

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.

Časť I. - Informačný systém

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 |

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 | 47 |
| 5 | Prototyp | 49 |
| | Funkcionalita prototypu | 49 |

| | | |
|------------|-------------------------------|-----------|
| 5.1 | Zmeny v dátovom modeli | 49 |
| 5.2 | Upload súboru | 50 |
| 6 | Záver | 52 |
| 7 | Použitá literatúra | 53 |

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 rozšíriteľnou 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.

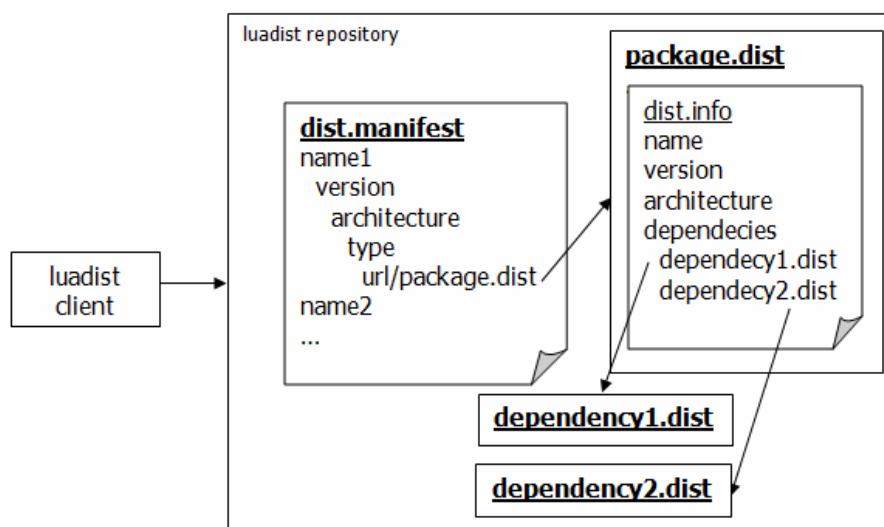
2.1.1 Hlavné výhody jazyka lua

Lua sa využíva hlavne v hernom odvetví. Napríklad vo videohrách je bežné, že multimedialne 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.

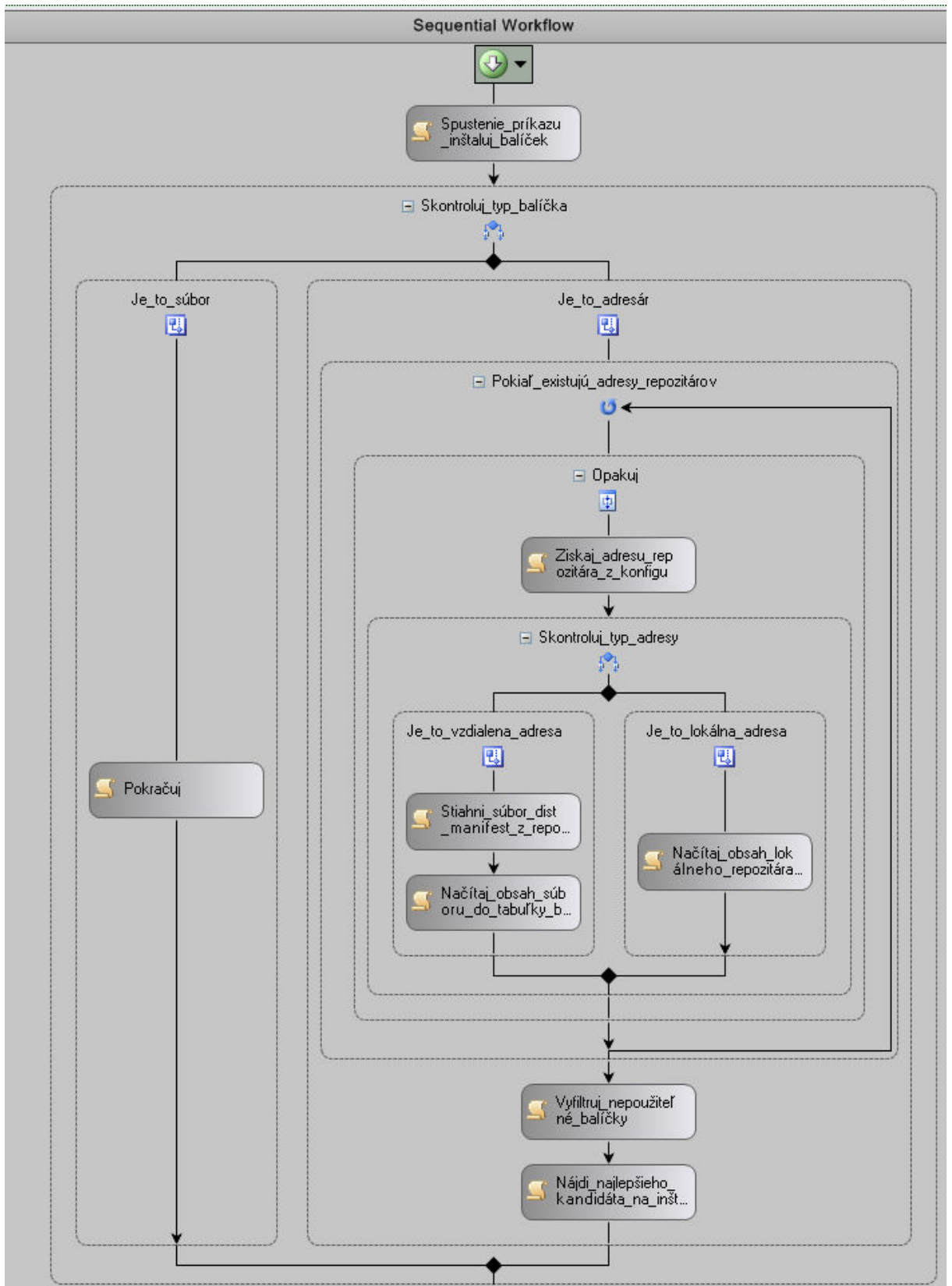
2.2.1 Inštalácia modulu

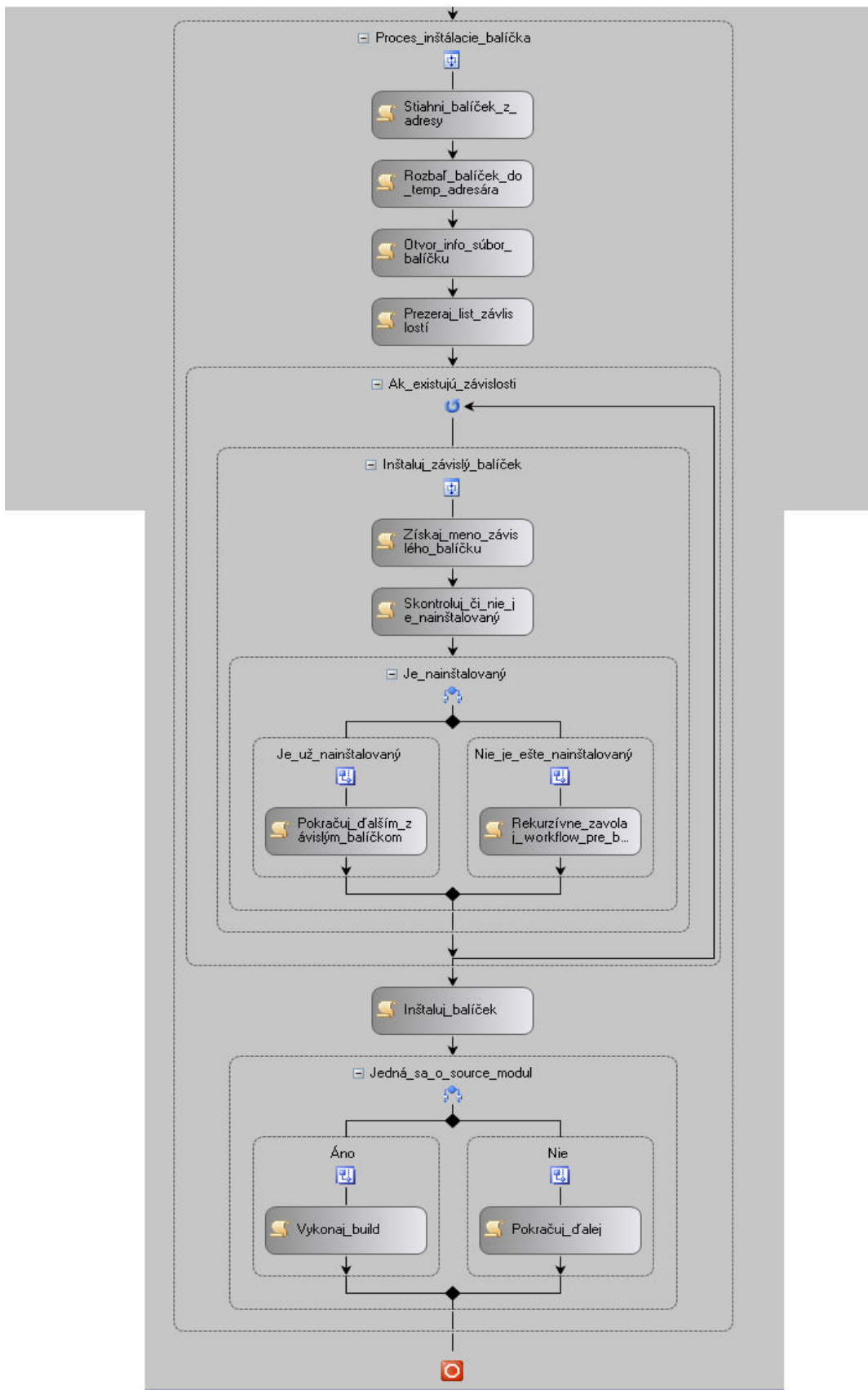
Následný diagram činností na obrázku č. 2 ukazuje, ako sa správa LuaDist počas inštalácie modulu a ako pri tomto procese využíva repozitáre. Inštaláciu ľubovoľného modulu začína užívateľ na svojom osobnom počítači jednoduchým príkazom „luadist install meno_modulu“.

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 vyhľadani 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.

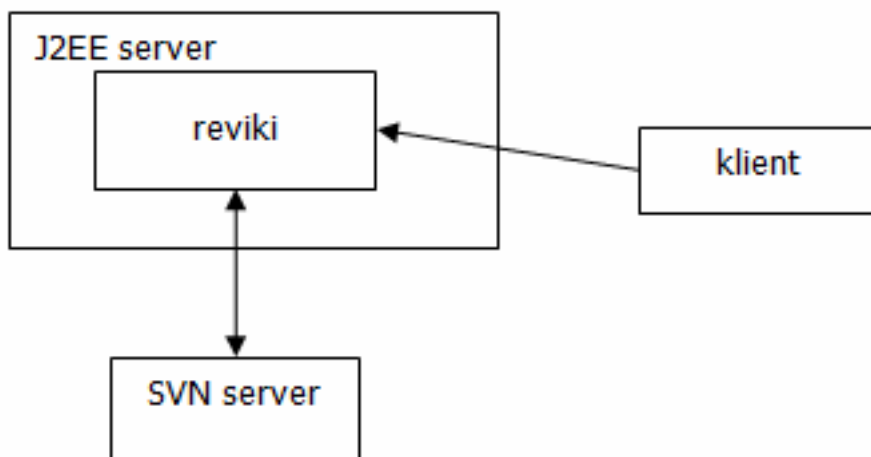
Tab. č. 2 - Výhody a nevýhody wiki systému Nanoki

| 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 uberá 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 technológii, • 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

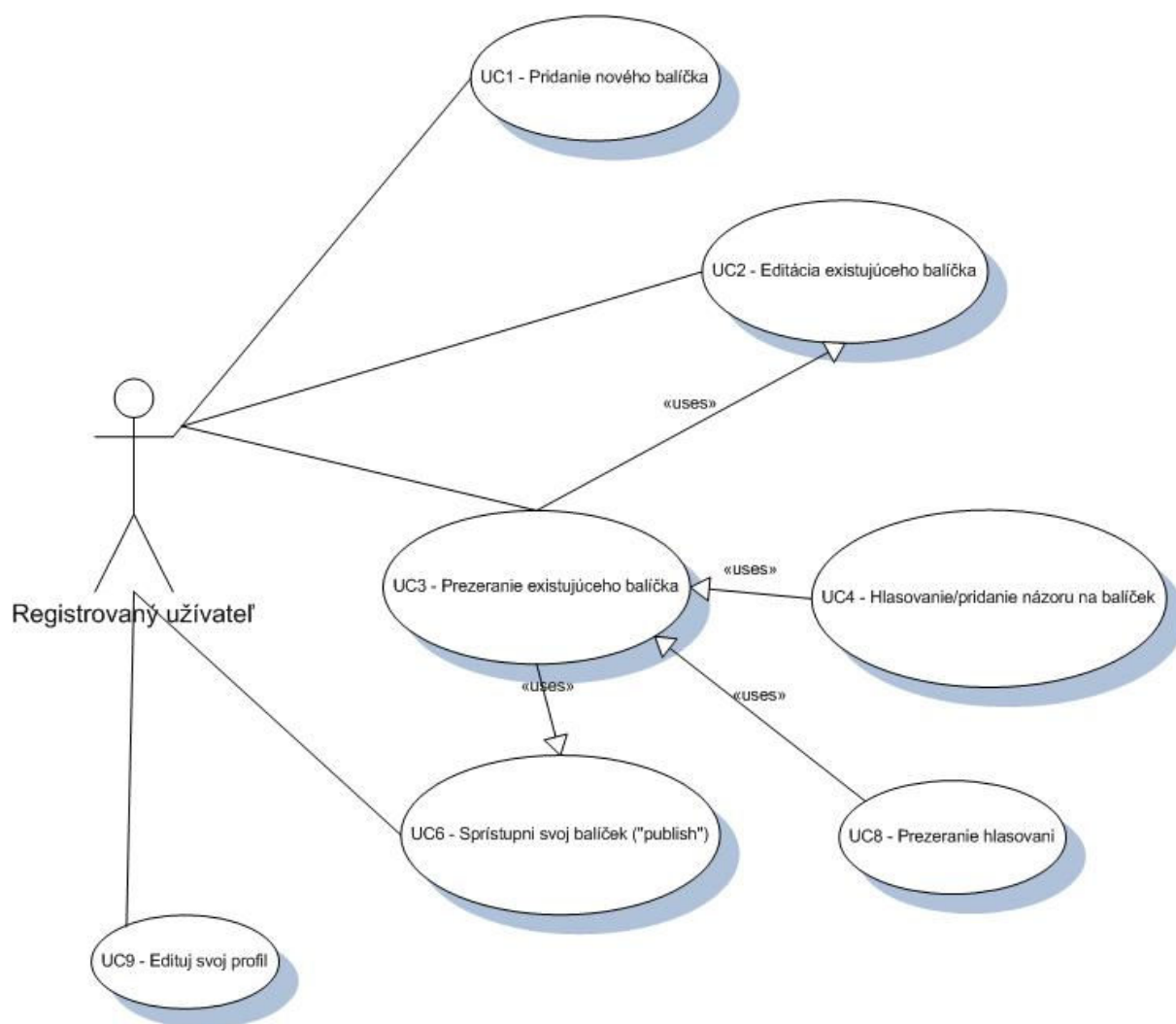
| Funkcia | Opis | Používateľ |
|---|--|--------------------------------------|
| Autentifikácia používateľa | na úvodnej stránke je formulár na vstupné údaje – prihlasovacie meno a heslo | všetci |
| Registrácia nového používateľa | na úvodnej stránke bude odkaz na registračný formulár | všetci |
| 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 |

| | | |
|--|---|--------------------------------------|
| Prezeranie konkrétneho balíčka <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 položkach alebo hľadať 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.

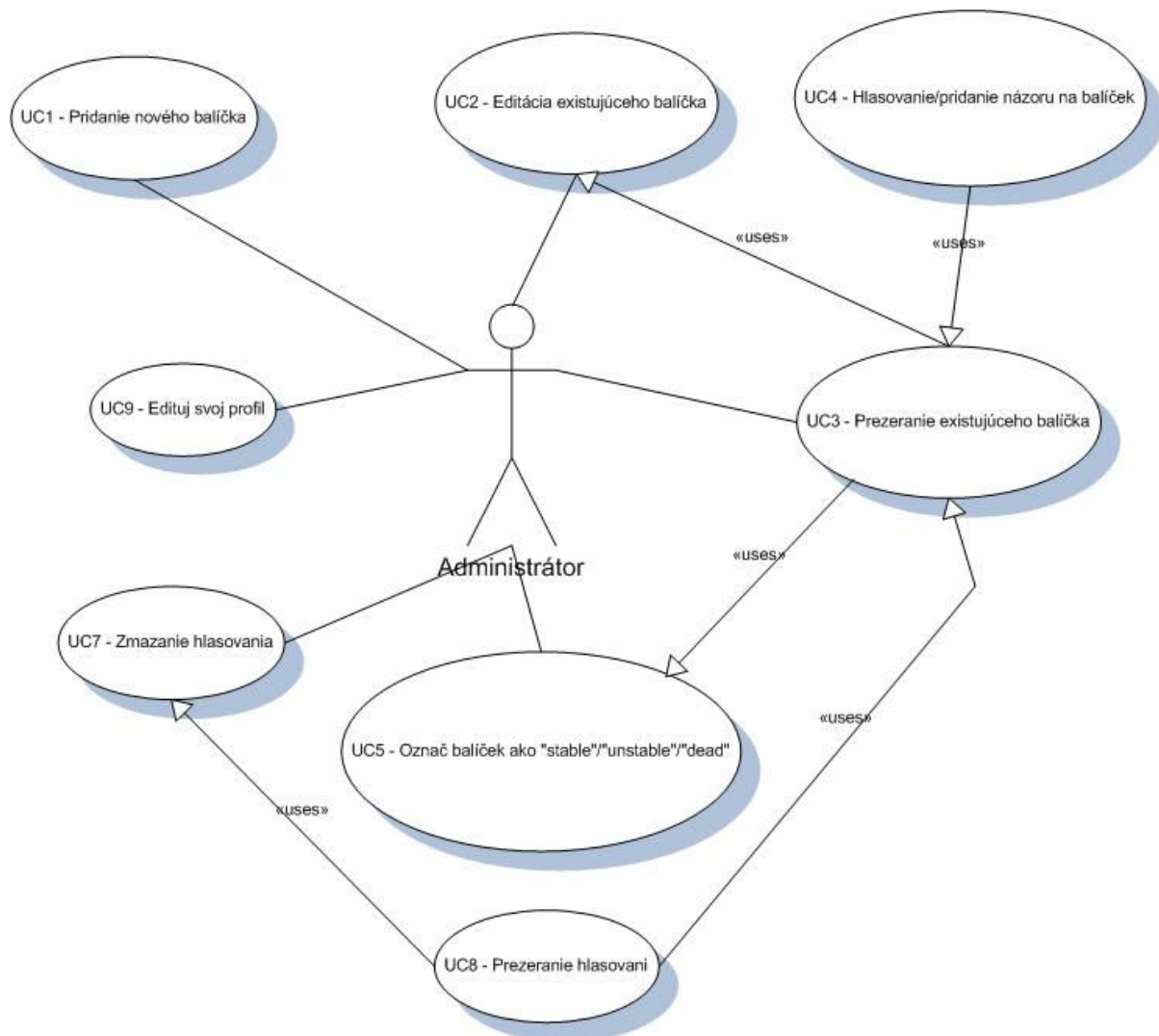


Obr. č. 4 Diagram prípadov použitia pre registrovaného užívateľa.

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

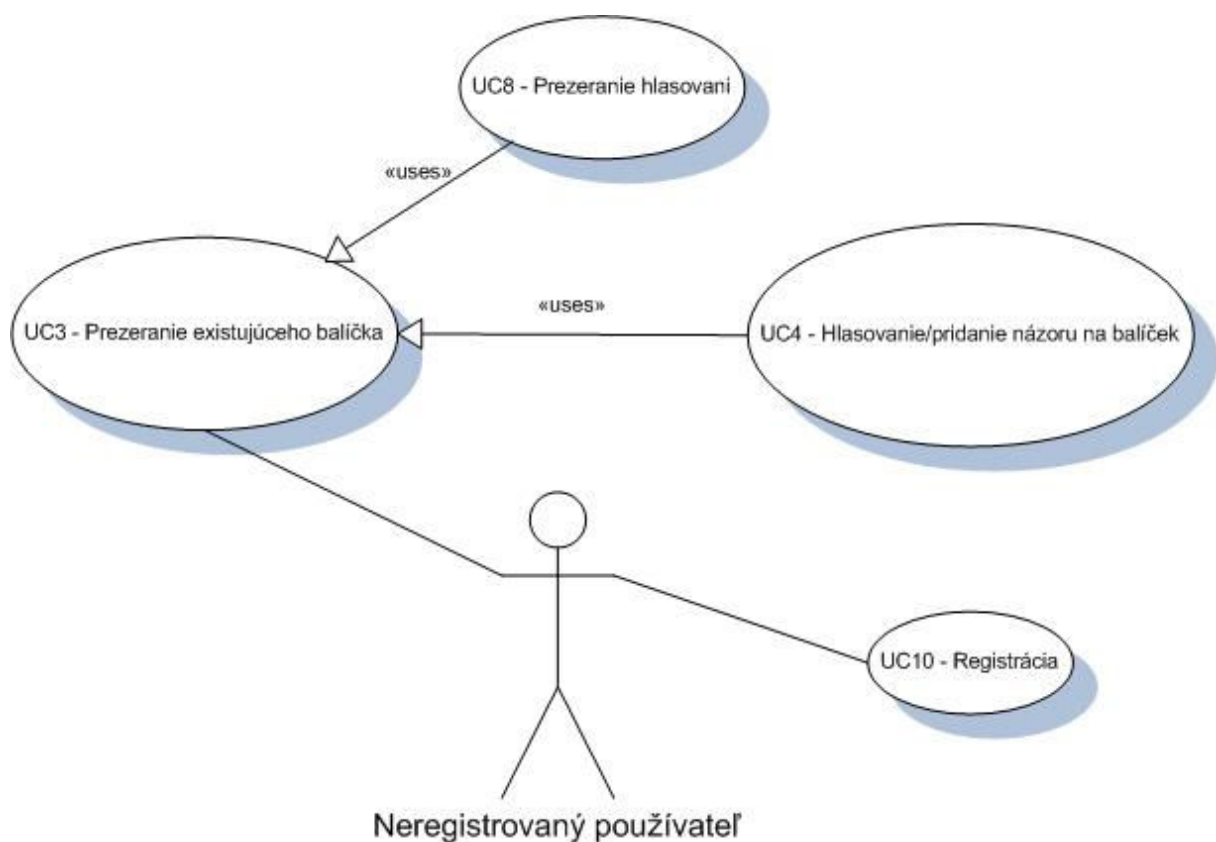


Obr. č. 5 Diagram prípadov použitia pre administrátora.

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, ktorý 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.

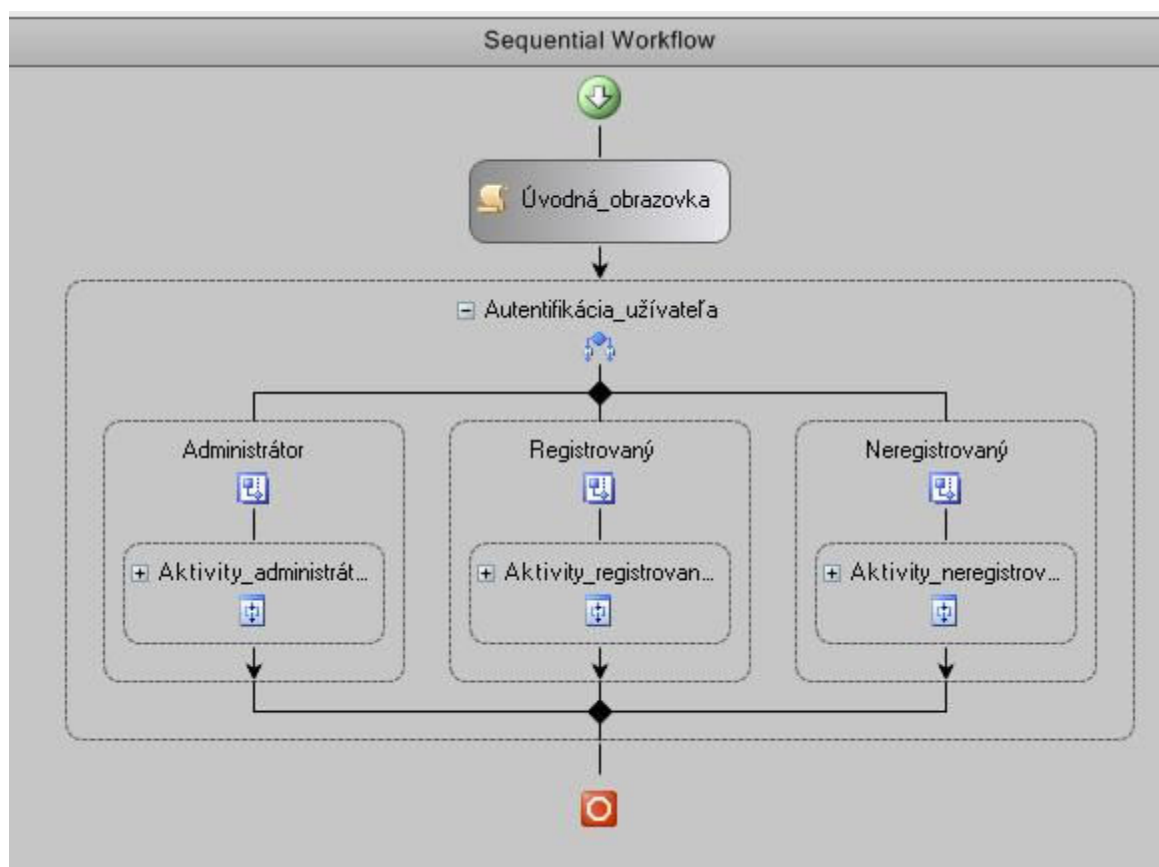
4 Návrh

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.

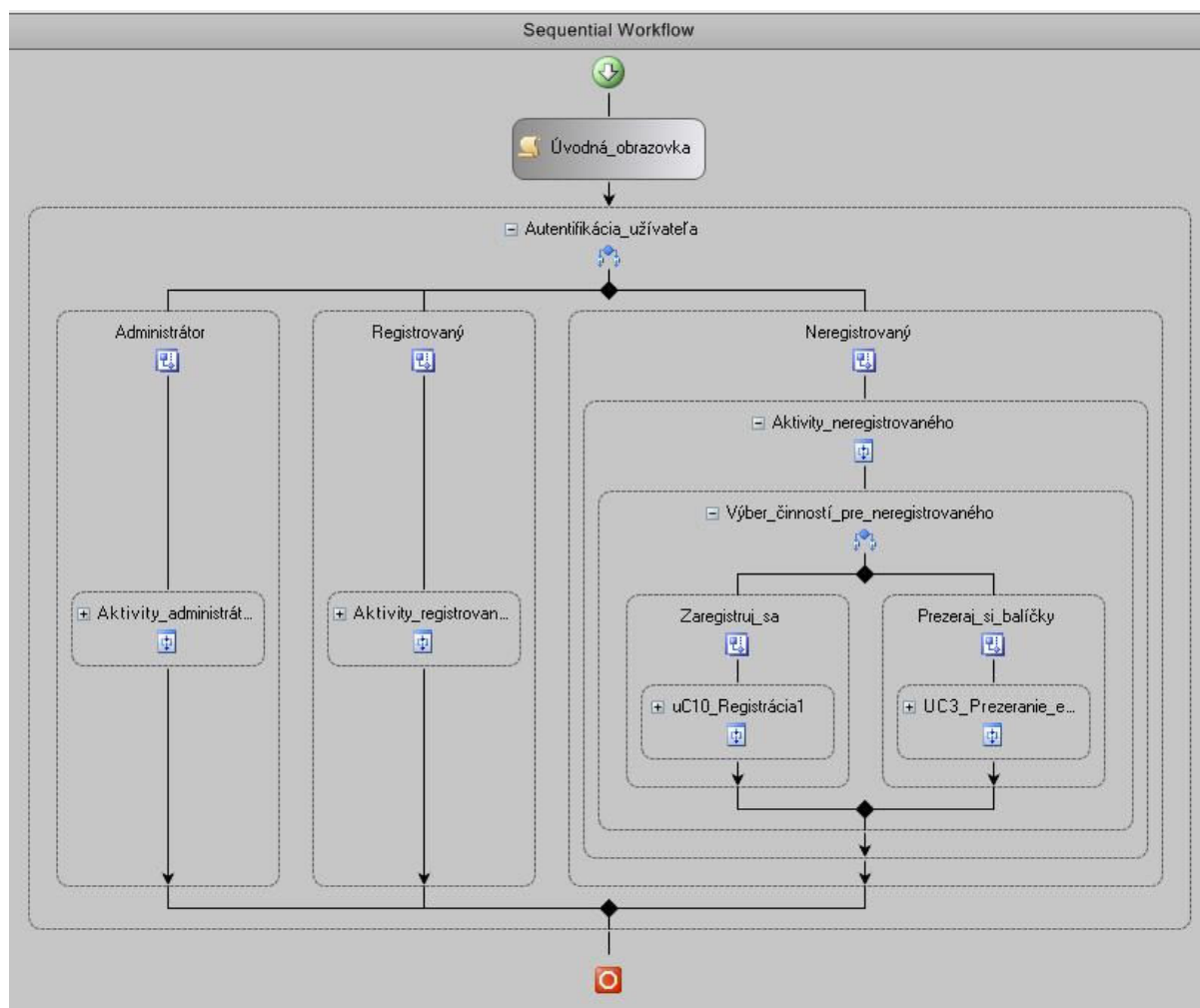
Prvý diagram zobrazuje vstup do systému. Každému užívateľovi bez ohľadu na práva sa zobrazí úvodná obrazovka a dá mu na výber, či sa prihlási alebo bude pokračovať ako neregistrovaný užívateľ s obmedzenými právami. Podľa jeho výberu bude systém pokračovať. Ak sa prihlási ako administrátor, zobrazí sa 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.

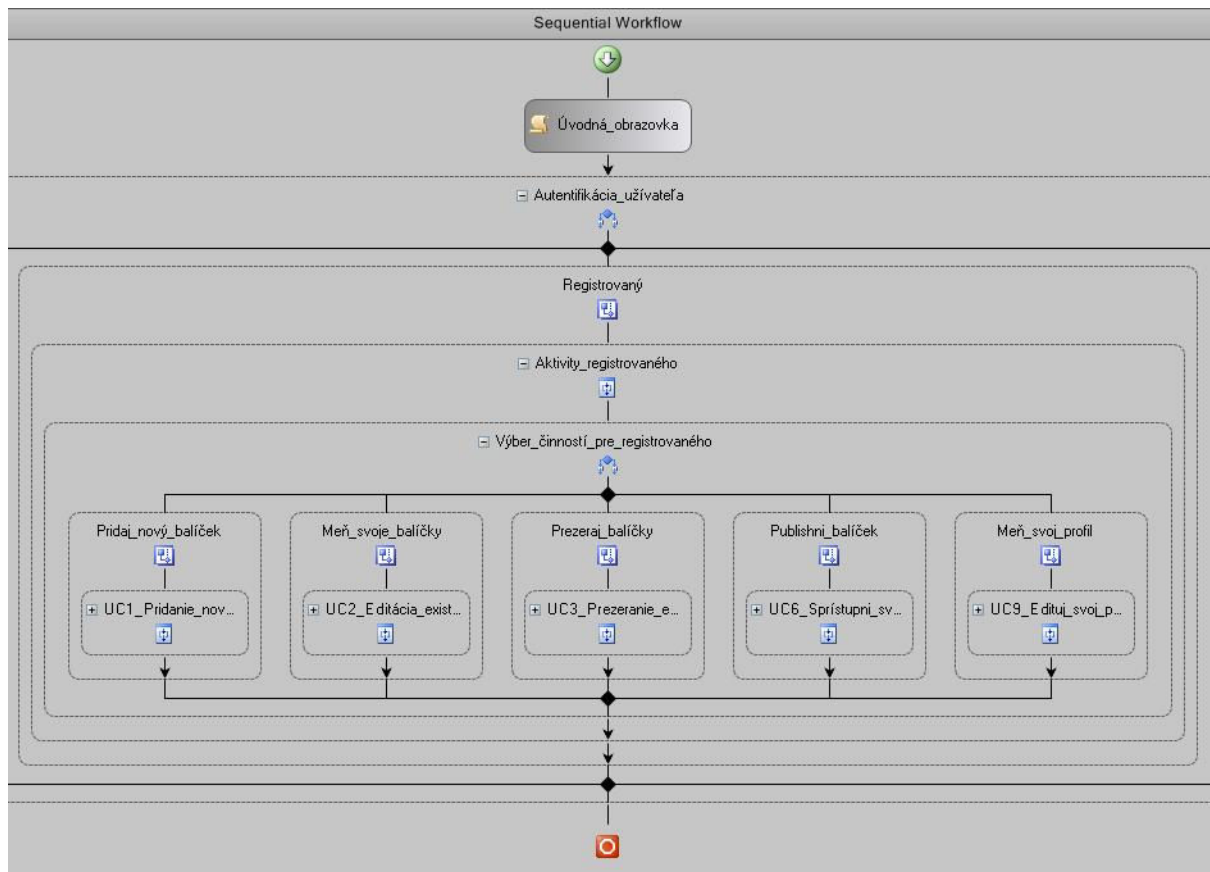
Ďalšie diagramy budú konkretizovať zobrazené možnosti, ktoré sa objavia jednotlivým užívateľom po autentifikácii.

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.

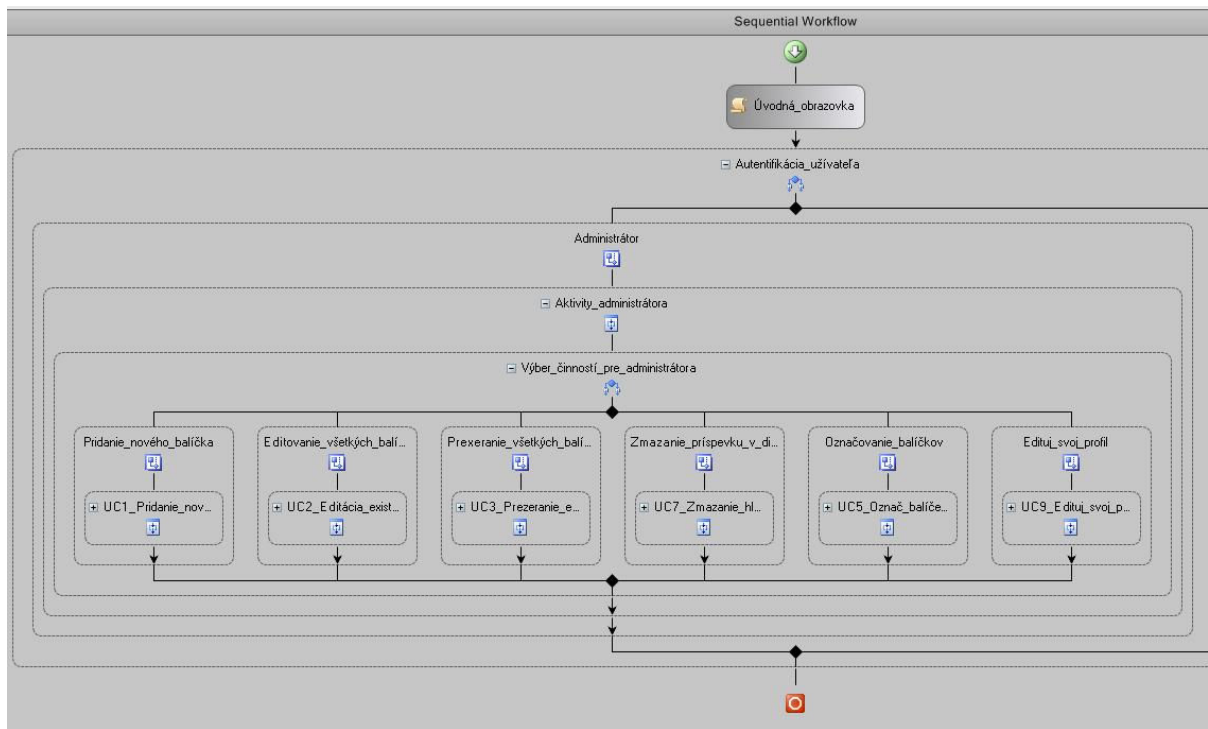
Registrovaní užívatelia majú po prihlásení na svojej úvodnej obrazovke k dispozícii viacero aktivít, ktoré môžu vykonať. Aktivity im umožňujú pridávať, prezerat' a upravovať balíčky, všetko v rámci pravidiel, ktoré im udáva ich rola v systéme. Taktiež môžu publikovať svoje balíčky a sprístupniť ich pre 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 užívatelia si budú môcť vytvoriť profil a meniť údaje v ňom. Taktiež môžu pridávať, prezerať a meniť všetky balíčky bez akýchkoľvek obmedzení.



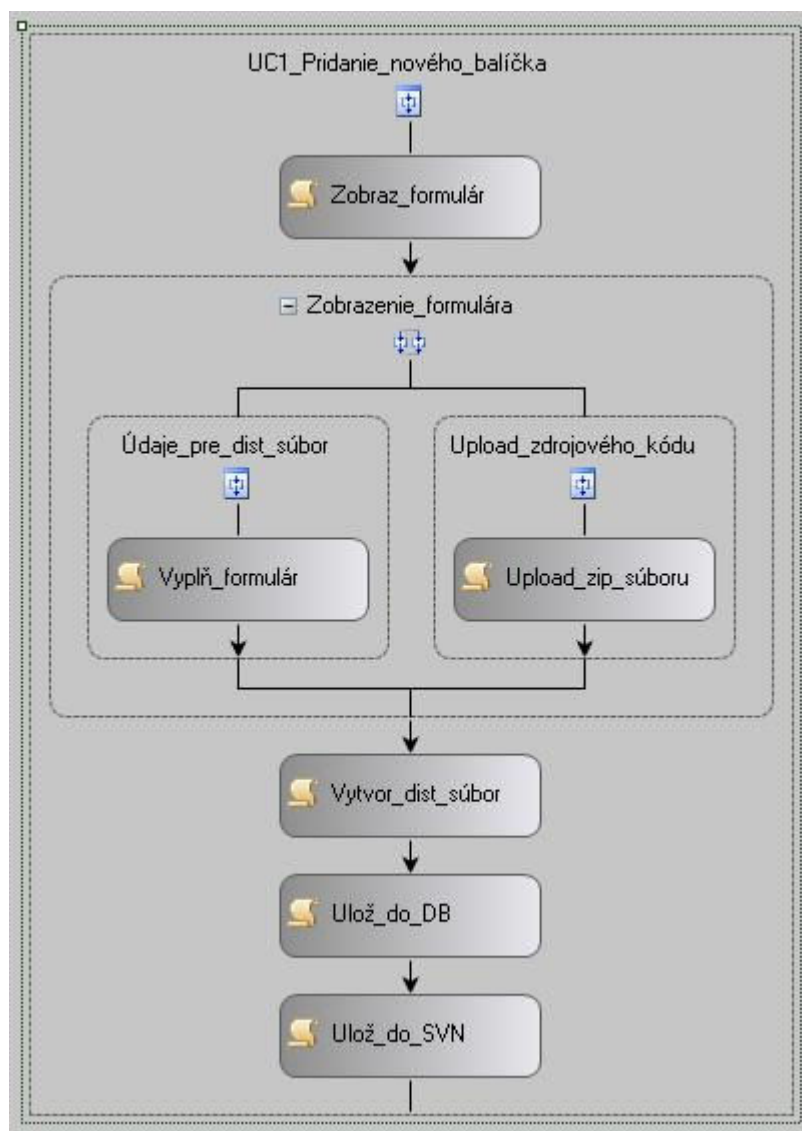
Obr. č. 10 Workflow diagram pre administrátora.

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.

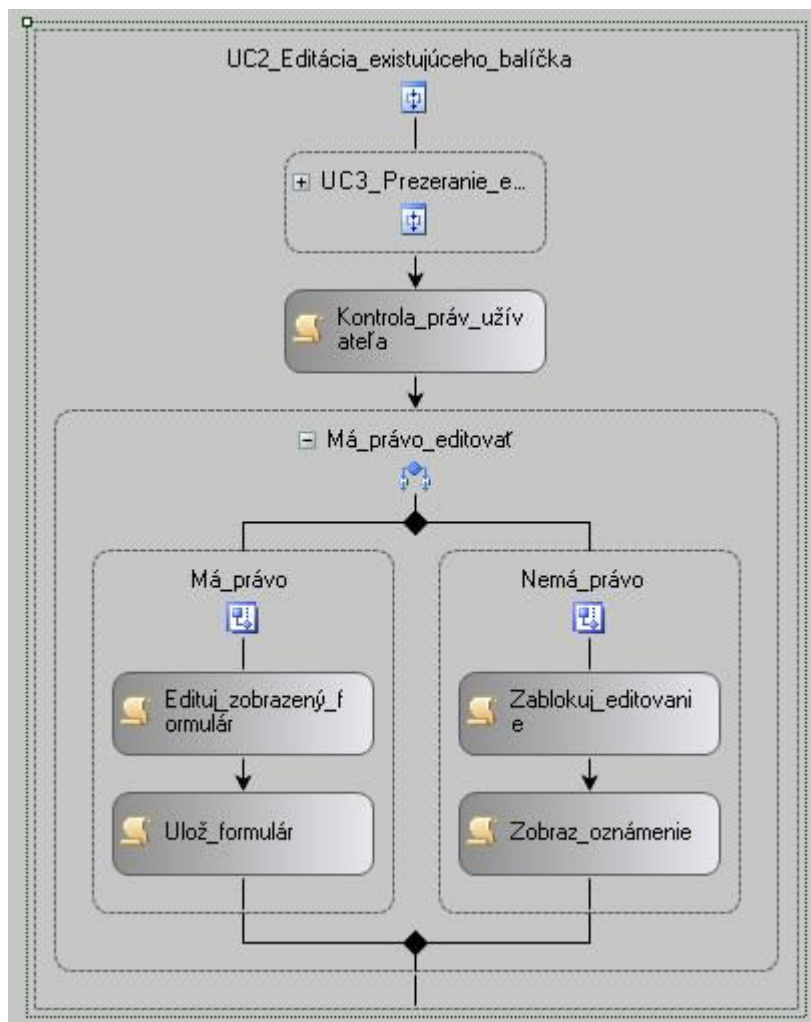


Obr. č. 11 Workflow diagram pre UC pridanie noveho balicka.

UC2 - Editacia existujuceho balicka

Editovanie existujuceho balicka je aktivita, ktora je zlozena z dalsich aktivita a doplnena o niektoré svoje casti. V prvom rade je nutné vybrať si balick, ktorý chceme editovať. Pre túto časť aktivity je použitý UC3, opísaný nižšie. Keď užívateľ vybral balick, ten sa mu zobrazil a editovanie je pripravené. Samozrejme na začiatku je skontrolované, či vôbec užívateľ môže daný balick editovať. Administrátori môžu meniť akýkoľvek balick, registrovaní užívatelia môžu meniť len nimi vytvorené balicky alebo akýkoľvek unstable balick. Nemôžu editovať prívatne balicky iných užívatel'ov, ani stable balicky, 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.

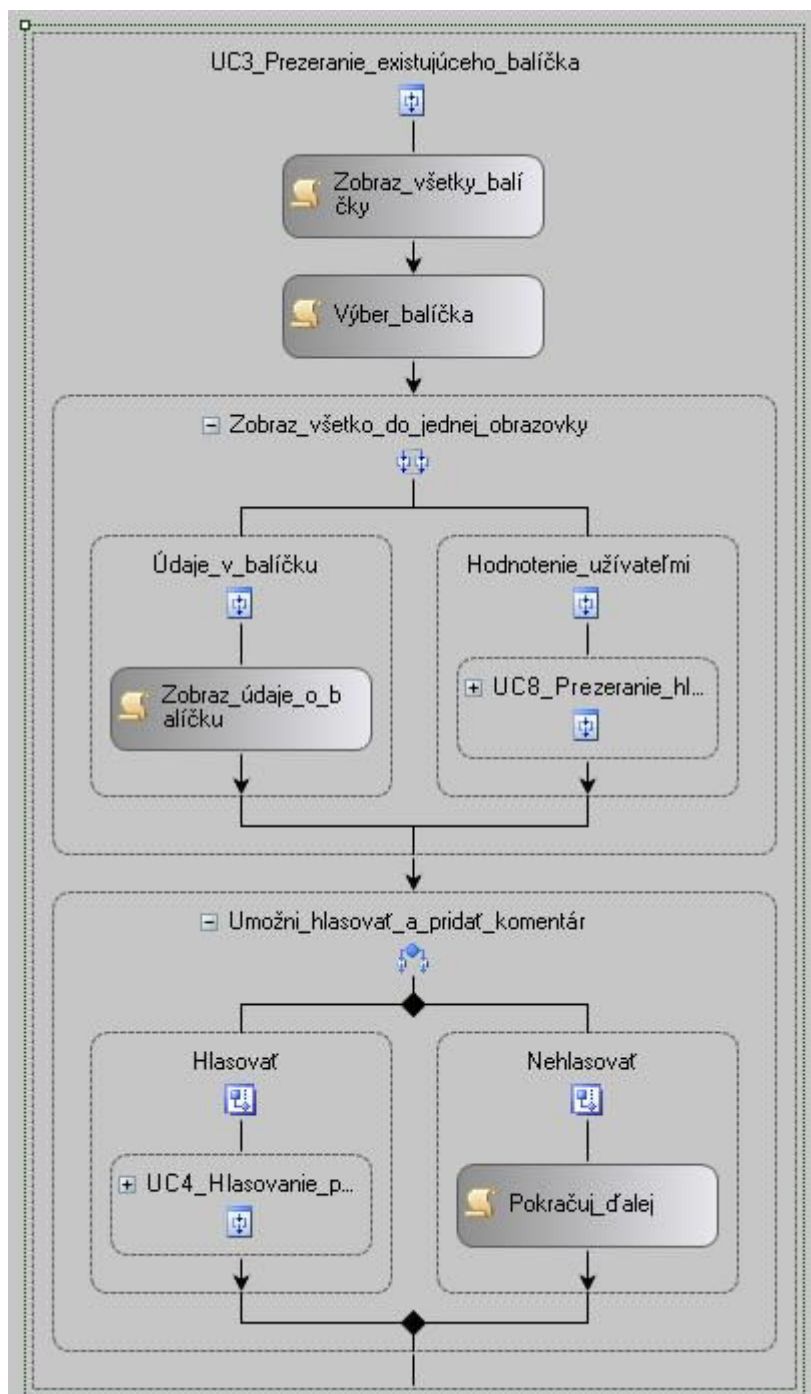


Obr. č. 12 Workflow diagram pre UC editovanie existujúceho balíčka.

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 balíčok ohodnotiť.

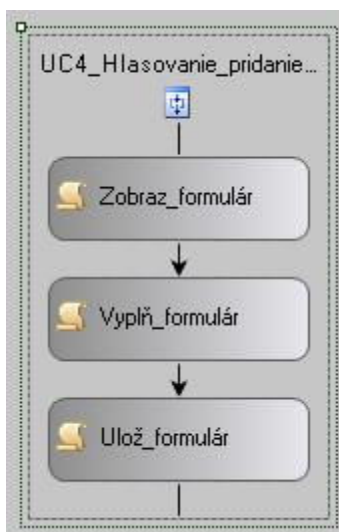


Obr. č. 13 Workflow diagram pre UC prezeranie existujúceho balíčka.

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

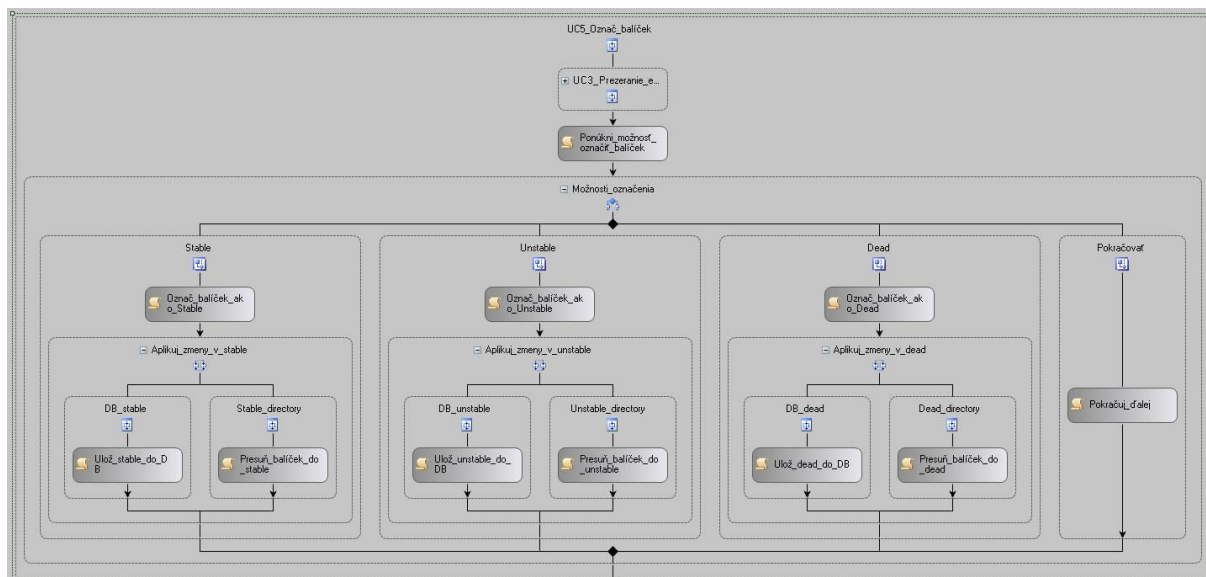
formuláru, do ktorého vpiše svoj názor. Po odoslaní príspevku sa uloží a pridá k danému balíčku.



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

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

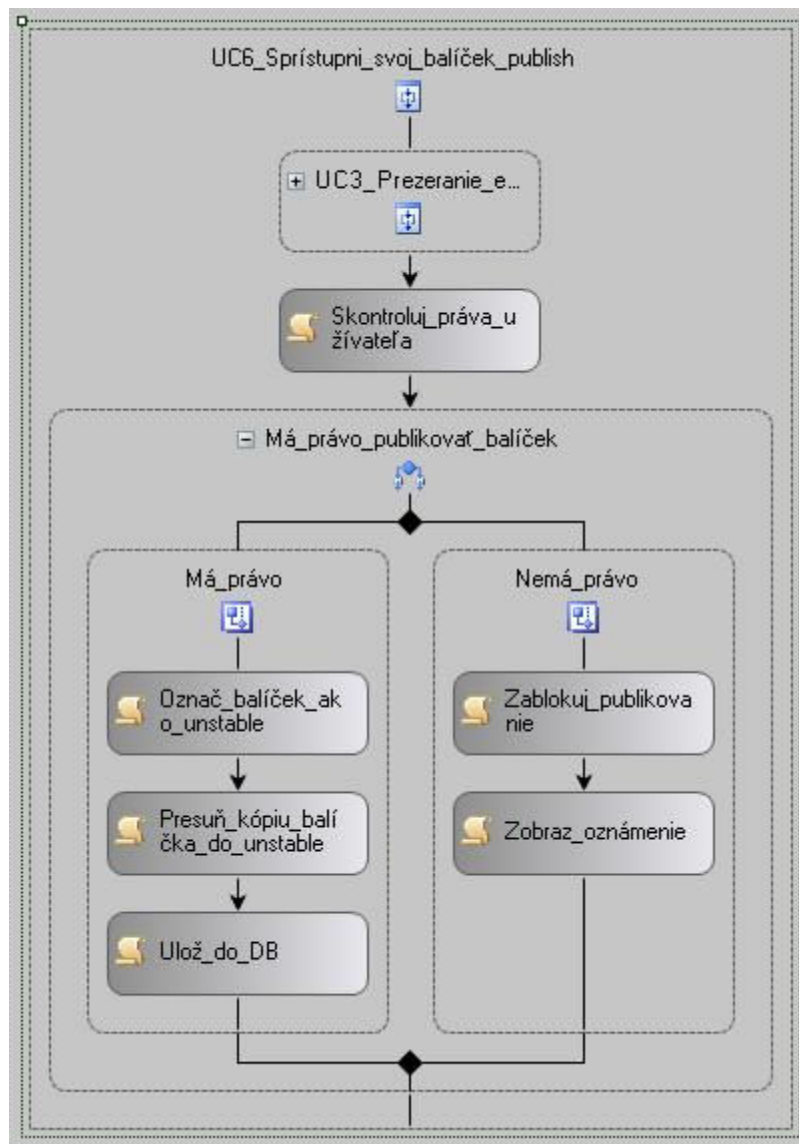
Aktivita, ktorá je najdôležitejšou pre administrátorov tiež pozostáva z viacerých aktivít. Administrátor, ktorý bude označovať nejaký balíček si najprv jeden vyberie. Potom zvolí operáciu označenia balíčka. Môže ho označiť ako stable, unstable, dead. Balíček sa dá označiť ako stable, čo znamená že jeho 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 balíčkom jeho typu. Druhým je uloženie zmien v databáze pre daný balíček.



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

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

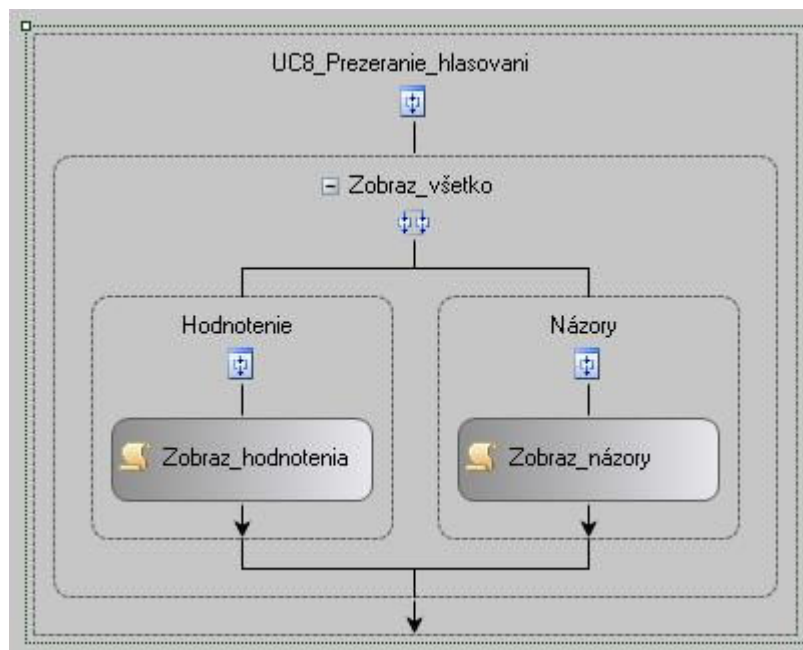
Táto aktivita slúži administrátorom, ktorý sú zároveň aj moderátormi diskusií k balíčkam. Je vhodné aby títo moderátori mohli regulovať správy v diskusiách, najmä neprimerané reakcie a reklamné, nevyžiadané správy. Samotný proces je veľmi jednoduchý, administrátor najprv vyberie z existujúcich názorov 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

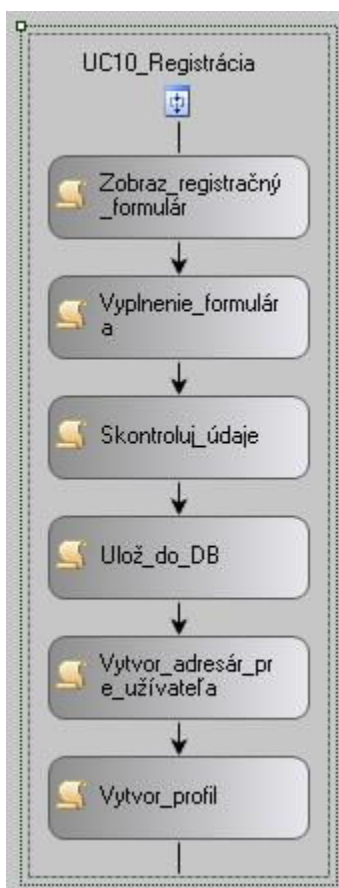
Editovať svoj profil môže ktokoľvek, kto má v systéme vytvorený účet. Jeho profil sa skladá z informácií, ktoré sú povinné, ako napr. užívateľské meno a heslo a z údajov, ktoré povinné nie sú, ako napr. meno, priezvisko, krajina, príp. zoznam iných členov, ktorí majú prístup k jeho privátnym balíčkom. 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 jedinú 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 vyhľadavací 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

win ▾

Architecture

architecture1 ▾

Status

stable ▾

Search

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 upravujú 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](#)

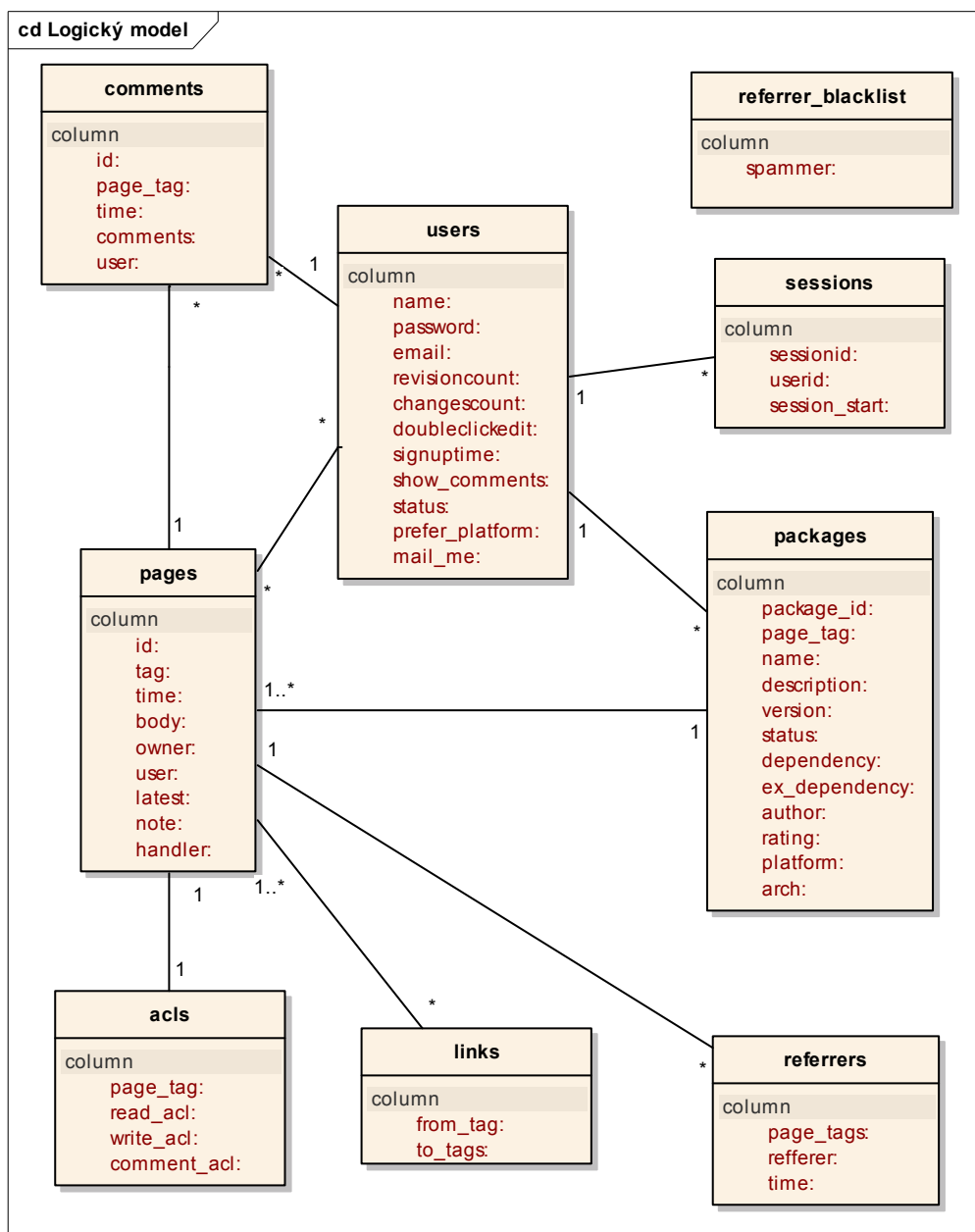
not bad

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. Ako je zrejmé z obrázka 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* – uložený jednotlivý zaregistrovaný používateľ 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 nežiaduce a odkazujú na systém len kvôli spamu

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 dvojitým 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 |
| | mail_me | TINYINT(1) | posielať e-maily o zmene používateľovho balíčka |

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) | ktorý používateľ 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) | na akej stránke sa nachádza |
| | name | VARCHAR (100) | názov |
| | description | TEXT | opis použitia |
| | version | VARCHAR (10) | verzia |
| | status | VARCHAR (10) | stav v systéme (private, unstable, stable) |
| | dependency | TEXT | závislosť na lokálnych balíčkoch |
| | ex_dependency | TEXT | závislosť na externých balíčkoch |
| FK | added | VARCHAR (75) | kto pridal balíček |
| | rating | FLOAT | hodnotenie používateľov |
| | platform | VARCHAR (100) | pre akú platformu |
| | arch | VARCHAR (100) | pre akú architektúru |

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 |

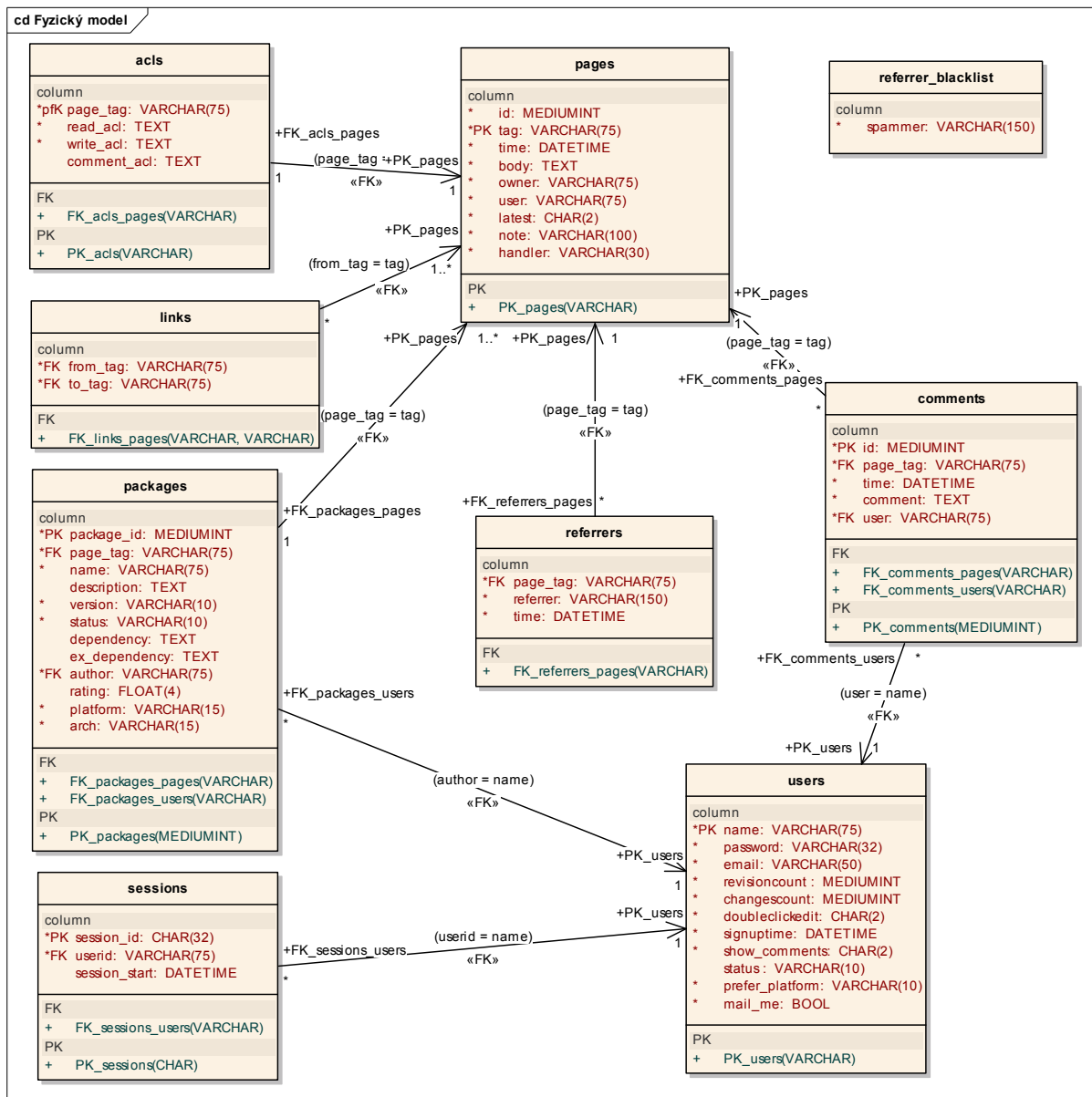
| | | | |
|--|----------|---------------|------------------------|
| | referrer | VARCHAR (150) | referujúca stránka |
| | time | DATETIME | dátum a čas pripojenia |

Tab. č. 13 - Opis atribútov tabuľky `referrer_blacklist`

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

4.3.4 Fyzický model údajov

Na obrázku č. 29 je znázornený fyzický dátový model, ktorý zobrazuje dátové typy databázy MySQL a použitie cudzích kľúčov. Na primárny kľúč hlavnej tabuľky *pages* sa cudzím kľúčom odvolávajú až 4 tabuľky, ktoré sú v relácií. Ako je zrejmé z obrázku, každý balíček je v relácií so stránkou, na ktorej je vytvorený a používateľom, ktorý je jeho autorom.



Obr. č. 29 Fyzický model údajov

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 obal'ovacie 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 Záver

Informácie v tejto dokumentácii odrážajú súčasný stav tímového projektu. V komunikácii so školiteľom sme ukončili identifikovanie požiadaviek na webový systém správy rozširujúcich modulov jazyka Lua. Taktiež sme prenikli do problematiky jazyka Lua a jeho distribúcie LuaDist, zvolili vyhovujúci wiki systém, ktorý budeme rozširovať, 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 hlavných obrazoviek používateľského rozhrania.

V ďalšej časti práce sme sa orientovali na implementáciu základných funkcií, ktoré budú tvoria prototyp..

7 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 rôznych platformách (<http://www.cmake.org/>)