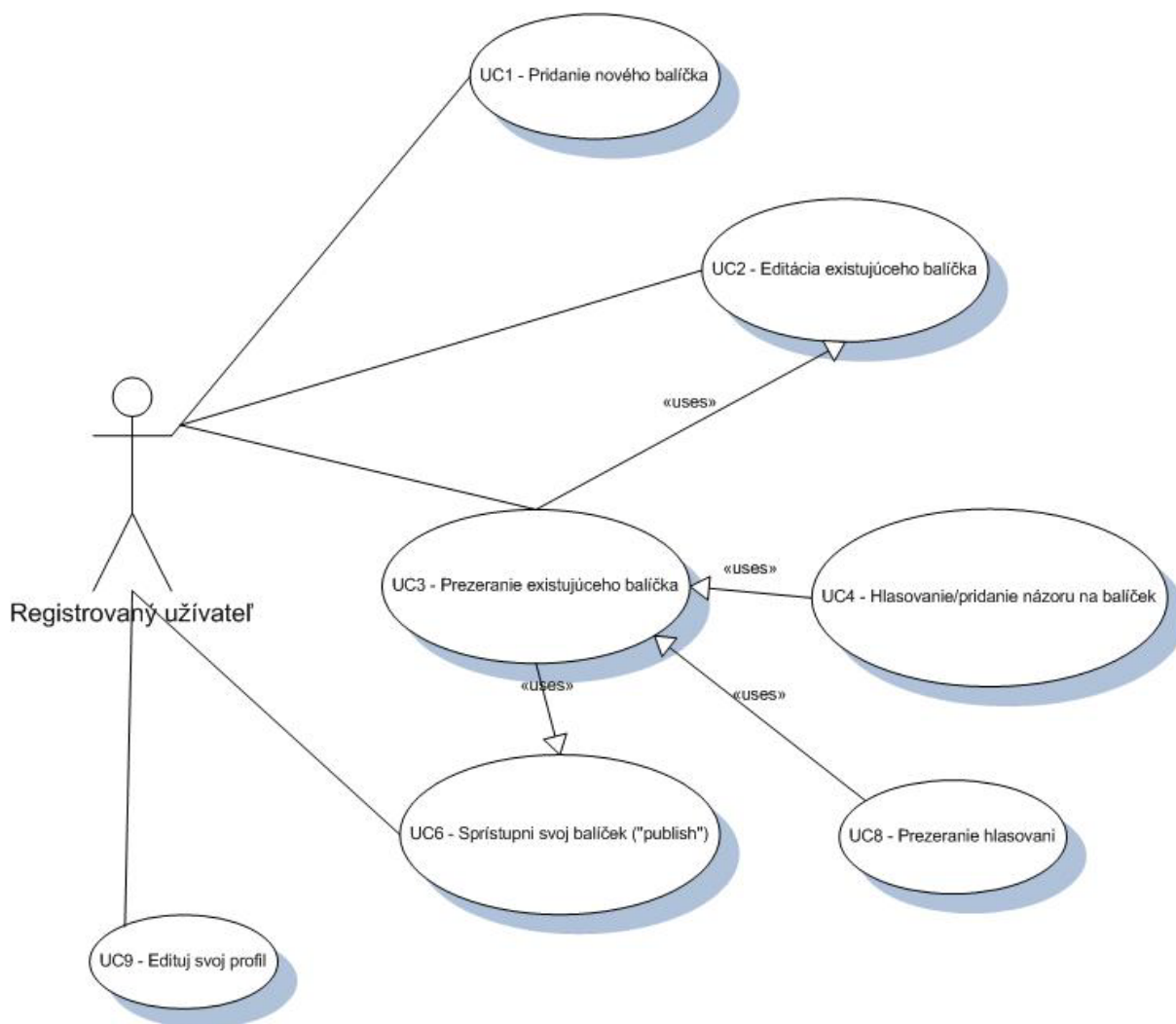
Funkcia	Opis	Používateľ
Autentifikácia používateľa	na úvodnej stránke je formulár na vstupné údaje – prihlasovacie meno a heslo	všetci
Registrácia nového používateľa	na úvodnej stránke bude odkaz na registračný formulár	všetci
Editácia vlastného používateľského profile	možnosť nastaviť si preferencie ako posielanie notifikačných e-mailov, zmena e-mailu, hesla, pridanie používateľov, s ktorými bude zdieľať svoje repozitár...	prihlásený používateľ, administrátor
Pridanie balíčku do databázy	odkaz pri zozname balíčkov	prihlásený používateľ, administrátor
Prezeranie zoznamu balíčkov	keď používateľ nechce hľadať balíčky pomocou vyhľadávania, ale potrebuje ich zobrazit' všetky. Pri názve balíčka sa zobrazuje aj platforma, opis, stav a taktiež počet stiahnutí, počet bodov v hodnotení a počet komentárov	všetci

<p>Prezeranie konkrétneho balíčka</p> <ul style="list-style-type: none"> • hodnotenie kliknutím na piktogram 	<p>počet bodov po zahlasovaní:</p> <ul style="list-style-type: none"> • prihlásený(+2 - 2), • administrátor(+3 -3), • neprihlásený(+1 -1) 	všetci
<ul style="list-style-type: none"> • zobrazenie základných informácií o balíčku 	názov, aktuálne hodnotenie, počet stiahnutí, opis, autor...	všetci
<ul style="list-style-type: none"> • zmena jeho stavu 	unstable → stable, dead	administrátor
<ul style="list-style-type: none"> • publikovanie balíčka 	private → unstable	prihlásený používateľ
<ul style="list-style-type: none"> • aktualizácia - t.z. pridanie novšej verzie 	<ul style="list-style-type: none"> – zobrazuje sa posledná stabilná verzia s možnosťou zobrazenia novších nestabilných s upozornením, že verzia ešte nebola otestovaná – ak by ešte nebola žiadna stabilná verzia, tak sa zobrazuje posledná pridaná s možnosťou si prezrieť všetky 	prihlásený používateľ, administrátor
komentovať jednotlivé balíčky	komentáre sa zobrazujú pod zobrazením jedného balíčka	prihlásený používateľ, administrátor
vyhľadávanie balíčka	user si bude moct vybrať fulltext vo vsetkych polozkach alebo hladat podľa platformy, nazvu, stable, unstable... proste advanced search	všetci
posielanie e-mailových upozornení autorom, keď niekto	automaticky pomocou systému	prihlásený používateľ, administrátor

vylepší ich balíček		
Vypĺňanie captcha pri potvrdzovaní formuláru na pridávanie nového balíčka	ešte pri registrácii nového používateľa a taktiež aj pri pridávaní novej verzie balíčka kvôli spam robotom	všetci
Načítanie informácií o balíčku z dist.info súboru	pri kopírovaní balíčka na server sa .zip súbor rozbalí a načíta práve tento súbor	prihlásený používateľ, administrátor

3.1 Prípady použitia

Nasledujúca kapitola opisuje možnosti využitia systému jeho používateľmi. Užívatelia sú rozdelení do troch kategórií, ktoré určujú ich pravidlá používania systému. Preto sú prípady použitia rozdelené do troch diagramov, pre každú kategóriu používateľov existuje jeden diagram.

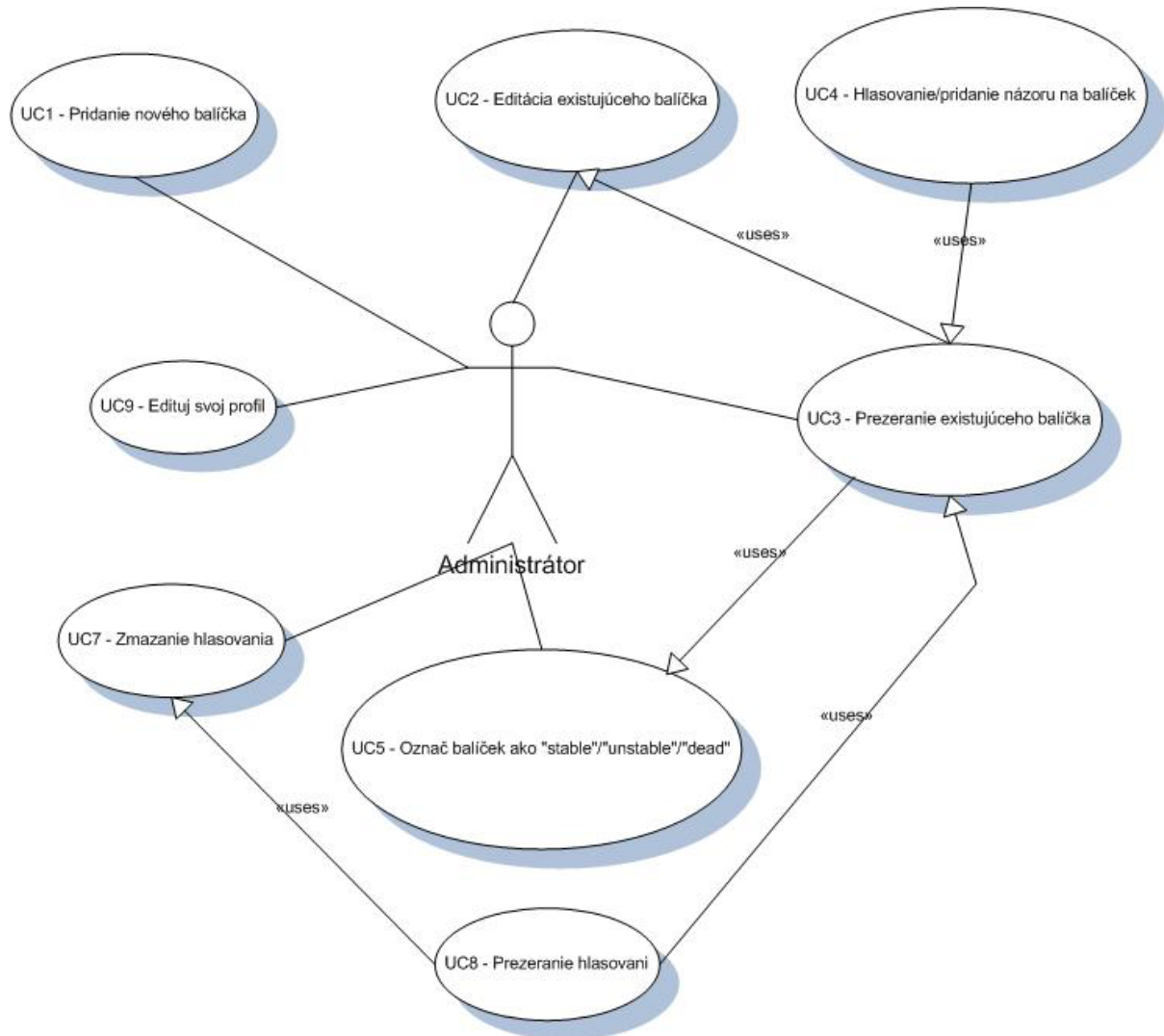


Obr. č. 4 Diagram prípadov použitia pre registrovaného užívateľa.

Registrovaný užívateľ je typ hráča, ktorý bude určený najmä pre vývojárov balíčkov, ktorý balíčky nie len sťahujú ale aj vytvárajú, príp. modifikujú už existujúce balíčky. Registrovaný užívateľom sa stane každý neregistrovaný užívateľ, ktorý sa zaregistruje prostredníctvom formulára (UC10 - Registrácia). Po tejto registrácii bude môcť využívať nasledovné možnosti systému:

- Pridanie nového vlastného balíčka – UC1
- Editovanie existujúcich balíčkov, ktoré sú mu prístupné – UC2
- Prezeranie existujúcich balíčkov, ktoré sú v systéme evidované – UC3 + UC4 + UC8
- Vyjadriť svoj názor v diskusií pre všetky balíčky – UC4
- Ohodnotiť akýkoľvek balíček známkou – UC4

- Sprístupniť svoj balíček ostatným užívateľom a umožniť im modifikovať ho – UC6
- Kontrolovať svoj profil a meniť údaje, ktoré sú v ňom – UC9

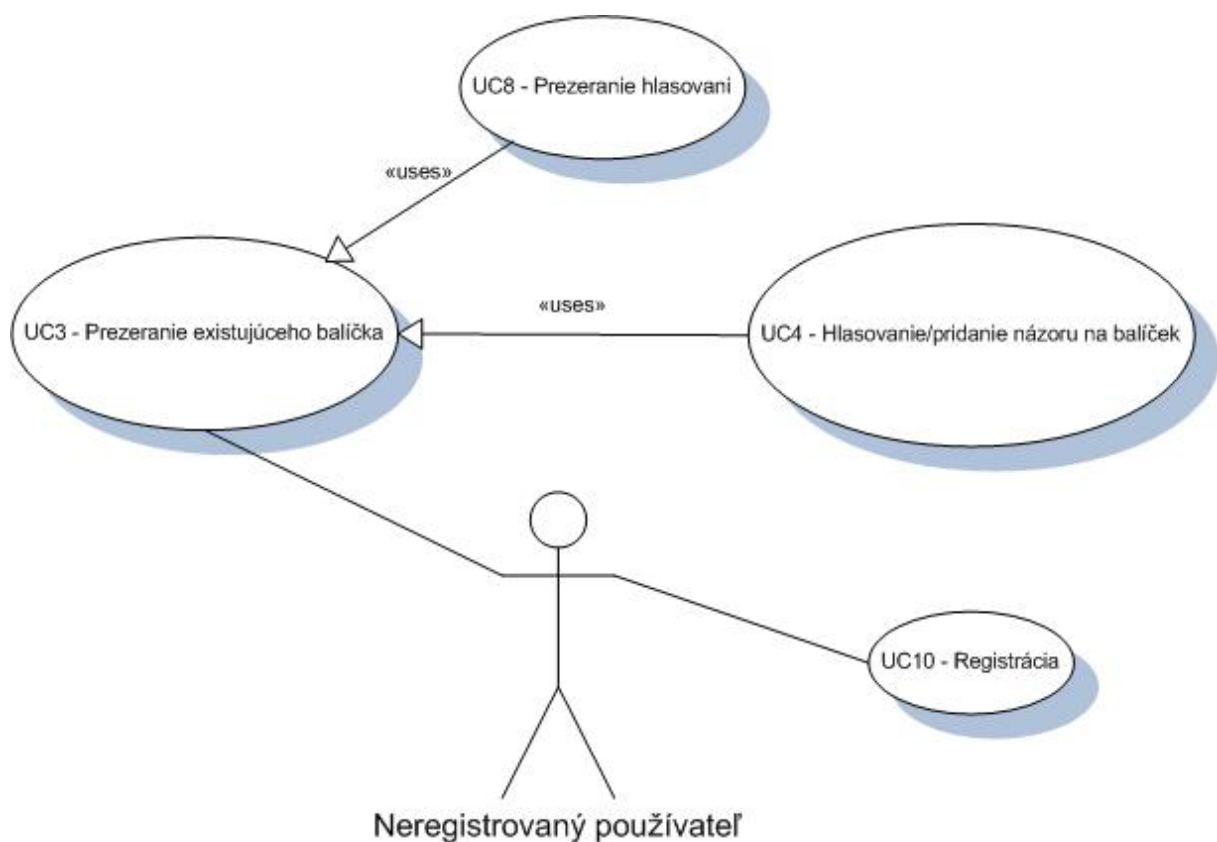


Obr. č. 5 Diagram prípadov použitia pre administrátora.

Administrátor je najdôležitejším hráčom, ktorý systém používa. Jeho hlavnou úlohou je systém spravovať a niesť zaň zodpovednosť. Preto by mal mať vyčlenených najviac právomocí. V konečnom dôsledku by to mal byť niekto, kto je pokročilým vývojárom a kto dokáže posudzovať výsledky práce iných užívateľov. Dostupné sú mu všetky možné činnosti, ktoré systém poskytuje:

- Pridanie nového vlastného balíčka – UC1
- Editovanie všetkých existujúcich balíčkov – UC2

- Prezeranie existujúcich balíčkov, ktoré sú v systéme evidované – UC3 + UC4 + UC8
- Vyjadriť svoj názor v diskusií pre všetky balíčky – UC4
- Ohodnotiť akýkoľvek balíček známkou – UC4
- Kontrolovať svoj profil a meniť údaje, ktoré sú v ňom – UC9
- Moderovanie diskusií a možnosť mazať príspevky do nich – UC7
- Označovať balíčky a meniť tým ich stavy – UC5



Obr. č. 6 Diagram prípadov použitia pre neregistrovaného užívateľa.

Neregistrovaný používatelia budú asi najčastejšie využívanou rolou systému, bude slúžiť pre všetkých užívateľov, ktorý systém len navštívia, príp. potrebujú len niektorý balíček stiahnuť. Preto sú činnosti pre nich obmedzené len na jednoduché úlohy:

- Prezeranie existujúcich balíčkov, ktoré sú v systéme evidované – UC3 + UC4 + UC8
- Vyjadriť svoj názor v diskusií pre všetky balíčky – UC4

- Ohodnotiť akýkoľvek balíček známkou – UC4
- Možnosť zaregistrovať sa v systémy a stať sa registrovaným užívateľom v prípade záujmu rozšírenia svojich možností práce so systémom – UC10

3.1.1 Opis jednotlivých prípadov použitia

UC1 – Pridanie nového balíčka

Táto činnosť je prístupná každému registrovanému užívateľovi a administrátorom a umožňuje pridanie nových balíčkov, ktoré boli užívateľmi vytvorené. Prebieha pomocou vyplnenia základných informácií do formulára a odoslaním („uploadom“) zip súboru do systému, kde sa zaznamenajú údaje o balíčku do databázy a zip súbor sa uloží do užívateľovho priečinka ako privátny.

UC2 - Editácia existujúceho balíčka

Táto činnosť je prístupná každému registrovanému užívateľovi a administrátorom a umožňuje im editovať balíčky, ktoré už boli užívateľmi vytvorené. Avšak pre rôznych hráčov tu platia rôzne podmienky editovania. Administrátori môžu editovať všetky balíčky, registrovaný používatelia len tie, ktoré sami vytvorili a sú označené ako privátne, alebo všetky balíčky, ktoré sú označené ako „unstable“. Privátne balíčky ostatných užívateľov editovať nemôžu. Pri editovaní balíčkov, ktoré boli označené administrátormi ako „stable“ sa vytvorí nová kópia balíčku v priečinku užívateľa aby „stable“ verzia ostala zachovaná.

UC3 - Prezeranie existujúceho balíčka

Túto činnosť môžu využiť všetci používatelia systému, slúži na prezentáciu už existujúcich balíčkov. Pre jednoduchú orientáciu v balíčkoch budú balíčky delené do kategórií ,ktoré sa užívateľovi zobrazia a taktiež bude k dispozícii filter na vyhľadávanie konkrétnych balíčkov. Užívateľovi umožní rýchlo a flexibilne prehľadávať zbierku balíčkov v systéme. Pri prezeraní akéhokoľvek balíčka budú mať používatelia možnosť vyjadriť svoj názor dvoma

spôsobmi, hlasovaním, čiže pridelovaním dobrých či zlých bodov alebo môžu využiť možnosť prídania komentáru do diskusie k danému balíčku.

UC4 - Hlasovanie/prídanie názoru na balíček

Táto činnosť je súčasťou všetkých prehľadov balíčkov a zobrazuje sa pri prezeraní každého z nich. Užívateľ má možnosť pridať balíčku hlas, príp. do diskusie pridať názor alebo otázku ak má nejaké nejasnosti ohľadom zobrazeného balíčka.

UC5 - Označ balíček ako "stable"/"unstable"/"dead"

Táto funkcionálnosť systému je špecifická a je veľmi dôležité aby bola neustále pod kontrolou, preto je priradená len pre administrátorov. Tí na základe svojho uváženia, hodnotenia, názoru a skúseností priradujú balíčkovi stavy, aby tým oznamovali ostatným užívateľom, či je balíček vhodný pre použitie. Každý balíček, ktorý užívatelia vytvoria bude časom overený či je správny a nespôsobuje pády aplikácií, ktoré ho používajú. Vtedy pomocou tejto činnosti administrátor priradí balíčku stav „stable“ – stabilný a tým povýši balíček na verejný, dostupný pre všetkých užívateľov, ktorí ho potom môžu využívať.

UC6 - Sprístupni svoj balíček ("publish")

Táto činnosť slúži na publikovanie svojej práce, svojich balíčkov pre všetkých užívateľov. Publikovaním sa balíček ocitne medzi „unstable“ verziami a ľubovoľný užívateľ môže tento balíček modifikovať k svojim potrebám.

UC7 - Zmazanie hlasovania

Tento prípad nastáva v momente, keď administrátor potrebuje moderovať diskusie k balíčkovi a potrebuje určitý príspevok zmazať. Obvykle táto situácia nastáva pri výskyte škodlivých botov, ktorý šíria prostredníctvom internetu reklamu alebo iné nevyžiadané

správy. Taktiež umožňujú moderátorom obmedzovať diskusiu v rámci noriem spoločenského a etického správania.

UC8 - Prezeranie hlasovaní

Táto činnosť nikdy nevystupuje samostatne, vždy je súčasťou iných prípadov použitia, najmä prezerania balíčkov. Vtedy užívateľovi jednoducho prezentuje názory iných užívateľov na daný balíček, ktorý si práve prezerá.

UC9 - Edituj svoj profil

Prípad použitia editácie profilu je potrebné pri správe svojich užívateľských nastavení v systéme. Konkrétne umožňuje užívateľovi meniť si osobné údaje, heslo a umožňuje správu zdieľania svojho súkromného priečinku s balíčkami aj s ďalšími užívateľmi. Tým dáva možnosť tímového vývoja svojich balíčkov.

UC10 – Registrácia

Tento prípad je dostupný len pre užívateľov, ktorý v systéme ešte zaregistrovaný nie sú. Môže ho využiť každý, kto sa rozhodne pre využitie ďalších možností systému. Registráciou získava privilégia registrovaného užívateľa a tým aj všetky jeho možnosti použitia jeho výhod. Počas registrácie si musí zvoliť unikátne prihlasovacie meno, primerané, netriviálne heslo a môže vyplniť niektoré údaje, ktoré budú zobrazené v jeho profile.

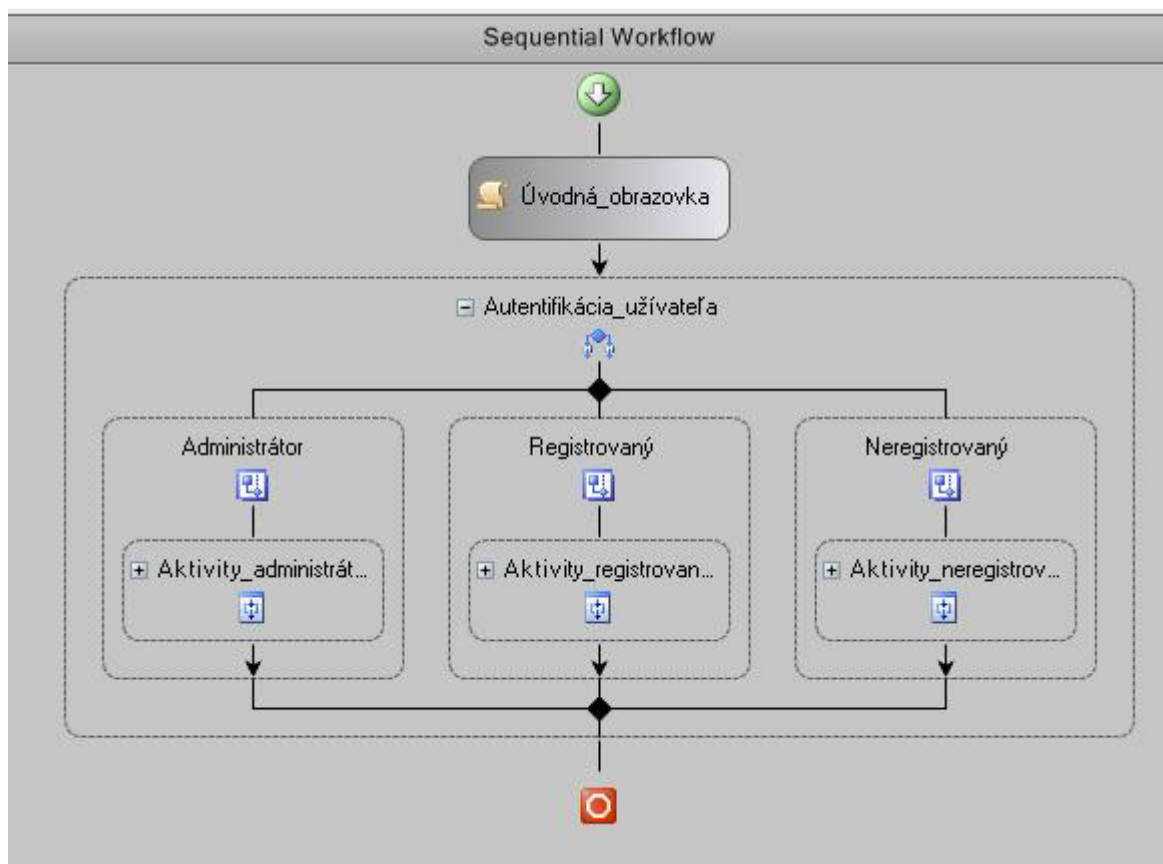
4 Návrh

Táto kapitola bližšie popisuje návrh riešenia systému cez workflow diagramy, model údajov a obrazovky používateľského rozhrania.

4.1 Workflow diagramy

Nasledujúca kapitola bližšie opisuje správanie systému pomocou workflow diagramov a prináša hlbší pohľad na aktivity užívateľov, ktoré im systém poskytuje.

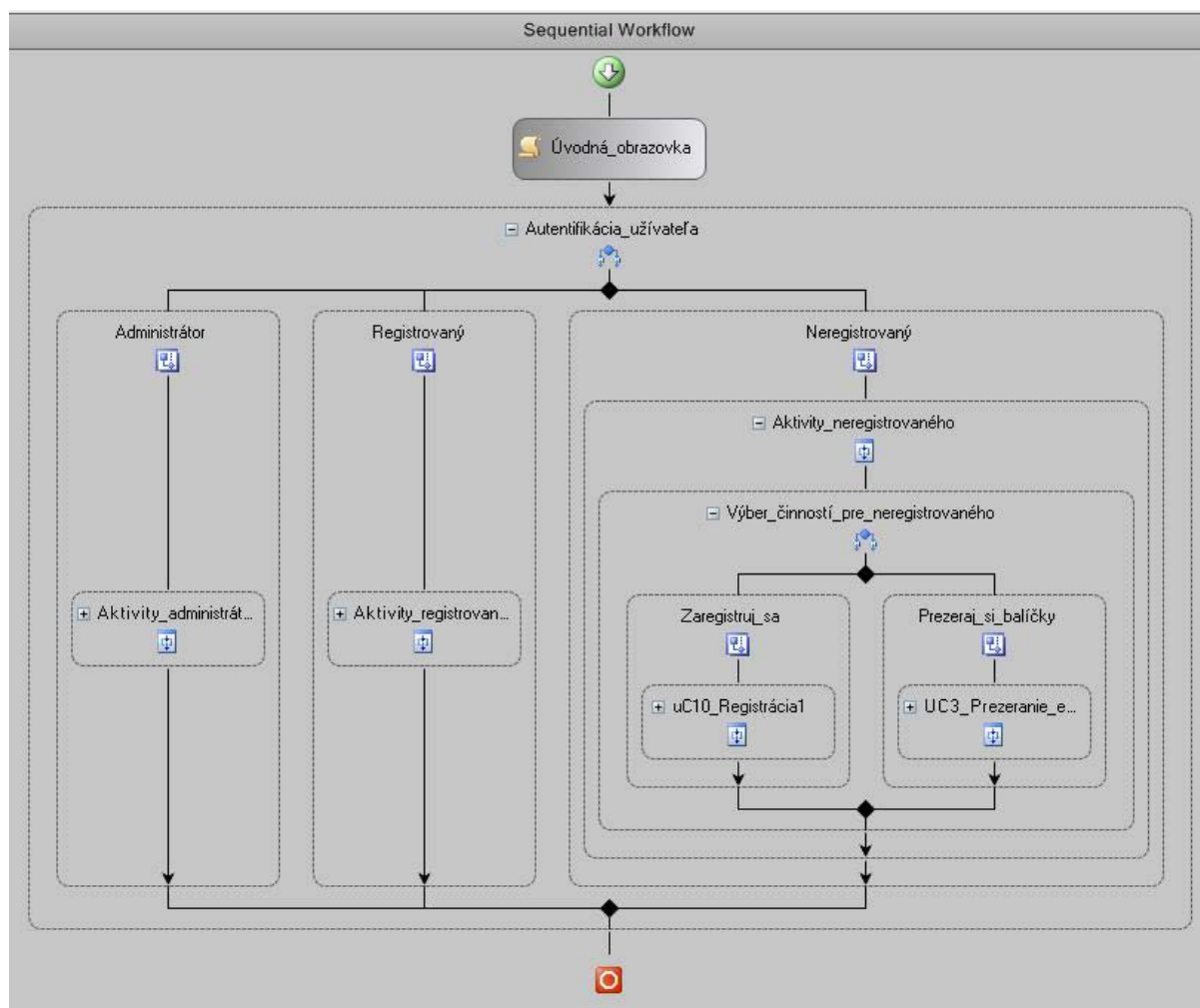
Prvý diagram zobrazuje vstup do systému. Každému užívateľovi bez ohľadu na práva sa zobrazí úvodná obrazovka a dá mu na výber, či sa prihlási alebo bude pokračovať ako neregistrovaný užívateľ s obmedzenými právami. Podľa jeho výberu bude systém pokračovať. Ak sa prihlási ako administrátor, zobrazí sa úvodná obrazovka pre administrátora a dá mu na výber možností vykonania aktivít určených pre skupinu administrátorov. Pri registrovanom užívateľovi sa zobrazí obrazovka určená pre nich a bude obsahovať príslušné aktivity. Neregistrovaný užívateľ bude mať svoje možnosti priamo na hlavnej úvodnej obrazovke.



Obr. č. 7 Workflow diagram demonštrujúci autentifikáciu užívateľov.

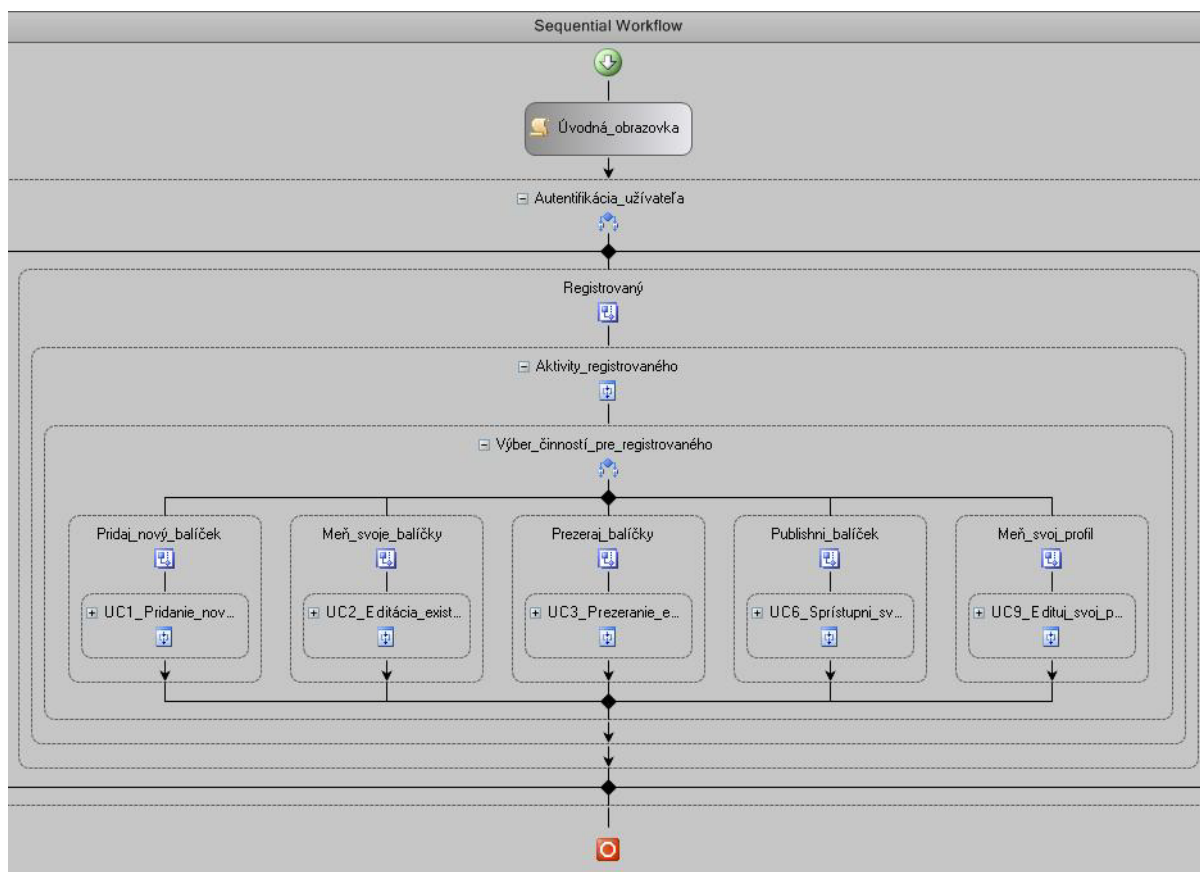
Ďalšie diagramy budú konkretizovať zobrazené možnosti, ktoré sa objavia jednotlivým užívateľom po autentifikácii.

Neregistrovaný užívateľ má podľa diagramy činností k dispozícii dve činnosti a to možnosť registrácie a prezerania existujúcich balíčkov, vytvorených ostatnými užívateľmi. Tieto aktivity plne postačujú bežným užívateľom, ktorí chcú balíčky len stiahnuť pre vlastné použitie. A taktiež tieto aktivity slúžia pre registráciu nových členov, ktorá nie je podmienená žiadnymi špeciálnymi podmienkami.



Obr. č. 8 Workflow diagram pre neregistrovaného používateľa.

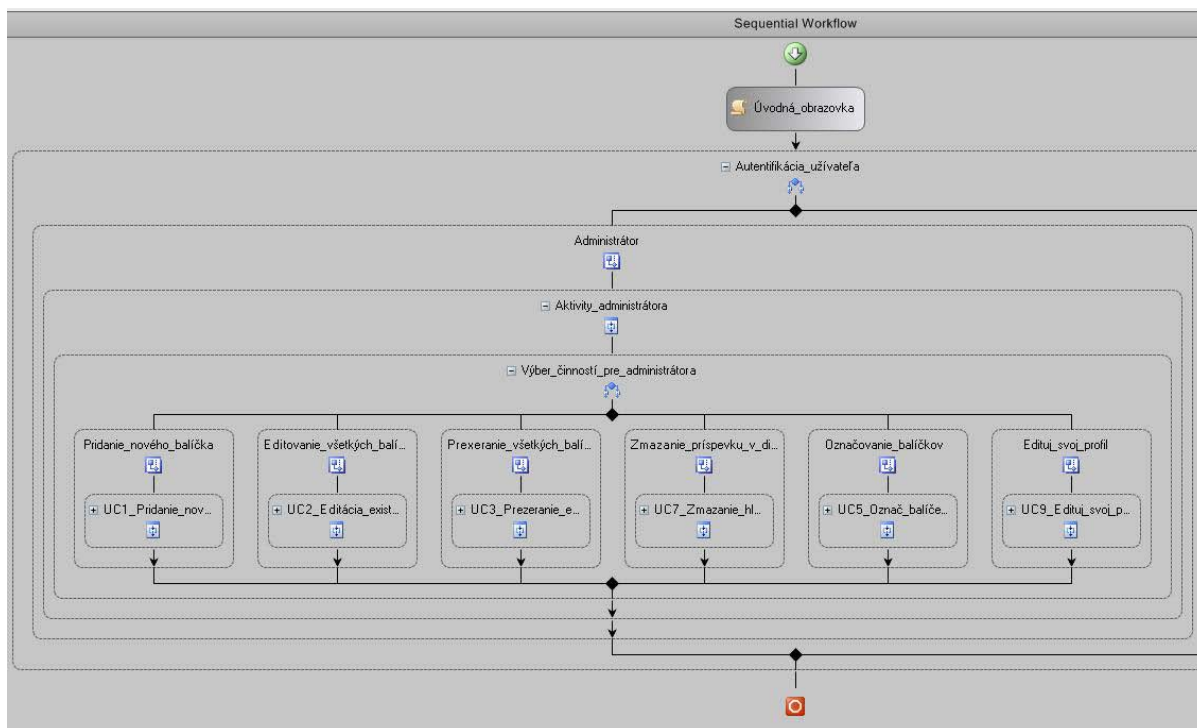
Registrovaní užívatelia majú po prihlásení na svojej úvodnej obrazovke k dispozícii viacero aktivít, ktoré môžu vykonať. Aktivity im umožňujú pridávať, prezerat' a upravovať balíčky, všetko v rámci pravidiel, ktoré im udáva ich rola v systéme. Taktiež môžu publikovať svoje balíčky a sprístupniť ich pre všetkých užívateľov systému. Ďalej môžu meniť, svoj profil, v ktorom majú všetky informácie o ich účte.



Obr. č. 9 Workflow diagram pre registrovaného používateľa.

Administrátorom ich úvodná obrazovka poskytne najviac možností aktivít. Okrem základných aktivít, ako sú pridávanie, editovanie a prezeranie balíčkov, majú k dispozícii aj veľmi dôležitú aktivitu označovanie balíčkov. Tá slúži na separáciu balíčkov do skupín a k uverejňovaniu balíčkov. Balíček sa dá označiť ako stable, čo znamená že jeho verzia je funkčná a plne kompatibilná so všetkými dependenciami uvedenými v dist súbore, ďalej môže byť dead, čo je označenie pre balíčky, ktoré už vyvíjané nie sú alebo podpora ktorých medzičasom skončila. Poslednou kategóriou je označovanie balíčkov ako unstable, v prípade že sa predsa len vyskytnú prípady, že stable balíček nebude spĺňať všetky náležitosti, ktoré mu prislúchať majú.

Ostatné aktivity sú skôr symbolické a nemali by byť veľmi často využívané, no sú administrátorom k dispozícii. Taktiež ako registrovaný užívateľ si budú môcť vytvoriť profil a meniť údaje v ňom. Taktiež môžu pridávať, prezerať a meniť všetky balíčky bez akýchkoľvek obmedzení.



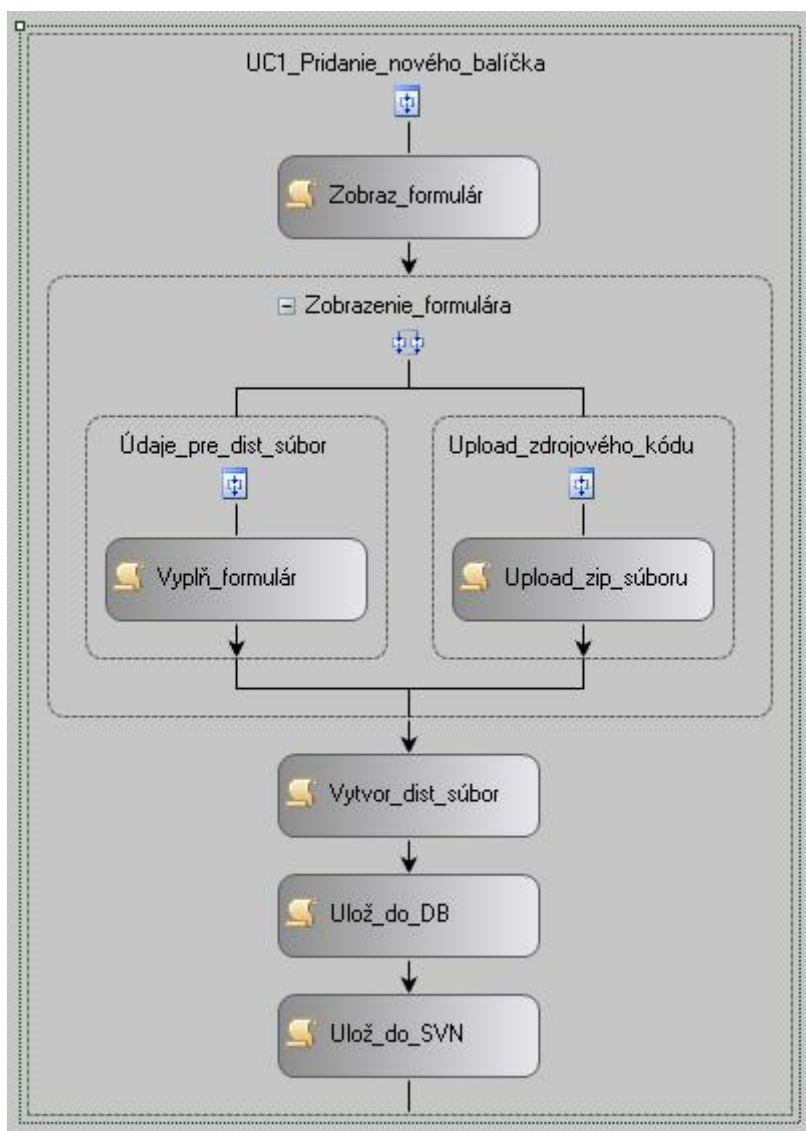
Obr. č. 10 Workflow diagram pre administrátora.

Prípady použitia

Nasledujúca kapitola bližšie určuje aký workflow nastane pri jednotlivých prípadoch použitia užívateľmi. Diagramy v predchádzajúcej kapitole rozoberali workflow-y z pohľadu systému a mali skôr abstraktný charakter a nekonkretizovali priamo aktivity, ktoré systém vykonáva na pozadí týchto prípadov použitia.

UC1 - Pridanie nového balíčka

Pridanie nového balíčka pozostáva z jednoduchého vyplnenia formulára. Na začiatku sa užívateľovi zobrazí formulár, v ktorom sa nachádzajú kolónky pre všetky údaje o novom balíčku, ktoré systém potrebuje vedieť. Tie musí užívateľ vyplniť. V prípade že by tak neučinil, bude vhodne na to upozornený. Dovtedy nie je možné balíček uložiť do systému. Vyplňovanie formulára taktiež obsahuje vyhľadanie súboru na užívateľovom pevnom disku, ktorý obsahuje zdrojový kód balíčku. Práve do tohto súboru systém pridá dist súbor, ktorého údaje vyplní z formulára. Na záver ak je všetko vyplnené v poriadku, nový balíček je uložený do SVN a informácie o balíčku sa pridávajú do databázy.

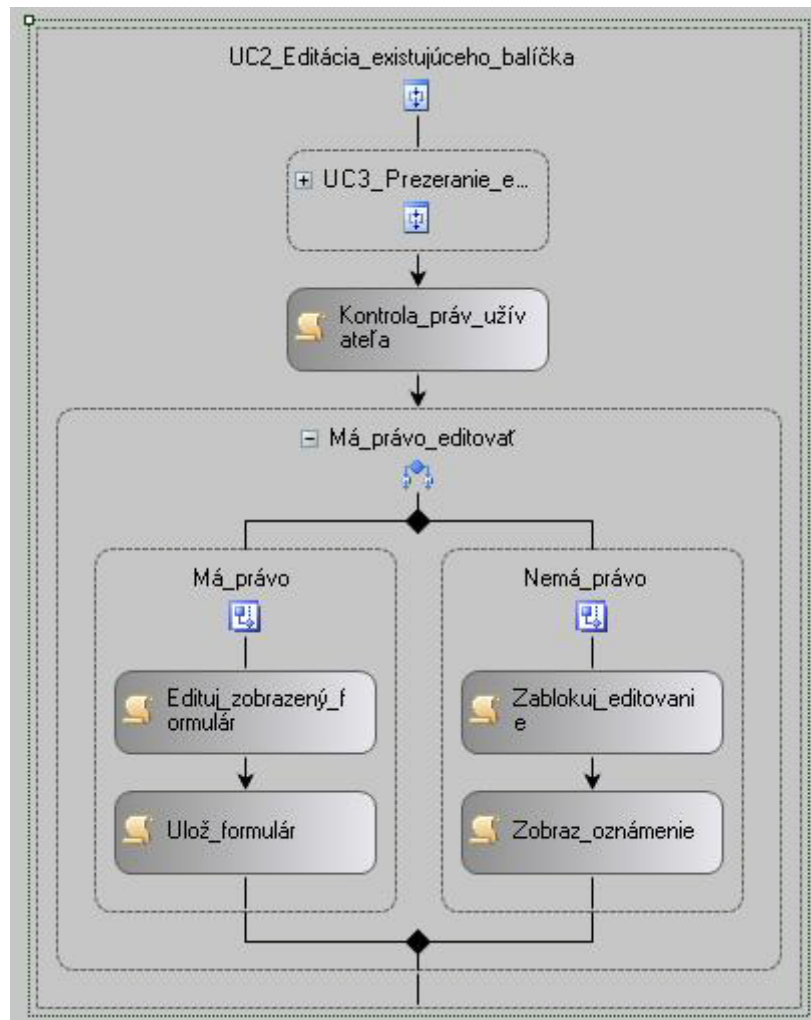


Obr. č. 11 Workflow diagram pre UC pridanie nového balíčka.

UC2 - Editácia existujúceho balíčka

Editovanie existujúceho balíčka je aktivita, ktorá je zložená z ďalších aktivít a doplnená o niektoré svoje časti. V prvom rade je nutné vybrať si balíček, ktorý chceme editovať. Pre túto časť aktivity je použitý UC3, opísaný nižšie. Keď užívateľ vybral balíček, ten sa mu zobrazil a editovanie je pripravené. Samozrejme na začiatku je skontrolované, či vôbec užívateľ môže daný balíček editovať. Administrátori môžu meniť akýkoľvek balíček, registrovaní užívatelia môžu meniť len nimi vytvorené balíčky alebo akýkoľvek unstable balíček. Nemôžu editovať prívátne balíčky iných užívateľov, ani stable balíčky, ktoré oni sami nevytvorili. Tieto podmienky sú zohľadnené v tejto kontrole, na základe ktorej workflow pokračuje buď povolením editovania alebo editovanie neumožní. Ak neumožní, užívateľ bude

o tom upovedomený a zobrazí sa mu dôvod prečo tomu tak je. V prípade, že má právo editovať, môže do zobrazeného formulára prevádzať zmeny, ktoré po odoslaní systém najprv skontroluje, či sú relevantné a v prípade že je všetko v poriadku, zmeny uloží.

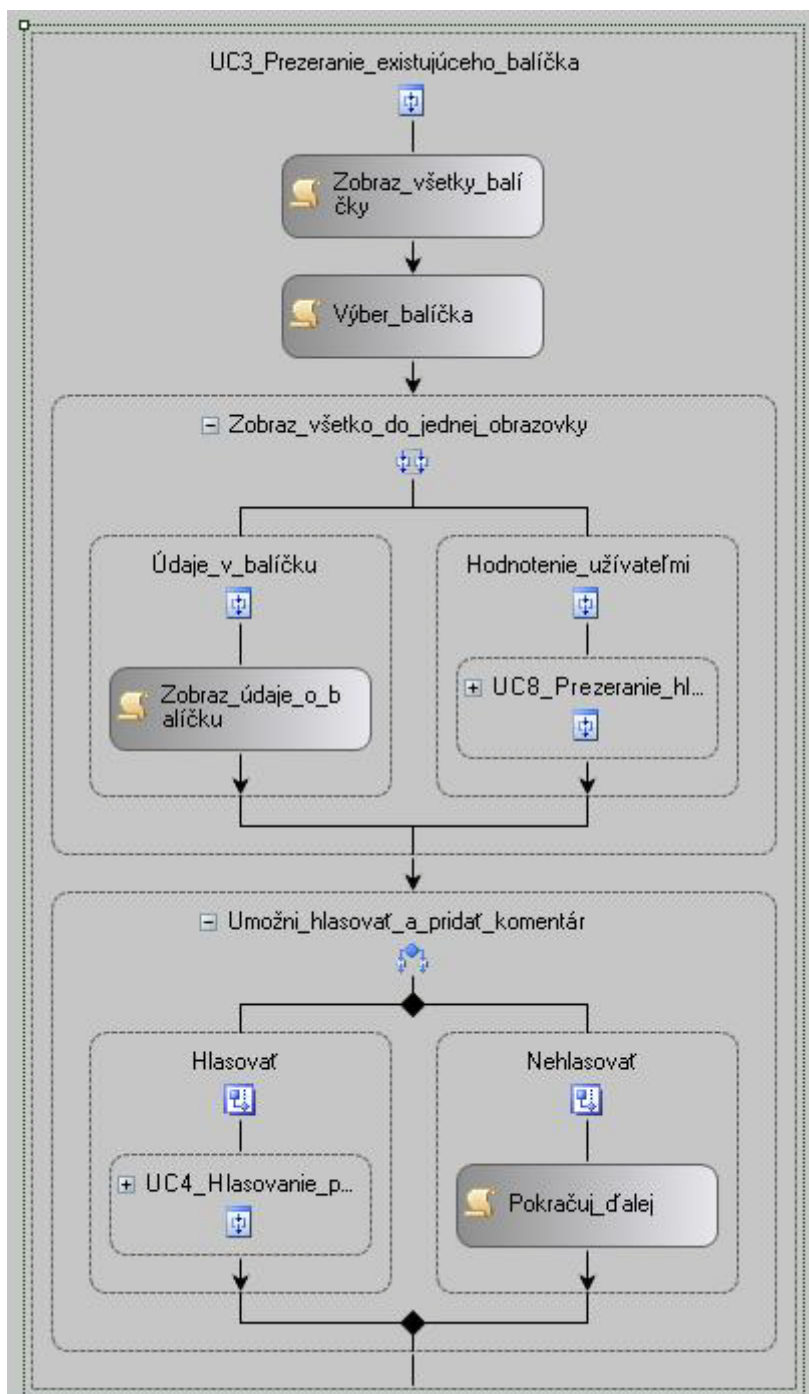


Obr. č. 12 Workflow diagram pre UC editovanie existujúceho balíčka.

UC3 - Prezeranie existujúceho balíčka

Táto aktivita môže byť vyvolaná samostatne užívateľom z úvodnej obrazovky alebo je súčasťou iných aktivít. Služi na prezeranie balíčkov a pomáha užívateľovi čo najjednoduchšie nájsť balíček, ktorý hľadá. Ak si teda užívateľ túto aktivitu vyberie, zobrazia sa mu všetky balíčky a filter, ktorým ich môže triediť. Filtrovať sa bude dať na základe kategórií balíčkov, mena balíčku, jeho autora, stavu balíčku, na základe verzie Lua-y alebo platformy, na ktorú bol balíček vytvorený. Po vybraní balíčka, ktorý užívateľ chce vidieť sa tento balíček zobrazí,

vrátane hodnotenia a názorov užívateľov. Taktiež je umožnené užívateľovi názor pridať alebo balíček ohodnotiť.

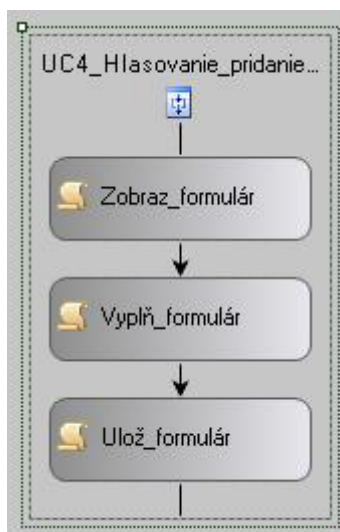


Obr. č. 13 Workflow diagram pre UC prezeranie existujúceho balíčka.

UC4 - Hlasovanie/pridanie názoru na balíček

Táto činnosť slúži najmä na získanie spätnej väzby od užívateľov pre autorov balíčkov. Činnosť je vždy súčasťou prezerania údajov o balíčku, kde užívateľ môže vyjadriť svoj názor na balíček, ktorý práve prezerá. Samotné hodnotenie prebieha vyplnením

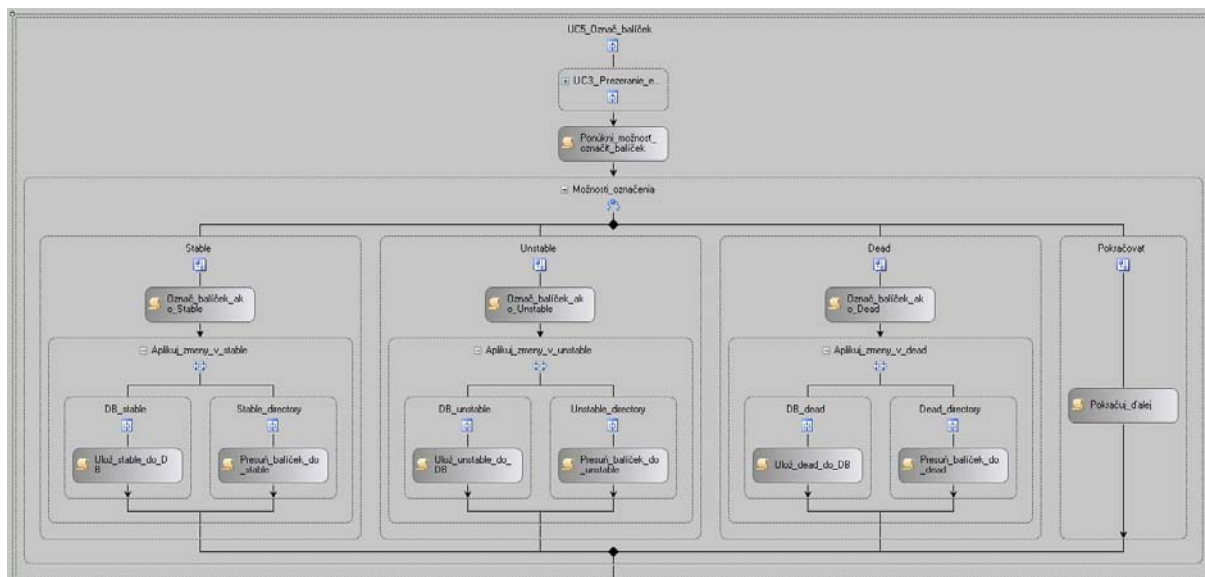
formuláru, do ktorého vpíše svoj názor. Po odoslaní príspevku sa uloží a pridá k danému balíčku.



Obr. č. 14 Workflow diagram pre UC hlasovanie/pridanie názoru na balíček.

UC5 - Označ balíček ako "stable"/"unstable"/"dead"

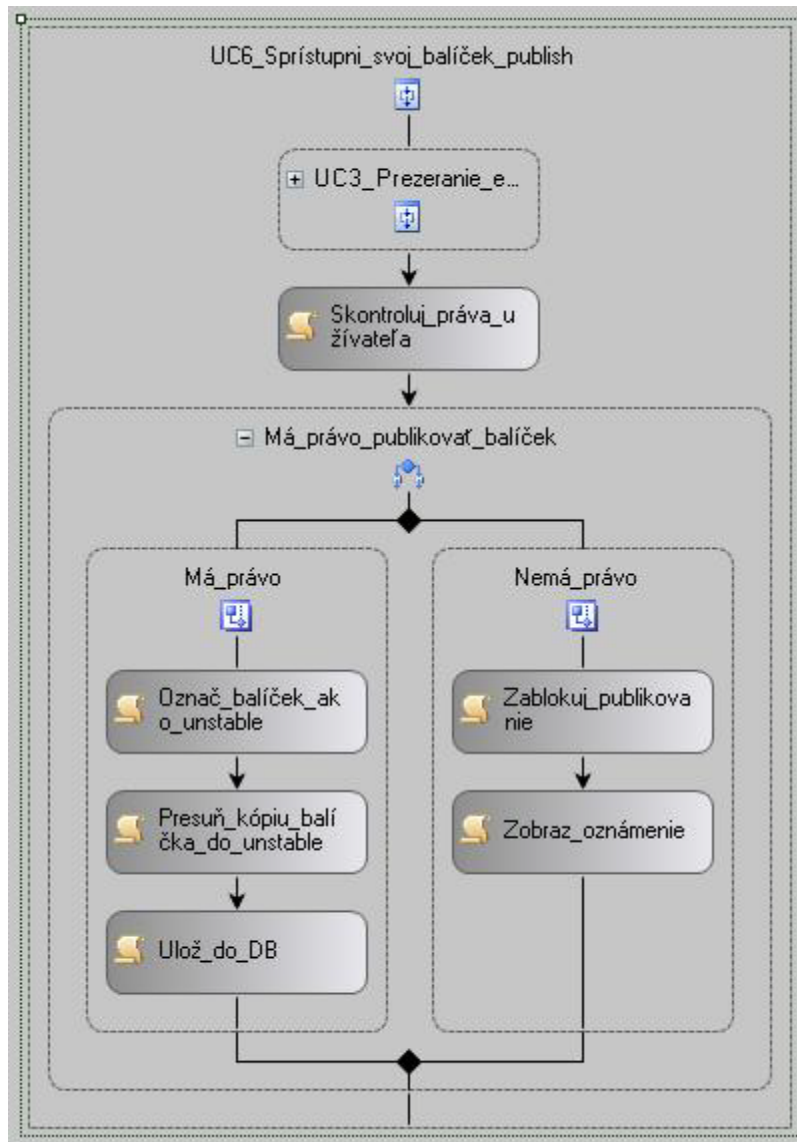
Aktivita, ktorá je najdôležitejšou pre administrátorov tiež pozostáva z viacerých aktivít. Administrátor, ktorý bude označovať nejaký balíček si najprv jeden vyberie. Potom zvolí operáciu označenia balíčka. Môže ho označiť ako stable, unstable, dead. Balíček sa dá označiť ako stable, čo znamená že jeho verzia je funkčná a plne kompatibilná so všetkými dependenciami uvedenými v dist súbore, ďalej môže byť dead, čo je označenie pre balíčky, ktoré už vyvíjané nie sú alebo podpora ktorých medzičasom skončila. Poslednou kategóriou je označovanie balíčkov ako unstable, v prípade že sa predsa len vyskytnú prípady, že stable balíček nebude spĺňať všetky náležitosti, ktoré mu prislúchať majú. Po označení a odoslaní sa vykonajú dva kroky. Prvým je premiestnenie balíčka do iného adresára, pretože sa mu zmenil typ označenia a preto je potrebné ho premiestniť k balíčkovi jeho typu. Druhým je uloženie zmien v databáze pre daný balíček.



Obr. č. 15 Workflow diagram pre UC označ balíčiek.

UC6 - Sprístupni svoj balíček ("publish")

Publikovanie balíčkov slúži najmä registrovaným užívateľom na poskytovanie svojich balíčkov ostatným užívateľom. Užívateľ jednoducho vyberie balíček, ktorý chce zverejniť. Systém skontroluje, či má na to právo, pretože užívatelia môžu zverejňovať len balíčky, ktoré sú nimi vytvorené a majú stav private. Ak užívateľ spĺňa tieto podmienky, vytvorí sa kópia tohto balíčka, následne je označená ako unstable a presunie sa do adresára s unstable balíčkami. V prípade, že užívateľ práva nemá, publikovanie sa zablokuje a zobrazí sa hlásenie a dôvod, prečo užívateľ nemôže balíček publikovať.



Obr. č. 16 Workflow diagram pre UC Sprístupni svoj balíček.

UC7 - Zmazanie hlasovania

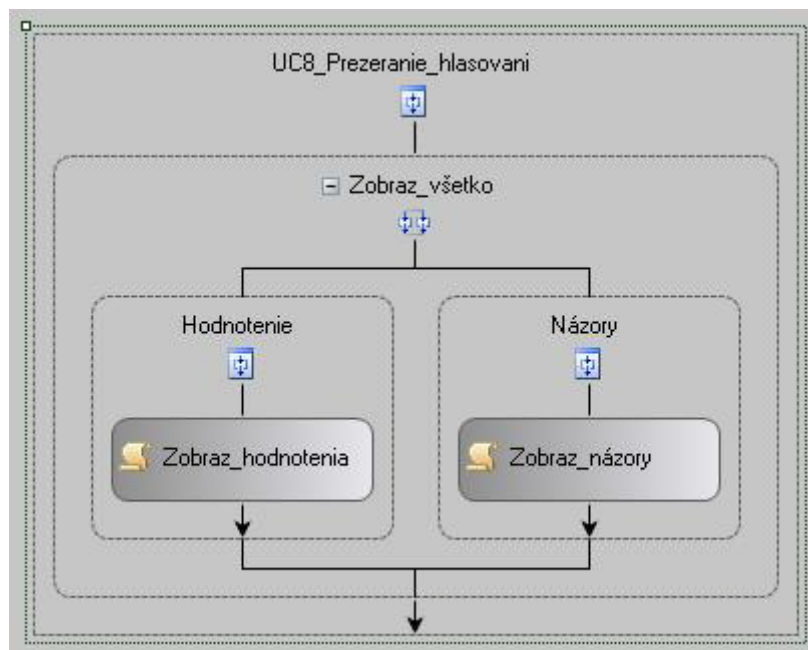
Táto aktivita slúži administrátorom, ktorý sú zároveň aj moderátormi diskusií k balíčkam. Je vhodné aby títo moderátori mohli regulovať správy v diskusiách, najmä neprimerané reakcie a reklamné, nevyžiadané správy. Samotný proces je veľmi jednoduchý, administrátor najprv vyberie z existujúcich názorov jeden, ktorý chce zmazať, klikne na tlačidlo zmazať a tým ho vymaže. Akcia sa automaticky prevedie aj v databáze.



Obr. č. 17 Workflow diagram pre UC Zmazanie hlasovania.

UC8 - Prezeranie hlasovani

Prezeranie hlasovani je dostupné pri každom prezeraní balíčkov. Pod všetky informácie o balíčku sa pridá jeho hodnotenie a názory naň. Aktivita následne môže pokračovať pridaním nového názoru alebo hlasovaním.



Obr. č. 18 Workflow diagram pre UC Zmazanie hlasovania.

UC9 - Edituj svoj profil

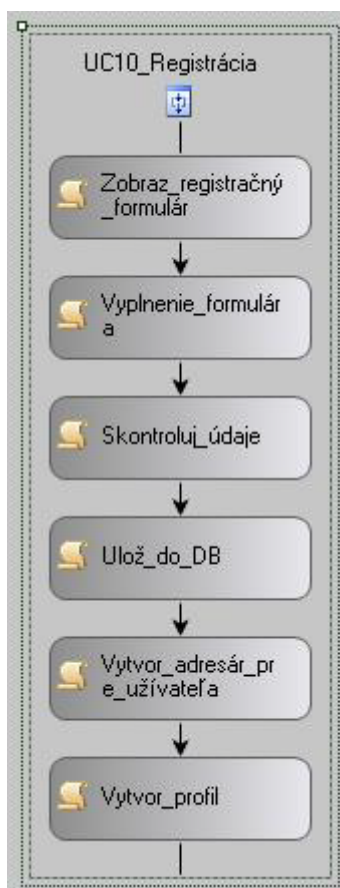
Editovať svoj profil môže ktokoľvek, kto má v systéme vytvorený účet. Jeho profil sa skladá z informácií, ktoré sú povinné, ako napr. užívateľské meno a heslo a z údajov, ktoré povinné nie sú, ako napr. meno, priezvisko, krajina, príp. zoznam iných členov, ktorí majú prístup k jeho privátnym balíčkom. Všetky tieto údaje je možné meniť pomocou tejto aktivity. Užívateľovi sa zobrazí jeho profil, užívateľ zmení údaje, ktoré potrebuje a formulár odošle na spracovanie a uloženie do databázy



Obr. č. 19 Workflow diagram pre UC Edituj svoj profil.

UC10 – Registrácia

Registrácia je základnou aktivitou, ktorú môže vykonať akýkoľvek neprihlásený užívateľ, majúci záujem stať sa registrovaným členom systému LuaDist. Samotná registrácia sa dá vyvolať z úvodnej obrazovky, ktorá sa zobrazí každému užívateľovi, ktorý navštívi systém. Užívateľ má len jedinou povinnosť a to vyplniť formulár, ktorý sa mu zobrazí, systém tieto údaje skontroluje, či sú správne a uloží ich do databázy. Následne systém vytvorí repozitár, pre tohto užívateľa a údaje ktoré zadal vloží do jeho profilu.



Obr. č. 20 Workflow diagram pre UC Registrácia.

4.2 Návrh obrazoviek

Nasledujúca kapitola popisuje a zobrazuje užívateľské rozhranie a obrazovky pomocou, ktorých bude používateľ systém obsluhovať. Obrazovky sú vytvorené na ovládanie špeciálnych funkcií, ktoré prislúchajú danému používateľovi. Niektoré obrazovky sú spoločné pre všetky tri skupiny používateľov a niektoré sú vytvorené pre každý typ používateľa zvlášť.

Prvou obrazovkou, ktorá je popísaná je obrazovka slúžiaca na registráciu používateľa. V nej používateľ vyplní údaje o logine, zadá heslo, a vyplní e-mail adresu na zaslanie potvrdzovacieho linku a môže si vybrať či chce, aby mu boli na mailovú adresu zasielané informácie o zmene v balíčkoch.

Registration

Login

Password

Confirm password

Email address

 Send me emails about changes in packages

Obr. č. 21 Návrh obrazovky na registráciu používateľov

Pokiaľ je používateľ už zaregistrovaný môže si upravovať svoj profil. Na tento účel slúži ďalšia obrazovka. Je veľmi podobná s obrazovkou registrácia ale je tu pridaná možnosť zmeny hesla a taktiež si môže nastaviť používateľov s ktorými chce zdieľať balíčky. Vykoná to kliknutím na odkaz „Edit AccessList“. Stlačením tlačidla uloží vykonané zmeny.

Edit Profile

Old password

New password

Confirm new password

Email address

 Send me emails about changes in packages

[Edit AccessList](#)

Obr. č. 22 Návrh obrazovky na editáciu profilu používateľa

Aj neregistrovaný používateľ má však niekoľko možností. Sú to prezeranie existujúcich balíčkov, hlasovanie za balíček a pridávanie komentárov k balíčku. Všetky tieto možnosti práce umožňuje nasledujúca obrazovka, ktorá slúži na prezeranie informácií o vybratom balíčku. Túto obrazovku využívajú na prezeranie balíčkov všetky tri skupiny používateľov systému. Nachádza sa tu názov balíčku, jeho krátky popis, stav v akom sa nachádza (stable, unstable, dead), pre akú platformu je určený, na akej architektúre je postavený, aká sú jeho dependencie (aj externé ak nejaké má) a poslednou položkou o balíčku je meno autora, ktorý tento balíček vytvoril. Nasleduje informácia o aktuálnom ratingu daného balíčka. Používateľ taktiež môže zahlasovať za daný balíček využitím radio buttonov s príslušnou bodovou hodnotou. Dôležitým je tlačidlo na stiahnutie práve prezeraného balíčka. Nasleduje ešte pole s odkazmi od používateľov a možnosť pridať vlastný odkaz.

Package

Name

NameOfPackage

Description

DescriptionOfPackage

Status

Stable/Unstable/Dead

Platform

Win/MacOS/Linux

Architecture

Architecture

Dependency

- [Dependency1](#)
- [Dependency2](#)
- [Dependency3](#)

External dependencies

- [External dependency1](#)
- [External dependency2](#)
- [External dependency3](#)

Author

AuthorName

Actual Rating 4.2

1 2 3 4 5

Comments

<john, IP:000.000.000.000, date 1.1.2009 at 00.01>

.....
very good

<smith, IP:000.000.000.000, date 1.6.2009 at 13.40>

.....
awesome

<carter, IP:000.000.000.000, date 13.9.2009 at 12.01>

.....
good work

<ma5, IP:000.000.000.000, date 12.11.2009 at 10.01>

.....
not bad

Add comment

Message



Write down the code from picture

Obr. č. 23 Návrh obrazovky na prezeranie balíčka

Na vyhľadavanie v existujúcich alíčkoch slúži vyhľadávací nástroj prístupný pre všetkých používateľov systému. Užívateľ si len vyberie podľa akého atribútu chce balíček vyhľadávať a následne zvolí (napíše) text, ktorý by chcel nájsť.

Search

Name of Package

Author

Platform

Architecture

Status

Obr. č. 24 Návrh obrazovky na vyhľadavanie balíčka

Prihlásený používateľ má ďalej právo pridávať nové balíčky. Na to slúži nasledujúca obrazovka. Užívateľ vyplní príslušné údaje o balíčku, vyberie .zip súbor zo svojho lokálneho adresára a uploadne ho. Týmto je balíček premiestnený k nemu do vlastného privátneho repozitára. Položka status slúži na rozhodnutie, či chce daný používateľ zobrazit' svoj balíček pre každého (public) alebo bude uložený len u neho v repozitári (private) a prístup k nemu budú mať len ľudia, ktorých v profile označil.

Add Package

Name

Description

Status

 ▼

Upload package

Upload binary

Obr. č. 25 Návrh obrazovky na pridanie balíčka

Používateľ môže ďalej svoje balíčky upravovať. Nasledujúca obrazovka mu umožňuje robiť zmeny v informáciách o jeho balíčku. Jednoduchým prepísaním príp. zmenou pôvodných hodnôt a následným uložením zmien sa upraví informácie o balíčku.

Edit Package

Name

Description

Status

Platform

Architecture

Dependency

External dependencies

Upload package

Upload binary

Obr. č. 26 Návrh obrazovky na úpravu balíčka používateľom

Používateľom s najväčšími právami je administrátor. Ten má prístupné všetky možnosti ako obyčajný používateľ ale navyše môže mazať komentáre ku konkrétnym balíčkom a mení stav balíčkov z unstable na stable, príp. dead. Všetky zmeny samozrejme uloží pomocou príslušného tlačidla.

Edit Package

Name

Description

Status

Platform

Architecture

Dependency

External dependencies

Author

Upload package

Upload binary

Actual Rating 4.2

1 2 3 4 5

Comments

<john, IP:000.000.000.000, date 1.1.2009 at 00.01> [DELETE](#)

very good

<smith, IP:000.000.000.000, date 1.6.2009 at 13.40> [DELETE](#)

awesome

<carter, IP:000.000.000.000, date 13.9.2009 at 12.01> [DELETE](#)

good work

<ma5, IP:000.000.000.000, date 12.11.2009 at 10.01> [DELETE](#)

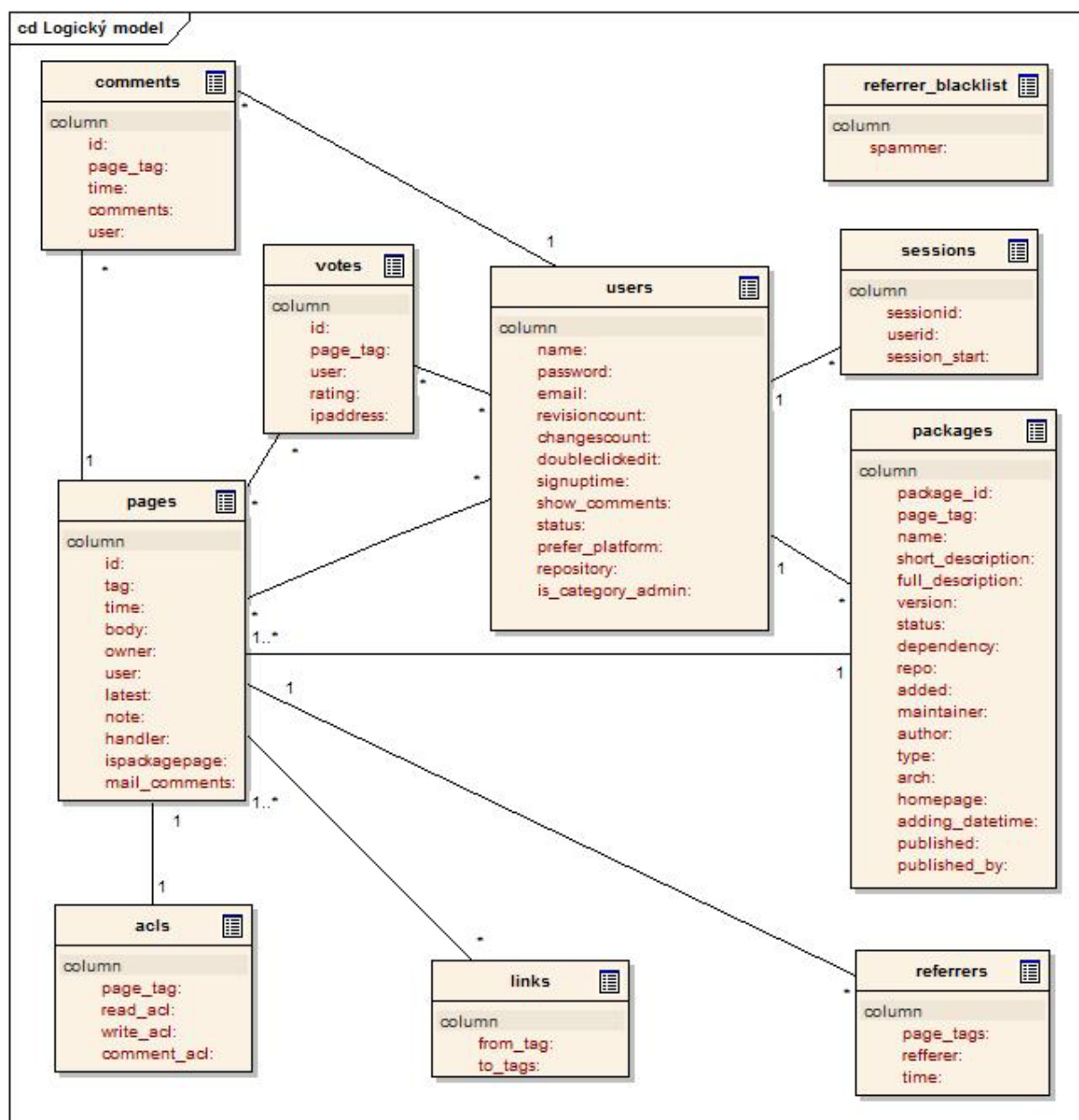
not bad

Obr. č. 27 Návrh obrazovky na editáciu balíčka administrátorom systému.

4.3 Model údajov

V tejto kapitole opisujeme model údajov navrhovaného systému založený na databázovom modeli Wikka wiki upravenom pre potreby nášho systému.

4.3.1 Logický model údajov



Obr. č. 28 Logický model údajov

4.3.2 Opis logického modelu údajov

Jadrom celého systému je databáza MySQL pozostávajúca z 10 tabuliek, ktorých logické prepojenie je znázornené na obrázku č. 27. Hlavná tabuľka sa volá *pages*, do ktorej sa ukladajú stránky vytvorené používateľmi. Ostatné tabuľky slúžia na nasledujúce funkcie:

- *acls* – uložené informácie o kontrole prístupu na jednotlivé stránky
- *links* – informácie o odkazoch v rámci stránky – aká stránka odkazuje na akú
- *users* – uložení jednotliví zaregistrovaní používatelia a informácie o nich
- *sessions* – aktuálne prebiehajúce relácie prihlásených používateľov
- *comments* – používateľ môže okomentovať každú stránku
- *packages* – tabuľka obsahujúca informácie o jednotlivých balíčkoch
- *referrers* – zoznam externých stránok, ktoré odkazujú na niektorú z podstránok systému
- *referrers_blacklist* – zoznam externých stránok, ktoré sú považované za nežiadúce a odkazujú na systém len kvôli spamu
- *votes* – zoznam hlasovaní za jednotlivé balíčky, používateľ môže ohodnotiť každý balíček

4.3.3 Opisy atribútov entít

V nasledujúcich tabuľkách sa nachádzajú opisy jednotlivých atribútov tabuliek databázy. Určený je aj opis a názov atribútu.

Tab. č. 6 - Opis atribútov tabuľky *acls*

Tabuľka <i>acls</i>			
Kľúč	Názov atribútu	Dátový typ	Opis
FK	page_tag	VARCHAR (75)	odkazuje na tabuľku <i>pages</i>
	read_acl	TEXT	kto si môže stránku prezerať
	write_acl	TEXT	kto môže meniť obsah stránky
	comment_acl	TEXT	kto môže komentovať obsah stránky

Tab. č. 7 - Opis atribútov tabuľky *links*

Tabuľka <i>links</i>			
Kľúč	Názov atribútu	Dátový typ	Opis
FK	from_tag	VARCHAR (75)	ktorá lokálna stránka odkazuje
FK	to_tag	VARCHAR (75)	na ktorú lokálnu stránku sa odkazuje

Tab. č. 8 - Opis atribútov tabuľky users

Tabuľka <i>users</i>			
Kľúč	Názov atribútu	Dátový typ	Opis
PK	name	VARCHAR (75)	meno používateľa
	password	VARCHAR (32)	kto si môže stránku prezerat'
	email	VARCHAR (50)	kto môže menit' obsah stránky
	revisioncount	INT (10)	počet revízií, ktoré používateľ vrátil späť
	changescount	INT (10)	počet zmien prevedených vo svojich stránkach
	doubleclickedit	ENUM ('Y', 'N')	povoľuje menit' obsah svojich stránok dvojitém kliknutím
	signuptime	DATETIME	dátum a čas registrácie
	show_comments	ENUM ('Y', 'N')	dá sa stránka okomentovať
	status	VARCHAR (50)	kategorizácia používateľov
	prefer_platform	VARCHAR (50)	preferovaná platforma
	repository	VARCHAR (100)	relatívna cesta k používateľovmu repozitáru
	is_category_admin	ENUM ('Y', 'N')	informácia o tom, či je daný používateľ myslený ako admin kategória

Tab. č. 9 - Opis atribútov tabuľky sessions

Tabuľka <i>sessions</i>			
Kľúč	Názov atribútu	Dátový typ	Opis
PK	sessionid	CHAR (32)	náhodne vygenerovaný identifikátor spojenia
FK	userid	VARCHAR (75)	používateľ, ktorý vytvoril spojenie
	session_start	DATETIME	dátum a čas vytvorenia spojenia

Tab. č. 10 - Opis atribútov tabuľky comments

Tabuľka <i>comments</i>			
Kľúč	Názov atribútu	Dátový typ	Opis
PK	id	INT (10)	identifikátor komentára
FK	page_tag	VARCHAR (75)	na ktorú stránku bol pridaný
	time	DATETIME	kedy
	comment	TEXT	samotný text komentára

FK	user	VARCHAR (75)	autor komentára
----	------	--------------	-----------------

Tab. č. 11 - Opis atribútov tabuľky packages

Tabuľka <i>packages</i>			
Kľúč	Názov atribútu	Dátový typ	Opis
PK	package_id	INT (11)	identifikátor balíčka
FK	page_tag	VARCHAR (75)	wikka stránka, na ktorej je opísaný balíček
	name	VARCHAR (100)	názov
	short_description	TEXT	krátky opis použitia
	full_description	TEXT	dlhý opis použitia
	version	VARCHAR (10)	verzia
	status	VARCHAR (10)	stav v systéme (private, unstable, stable)
	dependency	TEXT	závislosť na lokálnych balíčkoch
	repo	VARCHAR (100)	repozitár, v ktorom je balíček uložený
FK	added	VARCHAR (75)	kto pridal balíček
	maintainer	VARCHAR (75)	správca balíčka, ktorý ho udržiava
	author	VARCHAR (75)	autor balíčka
	type	VARCHAR (100)	typ architektúry (x86, all...)
	arch	VARCHAR (100)	pre akú architektúru (Windows, Mac...)
	homepage	VARCHAR (100)	domovská stránka balíčka
	adding_datetime	DATETIME	čas a dátum pridania
	mail_comments	TINYINT(1)	informácia o tom, či chce používateľ dostávať komentáre k balíčku formou e-mailov
	published	DATETIME	čas a dátum publikovania balíčka
FK	published_by	VARCHAR (75)	používateľ, ktorý publikoval balíček

Tab. č. 12 - Opis atribútov tabuľky referrers

Tabuľka <i>referrers</i>			
Kľúč	Názov atribútu	Dátový typ	Opis
FK	page_tag	VARCHAR (75)	na ktorú lokálnu stránku prišiel externý odkaz
	referrer	VARCHAR (150)	referujúca stránka
	time	DATETIME	dátum a čas pripojenia

Tab. č. 13 - Opis atribútov tabuľky referrer_blacklist

Tabuľka <i>referrer_blacklist</i>			
Kľúč	Názov atribútu	Dátový typ	Opis
FK	spammer	VARCHAR (150)	referujúca stránka považovaná za spam

Tab. č. 14 - Opis atribútov tabuľky pages

Tabuľka <i>pages</i>			
Kľúč	Názov atribútu	Dátový typ	Opis
	id	INT (10)	identifikátor stránky
PK	tag	VARCHAR (75)	unikátny názov stránky
	time	DATETIME	čas a dátum vytvorenia
	body	MEDIUMTEXT	telo stránky
	owner	VARCHAR (75)	vlastník stránky – pri publikovateľných kategóriách je to jej názov
	user	VARCHAR (75)	používateľ, ktorý ju vytvoril
	latest	ENUM ('Y', 'N')	informácia o tom, či je stránka najnovšia – posledný krát zmenená
	note	VARCHAR (100)	poznámka – zväčša zmienka o tom, že stránka bola automaticky vygenerovaná
	handler	VARCHAR (30)	súbor php funkcií, ktoré obsluhujú danú stránku
	ispackagepage	ENUM ('Y', 'N')	informácia o tom, či je stránka, kde je informácia o balíčku
	mail_comments	TINYINT(1)	informácia o tom, či posilať komentáre na e-mail

Tab. č. 15 - Opis atribútov tabuľky votes

Tabuľka <i>votes</i>			
Kľúč	Názov atribútu	Dátový typ	Opis
PK	id	INT (10)	identifikátor hlasovania
Fk	page_tag	VARCHAR (75)	stránka, za ktorú bolo zahlasované
FK	user	VARCHAR (75)	používateľ, ktorý hlasoval
	rating	INT (10)	hodnota udeleného hlasu (administrátori majú hodnotu hlasu 2)
	ipaddress	VARCHAR (40)	IP adresa, z ktorej bolo hlasované

4.3.5 Zmeny oproti modelu WikkaWiki

V súčasnej dokumentácii k systému WikkaWiki neexistujú diagramy, ktoré by znázorňovali logický, či fyzický model údajov. Dokonca nie je ani opis tabuliek a atribútov, ktoré sa používajú. Po preštudovaní kódu a tabuliek vznikol logický dátový model, ktorý sme doplnili o špecifickú tabuľku *packages* a *votes*. Do tabuľky *users* pridané nasledovné atribúty:

- *prefer_platform* - informácia o preferovanej platforme používateľa,
- *repository* - repozitár používateľa,
- *is_category_admin* - informácia, či je používateľ virtuálny a tvorí kategóriu, do ktorej sa dá publikovať balíček.

Tabuľku *pages*, v ktorej sú uložené všetky stránky sme rozšírili o dva potrebné atribúty:

- *ispackagepage* – informácia o tom, či je to stránka, ktorá obsahuje balíček
- *mail_comments* – informácia o tom, či sa majú vlastníčkovi stránky posielat' komentáre na e-mail

Vo fyzickom modeli bol zmenený primárny kľúč tabuľky *pages* z atribútu *id* na *tag*, ktorý je tiež unikátny a je použitý ako cudzí kľúč v ďalších šiestich tabuľkách. Nakoniec boli pozmenené niektoré dĺžky dátových typov (*varchar*, *mediumint*...).

4.3.6 Zmeny oproti zimnému semestru

V konečnom dátovom modeli je niekoľko zmien oproti jeho návrhu zo zimného semestra. Asi najdôležitejšou je pridanie ďalšej tabuľky *votes*, ktorá slúži na ukladanie hlasovaní používateľov za jednotlivé balíčky. Okrem toho sme zmenili niekoľko atribútov:

- tabuľka *packages*
 - zmazaný *rating* – počítanie hlasov sa získava z tabuľky *votes*
 - zmazaný *mail_comments* – informácia o posielaní komentárov na e-mail je obsiahnutá v tabuľke *pages*
 - zmazané *ex_dependency* – odstranené z implementácie samotného LuaDistu
 - pridané *full_description*, *maintainer*, *homepage*, *type* – atribúty obsiahnuté v súbore *dist.info* podľa implementácie LuaDistu
 - pridaný *adding_datetime* – aby sme vedeli zobrazit', keď bol balíček pridaný
 - pridané *repo* – relatívna cesta k repozitáru, kde je balíček umiestnený

- pridané *published* a *published_by* – aby sme vedeli kto a kedy publikoval balíček
- tabuľka *pages*
 - pridané *ispackagepage* – informácia o tom, či daná stránka obsahuje balíček
 - pridané *mail_comments* – informácia o tom, či sa majú posielat' nové komentáre na e-mail vlastníka stránky
- tabuľka *users*
 - zmazané *mail_me* – presunuté do tabuľky *pages*
 - pridané *repo* – repozitár používateľa
 - pridané *is_category_admin* – informácia o tom, či je to virtuálny používateľ čiže kategória, do ktorej je možné publikovať balíčky

5 Prototyp

Táto kapitola opisuje prototyp, ktorý je výsledkom práce tímu v zimnom semestri. Uvedené sú zmeny v pôvodnom systéme wikka wiki a takisto pridaná funkcionálnosť.

Funkcionálnosť prototypu

Prvou implementovanou funkciou bolo registrovanie a prihlasovanie používateľa. Po vyplnení požadovaných údajov sa novému používateľovi vytvorí na serveri konto, v rámci ktorého má vlastný adresár, do ktorého má prístup len on sám. Vytváranie týchto adresárov bola ďalšia implementovaná funkcionálnosť. Ďalšia funkcionálnosť pozostáva z možnosti uploadovania súborov do vlastných adresárov na serveri. Jednoduchým vybratím súboru z lokálneho disku a následným stlačením tlačidla sa vybraný súbor uloží do určeného adresára.

Ostatné implementované funkcie sú pre bežného používateľa skryté a vykonávajú sa len na pozadí. Jednou z nich je generovanie manifestu a získavanie informácií z *.dist súborov. Obe funkcie sú potrebné pre správne fungovanie systému a na ich základe sa vytvárajú súbory potrebné pre vyhľadávanie už uploadovaných súborov programom LuaDist. Funkcie využívajú funkciu `exec`, ktorá vykonáva postupne príkazy z príkazového riadku. Toto riešenie vyžaduje povolený shell na serveri kde sa celá aplikácia nachádza.

5.1 Zmeny v dátovom modeli

V súčasnej dokumentácii k systému WikkaWiki neexistujú diagramy, ktoré by znázorňovali logický, či fyzický model údajov. Dokonca nie je ani opis tabuliek a atribútov, ktoré sa používajú. Po preštudovaní kódu a tabuliek vznikol logický dátový model, ktorý sme doplnili o špecifickú tabuľku *packages*. Takisto boli do tabuľky *users* pridané atribúty *prefer_platform* (informácia o preferovanej platforme používateľa) a *mail_me* (zasielanie notifikačných e-mailov).

Vo fyzickom modeli bol zmenený primárny kľúč tabuľky *pages* z atribútu *id* na *tag*, ktorý je tiež unikátny a je použitý ako cudzí kľúč v štyroch ďalších tabuľkách. Nakoniec boli pozmenené niektoré dĺžky dátových typov (`varchar`, `mediumint`...).

5.2 Registrácia nového užívateľa

Na registráciu nových používateľov systému sme ponechali registračné formuláre wiki a doplnili ich o novú funkcionálnosť. Každý používateľ nášho systému totiž musí mať vyhradené miesto, kde systém bude udržiavať ním vytvorené moduly. Tento adresár sa vytvára hneď ako systém overí všetky náležitosti spojené s registráciou užívateľov, tzn. overenie užívateľského mena a dĺžke hesla. Hneď potom na základe užívateľského mena vygeneruje názov jeho repozitára a na to používa kryptografickú hash funkciu MD5.

Ak by pre chybu systému alebo pre iné zlyhanie nebol tento repozitár vytvorený, vytvorí sa počas pôsobenia používateľa v systéme. Ak používateľ zvolí ukladanie balíčka do repozitára, tak systém kontroluje, či tento repozitár existuje a ak nie, vytvorí ho.

5.3 Upload súboru

Jednou z hlavných možností prihláseného používateľa je uploadovanie súborov do vlastného adresára. Upload je vyriešený pomocou jednoduchého php skriptu, ktorý umožní používateľovi vybrať súbor z lokálneho disku a uložiť ho stlačením tlačidla do svojho vlastného adresára na serveri. Uploadovať súbor sa dá jednoducho stlačením „Upload“ na používateľovej hlavnej stránke v hornom menu.

Najdôležitejšou funkciou v napísanom skripte je php funkcia „`move_uploaded_file()`“, ktorá uloží uploadovaný súbor, ktorý je zatiaľ v pamäti iba ako dočasný súbor, na konkrétne miesto na disku. V tomto prípade je to „`repo/user/`“, pričom `user` znamená meno prihláseného používateľa.

5.4 Generovanie manifestu a získanie informácií z dist.info

Ako bolo uvedené v kapitole 5.1, funkcie pre generovanie manifestu a získavanie informácií z balíčkov sú implementované ako obalovacie funkcie (wrappre) samotného `luadistu`. Tzn. že pomocou `exec()` volajú `luadist` a funkcie, ktoré sme do neho dopísali. Dopísané funkcie sú de facto len vyňatými a používajú existujúce funkcie `luadistu`. Všetky tieto funkcie sú uložené v knižnici `luadist.lib.php`.

Pred volaním niektorej z funkcií `getDist*(path)`, je potrebné modul rozbaľiť do dočasného adresára funkciou `unzip` a následne pre funkciu `getDist*(path)` ako `path` parameter poskytnúť cestu k `dist.info` súboru, ktorý bol rozbalený v rámci balíčka. Po získaní potrebných informácií, sa rozbalený balíček z dočasného adresára zmaže funkciou `rmdir_recursive()`, ktorá je v súbore `rmrecursive.php`.

Všetky súbory, potrebné pre tieto funkcie sú v adresári luadist.

Pri testovaní tejto implementácie na rôznych platformách sme odhalili problémy s poskytovaným hostingom, kde na serveri labss2 beží neznáma verzia linuxu, takže bolo obtiažne vytvoriť binárny súbor lua a luadist. Preto sme sa rozhodli túto implementáciu v ďalšej časti prerobiť, kde budú všetky funkcie implementované priamo v php, takže implementácia nebude závislá na platforme (resp. budeme závislá minimálne).

6 Produkt

Táto kapitola opisuje produkt, ktorý je výsledkom práce tímu po dvoch semestroch. Uvedené sú zmeny oproti návrhu, implementovaná funkcionálna vo vybranom prostredí a možnosti ďalšieho rozšírenia.

6.1 Zmeny oproti návrhu

Oproti pôvodnému návrhu bolo zmenených viacero prípadov použitia: UC1, UC2 a UC6. Zrušený bol UC5 a pridané UC11 – zmazanie balíčka a UC12 správa *admin* kategórií.

- zmena UC1- vkladanie nového balíčka: užívateľ vytvorí .dist súbor sám a do systému ho už len vloží cez formulár. Z vloženého súboru sú následne načítané údaje, na základe ktorých je vytvorená základná informačná stránka ku balíčku,
- zmena UC2 – editácia existujúceho balíčka: užívateľ nemá umožnenú priamu editáciu existujúceho balíčka, môže však meniť informácie na stránke balíčka. Pokiaľ chce autor balíčka zmeniť samotný súbor, musí najprv zmazať existujúci balíček. Táto črta systému podporuje tvorbu verzií zo strany užívateľov,
- zmena UC6 – publikovanie (zverejnenie) balíčka (súvisí so zrušením UC5): publikovať balíček môže jeho autor alebo administrátor. Publikovanie znamená zaradenie daného balíčku do istej kategórie. Kategórie vytvárajú a spravujú administrátori. Balíček je po publikovaní prístupný všetkým užívateľom systému,
- UC11 – zmazanie balíčka: zmazať balíček môže len jeho autor alebo administrátor. Tento prípad použitia vznikol ako náhrada za časť UC5 – označ balíček ako *dead*. Prístup k tomuto prípadu použitia je zo zoznamu balíčkov (UC3),
- UC12 – správa *admin* kategórií: *admin* kategória je priečinok, do ktorého sa zverejňujú balíčky. Môže ich byť neobmedzené množstvo.

6.2 Výber implementačného prostredia

Ako základ implementácie systému sme vybrali WikkaWikki. Jej vlastnosti aj zdôvodnenie výberu sa nachádza v časti Analýza.

6.3 Realizácia a implementácia

V tejto časti sú opísané realizácie jednotlivých prípadov použitia, pričom sú spomenuté zaujímavé implementačné črty, ktoré tvoria systém. Technické detaily implementácie sú

lepšie zdokumentované v technickej dokumentácii. Tu je potrebné spomenúť, že pôvodná implementácia funkcií na získavanie informácií z dist.info súborov a funkcií pre generovanie manifestu bola zmenená oproti prototypu. Pôvodná implementácia založená na obalovaní luadist bola nahradená vlastnou php implementáciou týchto funkcií.

6.3.1 Realizácia UC1 – vloženie nového balíčka

Vloženie nového balíčka do systému (repozitára užívateľa) je implementované pomocou jednoduchého formulára, kde po vybratí .dist súboru z lokálneho disku, vyplnenie bezpečnostnej captcha a odoslání formulára tento súbor uložený na serveri (pokiaľ ide o dist balíček, teda obsahuje dist.info súbor). Spomínaná captcha pre administrátorov nie je zobrazovaná. Na základe dist.info súboru je pre balíček vygenerovaná wikki stránka a informácie o balíčku sú pridané do manifestu v repozitári užívateľa, ktorý tento balíček uložil. Vygenerovaná wikki stránka je prístupná len pre autora balíčka a administrátorov.

Dôležitou funkciou v napísanom skripte je php funkcia „`move_uploaded_file()`“, ktorá uloží uploadovaný súbor, ktorý je zatiaľ v pamäti iba ako dočasný súbor, na konkrétne miesto na disku. V tomto prípade je to `repo/user_hash/`. Nakoľko je potrebné, aby toto miesto bolo prístupné z internetu pre luadist samotný, `user_hash` je vytvorený ako MD5 hash mena užívateľa a `sol`, čo je iné náhodné číslo. Keby to nebol hash ale len meno samotné, bolo by ľahké ostatným užívateľom získať obsah repozitára, t.j. mohli by sa dostať aj k ešte nezverejneným balíčkom. Tento spôsob je relatívne bezpečný, hoci stále je možné URL adresára „uhádnuť“ generovaním hash, avšak je to časovo náročné. Do budúcnosti je však potrebné sa zamyslieť nad autorizácii prístupu k URL na úrovni web servera.

Informácie o balíčku sú pridané do manifestu aditívne, t.j. sú zapísané na koniec súboru.

Pridať balíček je ďalej možné pomocou utility napísanej v java. Utilita prehľadá daný adresár a vloží všetky nájdené .dist súbory do repozitára užívateľa na serveri. Prihlasovacie meno a heslo sa dávajú utilite ako vstupné parametre rovnako ako cieľová URL, ktorá musí byť adresa servera s požiadavkou na stránku `DistUploadAuto`. Utilita má aj GUI rozhranie pre jednoduchšie vloženie požadovaných parametrov. Viac v používateľskej príručke.

6.3.2 Realizácia UC2 – editácia existujúceho balíčka

Editácia existujúceho balíčka je obmedzená na editovanie vygenerovanej wiki stránky, čo je črta samotnej wiki. Samotná editácia vloženého súboru nie je umožnená. Pokiaľ chce užívateľ zmeniť tento súbor, musí existujúci balíček zmazať a znovu vložiť nový. Táto vlastnosť

systemu nepriamo podporuje vytváranie verzií. Teda pokiaľ chce používateľ opraviť balíček, je jednoduchšie ho vložiť s novou verziou.

6.3.3 Realizácia UC3 – prezeranie balíčkov a UC11 zmazanie balíčka

Prezeranie balíčkov je implementované ako stránka s odkazom v hornom menu: *Dist Index*. Na tejto stránke sú zobrazené všetky verejne prístupné balíčky a balíčky, ktorých autor je aktuálny používateľ, ak je prihlásený. Tiež sú na stránke zobrazené balíčky, na ktoré má užívateľ právo od ich autora. Na stránke je filter, pomocou ktorého je možné filtrovať balíčky podľa autora, resp. kategórie.

Vyfiltrované balíčky sú zobrazené v tabuľke, v ktorej sú nasledovné údaje: meno, verzia, popis, stav publikácie a v poslednom stĺpci sú akcie. Tabuľka je organizovaná podľa verzií balíčkov, t.j. pod menom a popisom balíčku sú v riadkoch zobrazené jednotlivé verzie. V stĺpci akcie je možnosť zmazať balíček. Nad tabuľkou je tlačidlo umožňujúce regenerovať manifest súbor v repozitári autora nasledujúcich balíčkov. Toto tlačidlo je zobrazené len pre administrátorov, resp. pre autora balíčkov. Rovnako je pre autora a administrátorov v stĺpci akcie zobrazené tlačidlo pre zmazanie daného balíčku. Tabuľka je štýlovaná pomocou CSS tak aby tvorila tzv. *tree-table*, kde kmeň je balíček a listy verzie balíčka.

Spomínaná regenerácia manifestu je implementovaná ako funkcia, ktorá si nájde informácie o balíčkoch v databáze. Pôvodný manifest je zmazaný a je vytvorený nový. Rovnaká funkcia (*createManifestFromDb()*) je volaná po zmazaní nejakého balíčku zo systému.

Z tabuľky je možné sa dostať na stránku balíčka, na ktorej sú informácie získané z *dist.info* súboru a odkaz na stiahnutie *.dist* súboru. Stránka ďalej umožňuje zverejniť balíček, hlasovať a pridať komentár.

6.3.4 Realizácia UC4, UC7 a UC8 – hlasovanie, komentovanie balíčka

Implementácia hlasovania je pre užívateľov dostupná vo forme troch tlačidiel na konci stránky balíčku. Ide o pridanie pozitívneho/negatívneho bodu a v prípade autora/administrátora o tlačidlo resetujúce hlasovanie. Každý užívateľ má len jeden hlas pre daný balíček.

Komentovať balíček je možné pomocou tlačidla po stlačení ktorého je zobrazený formulár. Odoslaním formulára sa ku stránke pridá komentár s menom autora a časom.

Zmazať komentár môže jeho autor alebo administrátor pomocou tlačidla, ktoré vymaže tento komentár z databázy. Hlasovanie môže resetovať len administrátor.

6.3.5 Realizácia UC6 – sprístupnenie/zverejnenie balíčka

Sprístupnenie balíčka sa deje cez jednoduchý výberový formulár na stránke balíčka, kde sa vyberie kategória, do ktorej bude balíček zaradený. Po odoslaní formulára je balíček skopírovaný do repozitára tejto kategórie a je pre neho vytvorená nová stránka ako kópia existujúcej. Táto stránka je editovateľná už len administrátormi. Pokiaľ však autor balíčku tento balíček zmaže, je zmazaný aj z tejto kategórie.

6.3.6 Realizácia UC10 – registrácia nových užívateľov

Na registráciu nových používateľov systému sme ponechali registračné formuláre wiki a doplnili ich o novú funkcionálnosť. Každý používateľ nášho systému totiž musí mať vyhradené miesto, kde systém bude udržiavať ním vytvorené moduly. Tento adresár sa vytvára hneď ako systém overí všetky náležitosti spojené s registráciou užívateľov, tzn. overenie užívateľského mena a dĺžke hesla. Hneď potom na základe užívateľského mena vygeneruje názov jeho repozitára a na to používa kryptografickú hash funkciu MD5.

Ak by pre chybu systému alebo pre iné zlyhanie nebol tento repozitár vytvorený, vytvorí sa počas pôsobenia používateľa v systéme. Ak používateľ zvolí ukladanie balíčka do repozitára, tak systém kontroluje, či tento repozitár existuje a ak nie, vytvorí ho.

6.3.7 Realizácia UC12 – správa kategórií

Správa kategórií je prístupná administrátorom systému cez položku menu *CreateCategory*. Tu vyplní formulár, kde vyplní meno kategórie, heslo a mail. Kategória je totiž implementovaná ako špeciálny typ užívateľa.

6.4 Možnosti rozšírenia

Systém má dobré možnosti rozšírenia. Pri samotnom rozširovaní je možné veľmi dobre využiť existujúcu architektúru základnej WikkaWikki, t.j. nadefinovať akciu napísanú v php a tú následne vložiť do wiki stránky pomocou *{action}*.

V budúcnosti je systém možné rozšíriť napr. o podporu menenia nezverejnených balíčkov bez nutnosti ich mazania, zlepšenie bezpečnosti, rozšíriť filter pre prezeranie balíčkov, vylepšenie utility na vkladanie balíčkov mimo web-rozrania, ...

7 Testovanie

Testovanie prebiehalo priebežne s implementáciou, niektoré UC zo špecifikácie testované neboli, pretože pri zmenách špecifikácie stratili svoj význam. V tejto kapitole sa nachádzajú finálne testy vo forme testovacích scénárov.

7.1 Testovacie scenáre

ID	1	Názov	Pridanie balíčka		
Prípado použitia	UC01	Úroveň splnenia testu	Musí – Mal by – Mohol by		
Rozhranie	DistUpload				
Účel	Overenie správnej funkčnosti pridávania balíčkov.				
Vstupné podmienky	Používateľ je zaregistrovaný a prihlásený.				
Výstupné podmienky	Balíček je nahraný, je vygenerovaná stránka pre balíček a .dist súbor sa nachádza v repozitári používateľa.				
Krok	Akcia	Očakávaná reakcia		Skutočná reakcia	
1	V menu kliknúť na položku Dist Upload	Systém zobrazí stránku pre nahrávanie balíčkov, v prípade obyčajného používateľa aj captchu, v prípade administrátora bez captche.		Systém zobrazí stránku pre nahrávanie balíčkov, v prípade obyčajného používateľa aj captchu, v prípade administrátora bez captche.	
2	Výber súboru pre nahratie.	Systém zobrazí vybraný súbor v poli pre zadanie cesty k súboru.		Systém zobrazí vybraný súbor v poli pre zadanie cesty k súboru.	

ID	1	Názov	Pridanie balíčka	
3	Potvrdenie pridania balíčka	Systém zobrazí informácie o balíčku a hypertextový odkaz ku stránke vygenerovanej pre tento balíček.		Systém zobrazí informácie o balíčku a hypertextový odkaz ku stránke vygenerovanej pre tento balíček.

ID	2	Názov	Publikovanie balíčka		
Prípado použitia	UC06	Úroveň splnenia testu	Musí – Mal by – Mohol by		
Rozhranie	Stránka pre balíček				
Účel	Overenie správnej funkčnosti publikovania balíčkov.				
Vstupné podmienky	Balíček je prítomný v repozitári používateľa				
Výstupné podmienky	Balíček je prístupný pre všetkých používateľov a zobrazuje sa v dist indexe kategórie, do ktorej bol publikovaný, takisto bol skopírovaný do repozitára kategórie do ktorej bol publikovaný.				
Krok	Akcia	Očakávaná reakcia		Skutočná reakcia	
1	Navigácia na stránku balíčka	Systém zobrazí stránku pre balíček.		Systém zobrazí stránku pre balíček.	
2	Výber kategórie	V listboxe sa zobrazí vybraná kategória.		V listboxe sa zobrazí vybraná kategória.	
2	Potvrdenie publikovania	Balíček je skopírovaný do repozitára kategórie a prístupný pre všetkých používateľov.		Balíček je skopírovaný do repozitára kategórie a prístupný pre všetkých používateľov.	

ID	3	Názov	Zmazanie hlasovania		
Prípado použitia	UC07	Úroveň splnenia testu	Musí – Mal by – Mohol by		
Rozhranie	Stránka balíčku				
Účel	Overenie správnej funkčnosti modulu hlasovania.				
Vstupné podmienky	Nenulový počet hlasov				
Výstupné podmienky	Počítadlo hlasov má hodnotu 0.				
Krok	Akcia		Očakávaná reakcia		Skutočná reakcia
1	Navigácia na stránku balíčka.		Systém zobrazí stránku pre balíček.		Systém zobrazí stránku pre balíček.
2	Resetovanie hlasovania		Systém vymaže hlasovanie pre tento balíček a počítadlo hlasov nastaví na hodnotu 0.		Systém vymaže hlasovanie pre tento balíček a počítadlo hlasov nastaví na hodnotu 0.

ID	4	Názov	Registrácia používateľa		
Prípado použitia	UC10	Úroveň splnenia testu	Musí – Mal by – Mohol by		
Rozhranie	Login/Register				
Účel	Overenie správnej funkčnosti registrácie používateľov.				
Vstupné podmienky	-				
Výstupné podmienky	Používateľ je zapísaný do databázy a má vygenerovaný repozitár.				
Krok	Akcia		Očakávaná reakcia		Skutočná reakcia
1	Vyplnenie formuláru pre registráciu a kliknutie na tlačidlo „Register“		Zobrazenie stránky s používateľovým profilom aj s linkou na jeho repozitár		Zobrazenie stránky s používateľovým profilom aj s linkou na jeho repozitár

ID	5	Názov	Zmazanie balíčka		
Prípado použitia	UC11	Úroveň splnenia testu	Musí – Mal by – Mohol by		
Rozhranie	DistIndex				
Účel	Overenie správnej funkčnosti mazania balíčkov.				
Vstupné podmienky	Balíček je nahraný v repozitári.				
Výstupné podmienky	Balíček nie je v repozitári a nezobrazuje sa informácia o ňom ani v DistIndex-e				
Krok	Akcia	Očakávaná reakcia		Skutočná reakcia	
1	V menu kliknúť na položku DistIndex	Systém zobrazí stránku pre prezeranie balíčkov.		Systém zobrazí stránku pre prezeranie balíčkov.	
2	Potvrdenie zmazania	Systém odstráni záznam z DB, súbor z repozitára a aktualizuje DistIndex		Systém odstráni záznam z DB, súbor z repozitára ale aktualizuje DistIndex až po obnovení stránky.	

8 Záver

Informácie v tejto dokumentácii odrážajú prácu celého tímu počas dvoch semestrov. V komunikácií so školiteľom sme ukončili projekt webového systému správy rozširujúcich modulov jazyka Lua nasadením do ostrej prevádzky.

Počas trvania projektu sme prenikli do problematiky jazyka Lua a jeho distribúcie LuaDist, zvolili vyhovujúci wiki systém a presne špecifikovali požiadavky, čím sme uzavreli časť analýzy. V návrhu sme identifikovali jednotlivé procesy systému, zakreslili ich do diagramov činností, doplnili a navrhli logický a fyzický model údajov a predstavili návrhy obrazoviek používateľského rozhrania. Počas implementácie sme rozšírili nasadený wiki systém Wikka o nové funkcionality, z ktorých najpodstatnejšie sú nahratie súboru s príponou .dist na server a jeho analýza a generovanie súboru dist.manifest na základe obsahu repozitára. Počas tejto fázy projektu sme zaznamenali aj zmeny požiadaviek, ktoré neboli obsiahnuté v špecifikácii (publikovanie balíčku do špeciálnych kategórií, automatické nahrávanie súborov na server), na ktoré sme vedeli flexibilne zareagovať a promptne rozšíriť existujúci dátový model a implementovať požadovanú funkcionality. Programovanie sme ukončili fázou testovania, na ktorej sa podieľal celý tím spoločne s pedagogickým vedúcim projektu.

Zdrojový kód spolu s rozsiahlou dokumentáciou sme odovzdali vedúcemu tímu, ktorý zastupoval v tomto prípade rolu zákazníka. Môžeme prehlásiť, že projekt bol úspešný nakoľko sa nám podarilo implementovať a nasadiť funkčný systém reflektujúci všetky požiadavky v stanovenom čase a s plánovanými zdrojmi. Systém bude reálne nasadený a bude slúžiť komunitě združenej okolo jazyka Lua a podporovať najmä projekt systému LuaDist.

9 Použitá literatúra

- [1] Bieliková, M. Softvérové inžinierstvo: Princípy a manažment. Slovenská technická univerzita v Bratislave. 220 s. 2000.

- [2] Bieliková, M.: Ako úspešne vyriešiť projekt. Slovenská technická univerzita v Bratislave. 158 s. 2000.

- [3] UML Use Case Diagrams: Tips and FAQ
<http://sds.hss.cmu.edu/courses/Syllabi/ids/271/umlfaq.asp>

- [4] Nanoki, wiki systém v Lue (<http://alt.textdrive.com>)

- [5] Reviki, wiki systém v Jave na báze SVN (<http://reviki.org>)

- [6] Wikka wiki, PHP wiki (<http://wikkawiki.org>)

- [7] Lua Dokumentacia: (<http://www.lua.org>)

- [8] Lua Reference manual (<http://www.lua.org/manual/5.1/>)

- [9] Ierusalimschy R., Programming in Lua, December 2003
ISBN 85-903798-1-7 (<http://www.lua.org/pil/>)

- [10] CPAN, System pre správu modulov pre jazyk perl (<http://search.cpan.org/>)

- [11] LuaRocks (<http://www.luarocks.org>)

- [12] LuaDist (<http://luadist.sf.net>)

- [13] Luaforge - Stranka zdruzujuca Lua projekty (<http://www.luaforge.net>)

- [14] Komunitne wiki (<http://lua-users.org/>)

[15] JIT compiler pre Luu (<http://luajit.org/>)

[16] Sputnik, wiki systém v Lue (<http://sputnik.freewisdom.org/>)

[17] MinGW (<http://www.mingw.org/>)

[18] Make system pre kompiláciu modulov na roznych platformach (<http://www.cmake.org/>)

Príloha A – Technická dokumentácia

1 Technická dokumentácia

Nasledujúca kapitola prináša hlbší pohľad na fungovanie aplikácie a procesy, ktoré v nej prebiehajú. Skladá sa z dvoch hlavných podkapitol. Prvá opisuje fungovanie samotnej wiki, ktorá zaoberá množstvo funkcionality počas každej žiadosti na server a druhá prinesie pohľad na našu prácu a zmeny, ktoré sme vo wiki uskutočnili.

1.1 WikkaWiki

Táto kapitola prináša komplexný pohľad na fungovanie samotnej wiki, na ktorej je naše riešenie postavené. Porozumenie tejto problematiky prinesie čitateľovi lepšie pochopenie ďalších častí dokumentácie.

Samotná wiki sa skladá z niekoľkých dôležitých častí. Patria sem moduly pre správu volaní na server, konfiguráciu, prácu s databázou a prácu so stránkami vo wiki. Všetky sú prehľadne a hierarchicky usporiadané v adresári wiki.

Ako wiki pracuje môžeme vidieť na sekvenčnom diagrame dole (**obr. č. 1**), ktorý zobrazuje priebeh práce wiki. Diagram ukazuje sekvenciu procesov od http žiadosti prijatej od klienta až po odoslanie odpovede klientovi naspäť. Každá žiadosť na server musí smerovať na `wikka.php` stránku, pretože práve tento php skript vykonáva všetky operácie spojené s wiki a stránkami v nej uloženými. Ďalším údajom, ktorý musí http žiadosť obsahovať je parameter `wakka`. Jeho hodnota obsahuje názov wiki stránky, ktorú chceme zobraziť. Ďalšie údaje v http žiadosti nie sú povinné ale môžu sa vyskytovať. Môžu byť oddelené znakmi `'&'` alebo `'/'`. V nasledujúcej tabuľke sú príklady formátu URL v http žiadosti a akcie, ktoré aplikácia vykoná.

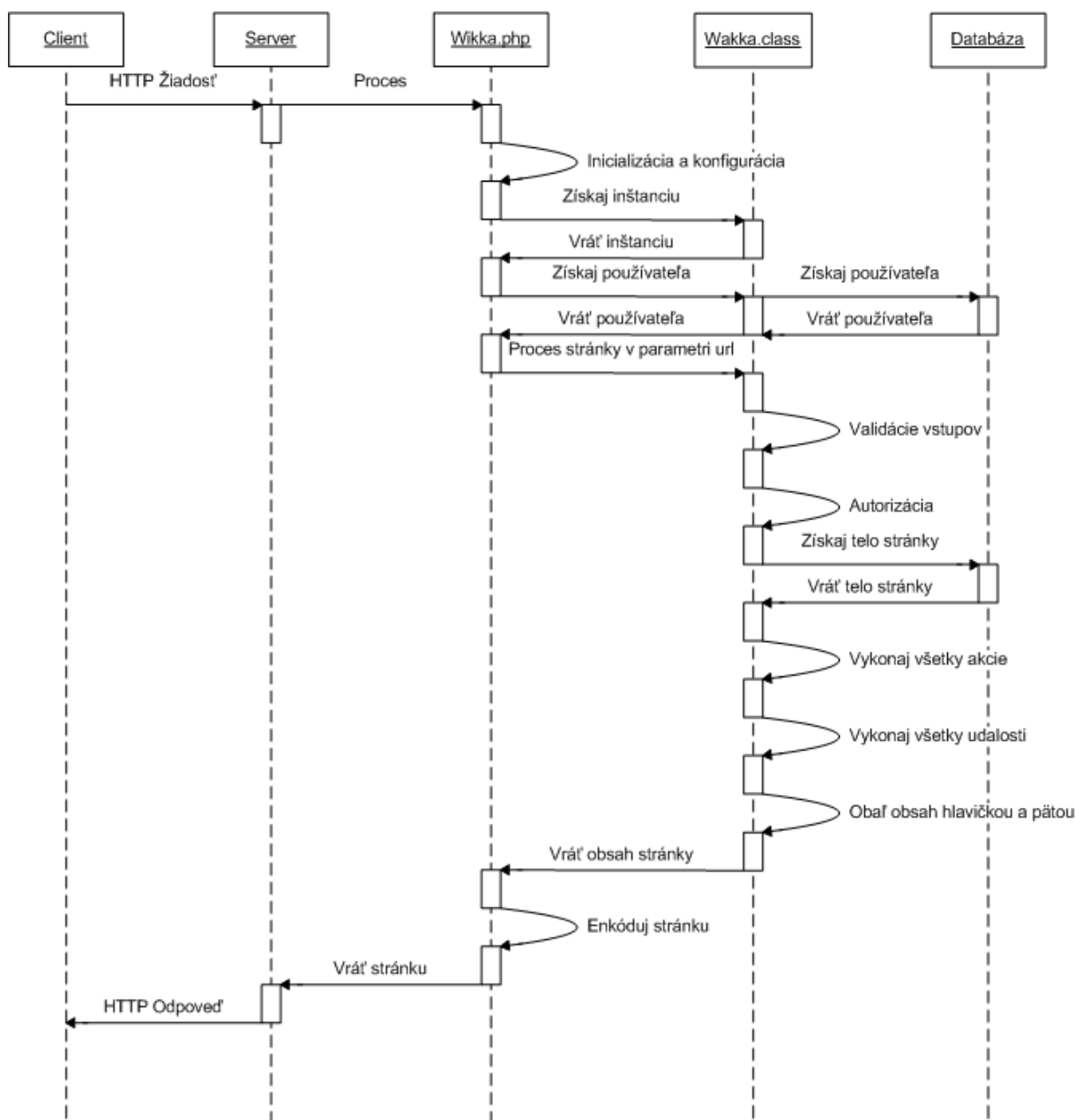
URL	Akcia
<code>http://server/</code>	Automaticky presmeruje žiadosť na URL http://server/wikka.php .
<code>http://server/wikka.php</code>	Automaticky načíta parameter <code>wakka</code> z konfiguračného súboru a presmeruje na http://server/wikka.php?wakka=DefaultPage .
<code>http://server/wikka.php?wakka=HomePage</code>	Zobrazí stránku s názvom „HomePage“.
<code>http://server/wikka.php?wakka=DistIndex&user=UserName</code>	Ukážka viacerých parametrov, jedná sa o stránku s menom „DistIndex“ ktorá v prípade zadania ďalších parametrov zobrazí filtrované údaje na základe týchto parametrov. V tomto prípade pre používateľa s menom „UserName“.

http://server/wikka.php?wakka=DistIndex/edit	Ukážka tzv. zachytávača udalosti ^[1] s menom „edit“, ktorý indikuje pre wiki že danú stránku chce používateľ editovať.
---	---

Ako je vidieť, wiki vždy pracuje so súborom `wikka.php` a v ňom prebiehajú jednotlivé procesy generovania stránky. V prvom rade pri každom volaní skript všetko inicializuje a načíta konfiguráciu, následne vytvorí inštanciu aplikačnej logiky pre stránky lokalizovanú v súbore s názvom „`Wakka.class.php`“. Potom overí aktuálneho používateľa a získa o ňom údaje z databázy. Nasleduje proces získania údajov z URL v http žiadosti a po ňom skript predá ďalšie úlohy na riešenie už pre inštanciu aplikačnej logiky.

Tá v prvom rade validuje vstupné parametre a autorizuje používateľa. Ak všetko prebehne v poriadku, získa z databázy telo stránky. V ňom sa môžu nachádzať tzv. akcie, ktoré sú špeciálnymi akciami wiki. Každá akcia je vlastne ďalším php skriptom a wakka tieto akcie postupne volá podľa toho v akom poradí sú v stránke zapísané. Výsledky týchto akcií vkladá do finálnej stránky, ktorá sa bude odosielať používateľovi v http odpovedi. Po akciách nasledujú už spomínané zachytávače udalostí, ktoré sú uvedené v parametroch URL http žiadosti a ak sa v nej nejaký zachytávač nachádza, wakka uplatní jeho php skript a stránku náležite podľa neho modifikuje. Proces vytvárania stránky je ukončený vložení hlavičky a päty dopredu, resp. dozadu do vygenerovanej stránky. Hlavička a päta každej stránky sú definované v šablónach taktiež ako php skripty.

Po skončení tohto procesu wakka vráti výsledok pre `wikka.php`. Tá už len stránku zakóduje a vráti serveru. Následne server vráti stránku pre používateľa.



Obr. č. 1 - Sekvenčný diagram práce wiki od žiadosti až po odpoveď.

Nasledujúce podkapitoly sú ukážkou práce jednotlivých častí wiki, ktorých všeobecný a skráteneý opis sme mohli vidieť v predchádzajúcom sekvenčnom diagrame.

1.1.1 Wikka.php

Ako už bolo spomínané, skript umiestnený v súbore **wikka.php** je srdcom celej aplikácie. Všetka funkcionlita sa vykonáva prostredníctvom nej. Stará sa o správne inicializovanie globálnych premenných, konfiguráciu aplikácie a taktiež o report chýb, ktoré sa vyskytnú počas generovania stránky.

Konfigurácia sa skladá z dvoch častí. Prvá je definovanie najdôležitejších premenných bez ktorých wiki nemôže fungovať a ich uloženie do pamäte ako pole konfiguračných hodnôt. Táto definícia je napísaná natvrdo v kóde. Následne je načítaný konfiguračný súbor a údaje z neho sú zlúčené s týmito premennými a výsledkom je kompletná konfigurácia. Pri zlučovaní platí pravidlo, že ak existuje konfiguračný záznam natvrdo napísaný v kóde a taktiež existuje preň záznam aj v konfiguračnom súbore tak wiki použije záznam z konfiguračného súboru. Pre zlučovanie využíva wiki php funkciu **array_merge()**, ukážka zdrojového kódu je na obrázku č. 2.

```
257 $wakkaDefaultConfig = array(  
258     'mysql_host'           => 'localhost',  
259     'mysql_database'     => 'wikka',  
260     'mysql_user'         => 'wikka',  
261     'table_prefix'       => 'wikka_',  
262     'root_page'          => 'HomePage'  
263 );  
264  
265 // load config  
266 $wakkaConfig = array();  
267 if (file_exists($configfile)) include($configfile);  
268  
269 $wakkaConfig = array_merge($wakkaDefaultConfig, $wakkaConfig); // merge defaults with config from file
```

Obr. č. 2 - Ukážka zlučovania konfigurácií pomocou metódy array_merge().

1.1.2 Wakka.class.php

Tento skript obsahuje triedu tvoriacu logiku aplikácie a to najmä pre správu stránok a prácu s nimi. Kľúčovou je jej metóda **Run()**, ktorá je vstupným bodom celej logiky a ktorá je volaná zo skriptu **wikka.php**. Stará sa o validácie vstupných údajov, overenie autorizácie používateľa a o doplnenie chýbajúcich údajov v prípade, že neboli zadané. Tieto údaje berie z konfigurácie aplikácie. Tú vytvoril už predtým práve skript **wikka.php** a konfiguráciu predal ako vstupný parameter pre konštruktor triedy aplikačnej logiky **Wakka.class**.

Zároveň táto trieda obsahuje množstvo pred pripravených metód, ktoré sú využívané samotnou wiki alebo rôznymi akciami a zachytávačmi udalostí počas generovania stránok.

1.1.2.1 Používateľ

Wakka.class má k dispozícii množstvo metód aj pre správu používateľov. Nasleduje zoznam tých najdôležitejších alebo často využívaných metód, určené hlavne na získavanie údajov o používateľovi, keď je potrebné rozhodnúť, či je používateľ autorizovaný na určitú činnosť alebo ktoré dáta sa mu môžu zobrazit' a ktoré nie.

LoadUser() je metóda, ktorá vráti všetky potrebné údaje o aktuálnom používateľovi, ktoré o ňom existujú v databáze. Využíva sa najmä pri prvotnej autorizácii, neskôr pri zobrazovaní používateľových nastavení. Počas behu wiki aplikácia často overuje, či je nejaký používateľ

autentifikovaný alebo nie, vtedy využíva jednoduchší variant tejto metódy **GetUser()**. Táto metóda pracuje s reláciou^[4] a údaje o používateľovi číta z nej, čiže priamo z RAM serveru a nemusí preto volať databázu.

```
1694 function LoadUser($name, $password = 0)
1695 {
1696     return $this->LoadSingle(
1697         "select * from ".$this->config['table_prefix'].
1698         "users where name = '".mysql_real_escape_string($name)."' ".
1699         ($password === 0 ? "" : "and password = '".
1700         mysql_real_escape_string($password)."'")." limit 1");
1701 }
```

IsAdmin() je metóda na určenie, či aktuálny používateľ systému má administrátorské privilégia v systéme. Je veľmi často využívaná a obsiahnutá v množstve akcií, pretože administrátori môžu vykonávať všetky úkony v systéme.

```
1816 function IsAdmin($user='') {
1817     $adminstring = $this->config["admin_users"];
1818     $adminarray = explode(',', $adminstring);
1819
1820     if(TRUE===empty($user))
1821     {
1822         $user = $this->GetUserName();
1823     }
1824     else if(is_array($user))
1825     {
1826         $user = $user['name'];
1827     }
1828     foreach ($adminarray as $admin) {
1829         if (trim($admin) == $user) return true;
1830     }
1831 }
```

UserIsOwner() je metóda, ktorá vráti informáciu, či je aktuálny používateľ vlastníkom stránky, ktorú si práve prezerá. Táto metóda je dôležitá napríklad pri určení, či používateľ môže editovať stránku alebo nastavovať práve pre prístup iných používateľov k nej.

```

1803     function UserIsOwner($tag = "")
1804     {
1805         // if not logged in, user can't be owner!
1806         if (!$this->GetUser()) return false;
1807
1808         // if user is admin, return true. Admin can do anything!
1809         if ($this->IsAdmin()) return true;
1810
1811         // set default tag & check if user is owner
1812         if (!$tag = trim($tag)) $tag = $this->GetPageTag();
1813         if ($this->GetPageOwner($tag) == $this->GetUserName()) return true;
1814     }

```

LoadACL() je metóda slúžiaca na načítanie používateľových práv, ktoré pre aktuálnu stránku má. Využíva sa pri autorizácii používateľa a taktiež pri zobrazovaní listov existujúcich stránok, aby sa v listoch nachádzali len údaje o stránkach, na ktoré má používateľ prístupové práva.

```

1856     function LoadACL($tag, $privilege, $useDefaults = 1)
1857     {
1858         if (!$acl = $this->LoadSingle(
1859             "SELECT ".mysql_real_escape_string($privilege).
1860             "_acl FROM ".$this->config["table_prefix"].
1861             "acls WHERE page_tag = '"
1862             .mysql_real_escape_string($tag)."' LIMIT 1"))
1863             && $useDefaults)
1864         {
1865             $acl = array("page_tag" => $tag, $privilege."_acl" =>
1866                 $this->GetConfigValue("default_".$privilege."_acl"));
1867         }
1868         return $acl;
1869     }

```

LogoutUser() je metóda na odhlásenie používateľa, ktorá sa stará o odstránenie všetkých stôp, ktoré používateľ v systéme zanechal a následne ho z neho odhlási.

```

1706     function LogoutUser ()
1707     {
1708         $this->DeleteCookie ("user_name");
1709         $this->DeleteCookie ("pass");
1710         // Delete this session from sessions table
1711         $this->Query (
1712             "DELETE FROM ".$this->config['table_prefix']."sessions
1713             WHERE userid='".$this->GetUserName ()."' AND sessionid='".$
1714             session_id()."'");
1715         $_SESSION["user"] = "";
1716         // This seems a good as place as any to purge all session records
1717         // older than PERSISTENT_COOKIE_EXPIRY, as this is not a
1718         // time-critical function for the user. The assumption here
1719         // is that server-side sessions have long ago been cleaned up by PHP.
1720         $this->Query (
1721             "DELETE FROM ".$this->config['table_prefix'].
1722             "sessions WHERE DATE_SUB(NOW(), INTERVAL ".
1723             PERSISTENT_COOKIE_EXPIRY." SECOND) > session_start");
1724     }

```

1.1.2.2 Stránky

Nasledujúca kapitola opisuje metódy pre správu stránok, ktoré sú neodmysliteľnou súčasťou správy stránok wiki.

SavePage() je metóda využívaná na ukladanie stránok do databázy. Každá nová stránka alebo každá zmenená existujúca stránka sa pomocou tejto metódy vkladá do databázy. Pri zmenených stránkach metóda najprv označí všetky predchádzajúce verzie stránok ako neaktuálne a následne vytvorí nový záznam v databáze s aktuálnou podobou stránky. Takýmto spôsobom vytvára verzie stránok a pomocou wiki nie je problém vrátiť stránku do predchádzajúcich stavov. Príklad zdrojového kódu nižšie je aj so zmenami vykonanými počas našej práce, kde táto metóda bola rozšírená o nepovinný parameter **is_package_page**, ktorý definuje typ vytváranej stránky. Tie sme v rámci návrhu rozdelili na wiki stránky a stránky zobrazujúce informácie o balíčkoch. Parameter udáva či stránka ktorú ukladáme do databázy je onou spomínanou stránkou pre balíček.

```

888     function SavePage($tag, $body, $note, $owner=null, $is_package_page='N')
889     {
890         // get current user
891         $user = $this->GetUserName();
892
893         // TODO: check write privilege
894         if ($this->HasAccess("write", $tag))
895         {
896             // If $owner is specified, don't do an owner check
897             if (empty($owner))
898             {
899                 // is page new?
900                 if (!$oldPage = $this->LoadPage($tag))
901                 {
902                     // current user is owner if user is logged in, otherwise, no owner.
903                     if ($this->GetUser()) $owner = $user;
904                 }
905                 else
906                 {
907                     // oh! page isn't new. keep owner!
908                     $owner = $oldPage["owner"];
909                     $is_package_page = $oldPage["ispackagepage"];
910                 }
911             }
912             // set all other revisions to old
913             $this->Query(
914                 "update ".$this->config["table_prefix"]."pages set latest = 'N' where tag = '".
915                 mysql_real_escape_string($tag)."'");
916             // add new revision
917             $this->Query("insert into ".$this->config["table_prefix"]."pages set ".
918                 "tag = '".mysql_real_escape_string($tag)."', ".
919                 "time = now(), ".
920                 "owner = '".mysql_real_escape_string($owner)."', ".
921                 "user = '".mysql_real_escape_string($user)."', ".
922                 "note = '".mysql_real_escape_string($note)."', ".
923                 "latest = 'Y', ".
924                 "body = '".mysql_real_escape_string($body)."', ".
925                 "ispackagepage = '".mysql_real_escape_string($is_package_page)."'");
926
927             if ($pingdata =
928                 $this->GetPingParams($this->config["wikiping_server"], $tag, $user, $note))
929                 $this->WikiPing($pingdata);
930         }
931     }

```

GetPageTag() je ďalšou veľmi vyťažovanou metódou, ktorá vráti aktuálny názov stránky, ktorý je taktiež uložený v databáze.

```

666     function GetPageTag()
667     {
668         return $this->tag;
669     }

```

1.1.2.3 Databáza

Wakka.class má veľa metód na získavanie údajov z databázy ale v tejto sekcii si predstavíme implementovanú logiku, ktorá pomáha týmto funkciám a zaoberá množstvo funkcionality. Tým robí kód znovu použiteľným. Hovoríme o nasledujúcich troch metódach:

Query() je metóda na vykonávanie všetkých dotazov v databáze. Zapuzdruje najzákladnejšie volanie na databázu – funkciu **mysql_query()**. Metóda sa stará aj o ošetrenie výnimiek a taktiež výpis

do debugu ak je povolený v konfiguračnom súbore. Metóda Query() je teda metódou, pomocou ktorej poľahky môžete volať SQL dotazy kdekoľvek v kóde a nemusíte sa starať o rôznu funkcionality spojenú s prácou s databázou.

```
77 function Query($query, $dblink='')
78 {
79     // init - detect if called from object or externally
80     if ('' == $dblink)
81     {
82         $dblink = $this->dblink;
83         $object = TRUE;
84         $start = $this->GetMicroTime();
85     }
86     else
87     {
88         $object = FALSE;
89     }
90     if (!$result = mysql_query($query, $dblink))
91     {
92         ob_end_clean();
93         die("Query failed: ".$query." (" .mysql_error().")");
94     }
95     if ($object && $this->config['sql_debugging'])
96     {
97         $time = $this->GetMicroTime() - $start;
98         $this->queryLog[] = array(
99             "query" => $query,
100            "time" => $time);
101     }
102     return $result;
103 }
```

LoadAll() je metóda, ktorá vráti pole výsledkov z databázy. Vstupný parameter predstavuje dotaz na databázu v jazyku SQL. Následne je tento dotaz vykonaný pomocou vyššie spomínanej metódy **Query()**. Výsledky dotazu sú vrátené ako pole riadkov databázy.

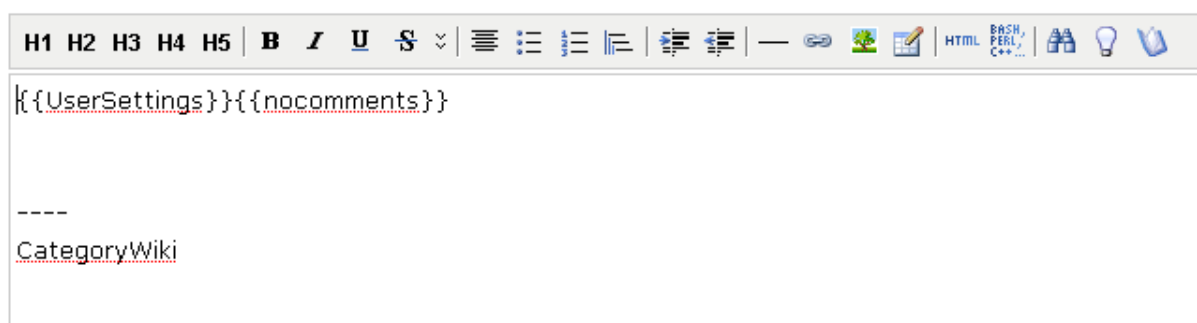
```
105 function LoadAll($query)
106 {
107     $data = array();
108     if ($r = $this->Query($query))
109     {
110         while ($row = mysql_fetch_assoc($r)) $data[] = $row;
111         mysql_free_result($r);
112     }
113     return $data;
114 }
```

LoadSingle() je metóda na získanie prvého vráteného riadku z databázy. Používa sa najmä v prípadoch, kde je potrebný len jeden výsledok z databázy a na ďalšie možnosti nie je potrebné prihliadať, napríklad pri načítavaní stránky z databázy, kde sa môže nachádzať viacero verzií stránky a pod. Využíva metódu **LoadAll()**, ktorá vráti pole výsledkov, následne zoberie prvý z nich a vráti ho ako výsledok volania.

```
104     function LoadSingle($query)
105     {
106         if ($data = $this->LoadAll($query))
107         {
108             return $data[0];
109         }
110     }
```

1.1.3 Akcie

Akcie sú špeciálnymi skriptami nachádzajúcimi sa v tele stránok a poskytujúcich rôznu funkcionality stránok. Na obrázku č. 3 je ukážka tela stránky ako ho vidí pri editovaní používateľ. Do tela stránky vložil dve akcie, ktoré poznáme podľa toho že sa nachádzajú uprostred dvojitéch zložených zátvoriek. Výsledkom takejto stránky po uložení do databázy a následnom zobrazení je, že sa najprv vykoná skript akcie **UserSettings**, ktorý zobrazí používateľove nastavenia a následne sa vykoná skript **nocomments**, ktorý zo stránky odoberie možnosť pridávať komentáre, pretože pre túto stránku si to používateľ neželá. Zdrojový kód skriptov akcií sa nachádza vždy v rovnomennom súbore, čiže napr. „Akcia.php“.



Obr. č. 3 - Ukážka použitia akcií v stránkach.

1.1.4 Udalosti

Udalosti^[5] sa týkajú najmä operácií nad stránkami, nezasahujú priamo do textu (tela) stránky ale slúžia na vyvolanie udalostí nad stránkami. Príkladom takejto udalosti, ktorá je najčastejšie vyvolávaná je udalosť „**edit**“. Po zavolaní stránky s touto udalosťou sa používateľovi otvorí okno pre editovanie stránky a zobrazí sa neformátovaná stránka ako editovateľný text.

Ďalšou typickou udalosťou je udalosť s menom „acl“. ACL je určená na správu prístupových práv k stránke. Tie sú uložené v databáze. Každá stránka môže mať štyri operácie, ktoré môžu byť pomocou ACL limitované:

Operácia	Opis
Write	Slúži na označenie práv používateľov meniť stránku a priamo zasahovať do jej obsahu.
Read	Slúži na označenie práv používateľov vidieť obsah stránky.
Comment	Slúži na označenie práv používateľov vložiť svoj komentár k stránke ak stránka má možnosť komentáre vkladať.
Set Page Owner	Umožňuje zmeniť majiteľa stránky a predať práva vlastníka stránky na iného používateľa.

Vyvolanie nejakej udalosti je možné prostredníctvom menu v päte každej stránky alebo manuálnym dopísaním do URL žiadosti. Základné štyri udalosti sú viditeľné pre všetkých registrovaných používateľov a administrátorov. Patria medzi ne už spomínané udalosti „edit“ a „ACL“ a ďalšie dve, slúžiace na zobrazenie referencií na stránku (udalosť „**referrers**“) a zobrazenie histórie stránky a posledných zmien (udalosť „**history**“). Kompletný zoznam udalostí a ich podrobný opis sa nachádza v dokumentácii k WikaWikki.

1.1.5 Šablóny

Nie je na škodu priblížiť aj túto časť wiki a tou sú šablóny. Vo wiki sú použité dve šablóny a to pre hlavičku a päť každej stránky.

Hlavička stránky obsahuje meno stránky a používateľské menu, ktoré sa generuje podľa toho či je používateľ prihlásený alebo nie. Odkazy v menu pre prihlásených a neprihlásených sa dajú konfigurovať v konfiguračnom súbore.

Päť stránky slúži na rôzne udalosti a operácie nad stránkou, ktorá je práve zobrazovaná. Okrem základných štyroch udalostí sa tu nachádza čas poslednej zmeny, odkaz na tzv. „**rss feed**“ sledujúci zmeny stránky, ďalej informácie o vlastníkovi stránky a možnosť vyhľadávať text v stránke. V pravom dolnom rohu sa tu taktiež vyskytujú informácie pre vývojárov, ako sú napríklad informácie o validnosti stránky a jej štýlov podľa noriem **W3C**, údaj o čase potrebnom na vygenerovanie stránky a pri zapnutom ladení^[7] SQL dotazov (cez konfiguračný súbor) aj údaje o všetkých SQL dotazoch, uskutočnených počas generovania stránky.

1.1.6 Konfigurácia

Konfigurácia bola už čiastočne spomínaná v časti **1.1.1** o `wikka.php`, kde sa jednalo hlavne o načítavanie a použitie konfigurácie. Nasledujúca kapitola má za úlohu priblížiť konfiguračný súbor, ktorý je voľne dostupný pre správcov systému a je priamo určený na konfiguráciu aplikácie.

Nasledujúca tabuľka uvádza zoznam najdôležitejších konfiguračných záznamov, ktoré sú pre používateľa systému k dispozícii a zlepšujú škálovateľnosť^[8] aplikácie.

Kľúč => hodnota	Vysvetlenie
'mysql_host' => 'luadist.org'	Adresa servera kde sa nachádza databáza wiki.
'root_page' => 'HomePage'	Názov stránky, ktorá je nastavená ako hlavná stránka a zobrazuje sa na začiatku alebo v prípade, že žiadna stránka nebola zadaná v parametri wakka v URL.
'action_path' => 'actions'	Cesta, kde sa nachádzajú skripty akcií wiki.
'handler_path' => 'handlers'	Cesta, kde sa nachádzajú skripty udalostí wiki.
'stylesheet' => 'wikka.css?d7f30'	Názov štýlu, ktorý sa použije ako základný štýl wiki.
'navigation_links' => 'PageIndex :: RecentChanges :: RecentlyCommented'	Odkazy v menu pre neprihlásených používateľov. Wiki pri ho pri načítavaní formátuje a vytvára z neho linky na existujúce stránky.
'logged_in_navigation_links' => 'PageIndex :: RecentChanges :: RecentlyCommented :: RecentlyPublished'	Odkazy v menu pre prihlásených používateľov. Wiki pri ho pri načítavaní formátuje a vytvára z neho linky na existujúce stránky.
'hide_comments' => '0'	Premenná indikujúca možnosti komentovať stránky používateľmi ak stránky komentovanie umožňujú.
'sql_debugging' => '0'	Premenná indikujúca možnosť zobrazit' všetky SQL dotazy použité pri generovaní wiki stránky. Účinné najmä pre vývojárov.
'admin_users' => 'KajoMarton, HakiBaki, RimmoN, Ma5, KubKub'	Označenie jednotlivých používateľov systému ako administrátorov. Používateľ samozrejme musí byť korektne zaregistrovaný a údaje o ňom musia existovať v databáze.
'default_write_acl' => '!*'	Predvolené nastavenie práv na zápis stránok.

	Nastaví obmedzenie, ktoré zakáže neprihláseným používateľom meniť obsah stránok, táto možnosť ostane len prihláseným a administrátorom.
'default_read_acl' => '*'	Predvolené nastavenie práv na čítanie stránok. Nastavenie povoľuje vidieť všetky vytvorené stránky všetkým používateľom, vrátane neregistrovaných
'default_comment_acl' => '+'	Predvolené nastavenie práv na pridávanie komentárov stránok. Nastaví obmedzenie, ktoré zakáže neprihláseným používateľom komentovať stránky, táto možnosť ostane len prihláseným a administrátorom.
'allow_user_registration' => '1'	Premenná, ktorá povoľuje neprihláseným používateľom registrovať sa.

1.2 Luadist

V predchádzajúcej kapitole sme sa pozreli na funkcionality, ktorú wiki poskytuje po svojom nainštalovaní. Samozrejme, pre náš projekt potrebujeme ďalšie, ktoré budú priamo pracovať s balíčkami a repozitármi našich používateľov.

1.2.1 Registrácia používateľa

Registrácia používateľa bola z časti ponechaná ale bolo pridané vytváranie používateľského repozitára, čiže adresára kde budú uložené jeho balíčky. Tieto zmeny boli prevedené do skriptu pre akciu **UserSettings**. Ak tento skript overí všetky vstupné údaje, ktoré zadal používateľ pri registrácii, vytvorí následne jeho používateľský adresár. Na tento úkon používa metódu **mkdir_r()**, ktorá rekurzívne vytvorí tento adresár.

```

131  if (!function_exists('mkdir_r')) {
132      function mkdir_r($dir)
133      {
134          if (strlen($dir) == 0)
135              return 0;
136          if (is_dir($dir))
137              return 1;
138          elseif (dirname($dir) == $dir)
139              return 1;
140          return (mkdir_r(dirname($dir)) and mkdir($dir, 0775));
141      }
142  }

```

Samotné volanie sa nachádza za ostatnými validáciami vstupných údajov a po úspešnom vložení používateľa do databázy. V prvom rade sa získa názov adresára pomocou „hash“ funkcie **md5()**, ktorá ako vstupný parameter zoberie používateľské meno a priradí k nemu náhodnú premennú. Vytvorením takéhoto „hash-u“ dosiahneme vyššiu bezpečnosť aby nebolo možné lokalizovať používateľský repozitár, ktorý je podľa špecifikácie privátny a tým pádom skrytý pre ostatných.

```
519     $path = "./repo/" . md5($this->GetUserName() . mt_rand());
520
521     if (!is_dir($path))
522     {
523         if (!mkdir_r($path))
524         {
525             echo "<strong>An error has been occurred during the action mkdir("
526                 . $path . ")!</strong>";
527         }
528     }
529     else
530     {
531         die("Couldn't create directory ".$path.
532             ", directory exists soon. Please contact system administrator.");
533     }
```

1.2.2 Generovanie manifestu

Manifest je generovaný 2 spôsobmi. Z databázy a pridaním záznamu na koniec existujúceho. Funkcie, ktoré implementujú túto funkcionality sú umiestnené v súbore `luadist.lib.php`.

```
41     function readDistArchive($dist) {
42         $unzip = unzip($dist, "tmp");
43         if($unzip <= 0){
44             return $dist . " is not a valid dist file.";
45         }
46         else {
47             $distName = getFileName($dist);
48             $distinfo_lines = file("tmp/" . $distName . "/dist.info", FILE_IGNORE_NEW_LINES | FILE_TEXT);
49             if(is_null($distinfo_lines))
50                 return false;
51             $toRet = array();
52             doPairs($distinfo_lines, $toRet);
53             $toRet["file_count"] = GetNumberOfFilesInArchive("tmp/" . $distName, true);
54             rmdir_recursive("tmp/" . $distName);
55             return $toRet;
56         }
57     }
```

Funkcia **readDistArchive()** najprv rozbalí daný dist súbor na danej ceste do tmp adresára, podadresára s menom súboru. O toto sa stará funkcia `unzip php pclzip.lib` knižnice. Následne z danej cesty si zistí meno samotného dist súboru, a podľa neho nájde v rozbalenom priečinku `dist.info` súbor a načíta ho celý do jedného reťazca. Následne je tento reťazec rozobratý funkciou `doPairs()`, ktorá vráti asociatívne pole hodnôt z `dist.info` súboru. Funkcia `doPairs()` prechádza reťazec pričom hľadá oddeľovače lua zápisu objektov – tabuliek.

```

181
182 function createManifestFromDb($page, $repository){
183     $sqlResultData = $page->LoadAllPackageInfosFromRepo($repository);
184     $manifest = fopen($repository . "/dist.manifest", 'w');
185     if ($manifest) {
186         foreach ($sqlResultData as $dist_info) {
187             if(is_array($dist_info)) {
188                 $dist_info["file_name"] = substr($dist_info["page_tag"], strpos($dist_info["page_tag"], "_") + 1);
189                 writeDistInfoToManifest($manifest, $dist_info);
190             }
191         }
192         fclose($manifest);
193         return true;
194     }
195     return false;
196 }
197 /**
198

```

```

163 function addToManifest($dist_info, $file_info){
164     //open the manifest file
165     $repository = $file_info["repo"];
166     $manifest = fopen($repository . "/dist.manifest", 'a');
167     if ($manifest) {
168         if(is_array($dist_info)){
169             $dist_info["file_name"] = $file_info["file_name"];
170             writeDistInfoToManifest($manifest, $dist_info);
171         }
172         else{
173             echo "error reading dist info of the ".$file_info["file_name"];
174         }
175
176         fclose($manifest);
177         return true;
178     }
179     return false;
180 }

```

Funkcia **addToManifest()** pridáva informáciu o dist balíčku do existujúceho manifestu. Funkcia **createManifestFromDb()** zmaže existujúci manifest súbor a vytvorí nový na základe údajov získaných z databázy.

Obe tieto funkcie používajú funkciu **writeDistInfoToManifest()**, ktorá vytvorí a zapíše formátovaný výstup do daného manifest súboru. Tento súbor otvorený na zapisovanie vo volajúcich funkciách.

```

150 function writeDistInfoToManifest($manifest, $distInfo) {
151     fwrite($manifest, $distInfo["name"] . " = {\n");
152     fwrite($manifest, "\t[" . $distInfo["version"] . "'] = {\n");
153     fwrite($manifest, "\t\t[" . $distInfo["arch"] . "'] = {\n");
154     fwrite($manifest, "\t\t\t[" . $distInfo["type"] . "'] = \t[[" . $distInfo["file_name"] . "]],\n");
155     fwrite($manifest, "\t\t},\n");
156     fwrite($manifest, "\t},\n");
157     fwrite($manifest, "}\n\n");
158 }

```

1.2.3 Wakka.class.php

Wakka.class ako miesto na lokalizáciu aplikačnej logiky na správu používateľov a stránok prešla taktiež viacerými zmenami, ktoré sme do nej implementovali. Pribudli hlavne metódy na prácu s balíčkami uloženými v databáze. Taktiež pribudli metódy na prácu s publikovanými balíčkami a napokon metódy pre kategórie balíčkov.

IsPackagePage() patrí medzi balíčkové funkcie a slúži na získanie informácie, či stránka, ktorú práve zobrazuje wiki je stránka s informáciami o balíčku. Takéto stránky majú v databáze nastavený stĺpec **ispackagepage** na 'Y'.

```
730     function IsPackagePage($tag)
731     {
732         $result = $this->LoadSingle("SELECT ispackagepage FROM ".$this->config["table_prefix"].
733             "pages WHERE tag='".$mysql_real_escape_string($tag)."'");
734         return $result['ispackagepage'] == "Y";
735     }
```

IsPagePublished() je metóda pre publikované balíčky a slúži pre zistenie informácie, či stránka, ktorá nesie informácie o balíčku bola používateľom publikovaná alebo nie. Táto informácia je dôležitá najmä pri zobrazovaní publikovaných balíčkov a pre vytvorenie menu pre publikovanie balíčka.

```
736     function IsPagePublished($tag)
737     {
738         $result = $this->LoadSingle("SELECT published FROM ".$this->config["table_prefix"].
739             "packages WHERE page_tag='".$mysql_real_escape_string($tag)."'");
740         return $result['published'];
741     }
```

LoadAllPackagePageOwners() je metóda, ktorá vráti všetkých používateľov, ktorý vlastní stránky s informáciami o balíčkoch. Potrebná je najmä pre filtrovanie údajov v rôznych zobrazeniach a prehľadoch o balíčkoch.

```

743     function LoadAllPackagePageOwners()
744     {
745         $query = "SELECT ".$this->config["table_prefix"]."packages.added AS added, ".
746             $this->config["table_prefix"]."packages.page_tag AS tag FROM ".
747             $this->config["table_prefix"]."pages LEFT JOIN ".
748             $this->config["table_prefix"]."packages ON ".
749             $this->config["table_prefix"]."pages.tag = ".
750             $this->config["table_prefix"]."packages.page_tag WHERE ".
751             $this->config["table_prefix"]."pages.ispackagepage = 'Y' AND ".
752             $this->config["table_prefix"]."pages.latest = 'Y' ".
753             "ORDER BY ".$this->config["table_prefix"]."packages.added";
754
755         $result = $this->LoadAll($query);
756         $users = array();
757         $i = 0;
758         foreach ($result as $res)
759         {
760             if ($this->HasAccess('read', $res['tag']))
761             {
762                 $users[] = $res['added'];
763             }
764         }
765
766         return array_unique($users);
767     }

```

IsCategoryPage() je metóda patriaca k metódam kategórií balíčkov. Slúži na získanie informácie o tom či zobrazovaná stránka zobrazuje všetky balíčky v danej kategórií. Metóda vytvorí spojenie dvoch tabuliek v databáze a porovnáva údaje v nich. Stránka má v sebe kategórie ak vlastník stránky je označený ako **is_category_admin**.

```

769     function IsCategoryPage($tag)
770     {
771         $result = $this->LoadSingle(
772             "SELECT ".$this->config["table_prefix"]."users.is_category_admin AS is_cat_admin FROM ".
773             $this->config["table_prefix"]."packages LEFT JOIN ".$this->config["table_prefix"]."users ON ".
774             $this->config["table_prefix"]."packages.added = ".$this->config["table_prefix"]."users.name WHERE ".
775             $this->config["table_prefix"]."packages.page_tag = '".mysql_real_escape_string($tag)."'
776         );
777
778         return $result['is_cat_admin'] == 'Y';
779     }

```

LoadAllPublishedPackages() slúži na správu všetkých balíčkov, ktoré boli publikované používateľmi. Metóda má dva nepovinné parametre, ktoré slúžia na filtráciu dát a to na zobrazenie publikovaných balíčkov len do určitého dátumu alebo zobrazenie publikovaných balíčkov len pre konkrétneho používateľa.

```

781 function LoadAllPublishedPackages($todate = null, $find_user = "")
782 {
783     if ($todate == null)
784     {
785         $query = "SELECT * FROM ".$this->config["table_prefix"].
786             "packages WHERE (published_by IS NOT NULL) AND (published IS NOT NULL) ";
787
788         if($find_user != "")
789         {
790             $query .= "AND (published_by LIKE '%" .
791                 mysql_real_escape_string($find_user)."%' )";
792         }
793
794         $query .= "ORDER BY published DESC";
795
796         return $this->LoadAll($query);
797     }
798     else
799     {
800         return $this->LoadAll(
801             "SELECT * FROM ".$this->config["table_prefix"].
802             "packages WHERE (published_by IS NOT NULL) AND (published IS NOT NULL) ".
803             "AND (DATEDIFF(published, '".$mysql_real_escape_string($todate).
804             "') >= 0) ORDER BY published");
805     }
806 }

```

LoadAllPublishers() patrí medzi používateľské metódy a je potrebná pri filtrovaní publikovaných balíčkov. Táto metóda vracia mená všetkých používateľov, ktorý publikovali nejaký svoj balíček.

```

807 function LoadAllPublishers()
808 {
809     return $this->LoadAll("SELECT published_by FROM ".$this->config["table_prefix"].
810         "packages WHERE (published IS NOT NULL) AND (published_by IS NOT NULL) ".
811         "GROUP BY published_by ORDER BY published_by");
812 }

```

GetAdminCategories() slúži na získanie údajov o všetkých kategóriách do ktorých sú balíčky triedené.

```

814 function GetAdminCategories()
815 {
816     return $this->LoadAll("SELECT * FROM ".$this->config["table_prefix"].
817         "users WHERE is_category_admin = 'Y' ORDER BY name");
818 }

```

IsPackageUnique() je potrebná pri publikovaní balíčka. Pred publikovaním totiž treba skontrolovať, či balíček, ktorý publikujeme náhodou v kategórii už neexistuje.

```

834     function IsPackageUnique($category, $package_name)
835     {
836         $result = $this->LoadSingle("SELECT COUNT(*) as count FROM ".
837             $this->config["table_prefix"]."packages WHERE page_tag = '".
838             mysql_real_escape_string($category."_".$package_name).
839             "' && added = '".mysql_real_escape_string($category)."'");
840
841         return ($result['count']) ? !($result['count'] > 0) : true;
842     }

```

LoadAllPackagePages() je v rebríčku najdôležitejších metód na najvyššom mieste. Využívaná je viacerými akciami, ktoré zobrazujú balíčky podľa rôznych klasifikácií, preto je potrebná jej určitá flexibilita, ktorú predstavujú tri nepovinné parametre. V prípade, že chceme získať z databázy len údaje o stránkach a nepotrebuje údaje o balíčkoch, ktoré akcie zobrazujú, tak nastavíme prvý parameter na hodnotu „false“. Sú akcie, ktoré ale potrebujú oboje údaje, vtedy vo volaní treba nastaviť prvý parameter na „true“. Ďalšie dva parametre slúžia na filtrovanie údajov. Druhý parameter slúži na filtrovanie výsledku podľa mena a tretí parameter indikuje či stačí aby meno autora iba obsahovalo zadaný reťazec alebo aby presne odpovedalo zadanému reťazcu.

```

855     function LoadAllPackagePages($include_packages_to_result = false, $find_user = "", $strict = false)
856     {
857         if ($include_packages_to_result)
858         {
859             $query = "SELECT * FROM ".$this->config["table_prefix"]."pages LEFT JOIN ".
860                 $this->config["table_prefix"]."packages ON ".
861                 $this->config["table_prefix"]."pages.tag = ".
862                 $this->config["table_prefix"]."packages.page_tag WHERE ".
863                 $this->config["table_prefix"]."pages.ispackagepage = 'Y' AND ".
864                 $this->config["table_prefix"]."pages.latest = 'Y' ";
865
866             if ($find_user != "")
867             {
868                 if (!$strict)
869                 {
870                     $query .= "AND ".$this->config["table_prefix"]."packages.added LIKE '%"
871                         .mysql_real_escape_string($find_user)."%' ";
872                 }
873                 else
874                 {
875                     $query .= "AND ".$this->config["table_prefix"]."packages.added = '"
876                         .mysql_real_escape_string($find_user)."' ";
877                 }
878             }
879             $query .= "ORDER BY ".$this->config["table_prefix"]."packages.page_tag";
880
881             return $this->LoadAll($query);
882         }
883         else
884         {
885             return $this->LoadAll("SELECT * FROM ".$this->config["table_prefix"]."
886                 "pages WHERE latest = 'Y' AND ispackagepage = 'Y' ORDER BY tag");
887         }
888     }

```

LoadAllAdminCategoryPages() je metóda, ktorá z databázy získa všetky stránky, ktoré patria do danej kategórie balíčkov.


```

890     function LoadAllAdminCategoryPages($category)
891     {
892         return $this->LoadAll("SELECT * FROM ".$this->config["table_prefix"].
893             "packages WHERE published_by IS NOT NULL AND added = '"
894             .mysql_real_escape_string($category)."' ORDER BY name");
895     }

```

LoadAllPackageInfosFromRepo() slúži na získavanie údajov o balíčkoch nachádzajúcich sa v konkrétnom repozitári, ktorý je zadaný ako vstupný parameter.

```

897     function LoadAllPackageInfosFromRepo($repo)
898     {
899         return $this->LoadAll("select * from ".$this->config["table_prefix"].
900             "packages where repo = '" .mysql_real_escape_string($repo)."'");
901     }

```

GetUserRepository() je ďalšia potrebná metóda, ktorá vráti adresu repozitára pre konkrétneho používateľa. Táto metóda je používaná pri vytváraní a regenerovaní manifestov. Používateľské repozitáre sú vytvárané pomocou „hash-u“ a náhodného čísla, pre to nie je možné získavať názov repozitára kedykoľvek počas behu len znovu vypočítaním cesty.

```

903     function GetUserRepository($username)
904     {
905         $repo = $this->LoadSingle("select repository from ".$this->config["table_prefix"].
906             "users where name = '" .mysql_real_escape_string($username)."'");
907         return $repo['repository'];
908     }

```

1.2.4 Akcie

Úlohou tejto sekcie je oboznámiť čitateľa s vytvorenými akciami, ktoré je možné vkladať do wiki stránok. Počas tvorby projektu bolo vytvorených 9 nových akcií, ktoré tvoria hlavnú časť projektu a jeho funkcionality, ktorú sme implementovali. Najviac práce počas implementácií bolo strávených práve pre tieto akcie.

1.2.4.1 DistUpload

Prvou akciou, ktorá bude priblížená bližšie bude akcia **DistUpload**. Je tvorená univerzálnym php skriptom. Univerzálny preto, lebo je používaný pre dve navonok rozdielne funkcie a to pre nahrávanie balíčkov na server a ich publikovanie používateľmi pre verejnosť. Tieto dve funkcie majú rovnaký základ práve v tomto skripte.

Nahrávanie balíčkov používa štandardné funkcie jazyka php ako **move_uploaded_file()** a definovanú premennú **_FILES**. Po úspešnom nahraní príde na rad proces začleňovania balíčka do systému. Prečítajú sa údaje o balíčku uložené v ňom a zapíšu sa do databázy. Nasleduje generovanie manifestu v používateľom repozitári kde bol balíček nahraný. Tento proces je podrobne opísaný

v kapitole 1.2.2. Nakoniec skript vygeneruje stránku pre balíček. Použije šablónu uloženú v adresári templates a vloží na miesta na to určené údaje o balíčku, ktoré získal. Takto vygenerovanú stránku uloží do databázy a nastaví prívátne práva na čítanie, zápis a komentovanie stránky. Proces registrácie zobrazuje nasledujúci zdrojový kód. Vidíme, že každá časť procesu je ošetrovaná proti výnimkám. Metódy **writePackageInfoIntoDb()** a **makeManifest()** boli už spomínané v predchádzajúcej kapitole 1.2.2 metódy generujúce stránku sa nachádzajú nižšie v časti o publikovaní.

```
38 function RegisterPackage($page, $dist_info, $file_info, $is_publishing = false)
39 {
40     if (writePackageInfoToDb($page, $dist_info, $file_info, $is_publishing))
41     {
42         echo ("Info about uploaded package was saved to database successfully.<br />");
43     }
44     else
45     {
46         echo ("<strong>Error: Saving to database!</strong><br />");
47         return false;
48     }
49
50     if (makeManifest($dist_info, $file_info))
51     {
52         echo ("Manifest generated successfully.<br />");
53     }
54     else
55     {
56         echo ("<strong>Error: Make manifest was failed!</strong><br />");
57         return false;
58     }
59
60     if (GeneratePackagePage($page, $dist_info, $file_info, $is_publishing))
61     {
62         echo ("New page for package created successfully.<br />");
63     }
64     else
65     {
66         echo ("<strong>Error: Cannot generate page for package!</strong><br />");
67         return false;
68     }
69
70     return true;
71 }
```

Pri publikovaní balíčka sa využívajú tieto metódy tiež, ale predchádza im iný proces. Publikovanie balíčka totiž nie je nič iné len opätovné nahratie balíčka na server ale do iného adresára, reprezentujúceho kategóriu balíčkov. Preto je možné tento kód znova využiť pri tomto procese. Publikovaniu predchádza proces overenia či balíček, ktorý ideme publikovať do určitej kategórie tam už neexistuje. Ak touto kontrolou prejde, nasleduje presne rovnaký proces ako pri nahrávaní balíčka. Prečítajú sa údaje z neho, uložia sa do databázy, pregeneruje sa manifest v adresári kategórie a vytvorí sa stránka. Tejto stránke sa už ale nenastavujú prívátne prístupové práva. Preto je v parametroch metódy **GeneratePackagePage()** údaj na identifikáciu procesu publikovania, ktorý mení toto správanie ako je vidieť v zdrojovom kóde nižšie. Ďalším rozdielom je, že pri publikovaní sa nepoužíva

aktuálny používateľ ale používateľ reprezentujúci kategóriu. Zabráni sa tým vytváraniu duplikátnych údajov o balíčkoch v databáze.

```
74 function GeneratePackagePage($page, $dist_info, $file_info, $is_publishing = false)
75 {
76     //get template
77     $template = GetTemplate("new_dist_page_template");
78
79     if ($template) {
80         //get body of the new page from template
81         $body = GetBody($page, $template, $dist_info, $file_info);
82
83         if ($body) {
84             //create page in db
85             $page->SavePage($file_info["page_name"], $body, "Autogenerated page.", $file_info["user"], 'Y');
86             //remove all permissions, only owner and admins could read/write/comment generated pages by default
87
88             if ($is_publishing)
89             {
90                 $page->SaveACL($file_info["page_name"], 'read', '!*');
91                 $page->SaveACL($file_info["page_name"], 'comment', '!*');
92             }
93             else
94             {
95                 $page->SaveACL($file_info["page_name"], 'read', '!*');
96                 $page->SaveACL($file_info["page_name"], 'comment', '!*');
97             }
98             $page->SaveACL($file_info["page_name"], 'write', '!*');
99
100             return true;
101         } else {
102             echo "Error: An error occured during creating package page content!<br />";
103             return false;
104         }
105     } else {
106         echo "Error: Cannot retrieve template for new package page!<br />";
107         return false;
108     }
109 }
```

1.2.4.2 DistIndex

DistIndex je akciou, ktorá má za úlohu zobrazovať balíčky viditeľné pre používateľa. Patria sem balíčky publikované, ním vytvorené alebo také, ktoré mu sprístupnili iní používatelia cez ACL.

DistIndex je taktiež php skript, ktorý na úvod vyčíta údaje z http metódy POST, kde sa môžu nachádzať informácie o filtrovaní zobrazovaných balíčkov. Používateľ môže v jeho menu voliť meno používateľa, ktorého balíčky chce vidieť.

Po úvodnom načítaní DistIndex zoberie vyfiltrované stránky a pokúsi sa balíčky vhodne zoskupiť podľa názvov. Môže sa stať, že jeden používateľ môže mať viacero balíčkov v rôznych verziách ale s rovnakým názvom. Vtedy sa tieto balíčky zoskupia a neskôr budú zobrazené ako jeden celok pre lepšiu orientáciu v nich. Zdrojový kód je znázornený nižšie. Metóda **GetGroupedPackages()** sa okrem zoskupovania stará aj o kontrolu či používateľ má práva stránky iných používateľov čítať. V prípade že nie, stránky sa späť do cyklu zobrazovania DistIndexu nevrátia. Taktiež metóda vyčlení zo skupiny balíčkov, ktorý používateľ označil na zmazanie. Túto možnosť DistIndex používateľom taktiež poskytuje.

```

248 function GetGroupedPackages($page, $pages)
249 {
250     if ($pages)
251     {
252         $result = array();
253         //search in array
254         foreach ($pages as $packagepage)
255         {
256             //only if user has read access
257             if (!$page->HasAccess('read', $packagepage['page_tag']))
258             {
259                 continue;
260             }
261             //calculate del_package string
262             $del_package = "del_package_".str_replace(".", "_", $packagepage['page_tag']);
263             //and check if is not deleted in current web request
264             if (!(isset($_GET[$del_package])))
265             {
266                 //compose a unique key for packages
267                 $key = $packagepage['added']."_".$packagepage['name'];
268                 //check if exists items for this package
269                 if ($result[$key])
270                 {
271                     //add item to existing array
272                     $packages[count($packages)] = $packagepage;
273                     $result[$key] = $packages;
274                 }
275                 else
276                 {
277                     //create a new array and set the unique entry
278                     $packages = array();
279                     $packages[0] = $packagepage;
280                     $result[$key] = $packages;
281                 }
282             }
283             else
284             {
285                 //a distindex needs to remember that one package is deleted now.
286                 $result['__package_to_delete__'] = $packagepage;
287             }
288         }
289     }
290     return $result;
291 }

```

Po zoskupení balíčkov sa začne proces zobrazovania. Každý používateľ, ktorý sa v zobrazovaných balíčkoch nachádza má svoju vlastnú tabuľku a do nej sa vkladajú údaje o jeho balíčkoch. Pre každý balíček sa kontroluje, či existujú práva pre používateľa zobrazujúceho DistIndex aby mohol balíček zmazať. To smú len administrátori a vlastníci balíčkov.

Mazanie balíčkov nastáva počas procesu zobrazovania. Ak používateľ predtým zvolil mazanie niektorého z balíčkov, je meno tohto balíčku zaznamenané a ak počas zobrazovania naň skript narazí, zmaže ho a nezobrazí. Mazanie reprezentuje kód nižšie.

```

209 //delete page in DB
210 $this->Query("delete from ".$this->config["table_prefix"]."pages where tag = '"
211     .mysql_real_escape_string($page['page_tag'])."'");
212 //get package repo
213 $repo = $this->LoadSingle("select repo from ".$this->config["table_prefix"].
214     "packages where page_tag = '" .mysql_real_escape_string($page['page_tag'])."'");
215 $repo = $repo['repo'];
216 //delete package in DB
217 $this->Query("delete from ".$this->config["table_prefix"]."packages where page_tag = '"
218     .mysql_real_escape_string($page['page_tag'])."'");
219
220 //delete childs if exists, applies to all admin category package pages
221 if ($page['published_by'])
222 {
223     $result = $this->Query("UPDATE ".$this->config["table_prefix"].
224         "packages SET published = NULL WHERE name = '"
225         .mysql_real_escape_string($page['name'])."' AND version = '"
226         .mysql_real_escape_string($page['version'])."' AND added = '"
227         .mysql_real_escape_string($page['published_by'])."' AND arch = '"
228         .mysql_real_escape_string($page['arch'])."'");
229 }
230
231 //get the name of the file (it has form f.e. KubKub_config-1.0.0-universal-all.dist, we dont want 'KubKub_'
232 unlink($repo ."/". substr($page['page_tag'], strpos($page['page_tag'], "_") + 1));
233 $repo = $this->GetUserRepository($current_username);
234 //create manifest in rpeo
235 createManifestFromDb($this, $repo);
236 //notify user
237 $index_output .= "Package and page ".$page['page_tag']." deleted. <br />";

```

1.2.4.3 RecentlyPublished

RecentlyPublished je posledná z dôležitých akcií, pretože zobrazuje pre používateľov všetky publikované balíčky. Prístup k nej majú aj neregistrovaní používatelia, taktiež sú tu možnosti filtrovať balíčky podľa dátumu publikovania alebo používateľského mena vlastníka balíčku.

1.2.4.4 CreateNewCategory

Akcia **CreateNewCategory** je prístupná len pre administrátorov a slúži pre vytváranie nových kategórií pre balíčky. Z časti využíva funkcionality existujúcej akcie **UserSettings**. Kategória je vlastne používateľ s administrátorskými právami, ktorý je ešte na viac označený v databáze ako **category_admin**. Každý takýto používateľ/kategória musí mať svoj repozitár a email. Preto pri vytváraní kategórie sa vlastne zobrazí formulár na tvorbu nového používateľa. Doňho sú vpísané meno, heslo a emailová adresa. Pri ukladaní do databázy sa taktiež vytvorí používateľský repozitár kategórie, pri tomto procese sa ale nepoužíva „hash“ funkcia, názov adresára odpovedá názvu kategórie. V tomto prípade totiž nepotrebuje skrývať adresár kategórie, obsah bude verejný. Posledným krokom je vygenerovanie stránky pre kategóriu a vloženie do tejto stránky akciu **ShowAdminCategories**, ktorá na tejto stránke vždy zobrazí všetky balíčky, ktoré budú do kategórie publikované.

```

125 default: //valid input, create user
126 {
127     $this->Query("INSERT INTO ".$this->config['table_prefix']."users SET ".
128         "signuptime = now(), ".
129         "name = '".mysql_real_escape_string($categoryname)."', ".
130         "email = '".mysql_real_escape_string(strtolower($email))."', ".
131         "repository = '".mysql_real_escape_string($path)."', ".
132         "password = md5('".mysql_real_escape_string($password)."', ), ".
133         "is_category_admin = 'Y'");
134
135     $path = "./repo/".$categoryname;
136
137     if (!is_dir($path))
138     {
139         if (!mkdir_r($path))
140         {
141             echo "<strong>Upload failed -> An error has been occurred during ".
142                 "the action mkdir(" . $path . ")!</strong>";
143         }
144     }
145
146     if (!GenerateAdminCategoryPage($this, "new_admin_category_page_template",
147         $categoryname, $email, $path))
148     {
149         $error = "Cannot generate page for admin category!";
150     }
151
152     echo "<p>Category created successfully.</p>";
153     echo "<p>Repository: <a href='".$repo.">".$this->htmlspecialchars_ent(substr($path,2)).
154         "</a></p>";
155     echo "<p>Page: ".$this->Format("[[".$categoryname."]]")."</p>";
156 }

```

1.2.4.5 Ostatné

ShowAdminCategories – akcia na zobrazenie všetkých existujúcich kategórií a adresy stránok, kde je možné prezrieť si ich obsah. Funguje na jednoduchom princípe. Zoberie z databázy všetky kategórie a zobrazí odkazy na ich stránky.

```

2     echo "<h3>Admin categories:</h3>";
3
4     if ($categories = $this->GetAdminCategories())
5     {
6         echo "<p><ul>";
7
8         foreach ($categories as $category)
9         {
10            echo "<li><strong>".$this->Format("[[".$category['name']."]]")."</strong></li>";
11        }
12        echo "</ul></p>";
13    }
14    else
15    {
16        echo "<p>No admin categories found. Try to create one.</p>";
17    }

```

AdminCategoryIndex – akcia zobrazujúca všetky balíčky danej kategórie.

Logout – akcia potrebná po zmene spôsobu odhlasovania používateľov. Cieľom vytvorenia tejto akcie bolo oddeliť odhlasovanie používateľov zo systému od stránky používateľských nastavení ako to bolo pôvodne.

DistUploadAuto – akcia vytvorená ako zjednodušená verzia akcie DistUpload. Bola z nej vynechaná funkcionálna na publikovanie balíčkov, pretože táto akcia slúži ako stránka pre externú aplikáciu na automatické nahrávanie balíčkov zo strany klienta. Taktiež nepotrebuje viacero validačných funkcií a potvrdzované výpisy o procese, pretože o validáciu sa stará už samotná externá aplikácia.

MailComments – akcia, ktorá po pridaní do stránky umožní posielanie informácií autorovi stránky prostredníctvom elektronickej pošty o nových komentároch pridaných do stránky inými používateľmi.

```
9      $this->query(  
10         'UPDATE '.$this->config['table_prefix'].  
11         "pages SET mail_comments = 1 WHERE tag = '"  
12         .mysql_real_escape_string($this->GetPageTag()).  
13         "'");
```

1.2.5 Šablóny

Úlohou tejto sekcie je oboznámiť s vytvorenými šablónami, ktoré boli pridané do wiki. Patria sem dve šablóny, ktoré sa používajú pre generovanie stránok. Jedná sa o generovanie stránok pri nahrávaní balíčka a vytváraní kategórie pre balíčky.

Pri nahrávaní balíčka sa zistia údaje o ňom a potom sa vytvorí stránka, kde sa tieto údaje zobrazia. Zoberie sa šablóna, ktorá sa stane telom budúcej stránky. Toto telo musí byť formátované ako wiki text. Pravidlá formátovania sa nachádzajú v dokumentácii k WikkaWiki. Nasleduje ukážka šablóny pre nový balíček.

```
1  =====%name%=====
2  ----
3  Uploaded:   **%date%** by: **%user%**
4  File size:  **%size% Kb**
5  Stored in:  ***"<a href="%repo%">%repo%</a>"***
6  Version:    **%version%**
7  Arch:      **%arch%**
8  Type:      **%type%**
9  Number of files in package: **%filesCount%**
10
11  **Description:**
12  %description%
13  ----
```

V šablóne sú použité rôzne formátovacie prvky wiki ako napríklad štyri pomlčky, ktoré značia vodorovnú čiaru (v html je to tag <hr>) alebo napríklad text oddelený dvojicami hviezdíčiek bude formátovaný ako tučné písmo (v html je to tag). V šablóne sa nachádzajú taktiež rôzne tzv. premenné, ktoré sú ohraničené znakom %. Tieto premenné budú potom nahradené údajmi z balíčka.

Pri generovaní stránky pre novú kategóriu sa využíva druhá šablóna. Jej obsah je strohejší ako v predchádzajúcom prípade, pretože nie je potrebné na týchto stránkach zobrazovať množstvo informácií. Na poslednom riadku šablóny vidíme, že v tele stránky sa bude nachádzať akcia **admincategoryindex**, ktorá pri každom zobrazení stránky pridá do stránky zoznam všetkých balíčkov v kategórií.

```
1 =====%user%=====
2
3 Admin: **%user%**
4 Email: **%email%**
5 Repo: ****"<a href="%repo%">repo%</a>"****
6
7 ----
8
9 {{admincategoryindex}}
```

Ako generovanie prebieha na strane servera nám znázorňuje nasledujúca časť zdrojového kódu. Vidíme ako postupne volané metódy **GetTemplate()** a **GetBody()**, ktoré získajú najprv šablónu zo súboru a potom do šablóny vložia údaje.

```
74 function GeneratePackagePage($page, $dist_info, $file_info, $is_publishing = false)
75 {
76     //get template
77     $template = GetTemplate("new_dist_page_template");
78
79     if ($template) {
80         //get body of the new page from template
81         $body = GetBody($page, $template, $dist_info, $file_info);
82
83         if ($body) {
84             //create page in db
85             $page->SavePage($file_info["page_name"], $body, "Autogenerated page.", $file_info["user"], 'Y');
```

Metóda **GetTemplate()** hľadá danú šablónu v adresári pre šablóny a následne šablónu načíta zo súboru ako text.

Po nej metóda **GetBody()** tento text zoberie a začne mapovať na jednotlivé premenné v šablóne údaje o balíčku. Tie metóda získa v dvoch poliach ako parametre. Polia sú preto dve lebo pri načítavaní informácií o balíčku sa získali údaje o zip archíve v ktorom je balíček zabalený a pri jeho rozbalení sa získali ďalšie údaje zo súboru dist.info priloženého pri balíčku.


```

113 function GetBody($page, $template, $dist_info, $file_info)
114 {
115     $template = str_replace("%name%", $dist_info["name"], $template);
116     $template = str_replace("%user%", $file_info["user"], $template);
117     $template = str_replace("%date%", $file_info["upload_time"], $template);
118     $template = str_replace("%size%", $file_info["file_size"], $template);
119     $template = str_replace("%repo%", $file_info["path"], $template);
120     $template = str_replace("%version%", $dist_info["version"], $template);
121     $template = str_replace("%arch%", $dist_info["arch"], $template);
122     $template = str_replace("%type%", $dist_info["type"], $template);
123     $template = str_replace("%filesCount%", $dist_info["file_count"], $template);
124     $template = str_replace("%description%", $dist_info["full"], $template);
125
126     return $template;
127 }
128
129 //Rimmon: give a content of template
130 function GetTemplate($template_name)
131 {
132     $template_dir = "./templates";
133     $template = file_get_contents($template_dir . "/" . $template_name .
134     ".disttmpl");
135     return $template;
136 }

```

1.2.6 Hlasovanie

Ďalšou implementovanou časťou, ktorú sme vytvorili je funkcionálna hlasovania používateľov. Hlasovaním pridávajú alebo odoberajú body balíčkom. Informácie o hlasovaní sa nachádzajú v päta každej balíčkovkej stránky. Každý používateľ môže hlasovať za jeden balíček len raz, preto bola do päty stránky pridaná kontrola stavu hlasovania pre používateľa. Autenticitu zabezpečuje IP adresa používateľa, ktorá je zaznamenávaná pre každé hlasovanie. Ak používateľ už hlasoval, v päte sú zobrazené len údaje o počte bodov, ak ešte nehlasoval, zobrazia sa mu dve ruky, so vztýčeným palcom resp. palcom ukazujúcim nadol. Po kliknutí na jeden z nich sa stránke pripočítajú resp. odčítajú body a obnoví sa aktuálny počet bodov.

Hlasovanie je implementované pomocou technológie AJAX aby sa po kliknutí nevyvolalo klasické odoslanie formulára a stránka sa celá nanovo nevygenerovala a znova nenačítala u klienta. Aby AJAX fungoval, museli sme do stránok kde sa hlasovania nachádzajú pridať kód v jazyku javascript, ktorý sa stará o asynchrónnu komunikáciu so serverom.

Keďže na túto komunikáciu potrebujeme v prvom rade podporu XML http žiadosti, vytvorili sme metódu, ktorá sa stará o vytvorenie takéhoto objektu. Metóda ošetruje aj rôzne verzie najpoužívanejších prehliadačov.

```

48  function GetXmlHttpRequestObject()
49  {
50      var xmlHttpRequest=null;
51      try
52      {
53          // Firefox, Opera 8.0+, Safari
54          xmlHttpRequest=new XMLHttpRequest();
55      }
56      catch (e)
57      {
58          //Internet Explorer
59          try
60          {
61              xmlHttpRequest=new ActiveXObject("Msxml2.XMLHTTP");
62          }
63          catch (e)
64          {
65              xmlHttpRequest=new ActiveXObject("Microsoft.XMLHTTP");
66          }
67      }
68      return xmlHttpRequest;
69  }

```

Keď bola základná funkcionálna pripravená bola implementovaná funkcia, ktorá je volaná po kliknutí na odkaz. V parametroch tejto funkcie sa nachádzajú údaje potrebné pre server keď bude poslaná XML http žiadosť naň. Funkcia najprv vytvorí XML http objekt pomocou metódy **GetXmlHttpRequestObject()**.

Následne poskladá adresu stránky, ktorá bude asynchrónne volaná prostredníctvom XML http žiadosti. Do parametrov stránky vloží údaje o používateľovi, názov aktuálnej stránky, IP používateľa a akcia hlasovania, ktorá závisí od toho či ide o pozitívny alebo negatívny hlas. Akcia môže byť taktiež aj typu „reset“, ktorú môže vyvolať len administrátor a v tomto prípade sa jedná o vymazanie údajov o ohlasovaní pre aktuálnu stránku.

Nasleduje proces, v ktorom funkcia zaregistruje metódu, ktorá bude čakať na odpoveď zo servera. Metóda **stateChanged()** je bližšie opísaná nižšie. Registrácia je nutná kvôli tomu, že volanie na server je asynchrónne, čiže nečaká na odpoveď ale poverí inú metódu čakaním na túto odpoveď.

Po registrácii sa odošle XML http žiadosť. Ako metódu http používa GET.

```

6 //xml representation
7 var xmlHttp;
8
9 //function called from html
10 function vote(user, tag, action, vote, ip)
11 {
12     xmlHttp=GetXmlHttpRequestObject();
13     if (xmlHttp==null)
14     {
15         alert ("Browser does not support HTTP Request");
16         return;
17     }
18
19     //prepare call to server
20     var url="ajax/vote.php";
21     url=url+"?action=" + action;
22     url=url+"&user=" + user;
23     url=url+"&tag=" + tag;
24     url=url+"&vote=" + vote;
25     url=url+"&ip=" + ip;
26     url=url+"&sid=" + Math.random();
27
28     //register asynchronous call back event
29     xmlHttp.onreadystatechange = stateChanged;
30
31     //call server
32     xmlHttp.open("GET", url, true);
33     xmlHttp.send(null);
34 }

```

Metóda **stateChanged()** sa stará o odpoveď zo servera. Ak nejakú odpoveď dostane, vyberie z html **DOM**-u element s identifikačným menom „votes“ a vpiše doňho text, ktorý mu server vrátil.

```

36 //callback function
37 function stateChanged()
38 {
39     //check if state complete
40     if (xmlHttp.readyState == 4 || xmlHttp.readyState == "complete")
41     {
42         //modify document
43         document.getElementById("votes").innerHTML= xmlHttp.responseText;
44     }
45 }

```

Funkcionalitu, ktorá sa vykonáva na serveri reprezentuje skript „**vote.php**“. Ten sa stará o vykonanie príslušných úprav v databáze na základe akcie, ktorú používateľ pri hlasovaní vykonal. Ak sa jedná o reset, zmaže všetky hlasovania pre stránku. V prípade hlasovania vloží nový údaj do tabuľky hlasovaní. Nakoniec získa aktuálny stav hlasovania po aplikovaní nových zmien a tento údaj vráti naspäť klientovi.

```

26  if ($action == 'reset')
27  (
28      $wakka->Query(
29          "DELETE FROM ".$prefix."votes WHERE page_tag = '"
30          .mysql_real_escape_string($tag)
31          ."'";
32      )
33  else
34  (
35      //SQL statement for inserting votes into DB
36      $wakka->Query(
37          "INSERT INTO ".$prefix."votes (page_tag, user, rating, ipaddress) VALUES ('"
38          .mysql_real_escape_string($tag)."', '"
39          .mysql_real_escape_string($user)."', '"
40          .mysql_real_escape_string($vote)."', '"
41          .mysql_real_escape_string($ip)
42          ."' )";
43      )
44
45      //get current state of voting
46      $result = $wakka->LoadSingle(
47          "SELECT SUM(rating) AS count FROM ".$prefix."votes WHERE page_tag = '"
48          .mysql_real_escape_string($tag)
49          ."'";
50
51      //check count
52      if (!$result['count'])
53      {
54          $result['count'] = '0';
55      }
56
57      //display count in content
58      echo '<span id="response"> Votes: '.$result['count'].'</span>';

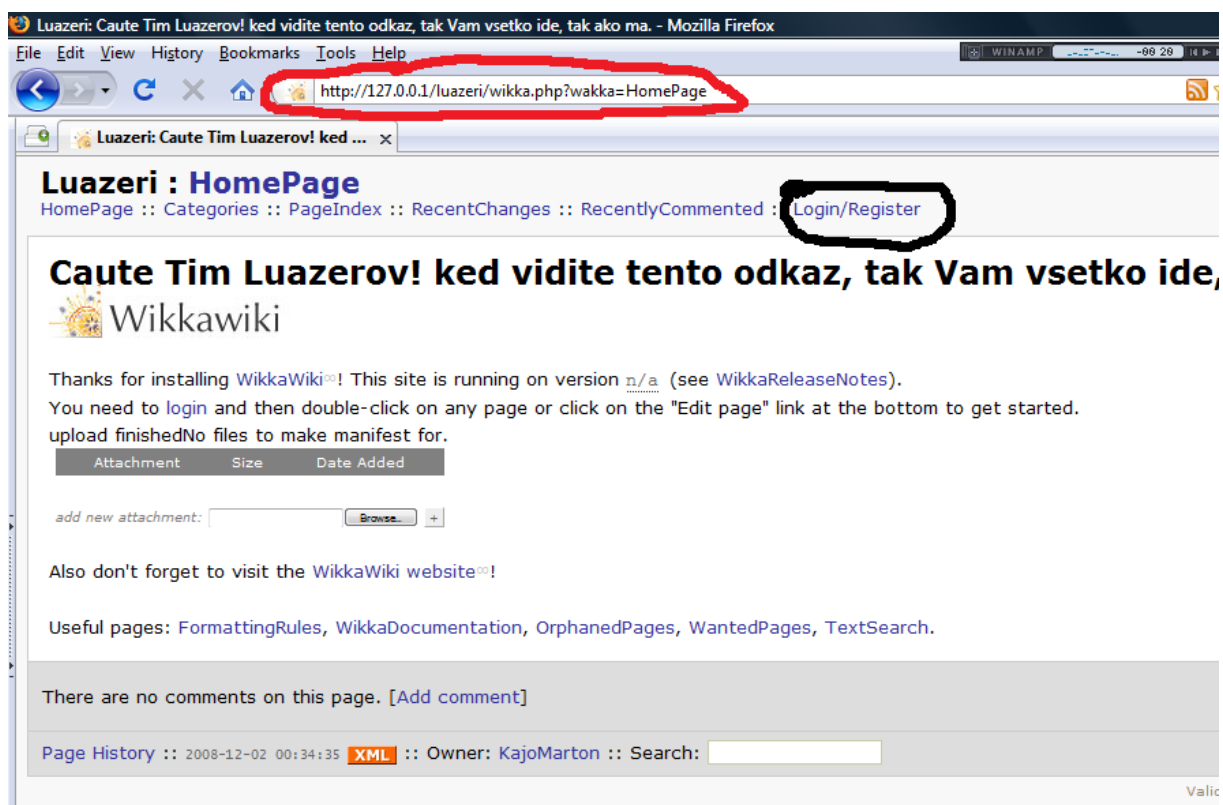
```

Príloha B – Používateľská príručka

1 Používateľská príručka

1.1 Prihlásenie a Registrácia užívateľa

Každému návštevníkovi nášho webového repozitára sa najprv zobrazí úvodná stránka, pomocou ktorej môže pristupovať k existujúcim stránkam, prihlásiť alebo zaregistrovať sa. Táto možnosť sa nachádza v hornej časti ako súčasť menu. Pre prihlásenie alebo registráciu stačí kliknúť na odkaz „Login/Register“.



Obr. č. 1 - Úvodná stránka.

Prihlasovanie a registrácia prebiehajú prostredníctvom jedného spoločného formulára. Ten sa skladá z troch častí:

- Prihlásenie pre existujúcich užívateľov
- Registrácia nového užívateľa
- Obnovenie zabudnutého hesla

Pre prihlásenie užívateľovi stačí vyplniť prvé dve textové polia, do ktorých užívateľ zadá svoje prihlasovacie meno a heslo. Ak sa registruje nový užívateľ, vyplní aj ďalšie dve polia, do ktorých potvrdí svoje heslo a zadá svoju emailovú adresu. Registrácia prebehne po stlačení tlačidla „Register“.

V prípade nekorektných údajov systém oznámi užívateľovi akým podmienkam tieto údaje nevyhovujú a požiada užívateľa o ich opravu.

Luazeri : UserSettings

[HomePage](#) :: [Categories](#) :: [PageIndex](#) :: [RecentChanges](#) :: [RecentlyCommented](#) :: [Login/Register](#)

Login/Register

If you're already a registered user, log in here!



Your WikiName:

Password (5+ chars):

Fields you only need to fill in when you're logging in for the first time (and thus signing up as a new user on this site).

Confirm password:

Your email address:

Forgot your password?

If you need a password reminder, click [here](#).

You can login here using your password reminder.

Your WikiName:

Password reminder:

Obr. č. 2 - Formulár pre registráciu a prihlasovanie užívateľov.

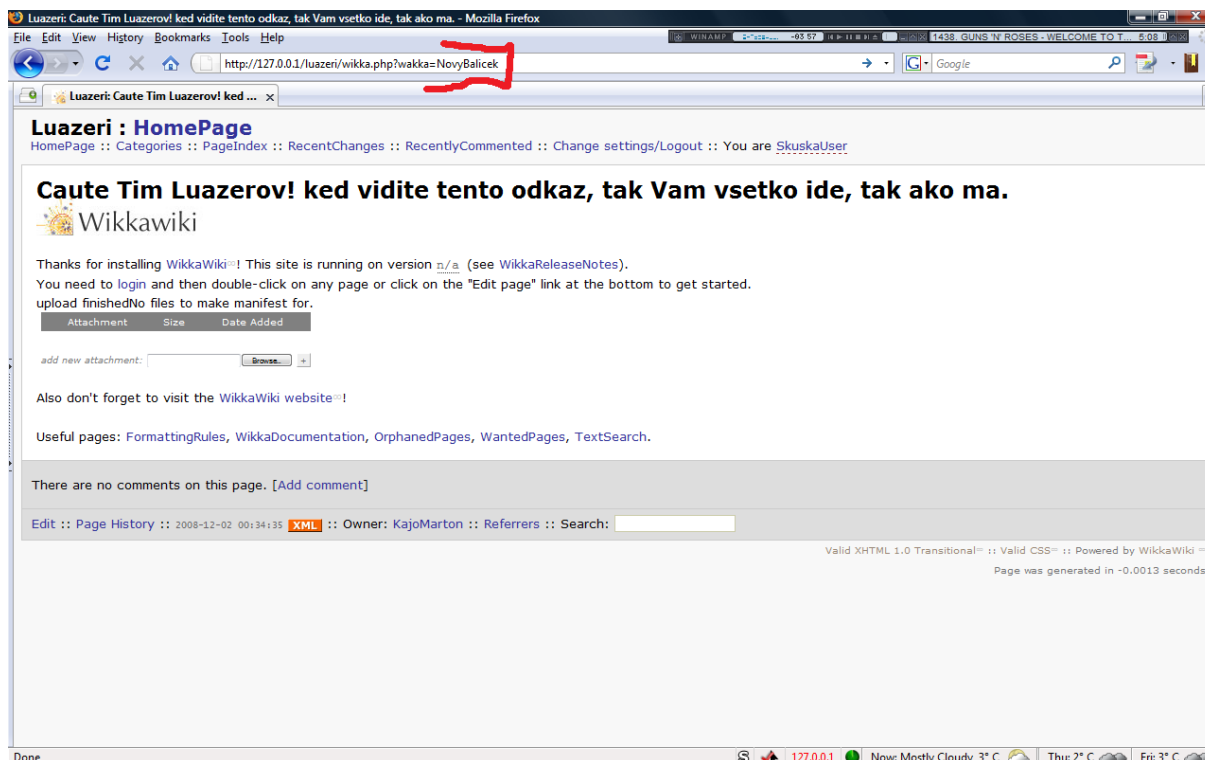
Používateľovi počas registrácie systém automaticky vytvoril jeho adresár, do ktorého môže vkladať svoje balíčky. Po úspešnej registrácii je jeho konto okamžite pripravené na používanie. Oznámenie, že je prihlásený sa zobrazuje v hlavnom menu v hornej časti stránky.



Obr. č. 3 - Koniec registrácie užívateľa a jeho prihlásenie v systéme.

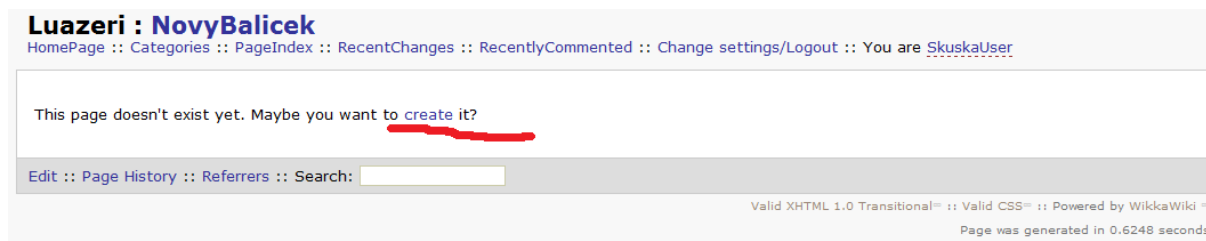
1.2 Vytvorenie novej stránky

Vytvárať nové stránky môžu všetci prihlásení užívatelia. Jej vytvorenie pozostáva z troch krokov. Prvým je vloženie názvu novej stránky do adresy wiki. Systém skontroluje, či stránka s takouto adresou neexistuje a ak nie, pokračuje ďalej v tvorbe.



Obr. č. 4 - Vkladanie názvu novej stránky do adresy webového prehliadača.

Ak stránka ešte neexistuje, zobrazí sa o tejto skutočnosti oznam a systém sa spýta užívateľa, či chce túto stránku vytvoriť. Pre pokračovanie v tvorbe treba kliknúť práve na tento odkaz.



Obr. č. 5 - Stránka ešte neexistuje, systém sa pýta užívateľa, či ju chce vytvoriť.

Po kliknutí na odkaz „create“ sa nám otvorí nová stránka a je možné ju okamžite meniť. Užívateľ má k dispozícii nástroje na úpravu textu a taktiež si môže zvoliť náhľad vytváranej stránky, aby videl ako stránka bude vyzerat' po uložení. Užívateľ teda môže vložit' informácie, ktoré chce na tejto stránke prezentovať a taktiež môže do stránky vkladat' rôzne akcie, ktoré sú ohraničené dvojitémi zloženými zátvorkami.

Luazeri : NovyBalicek

HomePage :: Categories :: PageIndex :: RecentChanges :: RecentlyCommented :: Change sett

H1 H2 H3 H4 H5 | **B** *I* U ~~S~~ | [List icons] | [Link icon] [Image icon] [Code icon] | HTML BASH PERL C++ [Lightbulb icon] [Hand icon]

====Môj nový fantastický balíček====

Ahojte všetci, mám pre vás nový balíček, keď už sú tie vianoce....

```
{{calendar}}
```

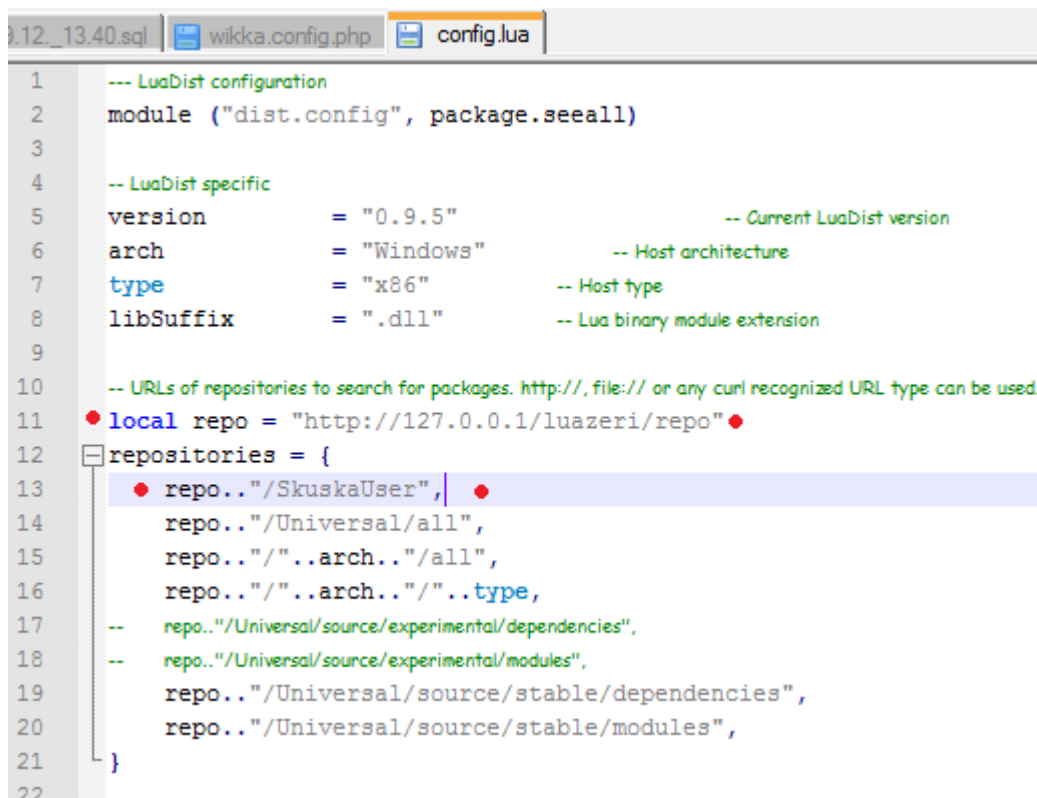
Please add a note on your edit

Store Preview Cancel

Obr. č. 6 - Editácia novej stránky, ukážka textu formátovaného ako H1 a pridanie akcie „calendar“.

1.3 Prístup k balíčkom

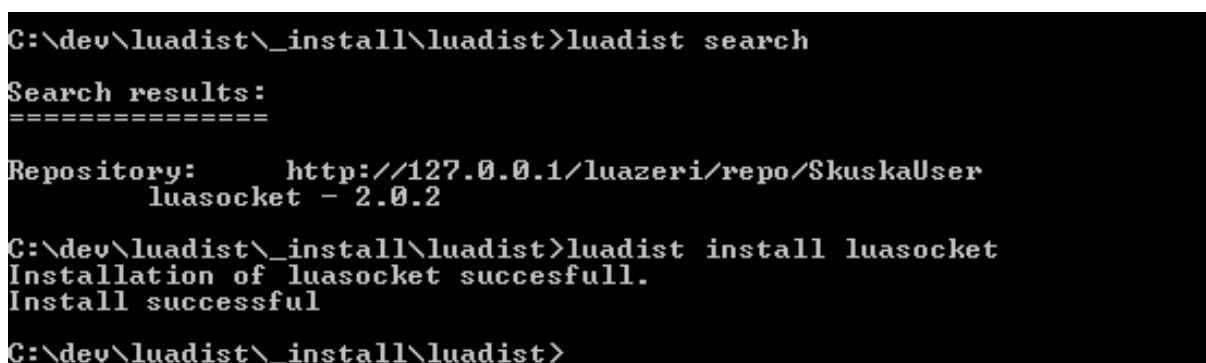
Pre prístup k nahraným súborom je potrebné najprv pridať adresár s balíčkami medzi repozitáre v konfiguračnom súbore luy.



```
1  --- LuaDist configuration
2  module ("dist.config", package.seeall)
3
4  -- LuaDist specific
5  version      = "0.9.5"                -- Current LuaDist version
6  arch        = "Windows"              -- Host architecture
7  type        = "x86"                  -- Host type
8  libSuffix   = ".dll"                 -- Lua binary module extension
9
10 -- URLs of repositories to search for packages. http://, file:// or any curl recognized URL type can be used.
11 local repo = "http://127.0.0.1/luazeri/repo"
12 repositories = {
13     repo.."/SkuskaUser",
14     repo.."/Universal/all",
15     repo.."/"..arch.."/all",
16     repo.."/"..arch.."/"..type,
17     -- repo.."/Universal/source/experimental/dependencies",
18     -- repo.."/Universal/source/experimental/modules",
19     repo.."/Universal/source/stable/dependencies",
20     repo.."/Universal/source/stable/modules",
21 }
22
```

Obr. č. 7 - Pridanie adresára medzi repozitáre.

Následne si môžeme nainštalovať tento balíček. Najprv zvolíme vyhľadať. LuaDist nám zobrazí všetky repozitáre a ich obsah. Potom môžeme nájdené balíčky inštalovať pomocou príkazu „luadist install <názov balíčka>“.



```
C:\dev\luadist\_install\luadist>luadist search
Search results:
=====
Repository:      http://127.0.0.1/luazeri/repo/SkuskaUser
                  luasocket - 2.0.2
C:\dev\luadist\_install\luadist>luadist install luasocket
Installation of luasocket succesfull.
Install successful
C:\dev\luadist\_install\luadist>
```

Obr. č. 8 - Inštalácia pridaného balíčka.

1.4 Ako uploadovať balíček

Tento návod má za úlohu oboznámiť používateľa wiki o uploadovaní balíčkov do jeho repozitára. Svoj repozitár a jeho adresu nájdete v nastaveniach (ChangeSettings). Každý uploadnutý balíček sa stáva automaticky skrytý pre ostatných používateľov, pre zverejnenie vid' tutoriál o [zverejňovaní](#)

[balíčkov](#). Pre každý nový balíček je vytvorená jeho stránka zo základnými informáciami, ktoré môžete ako jeho autor podľa potreby meniť.

1.5 Dist Upload

Upload balíčku je možný len cez stránku na to určenú (viď nižšie). Nájdete ju v menu ako 'Dist Upload'. Po otvorení tejto stránky máte k dispozícii formulár na uploadovanie balíčku.



Obr. č. 9 - Obrazovka na upload balíčka

Zvoľte tlačidlo 'Prechádzať' (Choose) a vyberte súbor s príponou dist. Prepíšte slová z captche. Následne odošlite súbor pomocou tlačidla 'Submit'.

1.5.1 Upload ukončený

Súbor je odoslaný na server a uložený vo Vašom repozitári. Pri ukladaní do repozitára sa v ňom automaticky pregeneruje súbor dist.manifest, do ktorého sa uložia údaje zo súboru dist.info odoslaného v balíčku. Automaticky je vytvorená stránka pre balíček, ktorá tieto údaje zobrazuje (viď nižšie).



Obr. č. 10 - Obrazovka po úspešnom uploadnutí balíčka

1.5.2 Ďalšie možnosti

Nakoniec môžete prejsť na stránku vytvorenú pre balíček, alebo môžete pokračovať v uploadovaní ďalších balíčkov.

1.6 Uploadovanie pomocou utility *luadistUp*

Túto utilitu je možné stiahnuť z <http://luadist.org/modules/luadist/luadistup.jar>. Je to aplikácia napísaná v Java, takže je potrebné JRE aby sa dala používať. JRE je potrebné stiahnuť a nainštalovať ak sa na vašom počítači ešte nenachádza.

Konečne je možné uploadovať balíčky buď priamo z konzoly alebo použitím GUI. Táto utilita umožňuje upload viacerých balíčkov naraz.

Pri použití konzoly použite nasledovný príkaz:

```
luadistup.jar <url> <login> <password> <directory_to_upload>
```

kde

- <url> je url wiki pre [DistUploadAuto](http://luadist.org/modules/wikka.php?wakka=DistUploadAuto) stránku. Pre tento krát, použite URL: <http://luadist.org/modules/wikka.php?wakka=DistUploadAuto>.

- `<login>` je login do wiki
- `<password>` je heslo k účtu vo wiki
- `<directory_to_upload>` môže sa jednať o jediný balíček alebo o adresár ktorý aplikácia rekurzívne prehľadáva a ukladuje všetky *.dist súbory.

Tento príkaz teda môže vyzerat' nasledovne:

```
luadistup.jar      http://luadist.org/modules/wikka.php?wakka=DistUploadAuto
WikkiUser guessMe mydistrepo
```

Ak spustíte aplikáciu bez parametrov, alebo s nesprávnymi parametrami, objaví sa GUI, kde je možné tieto parametre manuálne pridať. Potom po stlačení tlačidla 'Upload' sa začne upload balíčkov.

Poznámka: Pri jednom alebo druhom spôsobe, musí byť zadaná URL zhodná s URL wiki stránky kde sa DistUploadAuto nachádza. Ak je zadaná iná URL, upload jednoducho zlyhá.

Po úspešnom uploade sú stránky uploadnutých balíčkov automaticky vygenerované ako keby bol použitý jednoduchý DistUpload formulár priamo z wiki stránky.

1.7 Ako zverejňovať (publikovať) balíček

Publikovanie balíčkov slúži na zverejnenie balíčka pre ostatných používateľov. Balíček je pôvodne po pridaní v stave skrytý, sprístupniť ho verejnosti môže jeho majiteľ. Pri publikovaní sa balíček musí zaradiť do kategórie kam bude patriť. Túto kategóriu vyberie sám užívateľ. Publikovanie je **nevratná** operácia, zmeny v publikovaní môžu byť prevádzané len cez administrátorov, jednotlivých kategórií. Každá kategória má prideleného jedného administrátora, ktorý je uvedený na stránke kategórie (viď bod 5).

1.7.1 Vyber balíček

Otvorte zoznam svojich balíčkov z menu 'Your [DistIndex](#)'. Zo zoznamu vyberte nepublikovaný balíček a otvorte jeho stránku.

Your Dist Index

This is an alphabetical list of dist pages you can view on this server.

View All or or find user by name: Find

UserName

Name/Version	Page	Published	Actions
<input type="checkbox"/> lpeg <i>Parsing Expression Grammars For Lua</i>			
0.8.1	UserName_lpeg-0.8.1-Windows-x86.dist	2009-04-23 00:56:48	<input type="button" value="✖"/>

Obr. č. 11 - Obrazovka osobného DistIndexu (repozitára) používateľa

1.7.2 Vyber kategóriu

V ľavom dolnom rohu sa nachádza formulár na publikovanie balíčkov. V ňom si môžete vybrať z dostupných kategórií kam môžete balíček publikovať. Po zvolení kategórie balíček publikujte stlačením tlačidla 'Publish package'.

lpeg

Uploaded: 2009-04-23 00:44:30 by: [UserName](#)
 File size: 57 Kb
 Stored in: [./repo/3719712122bf694bc6ae4c48cd8425fb/lpeg-0.8.1-Windows-x86.dist](#)
 Version: 0.8.1
 Arch: Windows
 Type: x86
 Number of files in package: 7

Description:
 LPeg is a new pattern-matching library for Lua, based on Parsing Expression Grammars (PEGs). The nice thing about PEGs is that it has a formal basis (instead of being an ad-hoc set of features), allows an efficient and simple implementation, and does most things we expect from a pattern-matching library (and more, as we can define entire grammars).

There are no comments on this page. [\[Add comment\]](#)

[Edit](#) :: [Page History](#) :: 2009-04-23 00:44:30 :: Votes: 0 :: You own this page. :: [Edit ACLs](#) :: [Referrers](#) :: Search:

Obr. č. 12 - Obrazovka balíčka spolu s informáciami o ňom

1.7.3 Úspešne publikovaný balíček

Počas publikovania balíčka sa vytvorí jeho kópia v repozitári kategórie, ktorý je voľne dostupný pre ostatných užívateľov. Taktiež sa vytvorí nová stránka dostupná každému pre tento balíček, majúca v

názve stránky názov kategórie. V pravom hornom rohu sa zobrazí informácia o tom, že balíček bol publikovaný úspešne a formulár na publikovanie je nahradený informáciou o tom kedy bol balíček publikovaný.

lpeg
Package published successfully.

Uploaded: **2009-04-23 00:44:30** by: [UserName](#)
 File size: **57 Kb**
 Stored in: `./repo/3719712122bf694bc6ae4c48cd8425fb/lpeg-0.8.1-Windows-x86.dist`
 Version: **0.8.1**
 Arch: **Windows**
 Type: **x86**
 Number of files in package: **7**

Description:
 LPeg is a new pattern-matching library for Lua, based on Parsing Expression Grammars (PEGs). The nice thing about PEGs is that it has a formal basis (instead of being an ad-hoc set of features), allows an efficient and simple implementation, and does most things we expect from a pattern-matching library (and more, as we can define entire grammars).

There are no comments on this page. [\[Add comment\]](#)

Page was published 2009-04-23 00:56:48
 Edit :: [Page History](#) :: 2009-04-23 00:44:30 :: Votes: 0 :: You own this page. :: [Edit ACLs](#) :: [Referrers](#) :: Search:

Obr. č. 13 - Obrazovka zverejneného balíčka

1.8 Zobrazenie publikovaných balíčkov

Všetky naposledy publikované balíčky sa nachádzajú v menu zozname publikovaných balíčkov, kde si môžete aj vyhľadať publikované balíčky ostatných užívateľov alebo ich filtrovať podľa dátumu publikovania.

Recently Published Packages Index							
This is a list of published dist pages you can view on this server.							
View All or <input type="text" value="-- select a date --"/> or <input type="text" value="-- select an user --"/> or find user by name: <input type="text"/> Find							
Published	Name	Version	Short description	Published by	Category	Page	
2009-04-26 20:42:27	luafilesystem	1.4.1	File System Library for the Lua Programming Language	KajoTestik	AdminCategory05	AdminCategory05_luafilesystem-1.4.1-Windows-x86.dist	
2009-04-26 19:41:38	luafilesystem	1.4.1	File System Library for the Lua Programming Language	HakiBaki	AdminCategory03	AdminCategory03_luafilesystem-1.4.1-Windows-x86.dist	
2009-04-23 01:10:22	lpeg	0.8.1	Parsing Expression Grammars For Lua	Ma5	AdminCategory05	AdminCategory05_lpeg-0.8.1-Windows-x86.dist	
2009-04-23 00:56:48	lpeg	0.8.1	Parsing Expression Grammars For Lua	UserName	AdminCategory03	AdminCategory03_lpeg-0.8.1-Windows-x86.dist	
2009-04-22 17:52:08	bitlib	25	Tiny library for bitwise operations	RimmoN	AdminCategory03	AdminCategory03_bitlib-25-Windows-x86.dist	
2009-04-22 15:48:20	bitlib	25	Tiny library for bitwise operations	TestUser	AdminCategory05	AdminCategory05_bitlib-25-Windows-x86.dist	
2009-04-22 12:30:22	bitlib	25	Tiny library for bitwise operations	Ma5	AdminCategory04	AdminCategory04_bitlib-25-Windows-x86.dist	
2009-04-22 00:57:21	lpeg	0.8.1	Parsing Expression Grammars For Lua	RimmoN	AdminCategory04	AdminCategory04_lpeg-0.8.1-Windows-x86.dist	
2009-04-22 00:55:45	coxpcall	1.13.0	Coroutine safe xpcall and pcall	RimmoN	AdminCategory03	AdminCategory03_coxpcall-1.13.0-universal-all.dist	

Obr. č. 14 - Obrazovka posledne zverejnených balíčkov

1.9 Zobrazenie kategórií

Ak chcete prezrieť kategóriu, kde ste balíček publikovali alebo sa chystáte publikovať, zvolíte v menu Package Categories, kde sú zobrazené odkazy na stránky jednotlivých kategórií a zvolíte si kategóriu, ktorú chcete prezrieť.


Package Categories Index

This is an alphabetical list of package categories you can view on this server.

Name	Page	E-mail
▼ AdminCategory03	AdminCategory03	AdminCategory03@luadist.org
▼ AdminCategory04	AdminCategory04	AdminCategory04@luadist.org
▼ AdminCategory05	AdminCategory05	AdminCategory05@luadist.org
▼ AdminCategoryKajo	AdminCategoryKajo	kajo@ynet.sk

Obr. č. 15 - Obrázok zobrazujúca kategórie do ktorých sa môže zverejňovať balíček

1.10 Vymazanie balíčka


Vymazávanie balíčkov slúži na ich odstránenie buď so svojho vlastného repozitára alebo z celej databázy. Používateľ má právo mazať len balíčky, ktoré vlastní a teda nachádzajú sa na 'Your DistIndex' stránke. Vymazanie balíčka sa vykoná stlačením tlačidla  nachádzajúceho sa vedľa informácií o balíčku v sekcii 'Actions' (viď obrázok).

Your Dist Index

This is an alphabetical list of dist pages you can view on this server.

View All or or find user by name:

UserName

Name/Version	Page	Published	Actions
▼ lpeg			
Parsing Expression Grammars For Lua			
0.8.1	UserName_lpeg-0.8.1-Windows-x86.dist	2009-04-23 00:56:48	

Obr. č. 16 - Obrázok osobného DistIndexu (repozitára) používateľa

Po stlačení tlačidla sa vymazaný balíček odstráni zo zoznamu v 'Your DistIndex' a taktiež sa vymaže stránka daného balíčka. V prípade, že bol vymazávaný balíček už zverejnený, vykonajú sa všetky spomenuté operácie, avšak balíček zostane viditeľný pre verejnosť aj naďalej a taktiež zostáva nezmenená aj jeho stránka, ktorá bola vytvorená pri publikovaní tohto balíčka. Publikovaný balíček má právo vymazať jedine administrátor alebo vlastník balíčku.

Príloha C – Inštalačná príručka

1 Inštalačná príručka

Táto príručka obsahuje popis všetkých krokov, ktoré sú potrebné na spustenie a používanie tohto produktu.

1.1 Apache

Prvým hlavným krokom je nainštalovanie a nakonfigurovanie servera, na ktorom celý wiki systém pobeží. V našom prípade sa jedná o Apache HTTP Server, čo je softwarový webový server s Open source licenciou pre Linux, BSD, Microsoft Windows a iné platformy. V dnešnej dobe je najrozšírenejším na celom svete. Je možné ho získať zo stránky <http://www.apache.org/> a nainštalovať podľa návodov na stránke.

1.1.1 Základná konfigurácia Apache

Konfigurácia spočíva hlavne v editovaní súboru **httpd.conf**, ktorý bude v C:\Program Files\Apache\conf\. Otvorte si tento súbor v editore a môžeme sa pustiť do konfigurácie servera.

Hlavné direktívy servera

ServerRoot "C:/Program Files/Apache" - Táto direktíva nám hovorí, kde bude hlavný adresár servera, hľadať sa tam budú konfiguračné súbory a logy (/conf , /logs). Bude to aj adresár, od ktorého sa budú brať relatívne cesty.

DocumentRoot "C:/Program Files/Apache/htdocs/" - Táto direktíva nám hovorí, kde bude hľadať server súbory (php, html) aby ich vykonal. Čiže po zadaní <http://localhost> sa zobrazí obsah na ceste C:/Program Files/Apache/htdocs/.

ServerName 127.0.0.1 - Táto direktíva nám uvádza názov hostiteľa pre servera. Pre lokálne použitie nám bude stačiť lokálna IP adresa (127.0.0.1) alebo meno localhost.

Listen

80

Táto direktíva nám hovorí, na akom porte, alebo IP adrese bude Apache čakať požiadavky od klientov. Nechávame zvyčajne na Porte 80.

Viac sa dá získať na domovskej stránke produktu. Pre základný beh servera sú však tieto informácie postačujúce.

1.2 MySQL, PHP, E-Mail, JSON

Následne je potrebné do neho doinštalovať databázový server MySQL (<http://dev.mysql.com/downloads/>), PHP (<http://www.php.net/>) a podporu pre posielanie e-mailov (<http://commons.apache.org/email/>). Postupný popis inštalácií opäť nie je nutné bližšie špecifikovať, keďže sa jedná o inštalačné súbory, ktoré všetko spraví za vás (Windows). Externá knižnica JSON, ktorá sa používa na prácu so súborami .dist je pribalená priamo v základnej adresárovej štruktúre nášho produktu (luadist/JSON.php) a odkazujeme sa na ňu cez relatívne cesty. To znamená, že knižnica je už predinštalovaná a nie je potrebné ju inštalovať aj priamo do php na serveri.

1.3 Nastavenie práv na serveri

Následne je nutné nastaviť práva na zápis do repozitárov pre používateľa, pod ktorým funguje server Apache. Inak nebude možné nahrávať balíčky na server. Ideálnym riešením je nastaviť celej adresárovej štruktúre práva s označením 755 (platí pre unix servery).

1.4 Wikka.config nastavenia

Následne je možné nahrat' na server celý zdrojový kód nášho produktu (luadist_src/*) a importovať priložený sql súbor do vytvorenej databázy (luadist_src/db_dump/luadist_db_clean.sql).

Na koniec je potrebné upraviť zdrojový súbor wikka.config.php nasledovne (v zátvorkách sú odporúčané hodnoty):

'mysql_host'	adresa servera, kde beží mysql (localhost)
'mysql_database'	názov databázy (luadist)
'mysql_user'	meno používateľa, ktorý bude pristupovať k databáze a má vytvorené konto na mysql (luadist)
'mysql_password'	heslo patriace danému používateľovi pre prístup do databázy
'admin_users'	vymenovaní používatelia (prihlasovacie meno), ktorí majú práva administrátorov, oddelení čiarkou
'admin_email'	email na hlavného administrátora
'table_prefix'	predpona, ktorá sa dáva vytvoreným tabuľkám v databáze (luadist)

'root_page'	názov hlavnej stránky (HomePage)
'wakka_name'	názov celej wiki Wakka (LuaDist)

Nasledovné informácie nie je implicitne nutné meniť:

'meta_keywords' => " – meta informácie o stránke, kvôli SEO (Search Engine Optimization) optimalizácii

'meta_description' => " – meta informácie o stránke, kvôli SEO optimalizácii

'allow_user_registration' => '1' – prednastavené na '1' (dovoľuje sa registrovať novým užívateľom), ak sa zmení na nula nie je možné registrovať nových používateľov

'enable_version_check' => '1' – ak je nastavené na '1' tak sa kontrolujú nové verzie systému Wikka Wakka, ak je '0' kontrola nových verzií je vypnutá

'require_edit_note' => '0' – ak je nastavené na '1' tak po každej zmene stránky sa vyžaduje popis zmeny, v prípade '0' je možné meniť stránky bez popisov o zmenách

Ostatné nastavenia je doporučené nechať bez zmeny, keďže by sa tým mohla narušiť stabilita celej wikky.

2 Obsah priloženého CD

Celý zdrojový kód nášho produktu je zbalený do priloženého súboru LuaDist_Luazeri.zip. Po jeho rozbalení je potrebné nahráť celú jeho štruktúru na server. V adresári /__instal.guide sa nachádza inštaláčna príručka ešte raz kvôli tomu, že inštalátor nemusí mať pri sebe vždy dokumentáciu a je uložená v textovom formáte čitateľnom na hocijakej platforme. SQL skript, ktorý je potrebné spustiť na MySQL servery sa nachádza v adresári /__db_dump/luadist_db_clean.sql. Obsahuje súbor všetkých 10 tabuliek potrebných pre beh produktu spolu so základnými údajmi (wiki stránky, odkazy medzi nimi, základné prístupové práva...). Po rozbalení vznikne nasledovná adresárová štruktúra:

- 3rdparty - knižnice tretích strán, záležitosť WikkaWakka
- Actions - hlavné akcie pre wiki stránky, hlavný zdrojový kód
- Ajax - zdrojový kód funkcie hlasovania
- Css - kaskádové štýly, hlavný wikka.css
- Docs_wikka - základná dokumentácia k systému WikkaWakka
- Formatters - knižnice funkcií, záležitosť WikkaWakka

Handlers	-	triedy zahrňujúce funkcie pre správu obsahu, záležitosť WikkaWakka
Images	-	adresár obsahujúci všetky obrázky na stránke
Intranet	-	špeciálne funkcie WikkaWakka
Libs	-	triedy zahrňujúce funkcie pre správu revízií jednotlivých stránok wiki
Luadist	-	súbory a knižnice potrebné pre správu .dist súborov
Plugins	-	doplňky systému WikkaWakka
Repo	-	rezpozitáre všetkých používateľov
Setup	-	základné nastavenie systému WikkaWakka
Templates	-	šablóny obsahu WikkaWakka
Uploads	-	súboru nahrané na server od ostatných používateľov
__db_dump	-	sql skript s inštaláciou databázy
__instal.guide	-	používateľská príručka

Príloha D – Zápisnice zo stretnutí

Zápisnica zo stretnutia č.1

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
7.10.2008	19:00-20:15	Respírium na 2. poschodí bloku D	Bc. Karol Marton	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Predstavenie projektu a riešenej problematiky

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
	Dosiahnuté výsledky, výstupy, poznámky			

Priebeh stretnutia:

Na úvod nám Ing. Drahoš hlbšie predstavil celý projekt a predstavil prostredie komunity združujúcej sa okolo jazyka Lua. Naša úloha by sa dala rozdeliť na dve časti:

1. implementácia webového rozhrania pre repozitár modulov,
2. doprogramovanie používateľského rozhrania pre už existujúci inštalátor LuaDist.

Zopakoval svoje predstavy o úspešnom projekte (minimálne požiadavky, maximálna flexibilita a škálovateľnosť, prenositeľnosť), ale zároveň pripomenul, že máme voľné ruky a nebude nás prílišne obmedzovať vo výbere spôsobu tvorby web rozhrania repozitára. Na záver sme si zopakovali úlohy s termínmi ich predpokladaného ukončenia. Stretnutie trvalo 75 minút z dôvodu obmedzených zdrojov, nakoľko sme nemali prístup k potrebnému softvéru.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
1	Poslať e-mail Ing. Drahošovi s pripomenutím poslania odkazov na predstavované web stránky	Karol Marton	7. 10. 2008
2	Dohodnúť sa na mene tímu	Michal Dávid	10. 10. 2008
3	Preštudovať referenčný manuál o jazyku Lua spolu s príručkou k programovaniu	Miloš Cibulka	28. 10. 2008
4	Vytvoriť plagát tímu na základe mena tímu	Michal Dávid	21. 10. 2008
5	Zistiť možnosť vytvorenia webovej prezentácie v JAVE na adrese soft. štúdia	Jakub Marton	14. 10. 2008
6	Založiť si projektový denník	Karol Marton	14. 10. 2008
7	Preštudovanie odkazov od Ing. Drahoša a vniknutie do problematiky okolo jazyka Lua	Jakub Marton	28. 10. 2008
8	Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia	Marián Halaš	13. 11. 2008
9	Predbežný cieľ na Letný semester: Doprogramovanie používateľského rozhrania pre inštalátor	Karol Marton	dlhodobá

Zápisnica zo stretnutia č.2

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
14.10.2008	18:00-20:00	Labss2	Bc. Marián Halaš	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Hlbší úvod do problematiky LuaDist

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
1	Poslať e-mail Ing. Drahošovi s pripomenutím poslania odkazov na predstavované web stránky - kvôli nastaveniu na groups.google.com nebolo možné prijať odpoveď na mail.	Karol Marton	7.10.2008	splnená
2	Dohodnúť sa na mene tímu - meno tímu: Lu(a)zeri bolo jednohlasne schválené	Michal Dávid	10.10.2008	splnená
3	Preštudovať referenčný manuál o jazyku Lua spolu s príručkou k programovaniu	Miloš Cibulka	28.10.2008	začatá
4	Vytvoriť plagát tímu na základe mena tímu Mišo plagát vytvoril, ešte ho treba vytlačiť	Michal Dávid	21.10.2008	splnená
5	Zistiť možnosť vytvorenia webovej prezentácie v JAVE na adrese soft. štúdia - Jakub napíše mail Lackovi čím skôr	Jakub Marton	14.10.2008	nesplnená
6	Založiť si projektový denník	Karol Marton	14.10.2008	splnená
7	Preštudovanie odkazov od Ing. Drahoša a vniknutie do problematiky okolo jazyka Lua - linky boli poslané až 14.10. čiže nebolo kedy ich študovať	Jakub Marton	28. 10. 2008	ešte nezačatá
8	Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia - začali sme analýzou problémovej oblasti	Marián Halaš	13. 11. 2008	začatá
9	Predbežný cieľ na Letný semester:	Karol Marton	dlhodobá	

	Doprogramovanie používateľského rozhrania pre inštalátor			
--	--	--	--	--

Priebeh stretnutia:

Na začiatku stretnutia nám Ing. Drahoš predviedol inštaláciu a funkcionality jazyku Lua a projektu Luadist. Spresnil svoje požiadavky na systém:

- opis základnej funkcie systému – nahrávanie a sťahovanie balíčkov pomocou protokolu http
- predstavenie požiadavky minimálnych obmedzení pre používateľa – použiť nejaký funkčný wiki systém

Naša úloha by sa dala rozdeliť na dve časti. V prvej implementujeme web pre repozitár modulov a v druhej doprogramujeme používateľské rozhranie pre už existujúci inštalátor.

Po vysvetlení princípu Luadist aj na základe analógie s nástrojom Maven pre Javu sme zostali sami a diskutovali sme o ďalšom postupe. Dohodli sme sa na rozdelení nových úloh, na základe ktorých sa budeme môcť lepšie rozhodnúť v niektorých kľúčových veciach, predbežne sme sa dohodli na tom, že použijeme nejaké už existujúce PHP Wiki. Takisto sme sa predbežne dohodli, že súbory rozširujúcich modulov nebudeme ukladať do databázy, len relatívne cesty k nim, samotná súborová štruktúra je ešte otázná. Nakoniec sme si rozdelili úlohy do ďalšieho stretnutia.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
10	Spraviť prehľad existujúcich wiki a navrhnúť nejakých favoritov	Karol Marton Jakub Marton	21. 10. 2008
11	Workflow diagram server side, UseCases	Michal Dávid Miloš Cibulka	21. 10. 2008
12	Vytlačiť plagát	Michal Dávid	21. 10. 2008

Zápisnica zo stretnutia č.3

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
21.10.2008	19:00-20:30	Labss2	Bc. Miloš Cibulka	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Návrh riešenia projektu

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
3	Preštudovať referenčný manuál o jazyku Lua spolu s príručkou k programovaniu	Miloš Cibulka	28.10.2008	čiastočne splnená
7	Preštudovanie odkazov od Ing. Drahoša a vniknutie do problematiky okolo jazyka Lua	Jakub Marton	28.10.2008	čiastočne splnená
8	Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia - začali sme už s nákresom workflow diagramov	Marián Halaš	13.11.2008	začatá
9	Predbežný cieľ na Letný semester: Doprogramovanie používateľského rozhrania pre inštalátor	Karol Marton	dlhodobá	nezačatá
10	Spraviť prehľad existujúcich Wiki a navrhnúť nejakých favoritov	Karol Marton, Jakub Marton	21.10.2008	splnená
11	Workflow diagram server side, UseCases - počas návrhu a tvorby bolo odhalených viacero otázok, ktoré bolo nutné prekonzultovať s Ing. Drahošom	Michal Dávid, Miloš Cibulka	21.10.2008	čiastočne splnená
12	Vytlačiť plagát - formát A3, zalaminovaný – náklady 84 SK (2,80 Eur) preplatené.	Michal Dávid	21.10.2008	splnená

Priebeh stretnutia:

Na začiatku, po formálnych náležitostiach, sme s Ing. Drahošom prekonzultovali otázky, ktoré vznikli pri návrhu aplikácie a ktoré bolo treba spresniť pre ďalšiu tvorbu návrhu. Ďalej bola rozobraná problematika budúcich obrazoviek webu, kde Ing. Drahoš vysvetlil, čo všetko a aké konkrétne údaje by v nich mali byť obsiahnuté.

Vedúci školiteľ taktiež pokračoval vo vysvetľovaní princípov fungovania jazyka Lua a projektu LuaDist a vysvetlil niektoré otázky ohľadom problémov so spustením jazyka Lua na našich domovských počítačoch.

Na záver nasledovala tradičná diskusia členov tímu o aktuálnej situácii a stave projektu. Na základe rady Ing. Drahoša sme vybrali jedného člena tímu, ktorý bude mať na starosti výskum technológií vhodných pre projekt (*Jakub Marton*). Rozhodli sme sa pomaly začať pripravovať dokumentáciu projektu. Celkovo bolo vytvorených a pridelených šesť nových úloh.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
13	Vytvoriť návrh obrazoviek webu.	Michal Dávid	28.10.2008
14	Vytvoriť prezentáciu tímu formou webovej stránky na adrese soft. štúdia. Stránka musí spĺňať všetky náležitosti bližšie určené na stránke prof. Bielikovej.	Michal Dávid	24.10.2008
15	Začať s výskumom technológií potrebných k implementácii projektu.	Jakub Marton	28.10.2008
16	Začať s tvorbou dokumentácie pre projekt.	Marián Halaš	28.10.2008
17	Vybrať vhodný projekt na báze Wiki, ktorý bude základom pre web. Svoje názory bude konzultovať s Jakubom, zodpovedným za výskum.	Karol Marton	28.10.2008
18	Dokončiť workflow diagramy a vytvoriť use casey podľa návrhu a pripomienok, ktoré boli prekonzultované s Ing. Drahošom.	Miloš Cibulka	28.10.2008

Zápisnica zo stretnutia č.4

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
28.10.2008	19:00-20:30	Labss2	Bc. Jakub Marton	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Návrh riešenia projektu

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
3	Preštudovať referenčný manuál o jazyku Lua spolu s príručkou k programovaniu	Miloš Cibula	28.10.2008	splnená
7	Preštudovanie odkazov od Ing. Drahoša a vniknutie do problematiky okolo jazyka Lua	Jakub Marton	28.10.2008	splnená
8	Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia	Marián Halaš	13.11.2008	čiastočne splnená
9	Predbežný cieľ na Letný semester: Doprogramovanie používateľského rozhrania pre inštalátor	Karol Marton	dlhodobá	nezačatá
11	Dokončiť workflow diagramy a vytvoriť use casey podľa návrhu a pripomienok, ktoré boli prekonzultované s Ing. Drahošom	Miloš Cibulka	28.10.2008	splnená
13	Vytvoriť návrh obrazoviek webu - boli vytvorené predbežné verzie, zatiaľ nevieme ako budú vyzerat' detailne	Michal Dávid	28.10.2008	splnená
14	Vytvoriť prezentáciu tímu formou webovej stránky na adrese soft. štúdia.	Michal Dávid Jakub Marton	24.10.2008	splnená

	- stránka obsahuje všetky náležitosti bližšie určené na stránke prof. Bielikovej			
15	Začať s výskumom technológií potrebných k implementácii projektu.	Jakub Marton	28.10.2008	častočne splnená
	Výskum pokračuje ďalej, boli navrhnuté dve nanoki a reviki			
16	Začať s tvorbou dokumentácie pre projekt	Marián Halaš	28.10.2008	splnená
17	Vybrať vhodný projekt na báze wiki, ktorý bude základom pre web. Svoje názory bude konzultovať s Jakubom, zodpovedným za výskum.	Karol Marton	28.10.2008	splnená
	- výskum pokračuje ďalej, boli navrhnuté dva systémy - nanoki a reviki			
18	Dokončiť workflow diagramy a vytvoriť use casey podľa návrhu a pripomienok, ktoré boli prekonzultované s Ing. Drahošom.	Miloš Cibulka	28.10.2008	splnená

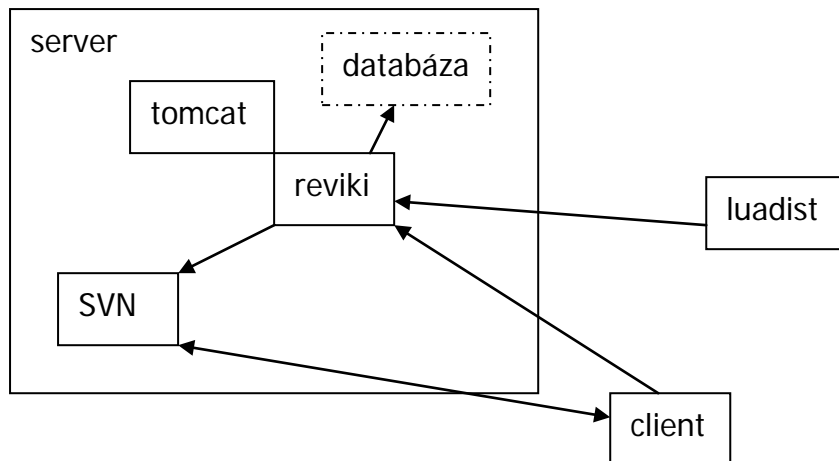
Priebeh stretnutia:

Na začiatku sme referovali o dvoch wiki, ktoré nás zaujali najviac. Prvá, **reviki**, implementovaná v Java, založená na systéme SVN, umožňuje priame ukladanie a spravovanie verzií obsahu. Druhá, **nanoki**, implementovaná v Lua, ma podstatnú výhodu v tom, že v sebe obsahuje http a file server, takže na jej deploynutie na web nie je potrebné nič viac. Teda, nie je potrebný hosting s apache serverom/tomcatom ako pri Java.

Následne sme rozoberali samotnú štruktúru a spôsob ako budú manažované balíky. Dostali sme sa k nasledovnej štruktúre:

- stable
 - lua.ver.x
 - lua.ver.x
- unstable
- user1
- user2
- ... ďalší používatelia.

Teda každý používateľ bude mať svoj repozitár, kde si bude môcť vyvíjať vlastné vetvy balíčkov. Po odchode Ing. Drahoša sme sa venovali návrhu technológií, a možných verzií návrhu. Prišli sme ku dvom: Použitie reviki a SVN:



Potreba databázy je ešte otázna. Klient, alebo náš používateľ bude môcť svoje vlastné vetvy balíčkov vyvíjať s overeným nástrojom SVN. Cez web rozhranie by boli manažované len samotné balíčky, diskusie a hodnotenia (práve tu by sa mala použiť databáza).

Pri použití nanoki by mohlo byť zložité spravovať verzie obsahu. Toto je potrebné zistiť, ako zložito je implementovaná, ako by sa dali spravovať verzie obsahu.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
15	Pokračovať v bližšom štúdiu reviki a nanoviki, rozhodnúť, ktorý systém sa použije	Jakub Marton Karol Marton	4.11.2008

Zápisnica zo stretnutia č.5

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
4.11.2008	18:00-19:30	Labss2	Bc. Michal Dávid	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Návrh riešenia projektu a tvorba dokumentácie

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
8	Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia	Marián Halaš	13.11.2008	čiastočne splnená
9	Predbežný cieľ na Letný semester: Doprogramovanie používateľského rozhrania pre inštalátor	Karol Marton	dlhodobá	nezačatá
15	Pokračovať v bližšom štúdiu reviki a nanoki, rozhodnúť ktoré sa použije	Jakub Marton Karol Marton	4.11.2008	splnená
- dohodli sme sa na použití wiki v PHP (WikkaWiki) s použitím SVN				

Priebeh stretnutia:

Na začiatku sme prekonzultovali výhody a nevýhody rewiki a nanoki. Nakoľko nanoki nemá vytvorenú dokumentáciu je problémom doimplementovať vlastné funkcie. Po vyjasnení nejasností a zhodnotení výhod a nevýhod sme sa rozhodli pre wiki napísanú v PHP s využitím SVN.

Následne nám Ing. Drahoš vysvetlil aké zmeny nastali v štruktúre súborov a adresárov v repozitároch. Zjednodušenie nastalo vo vynechaní .dist súborov a ich náhradou sa stali

súbory dist.info priamo v zip súbore package-u. Bola nám vysvetlené štruktúra a obsah jednotlivých súborov a zmeny, ktoré v nich nastali.

Po odchode Ing. Drahoša sme si ujasnili čo a ako budeme implementovať. Zhodli sme sa na štruktúre repozitárov spomenutej na poslednom stretnutí. Následne sme si rozdelili prácu do ďalšieho stretnutia. Hlavnou prioritou sa stalo dokončenie dokumentácie a jej prípravenie na odovzdanie

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
19	Poslať Ing. Drahošovi draft verziu dokumentácie.	Marián Halaš	10.11.2008
20	Nahodiť SVN na server a všetko nastaviť	Jakub Marton Karol Marton	18.11.2008
21	Dopracovať analýzu do dokumentácie	Jakub Marton Karol Marton	9.11.2008
22	Use Case, workflow diagramy zapracovať do dokumentácie	Miloš Cibulka	9.11.2008
23	Obrazovky a plán finálna podoba	Michal Dávid	9.11.2008

Zápisnica zo stretnutia č. 6

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
11.11.2008	18:30-20:00	Labss2	Bc. Karol Marton	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Diskusia k vytvorenej dokumentácii

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
8	Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia	Marián Halaš	13. 11. 2008	skoro dokončená
- chýba už len dopracovať mierne pripomienky Ing. Drahoša				
19	Poslať Ing. Drahošovi draft verziu dokumentácie.	Marián Halaš	10. 11. 2008	splnená
20	Nahodiť SVN na server a všetko nastaviť	Jakub Marton Karol Marton	18. 11. 2008	skoro dokončená
- chýba už len vyriešiť zdieľanie databázy				
21	Dopracovať analýzu do dokumentácie	Jakub Marton Karol Marton	9. 11. 2008	splnená
22	Use Case, workflow diagramy zapracovať do dokumentácie	Miloš Cibulka	9. 11. 2008	splnená
23	Obrazovky a plán finálna podoba	Michal Dávid	9. 11. 2008	splnená

Priebeh stretnutia:

Na začiatku stretnutia nám vedúci tímu predstavil najnovšie zmeny v projekte LuaDist a vysvetlil ešte raz veci, ktoré nám neboli jasné:

1. manifest súbor sa vytvára v pamäti počítača, je potrebné použiť explicitný príkaz v LuaDiste – *luadist make manifest*

2. manifest súbor je tabuľka → tabuľka verzií → tabuľka architektúr → tabuľka typov
(*symbol* → *označuje vzťah obsahuje*)
3. pripájanie na repozitáre:
 - v prípade vzdialeného (napr. cez http) sa vytvára súbor dist.manifest
 - keď ide o lokálny repozitár pracuje sa s vlastným manifestom uloženým v pamäti počítača, kvôli zjednodušeniu práce na localhoste
4. dokumentácia k LuaDist už existuje a dá sa nájsť v domovskom adresári LuaDist na adrese: /luadist/src/luadist-0.9/doc/install.html
5. obsah súboru dist.info:
 - povinné údaje sú len *name* a *version*
 - všetko ostatné je voliteľné

V ďalšej fáze sme diskutovali hlavne o dokumentácii, ktorú sme poslali Ing. Drahošovi na kontrolu a mal k nej zopár malých pripomienok, ktoré sú zhrnuté ako úlohy do nasledujúceho stretnutia.

Na záver sme hľadali možnosti zdieľania MySQL databázy, čo potrebujeme pre implementáciu nášho projektu. Boli navrhnuté viaceré možnosti:

- Naštartovať MySQL server na niektorom domácom počítači a cez tunel naň povoliť prístup z internetu, čo by mohlo byť pomalé (pri dynamickej IP adrese použiť službu dyndns - <http://www.dyndns.com>)
- Osloviť Ing. Lacka ako správcu softvérového štúdia o poskytnutie vzdialeného prístupu
- Nájsť verejný a zvonku prístupný MySQL server.

Predbežne sme sa dohodli na oslovení Ing. Lacka, ktorý by nám mohol poskytnúť požadovaný prístup na serveri labss2.fiit.stuba.sk

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
8	Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia	Marián Halaš	13. 11. 2008
	- výsledný .pdf súbor poslať Ing. Drahošovi		
20	Nahodiť SVN na server a všetko nastaviť	Jakub Marton Karol Marton	18. 11. 2008
	- vybaviť vzdialený prístup k databáze MySQL na serveri labss2.fiit.stuba.sk		
24	Upraviť zistené formálne nedostatky v dokumentácii	Marián Halaš	12. 11. 2008
	- vyhodiť logo na úvodnej strane príp. vymeniť za aktuálne		

	<ul style="list-style-type: none"> - preštylizovať krátke holé vety na úvodných stranách - odkazy a zdroje dať na koniec dokumentu v minimálnom počte 10 (podľa zaslaných linkov od Ing. Drahoša) - prepísať v celom dokumente LuaDist na LuaDist - nahodiť do výsledného dokumentu veci, ktoré ostatní členovia tímu spravia - slovník pojmov prerobiť na tabuľku 		
25	Do plánu doplniť zodpovednosti a k semestrálnym týždňom doplniť reálny dátum	Michal Dávid	12. 11. 2008
26	Napísať záver do dokumentácie	Karol Marton	12. 11. 2008
27	Prerobiť zápisnice podľa návrhov z cvičenia MSI	Karol Marton	12. 11. 2008
28	Spísať kroky, ktoré vykonáva LuaDist pri inštalácii balíčka	Jakub Marton	12. 11. 2008
29	Prerobiť obrázok č. 1 z dokumentácie	Jakub Marton	12. 11. 2008
30	Nakresliť workflow diagram podľa Jakubovho popisu inštalácie balíčka	Miloš Cibulka	12. 11. 2008
31	Doplniť popis modulu do dokumentácie	Marián Halaš	12. 11. 2008
32	Aktualizovať webové sídlo tímu o súbor výslednej dokumentácie – po odovzdaní	Michal Dávid	14. 11. 2008
33	Zaslanie opraveného draftu dokumentácie na posledné pripomienkovanie Ing. Drahošovi	Marián Halaš	12. 11. 2008

Zápisnica zo stretnutia č. 7

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
18.11.2008	18:30 - 20:00	Internát Mladosť pri vrátnici na C-D blokoch	Bc. Marián Halaš	Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Dokumentácia, posudok a funkcionálnosť systému

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
8	Odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek a návrh riešenia	Marián Halaš	13. 11. 2008	splnená
19	Poslať Ing. Drahošovi draft verziu dokumentácie.	Marián Halaš	10. 11. 2008	splnená
20	Nahodiť SVN na server a všetko nastaviť	Jakub Marton Karol Marton	18. 11. 2008	skoro ukončená
	- chyba už len vyriešiť zdieľanie databázy			
21	Upraviť zistené formálne nedostatky v dokumentácii	Marián Halaš	12. 11. 2008	splnená
	<ul style="list-style-type: none"> - vyhodiť logo na úvodnej strane príp. vymeniť za aktuálne - preštylizovať krátke holé vety na úvodných stranách - odkazy a zdroje dať na koniec dokumentu v minimálnom počte 10 (podľa zaslaných linkov od Ing. Drahoša) - prepísať v celom dokumente LuaDist na LuaDist - nahodiť do výsledného dokumentu veci, ktoré ostatní členovia tímu spravia - slovník pojmov prerobiť na tabuľku 			
25	Do plánu doplniť zodpovednosti a k semestrálnym týždňom doplniť reálny dátum	Michal Dávid	12. 11. 2008	splnená
26	Napísať záver do dokumentácie	Karol Marton	12. 11. 2008	splnená
27	Prerobiť zápisnice podľa návrhov z cvičenia MSI	Karol Marton	12. 11. 2008	splnená
28	Spísať kroky, ktoré vykonáva LuaDist pri inštalácii balíčka	Jakub Marton	12. 11. 2008	splnená
29	Prerobiť obrázok č. 1 z dokumentácie	Jakub Marton	12. 11. 2008	splnená

30	Nakresliť workflow diagram podľa Jakubovho popisu inštalácie balíčka	Miloš Cibulka	12. 11. 2008	splnená
31	Doplniť popis modulu do dokumentácie	Marián Halaš	12. 11. 2008	splnená
32	Aktualizovať webové sídlo tímu o súbor výslednej dokumentácie – po odovzdaní	Michal Dávid	14. 11. 2008	splnená
33	Zaslanie opraveného draftu dokumentácie na posledné pripomienkovanie Ing. Drahošovi	Marián Halaš	12. 11. 2008	splnená

Priebeh stretnutia:

Tentokrát sme sa stretli bez nášho pedagogického vedúceho, pretože takmer všetko potrebné už vieme. Hlavnou náplňou stretnutia bola diskusia o dokumentácii, priebehu jej tvorby, zhodnotili sme, že systém rozdelenia jednotlivých častí dokumentácie ľuďom podľa ich schopností a skúseností sa ukázal ako veľmi efektívny. Podobne sa ukázala ako vhodná dvojitá kontrola formátovania dvomi členmi tímu.

Ďalej sme sa bavili o veciach potrebných pre efektívny vývoj podľa zvolených technológií, hlavne systém zdieľanej databázy. Túto úlohu naďalej rieši Karol Marton. Diskutovali sme o funkciách systému, ktoré by bolo vhodné zahrnúť do prototypu, s ktorými sa predpokladajú ťažkosti. Dohodli sme sa, že do prototypu bude vhodné implementovať aspoň rozšírenie dátového modelu WikkaWiki a systém spracovania požiadavky na repozitár od LuaDistu. Na záver sme diskutovali o novej verzii LuaDistu, aby sme mali všetci jasno, čo sa zmenilo a čo zostalo nezmenené.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
20	Nahodiť SVN na server a všetko nastaviť	Jakub Marton Karol Marton	18. 11. 2008
	- vybaviť vzdialený prístup k databáze MySQL na serveri <i>labss2.fiit.stuba.sk</i>		
34	Napísať posudok na dokumentáciu tímu č. 20	Marián Halaš	20. 11. 2008
	-		
35	Nainštalovať si wikka wiki na lokálne počítače	Karol Marton	25. 11. 2008
36	Vybaviť prebratie dokumentácie od tímu č. 20 a zabezpečiť preberacie protokoly.	Marián Halaš	20. 11. 2008

Zápisnica zo stretnutia č. 8

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
25.11.2008	18:30 - 20:00	Labss2	Bc. Miloš Cibulka	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Dokumentácia, posudok a funkcionálnosť systému

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
20	Nahodiť SVN na server a všetko nastaviť	Jakub Marton Karol Marton	18. 11. 2008	splnená
34	Napísať posudok na dokumentáciu tímu č. 20	Marián Halaš	20. 11. 2008	splnená
35	Nainštalovať si wikka wiki na lokálne počítače	Karol Marton	25. 11. 2008	splnená
36	Vybaviť prebratie dokumentácie od tímu č. 20 a zabezpečiť preberacie protokoly.	Marián Halaš	20. 11. 2008	splnená

Priebeh stretnutia:

Stretnutie bolo opäť zvolané do softvérového štúdia a bolo už aj za účasti pedagogického vedúceho. Ten v úvode zhodnotil našu prácu na dokumentačnej časti projektu. Túto tému sme teda uzavreli a prešli sme na ďalšiu časť projektu - implementáciu. Ing. Drahoš špecifikoval požiadavky, čo všetko je potrebné implementovať v pilotnej verzii do Vianoc.

Pilot by mal byť zameraný na funkcionálnu a logickú časť, nie na užívateľské rozhranie a mal by obsahovať:

- Zdieľanie súborov
- Vytváranie užívateľov a ich adresárov
- Generovanie a pregenerovanie manifest súboru
- Čítanie dist.info súborov

Ďalej Ing. Drahoš odporučil, aby tím si rozdelil implementačnú časť do nasledovných oblastí. Na základe týchto odporúčaní sme k nim priradili ľudí z tímu.

- Jeden človek zodpovedný za dokumentáciu – Marián Halaš
- Traja ľudia implementujúci funkcionality – Karol Marton, Miloš Cibulka, Michal Dávid
- Jeden človek zaoberajúci sa problematikou Cmake a Luadist z dôvodu lepšieho poznania ich API, ktorý bude potrebný pre implementáciu – Jakub Marton

Na záver prebehla obvyklá porada členov tímu ohľadom rozdelenia úloh. Tím sa taktiež predbežne dohodol na stretnutí v sobotu o 13:00 hod. ak by počas implementácie vznikli nejaké problémy a bolo by nutné ich prediskutovať.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
37	Zistiť ako rozbaľiť dist.info súbor a zistiť z neho potrebné údaje.	Jakub Marton Marián Halaš	2. 12. 2008
38	Generovanie a pregenerovanie dist.manifest súboru. -	Karol Marton	2. 12. 2008
39	Doplniť existujúcu formu na pridávanie užívateľov z wikka wiki o údaje analyzované v dokumentácií.	Miloš Cibulka	2. 12. 2008
40	Vytváranie adresárov pre nových užívateľov.	Michal Dávid	2. 12. 2008

Zápisnica zo stretnutia č. 9

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
2.12.2008	9:00 - 10:30	D203, respírium 2. poschodie	Bc. Jakub Marton	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: **Implementácia UC1 Pridanie nového balíčka a UC10 – Registrácia**

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
37	Zistiť ako rozbaľiť dist.info súbor a zistiť z neho potrebné údaje.	Jakub Marton Marián Halaš	2. 12. 2008	splnená
	Napísané nové funkcie do luadist.lua, pre získanie informácií z dist.info. Napísané wrap php funkcie. Potrebné otestovať/vytvoriť tieto funkcie pre unix-based systémy.			
38	Generovanie a pregenerovanie dist.manifest súboru.	Karol Marton	2. 12. 2008	čiastočne splnená (90%)
	Podobne ako v 37, vytvorený wrapper, ktorý volá luadist a ten vytvorí manifest. Tiež potrebné otestovať na unix-based systémoch.			
39	Doplniť existujúcu formu na pridávanie užívateľov z wikka wikki o údaje analyzované v dokumentácií.	Miloš Cibulka	2. 12. 2008	zrušená
40	Vytváranie adresárov pre nových užívateľov.	Michal Dávid	2. 12. 2008	splnená
	Potrebné vyriešiť bezpečnosť. Zatiaľ majú do týchto adresárov prístup všetci.			

Priebeh stretnutia:

Stretnutie bolo neobvykle o 9,00 hodine ráno, nakoľko večer nebolo cvičenie z MSI. Počkali sme sa na 2. poschodí bloku D a po tom čo dorazili 3 meškajúci členovia z C4 15/3 sme prešli do kancelárie Ing. Drahoša.

Vedúci nás informoval o aktuálnom stave luadist, pre ktorý je pripravovaný release, nakoľko boli odstránené všetky väčšie chyby. Tu sme vedúceho informovali o stave implementácie. Jakub napísal nové funkcie do luadist.lua, ktoré slúžia na získanie informácií z dist.info. K týmto funkciám sú tiež napísané wrappre do php. Funkciou exec() volajú luadist. Pre generovanie manifestu je rovnako napísaná wrap funkcia, ktorá volá luadist a ten vygeneruje manifest v danom adresári. Ďalšou funkciou je funkcia unzip, ktorá rozbalí dist archív. Všetky tieto funkcie sú v súbore luadist.lib.php. Vznikli problémy ohľadom relatívnych ciest, a preto je momentálne táto knižnica uložená spolu s luadistom v jednom adresári. Potrebné vyriešenie tohto problému a upratanie súborov. Ďalšou úlohou je otestovanie/dopísanie týchto funkcií pre unix systémy, nakoľko väčšina web-hostingov je na tejto platforme. Ing. Drahoš upozornil na skutočnosť, že mnohé hostingy neposkytujú OS shell, ktorý je potrebný pre funkciu exec. Potrebné nájsť hosting poskytujúci OS shell, alebo prerobiť funkcie na čisté php. Miloš a Michal pracovali na registrácii nového používateľa a následnom vytvorení adresárov. Michal napísal php skript, ktorý tieto adresára vytvorí. Zatiaľ však majú ku nim prístup všetci. Potrebné je tieto adresáre zabezpečiť.

Pridanie nového balíčka bolo implementované ako vytvorenie novej wiki stránky a následného uploadu dist archívu. Tu narazili na problém, kde uploadovať súbory môže len používateľ s administrátorskými právami. To je momentálne len Karol. Je potrebné upraviť práva používateľov.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
41	Funkcie luadist.lib.php otestovať, resp. pre ne napísať unix skript, pre unix systémy.	Jakub Marton	9. 12. 2008
42	Nájdenie vhodného hostingu, ktorý by poskytoval OS shell.	Karol Marton	9. 12. 2008
43	Upraviť práva používateľom wiki, takže môžu uploadovať súbory.	Miloš Cibulka Michal Dávid	9. 12. 2008
44	Vytvorenie chýbajúcich tabuliek v databáze.	Karol Marton Marián Halaš	9. 12. 2008
45	Uloženie informácií o novom balíčku do db, uloženie balíčku do správneho adresára.	Karol Marton	16. 12. 2008

Zápisnica zo stretnutia č. 10

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
9.12.2008	18:00 - 19:30	Labss	Bc. Michal Dávid	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Predvedenie prototypu

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
41	Funkcie luadist.lib.php otestovať, resp. pre ne napísať unix skript, pre unix systémy.	Jakub Marton	9. 12. 2008	splnená
42	Nájdenie vhodného hostingu, ktorý by poskytoval OS shell. Projekt bude uložený na serveri u Ing. Drahoša.	Karol Marton	9. 12. 2008	splnená
43	Upraviť práva používateľom wiki, takže môžu uploadovať súbory.	Miloš Cibulka, Michal Dávid	9. 12. 2008	splnená
44	Vytvorenie chýbajúcich tabuliek v databáze.	Karol Marton, Marián Halaš	9. 12. 2008	splnená
45	Uloženie informácií o novom balíčku do db, uloženie balíčku do správneho adresára.	Karol Marton	16. 12. 2008	splnená

Priebeh stretnutia:

Stretnutie sa konalo v softvérovom štúdiu. Kým sme čakali na Ing. Drahoša Jakub doplnil hashovanie do názvov repo adresárov. Po príchode vedúceho sme mu predstavili náš prototyp a vysvetlili všetky potrebné funkcie. Až na malé chybičky bolo predstavenie prototypu úspešné bez vážnych výhrad. Ing. Drahoš akurát vysvetlil, čo ešte by tam malo byť. Konkrétne dorobiť výpis informácií o uploadovanom súbore, ktoré je potrebné vybrať z dist.info.

Následne Jakub riešil problémy s buildovaním wiki na labss serveri a problémy konzultoval s Ing. Drahošom. Následne sa zhodli, že daný postup je veľmi náročný a zdĺhavý a bude potrebné spraviť jemné zmeny v kóde. Hlavnou zmenou bude kompletne preprogramovanie generovanie dist.info nie cez exec funkciu ale priamo cez php a taktiež naprogramovať parser na dist súbor priamo v php.

Na záver nám Ing. Drahoš objasnil, čo je potrebné spraviť už na existujúcom prototypu a po jeho odchode sme si tieto úlohy rozdelili a ukončili sme stretnutie.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
46	Exec vymeniť za php parser. + generátor dist.info	Jakub Marton, Miloš Cibulka	1. 2. 2009
47	Upraviť dizajn wiki.	Michal Dávid	1. 2. 2009
48	Venovať sa hlbšiemu poznaniu luadistu.	Jakub Marton	1. 2. 2009
49	Rozdeliť úlohy na dokumentáciu k riadeniu a vytvorenie dokumentácie.	Marián Halaš	14. 12. 2008
50	Spraviť prezentáciu o prototypu.	Karol Marton	14. 12. 2008

Zápisnica zo stretnutia č. 11

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
25.02.2009	16:00 - 17:00	Labss	Bc. Karol Marton	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Plán na letný semester

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
46	Exec vymeniť za php parser. + generátor dist.info	Jakub Marton, Miloš Cibulka	1. 2. 2009	začatá
47	Upraviť dizajn wiki.	Michal Dávid	1. 2. 2009	začatá
48	Venovať sa hlbšiemu poznaniu luadistu.	Jakub Marton	1. 2. 2009	čiastočne splnená
49	Rozdeliť úlohy na dokumentáciu k riadeniu a vytvorenie dokumentácie.	Marián Halaš	14. 12. 2008	splnená
50	Spraviť prezentáciu o prototype.	Karol Marton	14. 12. 2008	splnená

Priebeh stretnutia:

Stretnutie sa konalo v softvérovom štúdiu po dlhšej pauze zapríčinenej skúškovým obdobím.

Hlavnou náplňou stretnutia bolo identifikovať úlohy, ktoré nás čakajú v letnom semestri. Na

úvod sa krátko zhodnotila práca v zimnom semestri. Ing. Drahoš poznamenal, že nemal žiadne vážnejšie nedostatky a že v letnom semestri sa máme viac orientovať na produkt ako na dokumentáciu. Ďalej sme debatovali o úlohách na najbližšie obdobie:

- Potreba plánu na letný semester – hlavne ako a kedy implementovať jednotlivé požiadavky
- Nájsť vhodné miesto na uloženie projektu na webe kvôli lepšiemu vývoju a testovaniu – Ing. Drahoš povedal, že o to sa postará on
- Upravenie dizajnu a kaskádových štýlov použitej wiki do želanej podoby
- Vytvoriť predgenerovanú wiki stránku pre práve pridaný balíček
- Vytvoriť parser pre .dist súbor
 - Je potrebné zistiť syntax tabuliek v Lua (Reference manual)
 - Možné použitie nejakého voľne dostupného PHP parsera na javascript
 - syntax je podobná systému kľúč – hodnota (angl. key - value) napr. externé závislosti (external dependencies)
- Vytvoriť generátor dist.manifest súboru – vysoká priorita

Po odchode pedagogického vedúceho sme si rozdelili úlohy a ukončili sme stretnutie.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
51	Exec vymeniť za php parser. + generátor dist.info	Jakub Marton	4. 3. 2009
52	Upraviť dizajn wiki – generovanie prednastavenej wiki stránky k novému balíčku	Michal Dávid	4. 3. 2009
53	Plán na letný semester	Karol Marton	4. 3. 2009

Zápisnica zo stretnutia č. 12

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
4.03.2009	15:00 - 16:00	Labss	Bc. Marián Halaš	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Implementácia funkcionality

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
51	Exec vymeniť za php parser. + generátor dist.info	Jakub Marton, Miloš Cibulka	4. 3. 2009	splnená
52	Upraviť dizajn wiki – generovanie prednastavenej wiki stránky k novému balíčku.	Michal Dávid	4. 3. 2009	začatá
53	Plán na letný semester	Karol Marton	4. 3. 2009	posunutá

Priebeh stretnutia:

Stretnutie sa konalo v softvérovom štúdiu. Hlavnou náplňou stretnutia bolo podať správu o napĺňaní identifikovaných úloh, ktoré nás čakajú v letnom semestri a dohodnúť sa na ďalšom postupe. Jakub Marton spravil funkčnú verziu parsera pre dist.info, ďalej sa bude venovať jeho optimalizácii pomocou Jason a Regexp. Ing. Drahoš nám zabezpečí webový server, kde budeme môcť náš produkt testovať. Miloš Cibulka dokončil skript na generovanie stránky pre nový balíček.

Treba ešte prerobiť dizajn stránok (možnosť použiť CSS zo stránky projektu LuaDist) a dorobiť funkcie na napĺňanie DB informáciami z dist.info a generovanie stránok z týchto

informácií (teda neparsovať znova balíček ale rovno použiť dáta z DB). Prioritou bude nasadiť produkt na testovanie na server, keď bude tento pripravený, dovedy vyvýjame na localhoste a prezentujeme z Jakubovho notebooku.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
53	Plán na letný semester	Karol Marton	11. 3. 2009
54	Pripraviť webový server pre nasadenie projektu	Ing. Drahoš	11. 3. 2009
52	Upraviť dizajn wiki	Míchal Dávid	11. 3. 2009
55	Generovanie dist.info z databázy	Miloš Cibulka	11. 3. 2009
56	Funkcia pre ukladanie dát z dist.info do DB	Karol Marton	11. 3. 2009

Zápisnica zo stretnutia č. 13

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
11.03.2009	15:00 - 16:00	Labss	Bc. Miloš Cibulka	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Implementácia funkcionality

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
	Dosiiahnuté výsledky, výstupy, poznámky			
53	Plán na letný semester	Karol Marton	11. 3. 2009	Splnená
54	Pripraviť webový server pre nasadenie projektu	Ing. Drahoš	11. 3. 2009	Posunutá
52	Upraviť dizajn wiki	Michal Dávid	11. 3. 2009	Čiastočne splnená
	Na stretnutí prebehli ďalšie doplňujúce pripomienky k úpravám.			
55	Generovanie dist.info z databázy	Miloš Cibulka	11. 3. 2009	Posunutá
56	Funkcia pre ukladanie dát z dist.info do DB	Karol Marton	11. 3. 2009	Posunutá

Priebeh stretnutia:

Stretnutie sa konalo v softvérovom štúdiu. Na úvod stretnutia sme prezentovali dosiahnuté výsledky našej práce a konzultovali ich s Ing. Drahošom. Karol Marton prezentoval náš plán na letný semester, Michal Dávid predviedol zmeny v dizajne stránok. Niektoré z nich budú ešte počas nasledujúceho týždňa inovované kvôli lepšej prehľadnosti. Miloš Cibulka predviedol generovanie stránok pre nové balíčky a upozornil na ďalšie

postupy, ktoré ohľadom generovania stránok ešte treba uskutočniť. Taktiež Ing. Drahoš doplnil aké ďalšie údaje sa budú automaticky na tieto stránky pridávať.

Našou hlavnou úlohou do ďalšieho stretnutia bude nasadiť projekt na virtuálny server, ktorý sprístupní Ing. Drahoš aby sme na nasledujúcich stretnutiach mohli prezentovať výsledky práce prostredníctvom neho. Popri tom bude pokračovať implementácia vkladania údajov o balíčkoch do DB a následne generovanie manifestu z týchto údajov.

Karol Marton taktiež informoval o novšej verzii wiki, ktorú jej tvorcovia vydali a rozhodli sme sa aktualizovať našu wiki touto novou verzou. Následne vznikla debata o potrebe changelog-u (záznamy zmien), do ktorého budeme zaznamenávať všetky zmeny, ktoré sme vykonali do zdrojového kódu wiki stránok, čím sa tento changelog stane základným návodom na budúce aktualizovanie wiki na novšie verzie. Zabráni sa tým prepisovaniu zmien, ktoré sme vo wiki urobili.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
54	Pripraviť webový server pre nasadenie projektu	Ing. Drahoš	18. 3. 2009
52	Upraviť dizajn wiki	Michal Dávid	18. 3. 2009
55	Generovanie dist.info z databázy	Miloš Cibulka	18. 3. 2009
56	Funkcia pre ukladanie dát z dist.info do DB	Karol Marton	18. 3. 2009
57	Prerobiť funkciu unzip na PHP	Jakub Marton	18. 3. 2009
58	Vytvoriť changelog	Marián Halaš	18. 3. 2009
59	Doplniť generovanie stránok o ďalšie údaje	Miloš Cibulka	18. 3. 2009
60	Aktualizovať wiki na novú verziu	Karol Marton	18. 3. 2009

Zápisnica zo stretnutia č. 14

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
18.03.2009	14:00 - 16:30	Labss	Bc. Jakub Marton	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Implementácia funkcionality

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
54	Pripraviť webový server pre nasadenie projektu	Ing. Drahoš	18. 3. 2009	Splnená
52	Upraviť dizajn wiki	Michal Dávid	18. 3. 2009	Splnená
55	Generovanie dist.info z databázy	Miloš Cibulka	18. 3. 2009	Posunutá
56	Funkcia pre ukladanie dát z dist.info do DB	Karol Marton	18. 3. 2009	Splnená
57	Prerobiť funkciu unzip na PHP	Jakub Marton	18. 3. 2009	Splnená
58	Vytvoriť changelog	Marián Halaš	18. 3. 2009	Posunutá
59	Doplniť generovanie stránok o ďalšie údaje	Miloš Cibulka	18. 3. 2009	Splnená
60	Aktualizovať wiki na novú verziu	Karol Marton	18. 3. 2009	Posunutá

Priebeh stretnutia:

Stretnutie sa konalo v softvérovom štúdiu. V priebehu prvej hodiny bolo nutné aktuálnu verziu wiki a databázy spustiť na notebooku Karola Martona, nakoľko externá databáza na serveri freshsql.com nebola dostupná v čase stretnutia. Všetko bolo v čase príchodu Ing. Drahoša a Jakuba Martona pripravené.

Na úvod nás ing. Drahoš informoval, že je pripravený hosting na luadist.org, kam umiestnime náš produkt. Databázu budeme riešiť tak ako doteraz, externe, s bezpečnostných dôvodov. Lokálna databáza na luadist.org bude používaná len na týždňové stretnutia a na testovanie. Následne Karol Marton predviedol aktuálnu verziu wiki, ktorá generuje stránky pre uploadnuté balíčky s ich základnými informáciami. Tieto informácie sú zapisované aj do databázy. Tu mal aj ing. Drahoš pripomienku ohľadom chýbajúceho zoznamu existujúcich balíčkov. Znova sa pootvorila otázka prístupu do používateľských repozitárov iným používateľom než autorovi. Táto otázka ostáva otvorená, v neskoršej fáze sa rozhodne, či sa bude bezpečnosť riešiť na úrovni *apache* servera. Rovnako ho zaujímal ďalší plán, ktorú funkcionálnosť chceme teraz implementovať. Podľa existujúceho plánu projektu sme si rozdelili úlohy: zlepšenie používateľského profilu, mail notifikácie autorov balíčkov pri ich zaraďovaní do stabilných, samotné zaraďovanie do stabilných, captcha, hodnotenie a komentáre balíčkov.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
55	Generovanie dist.info z databázy	Miloš Cibulka Jakub Marton	1. 4. 2009
58	Vytvoriť changelog	Marián Halaš	1. 4. 2009
60	Aktualizovať wiki na novú verziu	Karol Marton	1. 4. 2009
61	Presúvanie balíčkov do <i>stable</i> a <i>unstable</i> repozitárov	Karol Marton	1. 4. 2009
62	Zoznam balíčkov v repozitároch	Miloš Cibulka	1. 4. 2009
63	Hodnotenie balíčkov, komentáre	Michal Dávid	1. 4. 2009
64	Mail notifikácia, captcha	Marián Halaš	1. 4. 2009
65	Rozšírenie a editácia používateľského profilu	Jakub Marton	1. 4. 2009

Zápisnica zo stretnutia č. 15

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
25.03.2009	14:30 - 16:30	Labss	Bc. Michal Dávid	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Karol Marton

Téma stretnutia: Implementácia funkcionality

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
62	Zoznam balíčkov v repozitároch	Miloš Cibulka	1. 4. 2009	Čiastočne splnená
	Takmer dokončené, treba dorobiť filtre.			
61	Presúvanie balíčkov do <i>stable</i> a <i>unstable</i> repozitárov	Karol Marton	1. 4. 2009	Čiastočne splnená
55	Generovanie dist.info z databázy	Miloš Cibulka	1. 4. 2009	Začatá
63	Hodnotenie balíčkov, komentáre	Michal Dávid	1. 4. 2009	Čiastočne splnená
	Potrebné doladiť detaily.			
64	Mail notifikácia, captcha	Marián Halaš	1. 4. 2009	Čiastočne splnená
58	Vytvoriť changelog	Marián Halaš	1. 4. 2009	Čiastočne splnená
60	Aktualizovať wiki na novú verziu	Karol Marton	1. 4. 2009	Splnená
65	Rozšírenie a editácia používateľského profilu	Jakub Marton	1. 4. 2009	začatá

Priebeh stretnutia:

Stretnutie sa konalo v softvérovom štúdiu. Z dôvodu výpadku servera na FIIT bolo nutné presunúť wikku na súkromný web Karola Martona, keďže ju nebolo možné presunúť na hosting, ktorý zabezpečil Ing. Drahoš. Vyskytli sa však neočakávané problémy zapríčinené dočasným hostingovým serverom. Z tohto dôvodu nefungovali niektoré implementované funkcie.

Na úvod sme Ing. Drahošovi predstavili jednotlivé funkcie, ktoré boli doimplementované počas týždňa. Jednalo sa o captchu pri uploadovaní balíčka, hodnotenie balíčkov, page indexing uploadnutých balíčkov, publikovanie balíčkov z hidden verzie na unstable a zmena stavu balíčka z unstable na stable, ktorú môže vykonávať len administrátor. Po predstavení funkcií nám Ing. Drahoš vysvetlil svoje požiadavky, ktoré bude treba ešte do projektu dopracovať. Jednalo sa o vytvorenie automatického uploadera balíčkov na uľahčenie nahrávania väčšieho počtu balíčkov. Ďalej bude potrebné dopracovať filtre v indexe stránok o balíčkoch. Zobrazenie upozornenia adminovi pri uploadovaní balíčka s už existujúcim názvom. Upraviť vypisovanie modulov pre rôzne platformy (viditeľne zobrazit', že sa jedná o ten istý modul ale pre rôzne platformy). Na záver sme predstavili wikku aj Ing. Kapecovi a po rozdelení úloh sme stretnutie ukončili.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
55	Generovanie dist.info z databázy	Miloš Cibulka Jakub Marton	1. 4. 2009
58	Vytvorit' changelog	Marián Halaš	1. 4. 2009
61	Presúvanie balíčkov do <i>stable</i> a <i>unstable</i> repozitárov	Karol Marton	1. 4. 2009
62	Zoznam balíčkov v repozitároch	Miloš Cibulka	1. 4. 2009
63	Hodnotenie balíčkov, komentáre	Michal Dávid	1. 4. 2009
64	Mail notifikácia, captcha	Marián Halaš	1. 4. 2009
65	Rozšírenie a editácia používateľského profilu	Jakub Marton	1. 4. 2009
66	Dopísanie úvodnej stránky, systém schvaľovania, ako uploadnúť balíček.	Michal Dávid	1. 4. 2009
67	Automatické uploadovanie balíčkov.	Jakub Marton	1. 4. 2009
68	Do page indexu dorobiť komentáre o balíčku.	Karol Marton	1. 4. 2009
69	Dorobiť filtre do page indexu	Miloš Cibulka	1. 4. 2009
70	Zobrazovanie varovania pri rovnakom názve balíčka.	Karol Marton	1. 4. 2009

Zápisnica zo stretnutia č. 16

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
01.04.2009	14:30 - 15:30	Labss	Bc. Karol Marton	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Michal Dávid

Téma stretnutia: **Dokončovanie implementácie funkcionality**

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
62	Zoznam balíčkov v repozitároch	Miloš Cibulka	1. 4. 2009	splnená
61	Presúvanie balíčkov do <i>stable</i> a <i>unstable</i> repozitárov - presúvať sa bude no nových kategórií	Karol Marton	1. 4. 2009	zrušená
55	Generovanie dist.info z databázy	Jakub Marton	1. 4. 2009	splnená
63	Hodnotenie balíčkov, komentáre	Michal Dávid	1. 4. 2009	splnená
64	Mail notifikácia, captcha - e-mail notifikáciu je potrebné otestovať	Marián Halaš	1. 4. 2009	splnená
58	Vytvoriť changelog - zbytočná práca, postačujúci záznam z svn	Marián Halaš	8. 4. 2009	zrušená
65	Rozšírenie a editácia používateľského profilu	Jakub Marton	1. 4. 2009	splnená
66	Dopísanie úvodnej stránky, systém schvaľovania, ako uploadnúť balíček.	Michal Dávid	8. 4. 2009	začatá
67	Automatické uploadovanie balíčkov. - naprogramuje Ing. Drahoš	Jakub Marton	8. 4. 2009	zrušená
68	Do page indexu dorobiť komentáre o balíčku.	Karol Marton	1. 4. 2009	splnená

69	Dorobiť filtre do dist indexu	Miloš Cibulka	1. 4. 2009	zrušená
	- potrebné dorobiť textové pole na zadanie mena používateľa			
70	Zobrazovanie varovania pri rovnakom názve balíčka.	Karol Marton	1. 4. 2009	začatá
	- nové kategórie administrátorov tento problém mierne zľahčujú			

Priebeh stretnutia:

Stretnutie sa začalo referovaním splnených úloh, s ktorými vedúci vyslovil spokojnosť. Zároveň sme ho oslovili s malými požiadavkami na zmenu nastavení servera luadist.org/modules:

- Zmeniť práva webového servera, aby mohol pracovať so súbormi
- Zmeniť práva ftp, aby sa pri kopírovaní nastavovali potrebné práva (zatiaľ na 777)

Ďalej sa sám vyjadril k niektorým funkciám systému fungujúcim na školskom serveri, ktorý sám spravuje. Do nasledujúceho stretnutia by sme chceli mať dokončenú implementáciu aj s nasledovnými pripomienkami:

- Viditeľné balíčky pre prihlásených sprístupiť aj pre neprihlásených, a to aj na stiahnutie. Registráciu vyžadujeme iba pri nahratí nového balíčku.
- Vytvorenie špeciálnych skupín, do ktorých bude môcť používateľ nahráť a aktivovať (publish) svoj balíček. Takisto vytvoriť stránku pre administrátorov, kde sa takéto kategórie budú môcť vytvárať a vytvoriť administrátorské funkcie pre prácu s balíčkami (zmazanie, prekopírovanie do inej kategórie, hlásenie rovnakých balíčkov)
- Opraviť text pri nahrávaní súboru na server : choose dist.info → choose .dist file
- Mierne upraviť ostrosť gradientu v pozadí stránky

Ďalej bolo spomenuté, že programové riešenie automatického nahratia viacerých súborov bude vyriešené Ing. Drahošom. Pripomenul nám, že si máme nechať aj čas na potrebné testovanie balíčkov. Na záver sme si rozdelili úlohy.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
66	Dopísanie úvodnej stránky, systém schvaľovania, ako uploadnúť balíček – používateľská príručka	Michal Dávid	15. 4. 2009
70	Zobrazovanie varovania pri rovnakom názve balíčka.	Karol Marton	8. 4. 2009
71	Zobrazovanie hlasovania len na balíčkových stránkach.	Michal Dávid	8. 4. 2009
72	Zobrazovať posledne aktivované balíčky (recent published), ale aj všetky aktívne v dist index pre neprihlásených	Miloš Cibulka	8. 4. 2009
73	Špeciálne kategórie administrátorov, pridať admin	Miloš Cibulka	8. 4. 2009

	stránku na ich vytváranie	Jakub Marton	
74	Zmena gradientu pozadia na ostrejší	Michal Dávid	8. 4. 2009
75	Pridať textové pole na filter balíčkov, kde sa bude zadávať meno používateľa systému	Miloš Cibulka	8. 4. 2009
76	Opraviť generovanie odkazov na relatívne adresy	Jakub Marton	8. 4. 2009
77	Spísanie všetkých potrebných úloh a materiálov, ktoré je potrebné vypracovať k výslednému odovzdaniu	Marián Halaš	8. 4. 2009
78	Otestovanie posielania e-mailovej notifikácie	Marián Halaš	8. 4. 2009
79	Vymazanie balíčka a následné pregenerovanie manifestu z databázy	Jakub Marton	8. 4. 2009

Zápisnica zo stretnutia č. 17

Dátum	Čas	Miesto	Zapisovateľ	Prítomní
08.04.2009	16:00 - 17:00	Pracovňa ing. Drahoša	Bc. Marián Halaš	Ing. Peter Drahoš Bc. Miloš Cibulka Bc. Michal Dávid Bc. Marián Halaš Bc. Jakub Marton Bc. Michal Dávid

Téma stretnutia: **Dokončovanie implementácie funkcionality**

Pedagogický vedúci: Ing. Peter Drahoš

Členovia tímu: Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš
Bc. Jakub Marton
Bc. Karol Marton

Úlohy z predchádzajúceho stretnutia:

ID	Popis úlohy	Zodpovedný	Termín	Stav
Dosiahnuté výsledky, výstupy, poznámky				
66	Dopísanie úvodnej stránky, systém schvaľovania, ako uploadnúť balíček.	Michal Dávid	15. 4. 2009	začatá
Spraviť tak, aby to bolo možné použiť aj v dokumentácii ako používateľská príručka				
67	Automatické uploadovanie balíčkov. - naprogramuje Ing. Drahoš	Jakub Marton	8. 4. 2009	zrušená
68	Do page indexu dorobiť komentáre o balíčku.	Karol Marton	1. 4. 2009	splnená
69	Dorobiť filtre do dist indexu - potrebné dorobiť textové pole na zadanie mena používateľa	Miloš Cibulka	1. 4. 2009	zrušená
70	Zobrazovanie varovania pri rovnakom názve balíčka. - nové kategórie administrátorov tento problém mierne zľahčujú	Karol Marton	1. 4. 2009	splnená
71	Zobrazovanie hlasovania len na balíčkových stránkach.	Michal Dávid	8. 4. 2009	splnená
72	Zobrazovať posledne aktivované balíčky (recent published), ale aj všetky aktívne v dist index pre neprihlásených	Miloš Cibulka	8. 4. 2009	splnená
73	Špeciálne kategórie administrátorov, pridať admin stránku na ich vytváranie	Miloš Cibulka Jakub Marton	8. 4. 2009	splnená
74	Zmena gradientu pozadia na ostrejší	Michal Dávid	8. 4. 2009	splnená

75	Pridať textové pole na filter balíčkov, kde sa bude zadávať meno používateľa systému	Miloš Cibulka	8. 4. 2009	splnená
76	Opraviť generovanie odkazov na relatívne adresy	Jakub Marton	8. 4. 2009	splnená
77	Spísanie všetkých potrebných úloh a materiálov, ktoré je potrebné vypracovať k výslednému odovzdaniu	Marián Halaš	8. 4. 2009	splnená
78	Otestovanie posielania e-mailovej notifikácie	Marián Halaš	8. 4. 2009	posunutá
	Z dôvodu nenainštalovania funkcionality sendmail na serveri nebolo ako testovať			
79	Vymazanie balíčka a následné pregenerovanie manifestu z databázy	Jakub Marton	8. 4. 2009	splnená

Priebeh stretnutia:

Stretnutie sa premiestnilo z Labss do pracovne ing. Drahoša z dôvodu výpadku elektrického prúdu na celej fakulte. Preto nebolo možné demonštrovať implementovanú funkcionality. Preto sme len ústne zreferovali, že všetky podstatné veci už sú naimplementované podľa nových požiadaviek (kategórie ako špeciálni používatelia atď., pozri úlohy z minulého stretnutia)

Ing. Drahoš nám nakoniec dal za úlohu spraviť automatizované nahrávanie balíčkov na server z lokálneho repozitára. Potom sa už budeme sústreďovať na testovanie a prípadné odchyty chýb.

V testovaní funkcionality sa vyskytli problémy s nedostatočnými právami na serveri, bez týchto práv nie je napríklad možné vytvárať repozitáre a tým pádom je de facto nefunkčná celá wiki. Toto nám sľúbil vyriešiť Ing. Drahoš už minulý týždeň. Takisto je ešte potrebné na server doinštalovať modul *sendmail* kvôli emailovým notifikáciám.

Ing. Drahoš nám ešte odporučil, aby sme po beta otestovaní spravili snapshot wiki a takisto databázy (po nahraní už existujúcich balíčkov), ktorý bude mať vlastne úlohu výsledného produktu, ktorý by sme mali odovzdať na tomto predmete.

Na záver sme si rozdelili úlohy.

Úlohy do nasledujúcich stretnutí:

ID	Popis úlohy	Zodpovedný	Termín
66	Dopísanie úvodnej stránky, systém schvaľovania, ako uploadnúť balíček – používateľská príručka	Michal Dávid	15. 4. 2009
78	Otestovanie posielania e-mailovej notifikácie	Marián Halaš	8. 4. 2009
80	Spraviť utilitu na hromadný upload balíčkov na stránku cez http	Jakub Marton	15.4.2009
81	Poslať Ing. Drahošovi mail aby nastavil práva a doinštaloval sendmail na serveri	Karol Marton	15.4.2009
82	Vytvoriť SNAPSHOT wiki a databázy po ukončení testovania	Karol Marton	22.4.2009

Posudok analýzy, špecifikácie a návrhu tímu č. 20

Obsah

<i>1 Úvod.....</i>	<i>3</i>
<i>2 Obsah dokumentu.....</i>	<i>3</i>
<i>3 Hodnotenie obsahu.....</i>	<i>3</i>
<i>4 Formálna stránka.....</i>	<i>4</i>
<i>5 Záver.....</i>	<i>4</i>

1 Úvod

Dokument obsahuje posudok analýzy, špecifikácie a návrhu tímu č. 20. Dokument je posudzovaný z hľadiska obsahu, formy, súladu s požiadavkami a celkového dojmu. V závere dokumentu je celé hodnotenie textovo aj graficky zosumarizované.

2 Obsah dokumentu

Úvod dokumentu opisuje problematiku tvorby rozvrhov, zložitosť tohto problému vzhľadom k rôznym faktorom, podáva obraz o tom z akého stavu tím vychádza keďže nadväzuje na projekt z predchádzajúcich rokov, hlavne sa spomína práca tímu Fénix.

V kapitole 3 – analýza procesu tvorby rozvrhov sa tím podrobnejšie zaoberá jednotlivými aspektmi problematiky tvorby rozvrhov, kapitola je tematicky rozdelená na dve časti. Prvá pojednáva o tvorbe semestrálnych rozvrhov a druhá zase o tvorbe rozvrhov skúšok. V oboch častiach sa tím sústreďí na zber informácií o systéme. Je tu aj časť s úvahou o prepojení s akademickým informačným systémom, v tomto smere však autori vidia viacero prekážok. Nakoniec v tejto kapitole definujú jednotlivých hráčov systému vychádzajúc z už existujúceho rozdelenia ale s istou mierou prispôsobenia.

V kapitole 4 s názvom Analýza existujúceho systému je rozoberaný existujúci systém z predchádzajúcich rokov, tím analyzuje existujúce funkcie a technológie a pridáva alebo odoberá novú funkcionálnosť s ohľadom na zmenené podmienky od minulých rokov.

Funkcionálnosť systému následne špecifikuje kapitola 5, ktorá obsahuje okrem funkcionálnych požiadaviek aj nefunkcionálne. Funkcionálne požiadavky sú rozdelené podľa jednotlivých hráčov systému a následne uvedené v prehľadnej forme usecase diagramov.

Kapitola s návrhom je čo do obsahu zastúpená návrhom obrazoviek a formulárov pre zber údajov a ich prehľadné zobrazenie ako prioritu, ktorú si tím zvolil pre svoj projekt.

3 Hodnotenie obsahu

Dokument je veľmi dobre členený po obsahovej stránke, podrobná analýza problematiky tvorby rozvrhov je prehľadná a vhodne usporiadaná, čitateľovi je jasný problém.

Kapitola analýza existujúceho systému dobrým spôsobom uvádza do východiskového bodu tímu a jednoznačne určuje jeho štartovaciu pozíciu. Takisto je jasné, že systém podrobne zanalyzovali a pokúsili sa objaviť a nahradiť všetky jeho slabiny.

Tím si stanovuje jednoznačný cieľ projektu v rámci rozsahu tohto projektu – venujú sa hlavne časti systému pre zber a prehľadné zobrazenie dát. Je evidentné, že tím vie čo chce a v celom dokumente k tomu smeruje, v špecifikácii požiadaviek sa preto orientujú hlavne na túto oblasť takže to, že iné aspekty problematiky tvorby rozvrhov zostali neriešené nie je negatívom, naopak, určite je lepšie ak máme perfektne fungujúcu časť systému ako slabo, alebo vôbec nefungujúci systém.

V prílohe A sa nachádza konfliktná tabuľka, ktorej význam však nie je nikde opísaný a preto nie je jasné na čo presne slúži, resp. aký má presný súvis s navrhovaným systémom.

Možno by bolo vhodné v návrhu alebo v prílohe priložiť aj dátový model systému ako jednu z jeho veľmi podstatných častí.

4 Formálna stránka

Dokument je ucelene formátovaný a prehľadne členený. Formátovanie je jednotne dodržiavané v celom dokumente, kapitoly sú číslované, text členený do odstavcov. Obrázky vhodne dopĺňajú rozoberané problémy. Kapitola s odkazmi na použitú literatúru by však skôr mala byť na konci dokumentu. Takisto na konci dokumentu chýba záver.

Vo viacerých častiach dokumentu sa vyskytovali gramatické chyby, hlavne v kapitole 3.5. Obrázok č.1 ukazuje prepojenie servera na databázu cez JDBC avšak v texte pod ním je spomenuté, že technológia je PHP v kombinácii s PostgreSQL.

Funkcionálne požiadavky by mohli byť v prehľadnejšej forme, napríklad v tabuľkách. Spôsob zoznamu však tiež nie je zlý a neprehľadný.

5 Záver

Dokument jednoznačne preukazuje porozumenie tímu č.20 rozoberanej problematike, dostatočne rozsiahla analýza a presná špecifikácia nasledované prehľadným návrhom sú kompaktné a predstavujú dobrý východiskový bod pred prototypovaním a implementáciou projektu. Pre ucelenejší prehľad by bolo teda ešte vhodné doplniť dátový model a odstrániť formálne nedostatky spomenuté v kapitole 4.



SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4



Posudok na dokumentáciu tímu č. 18

System pre správu rozširujúcich modulov

jazyka Lua

Tímový projekt

Autori dokumentácie:

Tím č. 18

Bc. Karol Marton
Bc. Jakub Marton
Bc. Miloš Cibulka
Bc. Michal Dávid
Bc. Marián Halaš

Autori posudku:

Tím č. 20

Bc. Michal Drahoš
Bc. Jozef Grexa
Bc. Radoslav Halama
Bc. Roman Herbst
Bc. Tomáš Kramár
Bc. Rastislav Masaryk

Školský rok: 2008/2009

Obsah

1 Úvod	3
2 Formálna stránka	4
2.1 Forma dokumentov	4
2.2 Štylistika.....	5
3 Obsahová stránka	6
3.1 Dokumentácia k softvérovému systému	6
3.1.1 Analýza.....	6
3.1.2 Špecifikácia	6
3.1.3 Návrh systému.....	7
3.2 Dokumentácia k riadeniu projektu	8
4 Zhrnutie	8

1 Úvod

Cieľom dokumentu je posúdenie dokumentácie pre prvú etapu práce na tímovom projekte tímu č.18 „Luazeri“ v zimnom semestri. Posudzovaná dokumentácia obsahuje viacero kapitol, my však posudzujeme len tie najpodstatnejšie (analýza, špecifikácia, hrubý návrh architektúry). Dokumentácia je posúdená z pohľadu formálneho i pohľadu obsahového. V tomto posudku sa snažíme zhodnotiť celkový dojem nadobudnutý po prečítaní dokumentácie, nerozoberáme podrobne všetky detaily.

2 Formálna stránka

2.1 Forma dokumentov

Dokumentácia tímu č. 18 pôsobí na prvý pohľad uceleným dojmom. Autori oddelili samotnú *dokumentáciu vývoja systému* od *dokumentácie riadenia projektu*. Dokument o vývoji systému je rozdelený do kapitol podľa uvedených pokynov, z čoho je jasná celková štruktúra dokumentu.

Tím 18 sa snažil dodržať všetky dôležité pravidlá pre grafickú úpravu textu, pre členenie a zvýraznenie kapitol a podkapitol. V kapitole 3. *Špecifikácia*, na jej začiatku, však zabudli členovia zvýrazniť text „Funkcionálne požiadavky“, ktorý má tvoriť nadpis podkapitoly. Táto kapitola teda nie je uvedená ani v obsahu dokumentu.

Podľa nášho názoru by podkapitola 3.1. *Prípady použitia* by bola prehľadnejšia, keby bola členená na podkapitoly jednotlivých hráčov a ich prípadov použitia.

V dokumente sa ďalej nachádzajú chyby, ktoré vznikli pravdepodobne transformáciou dokumentácie z .doc formy do .pdf formy a následným neskontrolovaním vygenerovaného pdf súboru. Jedná sa najmä o

- zlé zarovnanie textu v tabuľkách (tabuľka 5.)
- veľa bielych znakov (tabuľka 5.)
- chýbajúce oddelenia stĺpcov v tabuľke čiarami (tabuľka 5.)
- veľa prázdnych riadkov pred a po kapitolách/podkapitolách/odsekoch.

Obrázky a tabuľky sú značené konzistentne a podľa pravidiel. Tabuľky sa síce nachádzajú vo vhodnej vzdialenosti od ich referencie, avšak k obrázkom vo väčšine prípadov chýba odvolávka v texte, čo spôsobuje mierny chaos hlavne v kapitole návrhu systému, kde sú obrázky hojne využívané. Tiež sme si všimli, že niektoré ilustračné obrázky sú už na hranici čitateľnosti, preto odporúčame väčšie obrázky radšej otočiť na výšku a rozťahnuť na celú stránku. Takisto je vidno, že na import obrázkov bol použitý stratový formát, ktorý znižuje výslednú kvalitu, preto odporúčame použiť formát s možnosťou bezstratovej kompresie (napr. png). Pre sprehľadnenie textu by možno bolo dobré uvážiť presun niektorých obrázkov do príloh.

Zistené nedostatky sú minoritného charakteru a preto nie je možné odoprieť snahu o vynikajúcu formálnu stránku dokumentu. Dôvodom nedostatkov je podľa nás fakt, že dokumentácia bola integrovaná z viacerých častí, kde každú napísal iný človek s vlastným

štýlom a taktiež k neskontrolovaniu formálnej stránky vygenerovaného pdf súboru, ktorý sme dostali k dispozícii na posudok.

2.2 Štylistika

Po stránke štylistiky to už je však horšie. Autori sa nevyhli bežným chybám ako sú preklepy, vynechané znaky, malé písmená na začiatku viet, nesprávne použitie čiarok a pod. Napríklad v kapitole 3. *Špecifikácia* to je „Možnosť zaregistrovať sa v systémy“ namiesto správneho „Možnosť zaregistrovať sa v systéme“. Takýchto prípadov bolo viac a nachádzali sa po celom dokumente vrátane ilustračných obrázkov.

Taktiež v niektorých prípadoch členovia tímu zrejme nemali presne dohodnutý spôsob označovania pojmov a tak sa stalo, že používateľ systému bol napísaný ako používateľ, ale na inom mieste to už bol užívateľ, inde zase „user“. Aj keď je zrejmé, že tieto slová majú rovnaký význam, bolo by vhodnejšie pridržať sa jednotného pomenovania, a to slovenského „používateľ“ (užívateľ je človek, čo niečo užíva, používateľ používa).

Veľmi častou gramatickou chybou však bolo nesprávne skloňovanie prídavných mien/zámen v nominatíve množného čísla mužského rodu (t.j. nesprávne použitie –ý namiesto –í). Najčastejšie sa táto chyba vyskytovala spoločne s predchádzajúcou chybou, napr. „...dostupný pre všetkých **užívateľov, ktorý** ho potom môžu využívať“, „...**registrovaný užívateľa**...“.

Veľmi rušivou chybou je časté používanie anglických výrazov, ktorých tvar sa na niektorých miestach autori snažia dokonca ohýbať (napr. na obr.č.9 „publishni“, v časti Prípady použitia „workflow-y“). Zarážajúci je hlavne fakt, že pre väčšinu výrazov existuje regulárny slovenský ekvivalent, ktorý navyše autori aj v dokumente na niektorých miestach (nevedomky) použili.

Spomínané nedostatky sa nachádzajú prakticky v celom dokumente, preto hodnotíme štylistickú časť negatívne. Celkovo sa nám zdá, ako keby autori písali dokumentáciu na poslednú chvíľu, pričom si ju po sebe ani neprečítali a opravu chýb zverili automatickým opravám v textovom editore.

3 Obsahová stránka

Celková dokumentácia skladá z častí: dokumentácia k vývoju systému a dokumentáciu k riadeniu, našou úlohou je posúdenie oboch dokumentácií.

Hneď na začiatku sme si všimli históriu vývoja dokumentu. Podľa nášho názoru je história z tohto uhla pohľadu internou záležitosťou tímu. Pokiaľ by sa jednalo o históriu zmien a opráv vo finálnom dokumente (po uvedení systému do prevádzky), tak by to bolo v poriadku, ale v tomto tvare si myslíme, že je úplne zbytočné ju tam uvádzať, a teda buď ju presunúť do dokumentácie k riadeniu alebo úplne odstrániť.

Takisto sme si všimli, že v dokumente sa používajú ilustračné obrázky zobrazujúce diagramy činností, ktoré však nie sú vytvorené pomocou štandardnej UML notácie. Preto odporúčame na začiatok dokumentu doplniť a vysvetliť použitú notáciu.

3.1 Dokumentácia k softvérovému systému

3.1.1 Analýza

Kapitola analýzy je prehľadne a podrobne spracovaná. Nezainteresovanému čitateľovi poskytuje okamžitý a ucelený prehľad o probléme.

Jednotlivé kľúčové pojmy (Lua, LuaDist...) sú vysvetlené v takom poradí, aby boli čo najjednoduchšie pochopiteľné, čo napomáha ich postupnému porozumeniu a celkovej prehľadnosti.

Vysvetlenie kľúčových pojmov a procesov je dostatočné aj vďaka použitým obrázkom a diagramom.

Chýba nám akási motivácia, prečo vlastne vytvárať takýto systém, a taktiež väčšie odôvodnenie, prečo sa zaoberali práve wiki systémami a nie napríklad nejakými inými.

Následná analýza jednotlivých wiki systémov je dostatočná a je cielene zameraná najmä na skúmanie vhodnosti pre použitie na danom projekte.

Pre každú možnosť sú podrobne rozpísané jej výhody a nevýhody, čo napomáha vytvoreniu si lepšej predstavy a taktiež najmä rozhodnutiu o výbere najvhodnejšej možnosti, ktorá bude implementovaná.

3.1.2 Špecifikácia

Kapitola sa zoberá špecifikáciou požiadaviek na vyvíjaný systém. Funkcionálne požiadavky sú spracované v dostatočnej miere, no v kapitole nám chýba aspoň naznačenie nefunkcionálnych požiadaviek.

V špecifikácii sú ďalej uvedené diagramy prípadov použitia. Diagramy sú prehľadné a prepracované, keďže nezachytávajú len vrcholové „use case“. Následne jednotlivé popisy prípadov použitia dávajú základnú informáciu o ich činnosti.

Po obsahovej stránke je kapitola napísaná dobre, formálne a štylistické chyby však kapitolu mierne zneprehľadňujú.

3.1.3 Návrh systému

Táto kapitola popisuje hrubý návrh riešenia systému. Kapitola je dostatočne obsiahla, vysvetľuje návrh systému z hľadiska činností vykonávaných v systéme, používateľského rozhrania a logického modelu údajov. V každej časti sú k textom doplnené aj ilustračné obrázky a diagramy, čo napomáha k pochopeniu celého návrhu.

Autori začali opis systému pomocou „workflow“ diagramov. V tejto časti sú podrobne popísané všetky možnosti pre jednotlivé kategórie používateľov. Diagramy sú po obsahovej stránke veľmi dobre spracované. Z tohto hľadiska nie je autorom čo vytknúť. Ďalej pokračoval opis činností pri jednotlivých prípadoch použitia. Opäť musíme pochváliť autorov za dobré vysvetlenie jednotlivých činností, ku ktorým sú takisto pripojené ilustračné diagramy. Tieto diagramy však odporúčame prerobiť na diagramy s partíciami (angl. aj tzv. swim-lanes), v ktorých bude jasne vidieť, kto vykonáva danú činnosť (systém, administrátor, používateľ,...).

V ďalšej časti sa autori venovali používateľskému rozhraniu, konkrétne návrhu obrazoviek. V tejto časti taktiež nie je čo autorom vytknúť. Obrazovky sú navrhnuté logicky a konzistentne. V tejto fáze návrhu neočakávame, že by autori mali dokonale premyslené grafické spracovanie systému, preto hodnotíme len obsahovú časť návrhu obrazoviek.

Nasledujúca časť sa zaoberá logickým a fyzickým modelom údajov. Je zrejmé, že fyzický model údajov by mal vychádzať z logického. Táto časť na nás však už nepôsobila tak prepracované ako predchádzajúce časti, pritom dátový model je jedna z najdôležitejších vecí pri tvorbe systému. Chýbali nám hlavne popisy jednotlivých atribútov entít. Ďalej fyzický model nekorešpondoval s logickým. Jedná sa najmä o vzťahy medzi entitami, kde vo fyzickom modeli jednak niektoré vzťahy úplne vypadli a niektoré zmenili kardinalitu. Nevieme, ako to autori mysleli, pretože chýba akékoľvek vysvetlenie. Nakoniec nám ešte chýbalo znázornenie referenčnej integrity vo fyzickom modeli. Túto časť odporúčame ešte prepracovať.

3.2 Dokumentácia k riadeniu projektu

V skratke, dokumentácia riadenia projektu spĺňa požiadavky na ňu kladené. Obsahuje všetky potrebné časti ako: *Ponuka, Plán projektu, Úlohy členov tímu a Zápisnice zo stretnutí*.

Po formálnej stránke sú tieto dokumenty v poriadku. Obsahovou stránkou sme sa podrobnejšie nezaoberali.

4 Zhrnutie

Posudzovaný dokument pôsobí aj napriek menším chybám, ako ucelený a detailne spracovaný dokument, plne poskytujúci informácie, čo a ako sa bude vytvárať.

V dokumente nám chýba akási motivácia, prečo vlastne vytvárať takýto systém. Fázy analýza, špecifikácia a návrh sú rozsahom dostačujúce. Nami zistené chyby a nedostatky by mali autorom poslúžiť na zlepšenie dokumentácie a tým aj celkovej úrovne projektu. Odporúčame hlavne autorom dôkladne celý text ešte raz prečítať a opraviť gramatické a štylistické chyby, ktoré zbytočne kazia celkový dojem z inak kvalitnej dokumentácie. Na základe prečítaného textu je zrejmé, že tím č. 18 má preštudovanú oblasť problematiky a premýšľal ako riešiť jednotlivé problémy.

Vyjadrenie k posudku tímu 20 na dokumentáciu analýzy, návrhu a špecifikácie tímu č. 18.

Autori posudku prehľadne a podrobne opísali vnímané klady, ale aj zápory dokumentácie analýzy, špecifikácie a návrhu vytváraného systému. V dokumente nám vyčítali hlavne časté pravopisné a štylistické chyby. Tie plynuli z toho, že na každej časti pracoval niekto iný a preto bolo ťažké zosúladiť nielen slovník pojmov, ale aj spomínané chyby, ktoré sa vyskytnú v každom dokumente. V budúcnosti tomu budeme venovať zvýšenú pozornosť. Tímu 20 chýbalo vysvetlenie motivácie pre výber wiki systému pre výsledný produkt, čo je však uvedené v dokumentácii na strane 9 v kapitole 2.3. Wiki systémy hneď na začiatku. Ďalšie výhrady smerovali k dátovému modelu údajov. Nekonzistentnosť logického a dátového modelu vznikla zlým využívaním aplikácie na tvorbu diagramov a neskontrolovaním vygenerovaného diagramu. Všetky zistené nedostatky sme sa snažili odstrániť hneď v ďalšej verzii dokumentu. Celkovo posudok hodnotíme veľmi kladne, pretože nám priniesol objektívny pohľad na dokument so všetkými pozitívami, ale aj negatívami, o ktorých sme dovtedy nevedeli. Všetky zistené nedostatky sa budeme snažiť čo najskôr odstrániť.