MASARYKOVA UNIVERZITA

FAKULTA INFORMATIKY

LMS pre Liferay portál

Diplomová práca

Bc. Ľubomír Lacika

Brno, jeseň 2014

Prehlásenie

Prehlasujem, že táto diplomová práca je mojim pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Vedúci práce: RNDr. Jaroslav Ráček, Ph.D.

Poďakovanie

Chcel by som sa poďakovať môjmu vedúcemu diplomovej práce RNDr. Jaroslavovi Ráčkovi, Ph.D. za ústretovosť, trpezlivosť a poskytnutie cenných rád pri tvorbe tejto práce. Taktiež ďakujem zamestnancom spoločnosti IBA CZ za získané vedomosti z programovania portletových aplikácií, ktoré som mohol nadobudnúť počas stáže.

Zhrnutie

Cieľom diplomovej práce je oboznámenie sa s vývojom aplikácií na platforme podnikových portálov, zoznámenie sa s princípmi firemného vzdelávania a tvorbou funkčných portletov určených pre elektronické vzdelávanie zamestnancov firmy. V analýze sme popísali princípy LMS systémov a analyzovali dostupné softvérové riešenia na trhu, ktoré sa v tejto oblasti používajú. Následne práca popisuje špecifikáciu, návrh a implementáciu prototypu takéhoto vzdelávacieho systému v prostredí Liferay portálu. Využili sme hlavne základné integrované komponenty portálu (DMS a CMS). Práca bola riešená v rámci priemyslového partnerstva FI MU s firmou IBA CZ, s.r.o..

Kľúčové slová

Portál, Portlet, JSR-168, JSR-286, Liferay, LMS, Firemné vzdelávanie, E-learning

Obsah

Úvod ... 1

1.Portály a portlety ... 3

1.1 Špecifikácia JSR-168 ... 4

1.2 Špecifikácia JSR-286 ... 7

1.3 Rozdelenie portálov ... 12

1.4 Liferay ... 14

2. Firemné vzdelávanie ... 21

2.1 E-learning ... 24

2.2 LMS ... 26

2.3 Dostupné produkty firemného vzdelávania ... 30

3. Analýza a návrh ... 33

3.1 Špecifikácia aplikácie ... 33

3.2 Dátový model ... 35

3.3 Návrh obrazoviek ... 37

3.4 Rozdelenie portletov ... 38

4. Realizácia a testovanie ... 42

4.1 Trojvrstvová architektúra ... 42

4.2 Použité technológie ... 44

4.3 Testovanie ... 50

4.4 Dokumentácia ... 54

Záver ... 57

Zoznam použitej literatúry ... 59

Prílohy ... 63

Príloha A ... 63

Príloha B ... 64

Príloha C ... 65

Úvod

V posledných rokoch došlo k rýchlemu rozmachu informačných služieb (IT) nie len vo svete, ale aj u nás v tuzemsku. Tieto IT začali integráciou do podnikových prostredí a postupne prenikali až do najrôznejších odvetví ľudskej činnosti. Veľkej modernizácií sa nevyhli ani vzdelávacie procesy vo firmách, či na univerzitách. Pre ich zefektívnenie a zlepšenie sa v organizáciách začali využívať nové spôsoby edukácie nazvané e-learning. K vzdelávaniu vo firmách často dochádza pomocou rôznych vzdelávacích programov, inštruktáží a rôznych skúseností, ktoré získava zamestnanec počas pracovných povinností. Efektivitu pracovných aktivít môže organizácia zvyšovať práve spomenutým elektronickým vzdelávaním. Vzdelávanie svojich pracovníkov patrí medzí kľúčové aktivity, ktoré zabezpečujú dlhodobé prežitie organizácie vo svete. Teda ide o vzdelávanie orientované na dospelých ľudí, čo vedie k rozvoju zručností, vedomostí a schopností, čím sa zvyšuje potenciál a použiteľnosť zamestnancov a taktiež rast firmy. Žijeme v dobe nových nápadov, výziev a rýchlo sa objavujúcich zmien, pomocou ktorých sa usilujeme posúvať našu spoločnosť vpred.

Cieľom diplomovej práce bolo navrhnutie a implementácia prototypu LMS[1] v prostredí firemného vzdelávania. Tento prototyp bol vyvíjaný na platforme Liferay CE[2] portálu a zameriaval sa na sprístupnenie študijných materiálov, testov a organizáciu kurzov. Medzi samozrejmosť patrila integrácia s DMS[3] a CMS[4] komponentov spomínaného portálu. Súčasťou práce bolo testovanie aplikácie a základná dokumentácia pre používateľov spomenutého systému na riadenie výučby. Pred praktickou časťou bolo nevyhnutnou potrebou analyzovať ponúkané softvérové riešenia v oblasti firemného vzdelávania a ozrejmiť základy a princípy systému na riadenie výučby (LMS) firemného vzdelávania, či podnikových portálov.

Prvá kapitola je vstupnou bránou do teoretickej časti práce, ktorá oboznamuje čitateľa s portálmi a portletovými špecifikáciami uľahčujúcimi kompatibilitu

---

1 Learning management system, wikipedia.org/wiki/Learning_Management_System

2 Liferay CE, https://www.liferay.com/community/liferay-projects/liferay-portal/overview

3 Document management system, http://cs.wikipedia.org/wiki/Spr%C3%A1va_dokument%C5%AF

4 Content management system, http://en.wikipedia.org/wiki/Content_management_system

medzi portletmi a samotným portálom. Taktiež sa tu môžeme dočítať o portáloch založených na Java EE[5] infraštruktúre, či ich kategorizovaní a tiež samotnom využití. Ďalej sme sa zamerali na podnikovú platformu Liferay Portal. Zaoberali sme sa hlavnými funkciami portálu, rôznorodými porovnaniami edícií, podporovanými technológiami a rozšíreniami, ktoré umožňuje spomenutý portál.

Druhá kapitola je pokračovaním teoretickej časti, kde je vysvetlená problematika firemného vzdelávania. Môžeme tu nájsť rôzne myšlienky autorov kníh zaoberajúcich sa marketingom a riadením ľudských zdrojov. Zamerali sme sa hlavne na identifikáciu potrieb firemného vzdelávania. Taktiež sme si ozrejmili pojem e-learning a pozreli sme sa i na ponúkané možnosti integrácie LMS systému do organizácie. Záverom tejto kapitoly bol vypracovaný prehľad dostupných produktov určených pre potreby firemného vzdelávania.

Tretia a štvrtá kapitola je venovaná praktickej časti diplomovej práce. Nájdeme tu analýzu a návrh spomínaného prototypu. Pri realizácii aplikácie sa môžeme dozvedieť akým spôsobom bola aplikácia rozdelená, aké technológie boli pri jej tvorbe do nej zahrnuté a aké služby ponúkané daným portálom sme využili. Neoddeliteľnou súčasťou poslednej kapitoly je testovanie a dokumentácia, v ktorej si môže lektor vzdelávania ozrejmiť niektoré nastavenia aplikácie, či napĺňanie kurzov študijnými materiálmi.

---

[5] Java Platform Enterprise Edition, http://www.oracle.com/technetwork/java/javaee/

1.Portály a portlety

Portál je webové prostredie, z ktorého môžu byť spustené všetky používateľské aplikácie. Tieto aplikácie sú integrované dohromady v súlade a systematickým spôsobom, takže spomínané webové prostredie slúži ako brána umožňujúca sprístupnenie a vytvorenie príslušného obsahu, a taktiež ako používanie aplikácií bežnej potreby pre príslušného používateľa. Portál sa pred klientom reprezentuje ako jednotná aplikácia skladajúca sa z viacerých stránok. [1]

Spoločnosť Sun Microsystems definuje portál vo svojej java portlet špecifikácii (JSR 286) nasledovne a to tak, že sa jedná o webovú aplikáciu, ktorá bežne poskytuje personalizáciu, autentifikáciu a obsah agregácie z rôznych zdrojov. Usporadúva aj obsah prezentačnej vrstvy informačného systému. Agregácia je akcia integrujúca obsah z rôznych zdrojov v rámci webovej stránky. Portál môže mať sofistikované funkcie na prispôsobenie obsahu špecifickým spôsobom pre daného používateľa. Portálové stránky môžu mať rozdielne nastavenie portletov vytvárajúcich obsah pre rozličných používateľov. [2]

Spomínaná špecifikácia portálu poukazuje na to, že sa jedná o druh webovej aplikácie agregujúcej obsah z rozličných zdrojov, či už z webových stránok, alebo webových aplikácií. Webovými stránkami generovaný obsah môže byť statický alebo dynamický. Ak portálové stránky vyžadujú prihlásenie používateľa, ide o sign-on, čo pre portál znamená, že v momente zadania prihlasovacích informácií používateľom získava prístup ku všetkým zobrazeným aplikáciám. Ako je uvedené v definícii od spoločnosti Sun Microsystems – portál poskytuje personalizáciu s dôrazom na slobodu voľby pri zobrazovaní jednotlivých aplikácií na osobnej stránke portálu pre daného používateľa. To znamená, že používateľ má možnosť kedykoľvek to uzná za vhodné nechcené aplikácie odobrať a žiadané pridať. [3]

Malé webové aplikácie napísané v programovacom jazyku Java sú známe pod pojmom portlety. Tieto portlety sú spúšťané na webových stránkach s cieľom poskytnutia konkrétneho obsahu, či služby. Ide o grafickú portálovú komponentu, pomocou ktorej sa klient orientuje, ďalej s portálom komunikuje a pracuje. Portál obsahuje zvyčajne celú radu portletov, ktoré umožňujú správcovi, resp. používateľovi pridať ich na stránku portálu. Na portáli sa nenasadzujú jednotlivé portlety, ale celé balíky portletov združujúce sa do portletovej apli-

kácie. Portletová špecifikácia definuje Portlet API, pomocou ktorého portál následne pracuje s portletmi a to znamená, že každý portál je v určitom zmysle odlišný. [4] Prvá špecifikácia portletov, vydaná v roku 2003, má označenie JSR 168. Bola však nahradená novou verziou JSR 286, z roku 2008. V súčasnosti sú pre verejnosť prístupné obe varianty. [5]

1.1 Špecifikácia JSR-168

Portletová špecifikácia JSR 168 určila smer vývoja portletov. Doposiaľ neexistoval žiadny štandard, ktorý by zabezpečoval spoluprácu medzi portletmi. Prenositeľnosť bez väčších úprav na iné portály nebola možná, pretože každý výrobca používal svoje vlastné napísané API. Takto napísaná aplikácia bola závislá na zvolenom portálovom riešení a to spôsobovalo i horší odchod k inému dodávateľovi. Ak k tomu došlo, celá aplikácia musela byť upravená, ba dokonca až prepísaná odznova. Nový vyvinutý štandard definoval portletový kontajner a životný cyklus celého portletu, jeho módy a stavy okien, tiež možnosť ukladania používateľských nastavení. Počas odlaďovania a testovania nového štandardu sa zistilo množstvo nedostatkov a chýb, ktoré dodávatelia začali riešiť a opravovať podľa vlastných metód, čo viedlo opäť k nekompatibilite. [2]

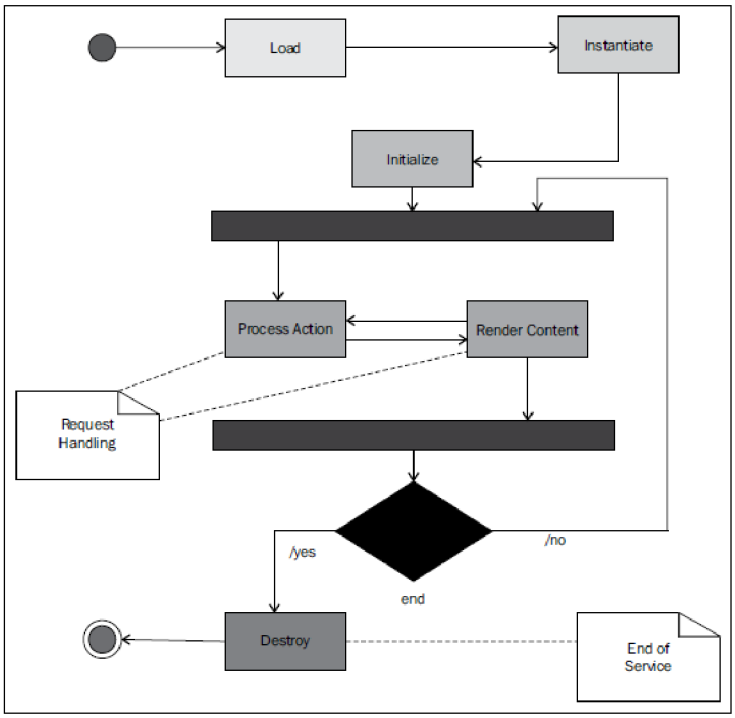
Portletový kontajner stojí medzi portálom a portletmi. Prijíma požiadavky zo strany portálu a poveruje dané portlety k ich obslúženiu. On sám zodpovedá za vykonávanie jednotlivých funkcií životného cyklu portletu. Životný cyklus pozostáva z troch základných fáz: [6]

Vytvorenie portletu - je prvou fázou životného cyklu portletu. Pomocou metódy nazývanej `init()` dochádza k zavedeniu všetkých potrebných tried a ich atribútov nadobúdajúcich základné hodnoty. Po nich sa následne zavolá konštruktor. Ak sú všetky kroky splnené, dochádza k vytvoreniu inštancií portletu a to buď pri zapnutí aplikácie alebo ak je daný portlet vyžiadaný k obsluhe niektorej z požiadaviek (viď obrázok č. 1).

Spracovanie požiadaviek - po úspešnej inicializácii je portlet pripravený plniť prichádzajúce žiadosti od používateľov. Požiadavky, ktoré môžu prísť, sú dvojakého charakteru. Žiadosť na zmenu stavu portletu dochádza prostredníctvom metódy `processAction()`. Pri tejto metóde sa napríklad spracovávajú údaje prichádzajúce z vyplneného formulára, či akcie požadujúcej spracovanie, uloženie alebo modifikovanie dát v dátovom úložisku. Druhá možnosť požiadavky

je žiadosť na vykreslenie portletu. Tá sa vykonáva pomocou metódy `render()` a má za úlohu generovanie obsahu stránky, na ktorej je umiestnená. V reťazci životného cyklu portletu je volaná vždy po metóde `processAction()` [7]. V špecifikácii verzie 2.0 (JSR 286) pribudla ďalšia fáza, ktorá ma za úlohu prijmať a reagovať na vzniknuté udalosti. Táto fáza sa vsunula medzi ukončením `action` a začatím `render` fázy. Portlet v takomto prípade musí mať naimplementované rozhranie `EventPortlet` s metódou `porcessEvent()`.

Odstránenie portletu - Ak portletový kontajner zistí, že daný portlet už nie je potrebný, zavolá metódu `destroy()`. Portlet automaticky uvoľní alokované zdroje a zruší všetky referencie. Pomocou spomenutej metódy dôjde k zrušeniu inštancie portletu a potom garbage collector[6] objekt uvoľní z pamäte. [8]



Obrázok č. 1: Životný cyklus portletu. Zdroj: [9].

[6] Garbage collector, http://javabook.compuware.com/content/memory/how-garbage-collection-works.aspx

Okrem toho, že portletový kontajner spravuje inštancie portletov a ich životný cyklus, poskytuje rovnaké možnosti ako servletový kontajner. Ide teda o jeho rozšírenie. Skoro všetky kontajnery poskytujú vlastné rozšírenia. Je len na voľbe vývojára, ktorý kontajner použije a či tieto ponúkané špecifikácie zahrnie do svojej práce, pretože pri ich využití portlet stráca možnosť kompatibility medzi inými kontajnermi.

Každý portlet môže mať rôzne režimy umožňujúce zobraziť odlišný obsah. Špecifikácia JSR 168 nám hovorí o troch štandardných režimoch, ktoré môžeme použiť. Tými sa zaoberá metóda `render()`, ktorá na základe nich vykreslí obsah portletu. Medzi spomínané tri módy patrí:

> *View* - ide o najdôležitejší a základný mód, ktorý je povinný a tým pádom ho musí každý portlet obsahovať. V ňom sa zobrazujú stránky a poskytnutá funkcionalita.
> *Edit* - tento mód slúži k nastaveniu daného portletu. Umožňuje tak používateľovi prispôsobiť si správanie portletu podľa svojich potrieb.
> *Help* - zobrazuje nám potrebné informácie o sebe a o tom, ako ho najlepšie používať.

Pri implementácii týchto módov musíme prekryť metódy ako `doView()`, `doEdit()` a `doHelp()` nachádzajúce sa v triede `GenericPortlet`. Portlet nemusí mať naimplementované posledné dve spomenuté módy. Tie patria medzi nepovinné a taktiež nás neobmedzujú v definovaní vlastných. [6]

Stavy okien nám hovoria o množstve miesta, ktoré zaberá daný portlet na portálovej stránke. Rovnako ako pri režimoch zobrazenia rôzneho obsahu (viď odsek nad textom) i stavy okien sú obstarávané portletovým kontajnerom. Tento kontajner posiela informáciu o stave okna portletu. Veľkosť portletu na portálovej stránke špecifikujú tri stavy okien. [6]

- *Normal* - jedná sa o bežné zobrazenie stránky, kde používateľ vidí viacero portletov. Jednotlivé portlety zdieľajú túto stránku s ostatnými.
- *Minimized* - pri použití tohto stavu sa zobrazí na portáli len hlavička portletu a celý jeho obsah je skrytý
- *Maximized* - portlet skryje všetky ostatné portlety a zobrazí sa cez celý portál. V takomto prípade ide o výhodu, ak chceme s portletom pracovať naplno, pretože pri takomto stave okna sa generuje širší obsah.

Portletové aplikácie sú založené na Java EE technológiách a svojou štruktúrou sú podobné webovým aplikáciám. Dôležitou súčasťou týchto aplikácií sú deskriptory `web.xml` a `portlet.xml`. Tie môžeme nájsť v adresári WEB-INF. Sú to popisovače obsahujúce základné informácie a nastavenia, ako meno a popis portletu. Obsahujú aj rôzne inicializačné nastavenia, módy portletu, medziportletovú komunikáciu (nastavenia verejných parametrov a udalostí), podporované jazyky a iné. Okrem týchto dvoch portletových deskriptorov sa v adresári WEB-INF môžu vyskytnúť popisovače závislé na dodávateľskom portáli, v ktorých môžeme nájsť rôzne špecifické nastavenia pre danú technológiu.

Xml[7] súbor `liferay-portlet.xml` nachádzajúci sa v tomto adresári je určený pre Liferay portlety. Sú v ňom uložené informácie o portletoch, CSS[8] a JS[9] súborov importovaných do hlavičiek stránok portletu alebo i nastavenia možností množstva zobrazenia portletu na danej stránke portálu. V spomenutom adresári sa taktiež môže nachádzať deskriptor `liferay-display.xml`. Ten nám slúži na zadelenie portletov do kategórie s ich popismi, ktoré nám sprehľadňujú množstvo portletov prichystaných pre import na stránky portálu.

Portlety sú nasadzované rovnako ako webové aplikácie, ktoré sú uložené v štandardnom archivačnom súbore jazyka Java[10], WAR[11]. Tento súbor obsahuje všetky portlety, JSP[12] súbory, deskriptory a iné použité pri tvorbe portletov. [7]

1.2 Špecifikácia JSR-286

Pri objavení nedostatkov v špecifikácii JSR 168 bola vyvinutá nová verzia JSR-286. Šlo o novú špecifikáciu, ktorá sa snažila pokryť výrazné problémy s predchádzajúcim štandardom. Nejednalo sa o nahradenie, ale o rozšírenie špecifikácie, ktorá je spätne kompatibilná. Vďaka potrebám zlepšenia a zjednodušenia JSR 286 prináša nové vylepšenia a to:

---

7 eXtensible Markup Language, http://www.xml.com/pub/a/98/10/guide0.html

8 Cascading Style Sheets, http://www.w3.org/Style/CSS/

9 JavaScript, https://developer.mozilla.org/en/JavaScript.

10 Java, http://www.java.com/.

11 Web Application Archiv files, https://docs.oracle.com/cd/E19830-01/819-4712/ablgz/index.html

12 Java Server Pages, http://www.oracle.com/technetwork/java/javaee/jsp/index.html

- medziportletovú komunikáciu pomocou verejných parametrov a udalostí,
- používanie cookies[13] a technológie AJAX[14],
- podporu pre bežné rámce určené na tvorbu web aplikácií,
- pridávanie javascriptu a CSS do hlavičiek stránok,
- lokalizáciu pomocou špeciálnych súborov,
- podporu dynamického poskytovania obsahu (získanie médií zo stránok bez nutnosti jej načítania),
- využitie anotácií pri spracovaní požiadaviek,
- zlepšenie podpory vyrovnávacej pamäti v portáli,
- obsluhu zdrojov,
- podporu pre binárne výstupy,
- portletové filtre a iné.

Medzi najväčšie problémy patrila obsluha alternatívnych zdrojov, medziportletová komunikácia a riešenie pomocou session. Obsluha zdrojov, napríklad vo forme obrázkov, PDF[15] a iné patrili medzi problematické. Riešené udalosti a reakcie na ne boli nedostatočné. Portlet verzia 1.0 (JSR 168) podporuje dve fázy spracovávania požiadaviek. Ide o `processActiono()` a `render()`. Vďaka anotáciám vo verzii 2.0 sa zjednodušilo ich používanie. Lokalizácia bola riešená pomocou deskriptorov (atribútov `xml:lang` v `portlet.xml`). V novšom spracovaní boli ukladané hodnoty do špeciálnych súborov - `resource bundles`. K výrazným vylepšeniam, ktoré si môžeme všimnúť pri JSR-286 oproti jeho predchodcu, patrí práca s udalosťami, s verejnými parametrami, s obsluhou zdrojov a portletovými filtrami. A práve preto sa budeme týmto témam v ďalších podkapitolách venovať detailnejšie. [10]

Zdieľanie dát v session (*PortletSession*[16]) – v predchádzajúcom štandarde bola riešená komunikácia za pomoci zdieľania dát v session a to prostredníctvom PORTLET_SCOPE a APLICATION_SCOPE. Ide o dva rôzne atribúty

---

13 Cookies, http://tools.ietf.org/html/rfc6265

14 Asynchronous JavaScript and XML, https://developer.mozilla.org/en/AJAX/Getting_Started

15 Portable Document Format Reference, http://www.adobe.com/devnet/pdf/pdf_reference.html

16 Portlet session, https://portals.apache.org/pluto/portlet-1.0-apidocs/javax/portlet/PortletSession.html

s odlišným rozsahom platnosti. Pri PORTLET_SCOPE sú objekty dostupné len v portlete, v ktorom boli vytvorené. Pri ďalšom menovanom atribúte sú dostupné pre všetky portlety, v rámci danej portálovej aplikácie pri použití rozhrania `PortletSession`. Toto rozhranie definuje obe možnosti zdieľania dát a tak môže dôjsť nielen ku kolíziám pri definovaní rovnakých mien, ale aj k zmenám v dátach uložených v session pri render fáze.

Verejné parametre (*public render parameters*) - umožňujú zdieľanie parametrov s ostatnými portletmi, dokonca i s portletmi z rôznych aplikácií. Ak chceme používať takýto spôsob komunikácie, musíme tieto parametre deklarovať v deskriptore (`portlet.xml`) a uviesť, ktoré portlety k ním môžu pristupovať a ktoré ich môžu meniť. Potom sú k dispozícii vo všetkých fázach životného cyklu. Pri takejto medziportletovej komunikácii sa zobrazia parametre na URL[17] stránke. Posielaný parameter môže byť len textový reťazec. *Public render parameters* nie je možné používať vo fáze spracovávania, ale len vo fáze vykreslenia portletu.

Udalosti (*events*) - ide o odlišné riešenie komunikácie medzi portletmi ako v predchádzajúcom riešení. Udalosti môžu byť vygenerované portálom, akciou používateľa, alebo i portálovým kontajnerom. Tak ako pri verejných parametroch, tak aj v deskriptore (`portlet.xml`) je dôležité vykonať nastavenie portletov, v ktorých definujeme aký portlet bude udalosť vytvárať a aký prijímať. Takáto zadefinovaná udalosť pozostáva z mena a posielaného objektu. Pri komunikácii pomocou udalosti je veľmi dôležité spomenúť, že umožňuje posielať i iný typ objektov, ako len textový reťazec. K obsluhe celej udalosti nám slúži metóda `processEvent()`.

Portletové filtre sú komponenty umožňujúce transformáciu informácií v požiadavkách a odpovediach portletu. Tieto filtre samé o sebe negenerujú výsledok, ale spracovávajú informácie a posúvajú ich ďalšiemu filtru, alebo portletu. Nastavenie sa definuje v `portlet.xml`. Využitie týchto filtrov si nájde svoje miesto pri zbere informácií o celej aplikácii alebo o činnosti portletu. Môžu byť zreťazené a použité vo viacnásobných životných cykloch.

Portlety na základe URL nie je možné adresovať, okrem adresy na seba samých. Je preto potrebné zabezpečiť iný mechanizmus. V predchádzajúcej verzii JSR-

---

[17] Uniform Resource Locator, http://en.wikipedia.org/wiki/Uniform_Resource_Locator.

168 nebolo možné dynamicky vygenerované zdroje poskytnúť. Novší štandard JSR-286 ponúka možnosti odkazovania sa na zdroje typu HTML[18], zdroje obrázkov a dokumentov, PDF formátov a iných. Výhodou tohto riešenia je špeciálna metóda (`createResourceUrl()`), ktorá vytvorí zdrojové URL pre prístup ku konkrétnemu zdroju. Taktiež máme na výber uloženie informácie do cache pamäte pre rýchlejší prístup. K dispozícií je i značka v knižnici JSP, ktorá slúži pre vytvorenie URL zdroja priamo na stránke. [10]

Najpoužívanejšie portály sú postavené na Java technológiách fungujúcich zvyčajne podľa špecifikácií JSR–168 alebo JSR – 286. Vďaka tejto štandardizácii je možné vytvoriť modul portálu (portlet) a nasadiť ho na akýkoľvek portál, kde bude fungovať bez nutnosti zmeny programového kódu. Firmy používajú portály Enterprise s množstvom nasadených aplikácií slúžiacich pre používateľovo ďalšie zameranie a prácu. Každá aplikácia je charakteristická svojim vlastným vzhľadom a umiestnením na serveri. Ak s ňou túži používateľ pracovať, je nevyhnutné, aby sa v rámci aplikácie osobne prihlásil. Adekvátnejším riešením by bol intranet s riešením single sign-on práve kvôli jednotnému prihlasovaniu sa. Používateľ by sa prihlásil len raz a to by mu stačilo na orientáciu sa vo všetkých potrebných aplikáciách. Portály obdobne ako intranety obsahujú single sign-on mechanizmus. [4]

Portály sa budujú nad existujúcu Java EE infraštruktúru, teda na webových kontajneroch. Pri nasadení portletovej aplikácie na portál sú portlety transparentne zabalené portálom do servletov. Z uvedeného vyplýva, že pre beh portletov nie je nevyhnutný aplikačný server. Pre efektívny beh portletov postačuje webový kontajner, ako je napríklad Apache Tomcat[19]. Isteže sa nejedná výlučne o Apache Tomcat, je možné využiť aj EJB[20] kontajner a i. Existuje obrovský počet naprogramovaných portálov podľa spomenutých štandardov. Niektoré boli vytvorené známymi firmami a iné sú ponúkané ako open source[21]. Tieto open source portály majú zvyčajne variantu s platenou podporou a garanciou.

---

18 Hypertext Markup Language, http://www.w3.org/html/

19 Apache Tomcat, http://tomcat.apache.org/tomcat-8.0-doc/config/resources.html

20 Enterprise JavaBeans, http://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/ejb.html

21 Open source, http://opensource.org/

Komerčné portály:

- IBM WebSphere Portal[22],
- Oracle WebLogic Portal[23],
- SAP Enterprise Portal[24] ,
- SharePoint[25].

Open source portály:

- Liferay Portal,
- Apache Jetspeed 2 Enterprise Portal[26],
- Exo Portal[27],
- JBoss Portal[28].

Medzi konkurenčné portály Liferay portálu stoja spomenuté štyri komerčné portály. Z nich za zmienku stojí SharePoint portál vyvinutý spoločnosťou Microsoft. Ide o webovú platformu slúžiacu ako centralizovaná náhrada viacerých aplikácií. Medzi jej typické prvky patrí prepojenie správy dokumentov a obsahu, vytváranie a správa intranetových a extranetových portálov. Ponúka množstvo nástrojov pre sociálne siete a vyhľadávanie podnikových informácií, plánovanie podnikových zdrojov (ERP[29]) a riadenie vzťahov zo zákazníkmi (CRM[30]), atď. Pre získanie viacerých informácií prikladáme domovskú stránku spomenutého portálu (https://www.discoversharepoint.com/).

---

22 WebSphere Portal, http://www-03.ibm.com/software/products/en/websphere-portal-family

23 WebLogic Portal, http://www.oracle.com/technetwork/middleware/weblogic-portal/overview/index.html

24 SAP Enterprise Portal, http://scn.sap.com/community/enterprise-portal

25 SharePoint , https://www.discoversharepoint.com/

26 Apache Jetspeed 2 Enterprise Portal, http://portals.apache.org/jetspeed-2/

27 Exo Portal, http://www.exoplatform.com/

28 JBoss Portal, http://jbossportal.jboss.org/

29 ERP, http://searchsap.techtarget.com/definition/ERP

30 CRM, http://en.wikipedia.org/wiki/Customer_relationship_management

1.3 Rozdelenie portálov

Portály je možné rozdeliť z rozmanitých hľadísk, no k základnej klasifikácii podľa kritéria použitia sem zaraďujeme tieto [3]:

Osobné portály - Tento druh je považovaný za jeden z najprijateľnejších. Ak klient nie je spokojný s predkonfigurovanými portálmi na internete a má v úmysle vytvoriť si vlastné stránky, osobný portál mu to umožňuje. Všetky stránky si spotrebiteľ dokáže prispôsobiť s ohľadom na svoje potreby a požiadavky. Vhodným príkladom sa môžu stať už existujúce portály ako je My Yahoo! [31], iGoogle [32], Seznam.cz [33] a mnohé iné.

Akademické portály - Akademické portály sa zameriavajú zvlášť na potreby študentov a učiteľov škôl. Elitným príkladom akademického portálu je uPortal[34] alebo SWITCH[35]. Pri porovnaní s Yahoo! a iGoogle, ktoré poskytujú prispôsobiteľné portály, uPortal poskytuje platformu pre spoluprácu. UPortal je verejne dostupný, spotrebiteľ si ho vie teda voľne stiahnuť. Navyše je založený na technológiách ako Java, XML, JSP a J2EE.

Regionálne webové portály - dodávajú také informácie, ktoré sa týkajú konkrétnej geografickej polohy. Regionálny webový portál ponúka údaje o podnikaní, životnom štýle, predpovedi počasia, o mapách ulíc, miestnom spravodajstve, nakupovaní a pod. Ukážkou opísaných portálov sú Rediff[36], South East of England[37], IN[38] .

Vládne webové portály - Vládne webové portály sú miestom poskytujúcim užitočné informácie týkajúce sa vlády a občanov, turistických údajov

---

31 Yahoo!, https://www.yahoo.com/

32 iGoogle, http://www.igoogleportal.com/portal/index.php

33 Seznam.cz, https://www.seznam.cz/

34 uPortal, https://www.apereo.org/uportal

35 Switch, https://www.switch.ch/

36 Rediff, http://www.rediff.com

37 South East of England, http://www.southeastofengland.com/

38 IN, http://www.in.gr

pre návštevníkov štátu, poznatkov pre podnikateľov a mnohé iné. Môže ísť o tieto portály: (http://www.usa.gov/ ; http://www.illinois.gov/).

Firemné webové portály - Firemné webové portály, alebo portály známe ako aj intranetové poskytujú ucelené informácie svojim zamestnancom a často im umožňujú prispôsobenie zobrazeného obsahu stránok. Súčasné podnikové, či firemné portály umožňujú reprezentatívnejšiu spoluprácu medzi divíziami spoločností, čím sa zvyšuje aj ich celková produktivita.

Doménovo–špecifické portály - Týmto názvom sa píšia portály, ktoré sú zamerané na konkrétne odvetvia obchodu, napríklad realitné s možnosťou uľahčenia nákupu a predaja nehnuteľností.

Športové portály - Tieto poskytujú znalosti pre všetkých horlivých nadšencov športu. Používateľ má možnosť sledovať zápasy naživo a v prípade, že ich nestihne, vyhľadá si reprízy, či výsledky z turnajov.

1.4 Liferay

Liferay Portál je podniková platforma určená pre budovanie Intranetových, ale aj Extranetových webových aplikácií založených na jazyku Java. Liferay platforma spĺňa štandard JRS-286 a predkladá používateľom robustnú platformu pre webové portály slúžiace všetkým klientom. Nezáleží na tom, či ide o desktopových alebo mobilných klientov. Liferay Portal je vyvinutá vďaka Open source metodiky jedincami z kútov celého sveta s distribuovanosťou pod licenciou GNU[39] Lasser General Public Licence. [11] Spomenutá webová platforma obsahuje vstavaný content management system, známy ako CMS umožňujúci spotrebiteľom vytvárať webové stránky a portály ako systém na správu obsahu, správu stránok a portletov pod spoločnou navigáciou. Jedná sa o výkonný, tzv. redakčný systém (CMS) bez ohľadu na to, či je obsah vo forme textu, obrázkov, fotografií, videa, zvuku alebo zdrojov informácií získavaných z externých zdrojov. CMS systém umožňuje majiteľovi stránok, resp. iným oprávneným osobám manažovať celý obsah bez potreby vedomostí v oblasti tvorby a produkcie stránok. Hovoríme o prezentačnej vrstve zabezpečujúcej zobrazenie obsahu prostredníctvom šablón, do ktorých je podľa stanovených podmienok automatický vkladaný obsah.

Hoci ponúka Liferay sofistikované programovacie rozhranie pre vývojárov, veľké programovacie zručnosti pre základnú inštaláciu a správu paradoxne nie sú potrebné. Okrem toho podporuje plug-in moduly zasahujúce do viacerých programovacích jazykov, ako je PHP[40], či Ruby[41]. Prednosťou tohto aplikačného rámca je starostlivosť o životný cyklus portletu, ktorý sa týka buď spustenia, ukončenia alebo komunikácie medzi klientom a portletom. Rovnako svoju starostlivosť prejavuje v oblasti interakcie medzi portletmi, samozrejme, ak to uzná za vhodné. Ako už bolo vyššie spomenuté, portál ktorý tu spomíname je založený na programovacom jazyku Java bežiaceho na každej platforme, na ktorej je spustený JRE[42] a aplikačný server. Liferay je k dispozícii na stránke https://www.liferay.com/downloads/liferay-portal/available-releases, s možnosťou stiahnuť si ho. Na tejto webovej adrese je ponúkaný s balíčkami obsahujú-

---

39 GNU, https://www.gnu.org/

40 Hypertext Preprocessor, http://php.net/

41 Ruby, https://www.ruby-lang.org/en/

42 Java Runtime Environment, http://en.wikipedia.org/wiki/Java_virtual_machine

cimi príslušné portálové servery, ktoré sú k portálu pribalené a nakonfigurované. Apache Tomcat, Glassfish[43], JBoss[44], Jetty[45] sú vhodnými príkladmi, ale samozrejme existujú ešte mnohé ďalšie. V balíčkoch je štandardne pribalená aj malá databáza Hypersonic slúžiaca pre ukladanie dát. Výhodou databázy je preferencia a jednoduchosť v zmysle inštalácie a konfigurácie, ktorá nie je v tomto prípade potrebná pri spustení. Pri nasadení systému do produkčného prostredia musíme databázu Hypersonic nahradiť inou. K dispozícii sú napríklad Derby[46], MySQL[47], Oracle[48], PostgreSQL[49] atď. Liferay svojou prenositeľnosťou umožní jednoduché spustenie a fungovanie celého balíčka. [12]

Liferay portál poskytuje maximálnu bezpečnosť, stabilitu, spoľahlivosť, optimálny výkon, servis a podporu. Portál je distribuovaný v dvoch edíciách [9] :

> **Liferay Portal Community Edition** (CE) - Ide o voľnú verziu s najnovšími funkciami a podporou prostredníctvom aktívnej komunity.
> **Liferay Portal Enterprice Edition** (EE) - Ide o verziu obdobnú Community Edition s navýšením v jej obsahu o ďalšie nástroje na monitorovanie, konfiguráciu portálu a platenú podporu zo strany výrobcu riešiaceho výskyt alebo zabezpečenie potencionálnych kritických, bezpečnostných a výkonnostných chýb. Zároveň zákazníkovi garantuje aktualizáciu, spätnú kompatibilitu a integritu portálu. Verzia Liferay Portal Enterprice Edition je spravidla k dispozícii pre nasadenie v rozpätí jedného až dvoch mesiacov po vydaní Community Edition.

Medzi hlavné funkcie portálu Liferay patria: [13]

> **Intengrácia frameworku** - Liferay Portal je platforma prezentačnej vrstvy, ktorá umožňuje užívateľom, správcom a vývojárom integrovať obsah a služby zo spodnej vrstvy (back-end), alebo starších aplikácií. Portál

---

43 Glassfish, https://glassfish.java.net/

44 JBoss, http://www.jboss.org/

45 Jetty, http://eclipse.org/jetty/

46 Derby, http://db.apache.org/derby/

47 MySql, http://www.mysql.com/

48 Oracle, https://www.oracle.com/database/index.html

49 PostgreSQL, http://www.postgresql.org/

podporuje služby pre integráciu ako SOAP[50], REST[51], RMI[52], XML-RPC[53], XML, JSON[54] a taktiež proprietárne API.

Dynamické Drag & Drop - Liferay portál patril medzi prvé portály, ktoré mali implementovanú funkciu potiahni a spusti. Táto technika sa používa v grafických prostrediach pri premiestňovaní objektov. Premiestňovanie objektov sa týkalo buď portletov alebo rôznych iných prvkov na portáli pomocou myši už spomínaným simplicitným potiahnutím a pustením.

Správa prístupu a schvaľovanie - táto funkcia umožňuje používateľovi vytvárať si autorské pracovné plány a zároveň definuje kvantum schvaľovacích obchodných požiadaviek a prevádzkových potrieb.

Single Sign On (SSO) - funkcia Single Sign On zabezpečuje jednotný prístup k obsahu portálu a aplikácií. Konkrétny používateľ je zvýhodnený v spôsobe prihlásenia sa, ktorý spočíva v jeho lineárnosti, čo značí jedno jediné prihlásenie sa z jeho strany. Následne webová platforma prepojí všetky používané systémy spolu takým spôsobom, aby boli k dispozícii po prihlásení sa prostredníctvom mechanizmu zabezpečenia SSO.

Auditácia a monitorovanie výkonu - úlohou auditácie a monitorovania výkonu je poskytnutie správcom portálu priame sledovanie a riadenie akýchkoľvek aktivít používateľov portálu. Výsadou tejto funkcie pre administrátorov je monitorovanie výkonnostných štatistík a optimalizovanie zdrojov pre portálovú stránku, takisto aj pre všetky portlety.

Liferay Sync - ponúka bezproblémovú synchronizáciu súborov z knižnice dokumentov do lokálneho počítača. Akékoľvek vykonané zmeny podniknuté v lokálnom adresári sa automaticky aktualizujú. Preferenciou Liferay Sync pre používateľov je publikácia a nezávislý prístup

---

50 Simple Object Access Protocol, http://www.w3.org/TR/soap/

51 Representational State Transfer, http://en.wikipedia.org/wiki/Representational_state_transfer

52 Java Remote Method Invocation, http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html

53 Remote Procedure Call protocol XML, http://en.wikipedia.org/wiki/XML-RPC

54 JavaScript Object Notation, http://www.json.org/

k dokumentom a súborom z viacerých prostredí a operačných systémov. Príkladom kompatibilného prostredia je Windows, či Mac OS. Eventualitou v operačných systémoch je iOS a Android, ktoré sú založené na mobilných platformách.

Vyhľadávanie a štítkovanie - ako už zo samotného názvu vyplýva, základom tejto funkcie je vlastné vyhľadávanie. Používatelia strategicky pátrajú po relevantných informáciách, ktoré voľne vyhľadávajú prostredníctvom osobitých a charakteristických kritérií v celom portáli. Hľadanie sa realizuje aj v rámci portletov a externých aplikácií.

Mnohojazyčná podpora - poskytuje viac ako 30 jazykov, z ktorých si dokáže používateľ ľubovoľne vyberať pomocou jednoduchého úkonu, čím je v podstate len prekliknutie jazyka. Taktiež má možnosť jednoduchého pridania ďalšieho potrebného jazyka.

OpenSocial - Podpora OpenSocial je inovatívnou subvenciou práve vďaka jej vlastnosti - vytvárať nové, jedinečné možnosti pre vývojárov. Správcovstvo a nasadzovanie sociálnych aplikácií priamo do stránok, či webových portálov patrí taktiež výlučne používateľom. V oblasti sociálnych aplikácií je žiaduce podotknúť ich postavenie na gadgets, teda miniaplikáciách.

Liferay Marketplace - Liferay Marketplace je verejným úložiskom aplikácií portálu Liferay. Zaujímavosťou je, že okrem nasadenia zdarma, ako v prípade ostatných funkcií, používateľom dáva navyše možnosť stiahnutia a nasadenia si rôznych portletových aplikácií aj za určitú finančnú čiastku.

Na istom fyzickom Liferay serveri existuje viacero portálov určených súčasne pre niekoľko druhov firiem. Vo vnútri každého portálu je daná možnosť definície používateľov do heterogénnych prístupových rol. Rozmanité rozdelenie rol je charakteristické štyrmi oblasťami a to rolou, organizáciou, komunitou a skupinou používateľov. Akákoľvek komunita, organizácia, či spoločenstvo môžu vlastniť verejné alebo súkromné stránky. Verejné, teda public stránky sú sprístupnené každému používateľovi portálu. Naopak súkromné stránky, teda private stránky umožňujú prístup výlučne používateľom špecifickej komunity, resp. príslušnej organizácii. Okrem už sprístupnených stránok existuje možnosť vytvárania aj jej vnorených hierarchií.

Pokiaľ by nebol spomínaný systém prideľovania práv dostačujúci, rozšírená podpora písania vlastných pravidiel od verzie 6 to umožní. Pravidlá je možné písať v jednoduchých programovacích jazykoch ako je napríklad Java. Konštantným spôsobom dokáže akákoľvek kompetentná persona limitovať niektoré portlety na základe vekovej kategórie, akceptovateľnosti podmienok používania alebo akejkoľvek inej ľubovoľnej vlastnosti.

Súčasnosť a jej etapa obľúbenosti sociálnych sietí umožnila portálu Liferay získať neobyčajnú funkcionalitu. Keďže hovoríme o sociálnych sieťach, význam Liferay sa odzrkadľuje v alternatíve publikovania jednotlivých portletov vrátane ich obsahu. Rozšírenie, ktoré nám podáva Social Office ponúka väčším firmám príležitosť na vybudovanie si obdobnej, no vlastnej autorskej sociálnej siete. Niet sa čo čudovať, že v dnešnej dobe mladými ľuďmi uznávaná a populárna myšlienka Marka Zuckenberga vytvoriť Facebook bola explicitnou inšpiráciou. Jedným z rozdielov medzi Facebookom a Social Office spočíva v tom, že orientácia Social Office sa koncentrovala hlbšie na oblasť prostredia podnikania. Okrem spomenutého umožňuje zamestnancovi náhľad na proces práce, či práve prebiehajúci projekt, v ktorom sú zaangažovaní i ostatní kolegovia vo firme alebo jej ďalší nezávislí spolupracovníci. Podobnosť Liferay a Facebooku je v zavedenom integrovanom účte Liferay-u ako aj stránky s novinkami silne pripomínajúcej už vychýrenú Facebook Wall.

Liferay sa ako platforma snaží ponúkať svojim vývojárom maximálnu kompatibilitu so všetkými Java technológiami. Práve pre portlety a Java EE špecifikácie každá portletová aplikácia dokáže použiť vlastnú sadu knižníc a technológií bez ohľadu na to, či ich samotný Liferay používa alebo nie. Spomenuté výhody poskytujú príležitosť pre portletové moduly, no moduly ako Ext takúto výhodu použitia Java EE technológií žiaľ nemajú. Ext moduly sú obmedzené iba na knižnice, ktoré používa jadro Liferay kódu Na druhej strane tie knižnice, ktoré môžu byť využité pri vývoji aplikácií poznáme pod názvami ako JQuery[55], Dojo[56], YUI[57], Vaadin[58] a iné.

---

55 jQuery, http://jquery.com/

56 DOJO, http://dojotoolkit.org/

57 YUI, http://yuilibrary.com/

58 Vaadin, https://vaadin.com/home

Na rozdiel od verzie 6 Liferay poskytuje vlastnú knižnicu s názvom AlloyUI[59], založenú na YUI3, ktorá ponúka veľkú sadu komponentov určených špeciálne pre efektívne a interaktívne aplikácie nasadené do prostredia spomenutého portálu. Jeho podpora siaha až k toolkit Vaadin – u, pomocou ktorého dokáže vývojaschopný človek vyprodukovať simplicitné aplikácie v AJAX – i bez nutnosti písania HTML kódu, alebo JavaScriptu. JavaScript je generovaný toolkitom a Liferay teda nevyžaduje žiadne špecifické a mimoriadne požiadavky potrebné pre použitie niektorých z frameworkov. Výber dostupných frameworkov a technológií umožňujúcich Liferay nasadiť pri vývoji aplikácií je neskutočne širokospektrálny. Na základe jeho šírky výberu si vývojár dokáže zvoliť nástroje, ktoré vie ovládať, na ktoré je zvyknutý a dokáže s nimi narábať pri vývoji už konkrétnych produktov. [9]

Medzi rozšírenia portálu Liferay patria tieto súčasti:

Liferay Themes - sú používané najmä v prípadoch, ak ide o zmenu vzhľadu. Akákoľvek aplikácia môže vlastniť minimálne jedno Theme skladajúce sa z CSS súborov a Velocity šablón.

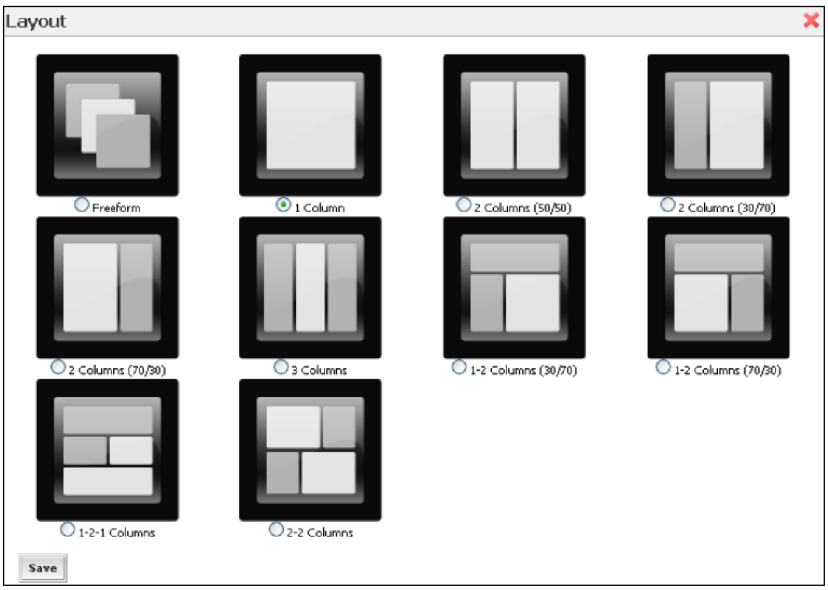
Layout - ich využitie nachádzame v mieste rozmiestnenia komponentov stránok. Liferay ponúka rôznorodé rozvrhnutia stránok, teda layout, rozmiestňujúcich portlety po stránke. Pre príklad bol použitý obrázok č. 2.

Hook rozšírenia - Hook rozšírenia slúžia programátorom pre prepísanie niektorých častí jadra Liferay s vlastnou, jedinečnou implementáciou. Podobný návrh pre rovnaký spôsob použitia mali Ext rozšírenia s tým rozdielom, že pomocou Ext modulu dokáže persona upraviť čokoľvek v jadre portálu a to len preto, že čítač pracuje v rovnakej triede a rozšírenia sú modulmi, ktorých prístup je ku všetkým jeho hlavným triedam zamietnutý. Je preto oveľa jednoduchšie spomenuté prepojenia napísať, nasadiť, prípadne zrušiť alebo udržiavať. [1]

---

[59] Alloy UI, http://alloyui.com/

Ext rozšítenie - poskytuje flexibilnú zmenu jadra Liferay portálu s dôrazom na zámenu akejkoľvek triedy za vlastnú implementáciu. Vývojárovi dodáva podstatne dostačujúcu silu nad celým portálom. Zároveň umožňuje prístup ku



Obrázok č. 2: Layout zobrazujúci rôzne možností rozdelenia portletov na stránke.

všetkým interným API Liferay. V podstate jedinou nevýhodou rozšírení, o ktorých je tu reč je, že pri upgrade portálu nemusia v novej verzii fungovať. Pri ich nasadení sa teda musí server reštartovať, aby sa všetky zmeny mohli dostatočne prejaviť. [1]

2. Firemné vzdelávanie

Vzdelávanie v dnešnej dobe patrí medzi dôležité aspekty potrieb jedinca i spoločnosti. Nielenže sa výchovou a vzdelaním zvyšujú možnosti obyvateľov pri uplatnení na trhu práce, ale znižujú sa aj náklady na školenie, ktoré financujú organizácie a podnikateľské subjekty. Medzi tvorivé prvky v organizáciách patria ľudské zdroje. Práve oni sa starajú, či už o výrobu alebo uplatnenie produktov na trhu, aj o plánovanie stratégie celej firmy. Navrhujú tovary a kontrolujú dosiahnutú kvalitu a veľa iných procesov nachádzajúcich sa vo firemnom prostredí. Ak chceme, aby sa spoločnosť presadila a prosperovala na trhu práce, musí poskytovať svojim zamestnancom získavanie nových, či už teoretických alebo praktických poznatkov. V tejto kapitole sa budeme zaoberať cieľmi a úlohami firemného vzdelávania a hodnotou ľudského kapitálu. Dotkneme sa aj metód vzdelávania a vzdelávania pomocou e-learningu.

M. Amstrong definuje firemné vzdelávanie ako „plánovaný proces modifikácie postojov, znalostí a skúseností smerujúcich k dosiahnutiu efektívneho výkonu v určitej činnosti, či okruhu činností. Ich cieľom z hľadiska práce je rozvinúť schopnosti jedinca a uspokojiť súčasné a budúce potreby organizácie týkajúce sa pracovnej sily“. [14]

V dnešnej rýchlo meniacej sa dobe sa ďaleko častejšie menia techniky v podnikaní, objavujú sa nové technológie a poznatky. Existuje niekoľko dôvodov prečo treba vzdelávanie zamestnancov v podnikoch realizovať. Patrí medzi ne nutnosť znižovania nákladov podniku, boom veľkého rozvoja informačných technológií, globalizácia a nutnosť pohybu. Pretože je premenlivosť ľudských potrieb na trhu čoraz výraznejšia je dôležité spomenúť, že starostlivosť o zamestnancov vytvára meno organizácie. Cieľom je poskytovanie vedomostí a zručností, ktoré zlepšujú splnenie požiadaviek a potrieb zákazníka. Pre uľahčenie angažovanosti pracovníkov má vzdelanie a príprava pracovníkov zahŕňať: [15]

- víziu a budúcnosť organizácie,
- ciele a politiky organizácie,
- vývoj a zmenu organizácie,
- zavádzanie inovácií a nových procesov.

O tom, že sa ľudia stávajú akousi hnacou silou celej organizácie a ovplyvňujú jej úspešnosť už nik nepochybuje. Práve preto si firemné vzdelávanie kladie za cieľ prípravu zamestnancov tak, aby táto úspešnosť bola pozitívna.

Ľudský kapitál je určitá hodnota, ktorej cena závisí od toho, ako sa používa. E. Hvizdová hovorí o tom, že inteligentná a efektívna pracovná sila môže patriť medzi hlavné konkurenčné výhody firmy, čo predstavuje potenciál tvorby zisku. Pri rozvoji ľudského kapitálu pozorujeme výchovno-vzdelávacie činnosti, prácu s talentovanými a profesionálnymi pracovníkmi a taktiež so stále „driemajúcimi" silami v podniku. Tento kapitál zahŕňa znalosti, ktoré jedinec buď na základe školského vzdelania, životných a pracovných skúsenosti alebo zámerného vlastného vzdelávania získal. [16]

P. Senge vo svojom diele „Piata disciplína" popisuje teóriu, že učiť sa môžu aj organizácie a to prostredníctvom jedincov. Ide teda o zamestnancov, ktorí nepretržito zvyšujú svoju kapacitu dosiahnutia želaných výsledkov. Sú podporované ich nové spôsoby myslenia a ambície sú kolektívne. [17]

Napriek tomu, že firemné vzdelávanie hrá veľký význam pri tvorbe zisku, vznikajú obavy týkajúce sa investovania a návratnosti finančných prostriedkov vložených v tejto oblasti. Ide hlavne o riziká odchodu pracovníka ku konkurencii vrátane vedomostí a schopností, ktoré nadobudol prostredníctvom rôznych školení financovaných danou organizáciou. Tým pádom je takáto investícia neefektívna a pre podnik stratová.

A. Kuchrčíková vo svojej publikácii „Ako prispieť ku konkurenčnej schopnosti podniku" radí, že ak sa podnik rozhodne investovať do vzdelávania, tak prispieva k rastu, výkonnosti a konkurencieschopnosti celého podniku. Taktiež zamestnávateľ musí myslieť na motiváciu svojich pracovníkov, čím zabraňuje odchodu vyškoleného pracovníka a tým zabezpečuje návratnosť investícií. V každom prípade je pre podnik výhodné investovať do vedomostí a zručností, ktoré sa dajú využiť len v danej organizácii. Tieto nadobudnuté zručnosti nezvyšujú hodnotu pracovníka pre iných zamestnávateľov, ale vzniknuté hodnoty zvýšia rozhodnutie zotrvať u školiaceho zamestnávateľa. [18]

Pri identifikácii vzdelávacích potrieb sa vychádza zo strategických cieľov organizácie, požiadaviek pracovného miesta, individuálnych kompetencií zamestnanca a z pracovného výkonu. Existuje veľká škála vyskúšaných a efektívnych

edukačných metód. Každá metóda má svoje klady aj zápory a preto musíme brať pri výbere metódy do úvahy rôzne faktory. Medzi základne faktory ovplyvňujúce voľbu patrí veľkosť, lokalita firmy a rozpočet. Faktory ovplyvňujúce výber vzdelávacích metód podľa Kucharčikovej A. sú: [19]

Cieľ učenia: vedomosti, schopnosti, zručnosti, postoje a hodnoty.
Ľudské faktory: lektori, prostredie a účastníci.
Predmet a obsah: požiadavky firmy, interdisciplinárne problémy.
Časové a materiálne faktory: čas, financie, vzdelávanie.
Princípy učenia: motivácia, aktívne zapojenie, individuálny prístup, spätná väzba, prenos.

J. Koubek rozdeľuje metódy vzdelávania do dvoch základných skupín. [20]

Vzdelávanie na pracovisku „on the job" - Formy vzdelávania, tzv. „on the job" sú vhodnejšie pre také spoločnosti, ktoré si môžu dovoliť na určitý čas zamestnať ľudí, ktorí by zabezpečovali školenia. Tieto školenia prebiehajú na konkrétnom pracovnom mieste a pri vykonávaní bežných pracovných povinností. Medzi výhody patrí skutočnosť, že pracovník uplatňuje potrebné informácie bezprostredne v praxi a nie je nutné aby kvôli rekvalifikácii niekam cestoval. Takýmto spôsobom sa uplatňuje individuálny prístup ku školenému zamestnancovi, čím sa dosahuje oveľa väčšia efektivita. Je potrebné, aby bol v takomto prípade vypracovaný individuálny vzdelávací plán, ktorý je šitý danému zamestnancovi na mieru. Musíme spomenúť, že efektívnosť takéhoto vzdelávania závisí od správneho výberu školiteľov, pretože môže dôjsť k odovzdaniu nevhodných pracovných návykov. Medzi najbežnejšie metódy, s ktorými sa môžeme stretnúť na pracovisku patrí:

- inštruktáž pri výkone,
- asistovanie,
- rotácia práce (cross training),
- koučovanie (coaching),
- mentorovanie (mentoring),
- vzájomné konzultovanie (counselling),
- pracovná porada.

Vzdelávanie mimo pracoviska „off the job" - Pri vzdelávaní „off the job" je zamestnanec vyslaný mimo pracoviska, aby zvýšil svoju kvalifikáciu a zároveň

získal nové poznatky, čím by udržal krok s novými trendmi v jeho špecializácii. Väčšinou je takéto vzdelávanie vhodnejšie pre manažérov a špecialistov príslušnej organizácie. Medzi výhody formy „off the job“ vzdelávania patrí zameranosť školení na konkrétnu problematiku a tiež prednášajúci špecialisti s dlhoročnou praxou a skúsenosťami. Takýto druh edukácie je menej formálny a zamestnanci sa stávajú prístupnejšími pre nové myšlienky, nasmerovanie, či nové otázky a témy. Zamestnanci sa dokážu rýchlejšie uvoľniť a nabrať vedomosti potrebné pre svoj odbor. Tak ako pri predchádzajúcej forme vzdelávania má aj „off the job“ nevýhodu a to z finančného hľadiska. Je náročnejšia na čas, počas ktorého sú títo pracovníci školení a svoju prácu nemôžu vykonávať.

Medzi typy metód používané mimo pracoviska patria:

- prednášky a semináre,
- metóda demonštrovania,
- samoštúdium,
- workshop,
- brainstorming,
- simulácie, hranie rolí a hier,
- prípadová štúdia,
- vzdelávanie pomocou počítača.

Edukácia takéhoto druhu je v prvom rade o poznaní potenciálu svojich zamestnancov, pretože bez tejto informácie nie je nik schopný sa správne a objektívne rozhodnúť. Teda zvolenie správnej formy podnikového vzdelávanie je veľmi dôležité.

2.1 E-learning

E-learning resp. vzdelávanie pomocou počítača sa za posledné roky vďaka rýchlo napredujúcim informačno-komunikačným technológiám dynamicky rozvinul. Vzhľadom k takému dynamickému vývoju je e-learningový proces vzdelávania neustále inovovaný.

Profesor M. Turčáni vo svojich prednáškach definuje e-learning ako fenomén vzdelávania 21. storočia. Nielenže nás ohromuje svojim širokým záberom, ale priťahuje nás taktiež rôznymi technickými možnosťami. E-learning nazval „kamarátom, ktorý šetrí náš čas i peniaze, ktorý prináša dosiaľ netušené a nepoznané možnosti rozvoja našej vlastnej osobnosti“. [21]

Autori Pedagogického slovníka predstavujú definíciu e-learningu ako typ učenia, pri ktorom sú získané a používané vedomosti uľahčené a distribuované prostredníctvom elektronických zariadení. [22]

Taktiež sa môžeme stretnúť s inými publikáciami, ktoré tvrdia že e-learning nie je nič prevratné. Ide len o efektívne využívanie informačných technológií v procese edukácie. Pri zhrnutí rôznych definícií môžeme takéto vzdelávanie nazvať ako elektronické vzdelávanie, ktoré sa dá realizovať vďaka rôznym informačným technológiám a môžeme ho zaradiť medzi účinné prostriedky vzdelávania a rozvíjania študentov.

Základné výhody elektronického vzdelávania pri porovnaní s bežným vzdelávaním sú:

- zníženie nákladov na prenájom učební, tlače a distribúcie študijných materiálov, dopravy školiteľa alebo školeného a cena lektora,
- štúdium je časovo nezávislé a školiaci pracovník si sám vyberá čas, miesto a tempo vzdelávania,
- všetci študenti sú hodnotení podľa rovnakých kritérií a zvyčajne získajú okamžitú informáciu o svojich dosiahnutých výsledkoch,
- využitie internetu ako komunikačno-vzdelávacieho kanálu.

Medzi najčastejšie uvádzané nevýhody elektronického vzdelávania patria:

- nenahraditeľný prístup k informačným technológiám (internet a výpočtová technika),
- väčšie počiatočné náklady spájajúce sa so školením lektorov a s vytváraním kurzov,
- keďže väčšinou ide o písomnú formu, chýba osobný kontakt s kolegami a tútorom,
- náročnejšie na preverenie kvality vedomostí, ktoré študujúci nadobudol počas školení.

Slovo e-learning sa veľmi často nespája len s univerzitami pri externej či dištančnej forme štúdia. Dnešná doba ponúka populárne využitie v rámci celoživotného vzdelávania. Účastníci sú často pracujúcimi ľuďmi a preto nemôžu stále cestovať za štúdiom a venovať sa mu v pevne stanovený čas. Takýmito možnosťami elektronického vzdelávania sa zaoberá i podnikateľský sektor.

O zavedenie spomenutého druhu vzdelávania uvažujú najmä organizácie, ktoré potrebujú rekvalifikovať alebo vyškoliť väčší počet pracovníkov v určitom časovom intervale. Medzi najznámejšie spoločnosti, ktoré sa zapojili do využívania moderných metód vzdelávania patrí Hewlett-Packard, Cisco, IBM. Na domácom trhu našli jeho využitie aj firmy zabezpečujúce školenia pracovníkov o ochrane a bezpečnosti zdravia pri práci (BOZPO), T-mobile, Allianz Slovenská poisťovňa, Československá obchodná banka a iné. Tieto spoločnosti nielenže vzdelávajú svojich vlastných zamestnancov, ale otvorili svoje brány vzdelávania i pre širokú verejnosť. Keď zhrnieme možnosti využitia elektronického vzdelávania v podnikateľskom sektore, výsledkom budú školenia pri prijatí nových pracovníkov, rôzne vnútorné firemné školenia a školenia o nových predpisoch, či zavádzaní nových produktov na trh.

2.2 LMS

Aby sa dosiahla lepšia efektívnosť pri správe vzdelávacích procesov je potrebné nevyhnutne postupovať podľa určitých overených návrhových vzorov. Preto vznikol aj z takýchto dôvodov systém na riadenie výučby (LMS). Za LMS môžeme označiť ľubovoľný softvér, alebo balík aplikácií s cieľom integrovať nástroje pre vytváranie virtuálneho vzdelávacieho prostredia. Väčšina LMS produktov je založená na web technológiách, ktoré sú dodávané rôznymi výrobcami, čo môže mať za následok rôznorodosť k poskytovanej funkcionalite.

Medzi základné úlohy môžeme považovať nasledujúce činnosti:

- evidencia a správa študentov,
- evidencia a správa kurzov a zoznamov vzdelávacích objektov,
- správa študijných materiálov,
- testovanie a hodnotene študentov,
- sledovanie progresu študenta v procese vzdelávania,
- spravovanie prístupových práv,
- komunikačné nástroje určené pre riadenie výučby (chat, diskusné fórum, wiky, nástenka).

Pre lepšie porozumenie softvérových služieb v oblasti vzdelávania a podpory výučby vo virtuálnom svete prikladáme obrázok č. 3.

Obrázok č. 3: Softvérové služby v oblasti vzdelávania a výučby[60].

Medzi výrazné rysy systému na správu výučby patrí jednoduchosť ovládania systému. Študentom je potrebné zaistiť nepreťažovanie štúdiom takéhoto LMS systému. Mohlo by to byť na úkor času, ktorý dokážu naplno venovať štúdiu kurzu samotného. Pre študenta takýto vzdelávací systém predstavuje vlastné virtuálne prostredie, v ktorom nájde všetky dostupné prostriedky (ako kurzy, pokyny pre štúdium, diskusné fóra k jednotlivým témam, testy a iné), ktoré by našiel i v skutočnej triede.

Systém na správu výučby je vďaka medzinárodnej sieti Internet široko dostupný. Nie je viazaný na konkrétnu organizáciu zabezpečujúcu vzdelávanie, či región alebo počítač. V praxi to pre nás znamená, že takéto počítačové aplikácie nám slúžia pre získavanie a výmenu informácií.

J. Dlouhý a A. Jančařík kladú veľký dôraz na výber LMS platformy. Organizácia poskytujúca vzdelávanie musí posúdiť množstvo atribútov, ktoré ovplyvňujú výber konkrétneho systému. Medzi prvé miesto sa zaraďuje: [23]

---

[60] Zdroj: http://www.icsilhealth.in/smarteachlms.html

- finančná dostupnosť (komerčný softvér vs. Open Source),
- nároky na hardvér a softvér (webový server, dátové úložiská a pod.),
- nároky spájajúce sa s odbornými znalosťami pri inštaláciách alebo aktualizáciách produktu,
- možnosť zmeny jazyka do vyžadovaných jazykov,
- stabilitu a odolnosť systému voči softvérovým útokom,
- intuitívnosť a jednoduchosť obsluhy,
- dostupnosť technickej pomoci,
- možnosti vkladania hotových elektronických modulov z iných LMS systémov.

Výber vzdelávacích modulov je dnes veľmi veľký. Medzi najrozšírenejšie systémy svetových výrobcov spĺňajúce všetky hore popísané funkcionality patria LMS uvedené v prílohe B.

LMS môžeme rozdeliť podľa nasledovných systémov:

Management Systems (CMS) - systém pre riadenie kurzov sa najviac používa na vysokých školách. Tieto systémy sú podkategóriou LMS a sú primárne určené pre on-line vzdelávanie. To znamená, že poskytujú umiestnenie študijných materiálov a uľahčujú správu kurzov, všetkých dostupných vzdelávacích a podporných nástrojov v prostredí internetu. Medzi značné výhody oproti LCMS a ELMS sa považuje relatívne nízka cena a použiteľnosť. [24]

Learning Content Management Systeme (LCMS) - systém pre tvorbu obsahu štúdia poskytuje multi-používateľské prostredie, v ktorom vývojári, autori a celé tímy realizátorov e-kurzov môžu vytvárať, ukladať, spracovávať, opätovne používať a dodávať digitálny vzdelávací obsah z centrálneho úložiska. LCMS sa zameriava na vývoj, správu a publikovanie obsahu, ktorý je zvyčajne dodávaný cez LMS. Takýmto spôsobom môže byť výučba oveľa lepšie prispôsobiteľná podľa potrieb pre konkrétnych študujúcich bez zbytočných duplicít. Technológia LCMS býva zvyčajne súčasťou systémov určených na riadenie podnikového vzdelávania alebo môže byť použitá ako samostatná aplikácia pre iniciatívne učenie, ktoré si vyžaduje rýchly vývoj a distribúciu vzdelávacieho obsahu. [25]

Enterprise Learning Management Systeme (ELMS) - systémy na riadenie podnikového vzdelávania sa presadzujú vo väčších firmách. Poskytujú kontinuálne

vzdelávacie príležitosti, ktoré zlepšujú pracovné sily prostredníctvom monitorovania a poskytovania vzdelávacích akcií, ako sú elektronické školenia. Sú to rôzne vzdelávacie aktivity slúžiace na prispôsobenie pracovnej sily danej organizácie v meniacom sa konkurencie schopnom prostredí. Nie len že efektívne uľahčuje správu vzdelávania, ale poskytuje aj jasný smer k organizačným cieľom, podporuje rozvoj zamestnancov, čo má za následok zvýšenie úrovne a výkonnosti organizácie. Pri porovnaní CMS a ELMS sa ceny pohybujú dosť vysoko. Podnikom ale i zamestnávateľom poskytujú oveľa viac možností pre rozvoj vzdelávania a integrácie do svojich podnikových informačných systémov.

Hranice medzi ELMS, LCMS, CMS nie sú pevne vymedzené. Môže sa nám zdať, že ide o veľmi obdobnú záležitosť. Tieto tri spomínané systémy pre vzdelávanie či už v malej alebo veľkej miere majú integrované rôzne softwarové nástroje pre podporu elektronického vzdelávania.

Pri rozhodnutí integrácie vzdelávacieho systému do firemného alebo univerzitného prostredia sú nám poskytnuté tri možnosti. Ide o možnosť využiť buď cudzí systém riadenia výučby, možnosť zakúpenia komerčného systému alebo vývoj vlastného.

Prenajatie LMS v cudzom prostredí nám môže ušetriť náklady, ktoré sú spájané s prevádzkou a údržbou takéhoto systému. Avšak pri takomto výbere strácame plnú moc nad spravovaním celého systému a serveru, na ktorom je nainštalovaný. Toto riešenie je vhodné najmä pre menšie inštitúcie, ktoré používajú školenia pre hromadné alebo jednorazové aktivity. Spomenuté riešenie možno využiť aj v prípade záujmu o doplnenie štandardného školenia použitím LMS.

Zakúpenie LMS organizácie sa väčšinou volí pre pravidelne sa opakujúce školenia. V súčasnej dobe na trhu pôsobí veľké množstvo vyvinutých komerčných, alebo Open Sourse systémov, ktoré ponúkajú veľké množstvo modulov a nástrojov. Takouto širokou škálou ponúkaného príslušenstva dokážu splniť požiadavky i u náročných používateľov. Firma zaoberajúca sa systémami na riadenie výučby dokáže podrobne analyzovať a navrhnúť optimálne riešenia pre zákazníkov.

Vyvinutie vlastného LMS je vhodný len pre veľké organizácie, ktoré sa rozhodli takýto projekt podporovať dlhodobo a vo veľkom rozsahu. Ide o náročný

projekt vyžadujúci si silné programátorské zázemie. Medzi výhody pri zvolení tohto spôsobu patrí možnosť zohľadnenia všetkých zistených požiadaviek na vytvorenie LMS.

2.3 Dostupné produkty firemného vzdelávania

V podnikateľskej brandži sa stále viac a viac uplatňujú rôzne profesionálne vzdelávacie systémy. Tie sa stávajú nenahraditeľnou súčasťou ich informačných a podnikových portálov. Pri pohľade na celosvetových producentov LMS systémov dostaneme dostatočne širokú ponuku produktov. Medzi nimi môžeme nájsť systémy vyvíjané veľkými firmami (napr. IBM), rôznymi komunitami (nájdeme ich podľa značky open source), univerzitami a menšími firmami vyvíjajúcimi svoje vlastné riešenia. Avšak nie každý spomenutý produkt je z týchto skupín vhodný pre všetky typy vzdelávania. Niektoré zahŕňajú potreby vysokých škôl a iné sa špecializujú pre pokrytie potrieb špeciálne pre vzdelávanie zamestnancov vo firmách. Medzi prvú štvoricu najznámejších LMS systémov patria: [26]

- Moodle (cca 73 mil. používateľov),
- Edmodo (cca 20 mil. používateľov),
- Blackboard (cca 20 mil. používateľov),
- SkilSoft (cca 19 mil. používateľov).

K najznámejším českým dodávateľov komerčných LMS systémov môžeme zahrnúť tieto produkty:

eDoceo[61]- je prevádzkovaný v komerčných organizáciách a v štátnej správe. Firma taktiež disponuje konzultáciami a zavádzaním vzdelávacích procesov do organizácií. Organizácie využívajúce služby eDoceo sú napríklad: České aerolínie, Ministerstvo vnútra ČR, UniCredit Bank Czech Republic, Škoda Auto, UVN Střešovice.

Eden[62] - vznikol v spolupráci spoločnosti Renel s. r. o. s Univerzitou Karlovou v Prahe. Medzi hlavné referencie firmy patrí Fakultní nemocnica Brno, Rezort životního prostředí, DAGO .

---

61 eDeceo, http://www.edoceo.cz/

62 Eden, http://www.eden-online.org/

iTutor[63] - Dodávateľ Kontis poskytuje širokú škálu služieb v oblasti firemného vzdelávania. Ponúka hotové kurzy, vývoj kurzov na kľúč, implementáciu, systémovú integráciu a iné. Patrí medzi najrozšírenejšiu e-learningovú platformu v Českej a Slovenskej republike (cca 70% trhu). [27]

V Slovenskej republike môžeme nájsť zastúpenie vzdelávacieho systému iTutor vo firmách: Aegon, ČSOB Banka, Generali, Orange, T-COM, T-Mobile Slovensko, VUB, Východoslovenská energetika, Železnice Slovenskej Republiky a iné.

V Českej republike ide o pomerne väčšie používanie spomenutého vzdelávacieho systému: AVG Technologies, Cambridbe Business School, CocaCola, Česká národní banka, České dráhy, ČEZ, ČSOB, Etiopska armáda, KIA Motors, Lázne Drakov, McDonald´s, Ministerstvo obrany ČR, Škoda, T-Mobile Tesco a iné.

Spôsobmi implementácie LMS systémov sme sa zaoberali v kapitole 2.2 LMS. V tejto časti sa zameriame na porovnanie komerčných a open source vzdelávacích systémov.

Open source je technológia takzvaného „otvoreného" zdrojového kódu, ktorý je k dispozícii pre širokú verejnosť a vývojári si ho môžu podľa svojho účelu prispôsobiť. Medzi hlavné dôvody pre zvolenie LMS systému patrí:

- firma používajúca komerčné riešenie LMS systému, ktorá sa musí spoliehať na údržbu, servis a odstránenie vyskytnutých chýb dodávateľom,
- dostupnosť kódu širokej verejnosti vývojárom, ktorí čítajú a skúmajú kód, čo má za následok rýchlejšie identifikovanie a odstránenie chýb,
- flexibilnejšie riešenie pre prispôsobenie ku konkrétnej firemnej politike,
- technológia nie je zaťažená rôznymi licenčnými alebo hardwarovými nákladmi.

Riešenia Open source majú svojich priaznivcov, ale aj odporcov. I keď je takéto riešenie finančne oveľa menej náročné, nemôžeme očakávať nulové náklady. Je potrebné zabezpečiť a vyškoliť zamestnancov, ktorí budú zabezpečovať technickú časť systému a lektorov, ktorí budú vytvárať a spravovať kurzy. Pri nasa-

---

[63] iTutor, http://www.itutor.com/

dzovaní môže dôjsť k rôznym komplikáciám a takéto riešenie môže byť pre neskúsených technikov neefektívne a pomalšie pri porovnaní s komerčným riešením. Pre prehľad open source produktov určených k vzdelávaniu bol vypracovaný zoznam niektorých používaných systémov. Ide o nasledovný zoznam:

- Atutor - www.atutor.ca,
- Dacebo - www.dacebo.org,
- Dokeos - www.dokeos.com,
- dotLRN - www.dotlrn.org,
- eFront - www.efrontlearning.net,
- Ilias - www.ilias.de,
- Moodle - www.moodle.org,
- Sakai - www.sakaiproject.org. [28]

Za výhody komerčných vzdelávacích systémov môžeme pokladať nadobudnuté skúsenosti a odborné znalosti v tvorbe, implementácii a nasadzovaní vzdelávacích systémov. Firmy predkladajúce spomenuté riešenia ponúkajú rôzne balíčky služieb, ktoré zahŕňajú široké spektrum od vytvárania kurzov a školenia používateľov, cez analýzu a monitoring celého vzdelávacieho procesu až po nasadenie do podniku.

Ich najviditeľnejšou nevýhodou je cena a možné komplikácie, ktoré môžu nastať pri nekompatibilite ponúkaných komponentov a nástrojov do požadovaných firemných informačných systémov. Na záver pridávame zoznam niektorých komerčných LMS systémov:

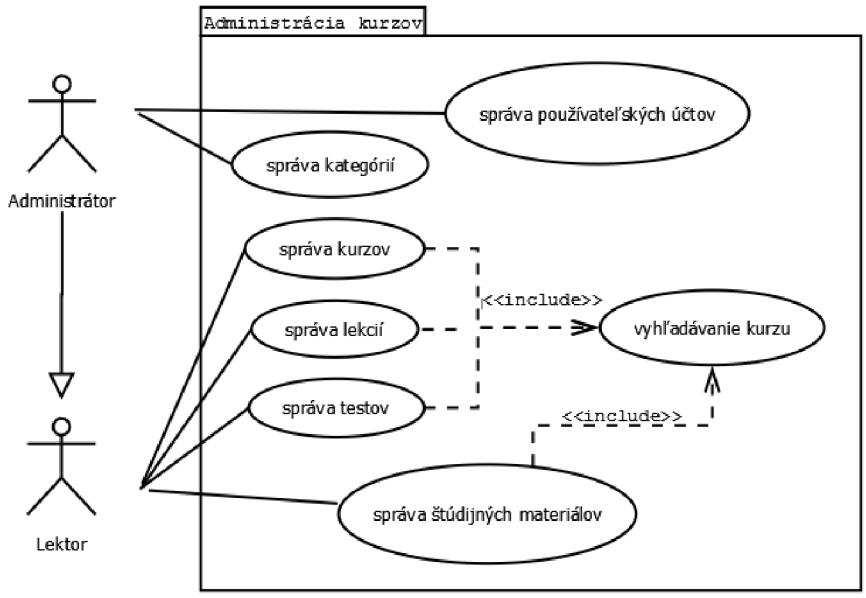
- Blackboard - www.blackboard.com,
- CertpointVLS - www.certpointsystems.com,
- Desire2Learn - www.desire2learn.com,
- iTutor - www.e-learn.cz,
- Joomla LMS - www.joomlalms.com,
- Mzinga - www.mzinga.com,
- Oracle iLearning – ilearning.oracle.com,
- Plateau - www.plateau.com,
- Saba - www.saba.com,
- SharePointLMS - www.sharepointlms.com.

3. Analýza a návrh

Na základe teoretickej časti a rozboru LMS systémov sme sa dozvedeli o základnej funkcionalite e-learningových kurzov pre podnikové vzdelávanie. Prostredníctvom získaných vedomostí môžeme vytvoriť návrh celej aplikácie. Výsledkom nebude plnohodnotná aplikácia. Bude sa jednať o prototyp, ktorý bude obsahovať základné jadro funkcií, ktoré môžeme nájsť vo väčšine vzdelávacích systémov. Pre analýzu požiadavky využijeme štandardný jazyk UML[64], ktorý nám bude slúžiť pre vizualizáciu, špecifikáciu a konštrukciu našej aplikácie. Tento jazyk je čitateľný pre analytikov, dizajnérov, programátorov a môže byť použitý pri všetkých procesoch životného cyklu vývoja softvéru. V tejto kapitole sa budeme ďalej zaoberať analýzou a návrhom riešenia praktickej časti diplomovej práce.

3.1 Špecifikácia aplikácie

Aplikácia určená pre zabezpečovanie vzdelávacieho procesu vo firemnom prostredí bude obsahovať kurzy s možnosťou zadelenia do príslušných kategórií. Administrátor e-learningu bude mať možnosť výberu. Buď bude celá kategória



Obrázok č. 4: Stručný diagram prípadov použitia pre administráciu kurzov.

64 Unified Modeling Language, http://www.uml.org/

kurzov určená pre voľné pridávanie kurzov (jedná sa o doplnkové štúdium pre používateľov, ktorí majú záujem o zvýšenie svojej úrovne vzdelania) alebo budú kurzy pridávané administrátorom manuálne podľa potreby. Každý kurz bude obsahovať banku dokumentov, banku testov, lekcie a skupinu používateľov, ktorým bol pridaný kurz (viď obr. č. 4).

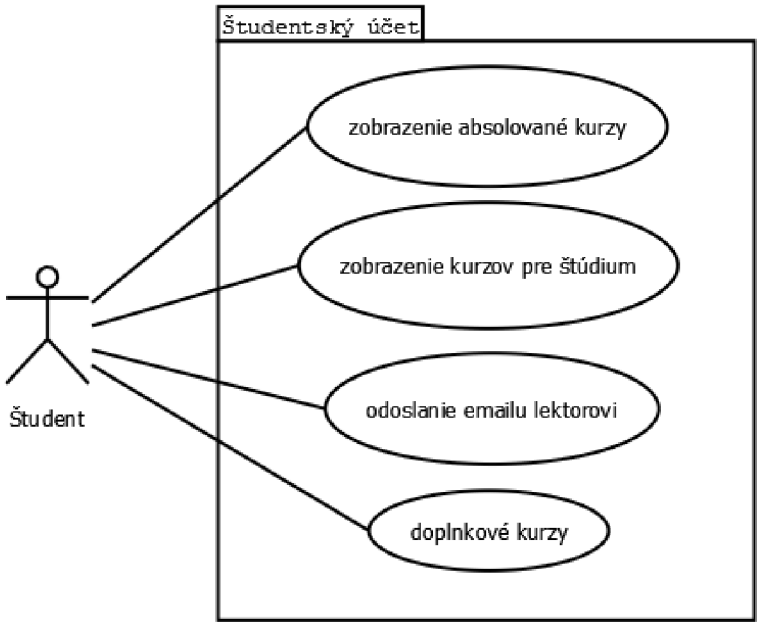
V banke dokumentov budeme môcť zoskupovať študijné materiály do balíkov. Dokumenty a médiá vkladané do týchto balíkov budú nahrávané do DMS Liferay pomocou integrovaného systému v platforme. Lektor pri zdieľaní dokumentov určených k vzdelávaniu, ktoré sú uložené mimo portálu, využije odkaz. Odkaz vloží daný URL link do banky dokumentov. Túto funkciu bude môcť využiť pre súbory, ktoré sú v podniku už nahrané v repozitári Dokumenty a Médiá. K dispozícii bude i tvorba lekcií a ich náhľadu v podobe knihy. Napĺňanie obsahu lekcií poskytnutým portálom bude zabezpečovať wysiwyg[65] editor. Pri tvorbe testovacích otázok bude k dispozícii banka otázok, do ktorej budú vkladané vytvorené otázky. Tieto otázky budú podľa potreby pridávané do bánk otázok a tie následne priraďované kurzom. Kurzy budú obsahovať viacero testovacích balíčkov. Každý takýto balíček bude obsahovať vlastné nastavenie, v ktorom môže lektor zvoliť spôsob výberu a miešania otázok. Všetky kurzy budú mať spôsob ukončenia na základe testov. Pri každom kurze budeme môcť zistiť používateľov zapísaných do kurzu. Aplikácia bude napojená na skupiny používateľov portálu, na základe ktorých bude aktivovať vzdelávacie účty a spravovať ich prístupy k celému vzdelávaniu. Prístup k spravovaniu kurzov bude rozdelený do dvoch správcovských rol. Prvá rola bude určená pre administrátora vzdelávania, ktorý bude mať prístup ku všetkým funkciám tvorby kurzov. Druhá rola bude určená lektorom, ktorým sa budú zobrazovať iba prístupy nimi vytvorených entit. Je nevyhnutné spomenúť, že v aplikácii budeme môcť upravovať a spravovať všetky spomenuté špecifikácie. Všetky vymenované entity budú spravované (vytvárané, mazané, zobrazované a upravované) v rámci administračného prostredia LMS.

Z pohľadu používateľa - študenta budú spomenuté entity zobrazované na základe ich vlastných nastavení. Každý kurz bude graficky označený pre rýchlu identifikáciu stavu, v ktorom sa kurz nachádza (vykonávaný, doposiaľ nena-

---

[65] Wysiwyg, http://en.wikipedia.org/wiki/WYSIWYG

vštívený, absolvovaný a uzamknutý z dôsledku prekročenia časového limitu). Študent bude mať k dispozícii možnosť zaslania správy pre lektora pre získanie informácie tykajúcej sa štúdia. Taktiež bude mať prístup ku kategóriám určeným pre voľné pridávanie kurzov. Lepší náhľad funkcií určených pre študenta môžeme nájsť na obrázku č. 5. Pre lepšiu zrozumiteľnosť a prehľad funkcií jednotlivých častí systému boli vypracované diagramy prípadov použitia, ktoré čitateľ môže nájsť prílohe na CD.



Obrázok č. 5: Diagram prípadov použitia pre študentský účet.

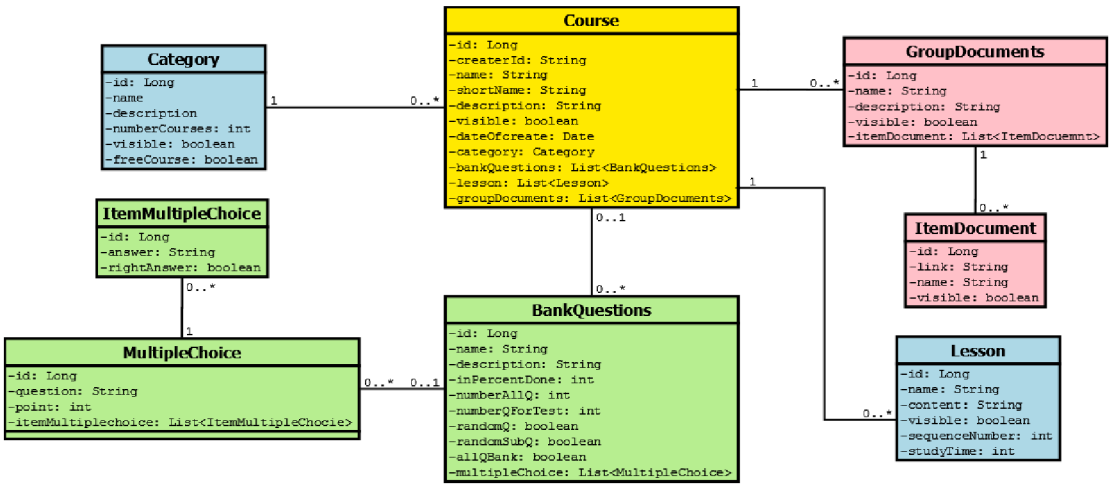
3.2 Dátový model

Na základe analýzy a špecifikácie boli identifikované nasledujúce potrebné entity, s ktorými budeme môcť pracovať v rámci celej aplikácie vzdelávania pracovníkov. Entity určené na spravovanie:

- kurzov (môžeme tu nájsť rôzne atribúty spojené so spravovaním kurzov),
- kategórií (entita, ktorá zlepšuje vyhľadávanie kurzov podľa danej kategórie a obsahuje informácie o možnosti voľného priradenia kurzu daným študentom),
- lekcií (obsahuje študijné informácie, ktoré sú zviazané s konkrétnym kurzom),

- banky dokumentov (spája študijné materiály do jednotlivých skupín, ktoré sú následne priraďované ku konkrétnym kurzom),
- dokumentov (entita, ktorá reprezentuje odkazy na študijné materiály uložené mimo portálu ale i odkaz na vložené súbory v integrovanom DMS portáli),
- banky otázok (entita ktorá uchováva informácie o nastavení testu, udržuje viacero otázok v jednom balíku),
- otázok (entita, ktorá reprezentuje otázku s bodovým ohodnotením),
- odpovedí (môžeme tu nájsť odpovede ktoré sú priradené k jednotlivým otázkam),
- učtu používateľa (nájdeme tu informácie o tom, aké kurzy má používateľ absolvované, na ktorých ešte pracuje alebo dátum jeho registrácie do LMS systému),
- položky kurzu (entita obsahuje informácie o jednotlivých kurzov, ktoré má študent pridané vo svojom študijnom pláne),
- pokusov testov (entita, ktorá sa viaže k testovaniu a obsahuje atribúty o konkrétnom pokuse a získaných bodov pri testovaní).

Pre celistvý pohľad na objekty spojené s kurzom bol vypracovaný diagram entít, ktorý je k dispozícii na obrázku č. 6. Diagram zobrazujúci entity určený k správe účtov je priložený v prílohe na CD. Entity zahrnuté do diagramov predstavujú základné objekty ukladané do databázy a každá z nich reprezentuje relačnú tabuľku s atribútmi.



Obrázok č. 6: Diagram tried zobrazujúci entity spojené s kurzom.

3.3 Návrh obrazoviek

V ďalšom postupe pri návrhu aplikácie sme sa zaoberali vytvorením prototypu obrazoviek. Obrazovky pomáhajú zobraziť do reálnej podoby získané požiadavky pri analýze. Prototypovanie býva vykonávané rôznymi spôsobmi. Spomenieme zopár základných spôsobov, medzi ktoré patrí: zošit, grafický program alebo nástroj určený na návrh grafických používateľských rozhraní.

Kčasto používaným spôsobom patrí ceruzka a papier. Návrhári po ňom často siahnu, pretože rukou nakreslia čo chcú a jedná sa o veľmi rýchly proces. Avšak pri editácii a prevedení nakresleného návrhu na digitálny sa často vyskytuje problém.

K ďalším bežným spôsobom patrí kreslenie vzhľadov obrazoviek pomocou grafických programov, ako je napríklad PhotoShop[66]. Nevýhodou je licencia a nutnosť grafických skúseností. Za výhodu sa dá považovať fakt, že takto vykonaný návrh je už veľmi jednoduché dokončiť do výslednej podoby, ktorá sa použije pri vytváraní grafických rozhraní.

Poslednou možnosťou vytvárania návrhu obrazoviek sú nástroje určené na tvorbu prototypov. Ide o nástroje, ktorými dokáže rýchlo a efektívne či už vývojár, analytik alebo manažér zaznamenať podobu GUI[67]. Takýto program ponúka komponenty, ktoré môžu byť jednoducho editované a pridané na kresliacu plochu. Pri návrhu prototypov obrazoviek bol použitý práve takýto nástroj, Balsamiq[68]. Jedná sa o pomerne jednoduchý a rýchly program, ktorý je dostupný pre Windows, Mac OS X aj Linux. Obsahuje podporu pluginov pre JIRA[69], Confluence[70] a iné. Výhodou je možnosť exportovania výsledných obrazoviek do PNG[71] alebo PDF formátu.

---

66 Photoshop, http://www.photoshop.com/

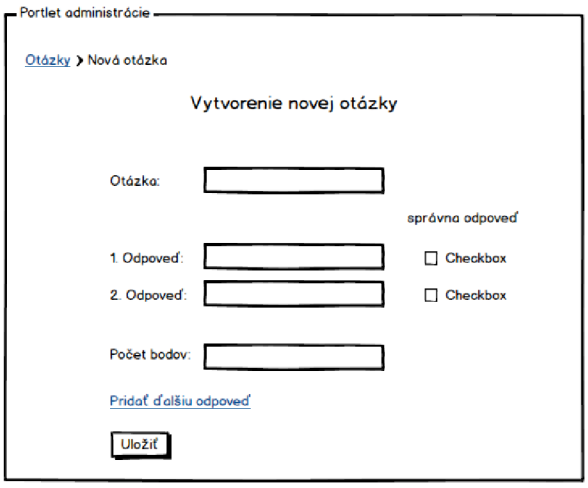
67 GUI, http://en.wikipedia.org/wiki/Graphical_user_interface

68 Balsamiq, www.balsamiq.com

69 Jira, https://www.atlassian.com/software/jira/?clicked=j.a.com_introduction

70 Confluence, https://www.atlassian.com/software/confluence

71 Portable Network Graphics, http://www.w3.org/TR/PNG/



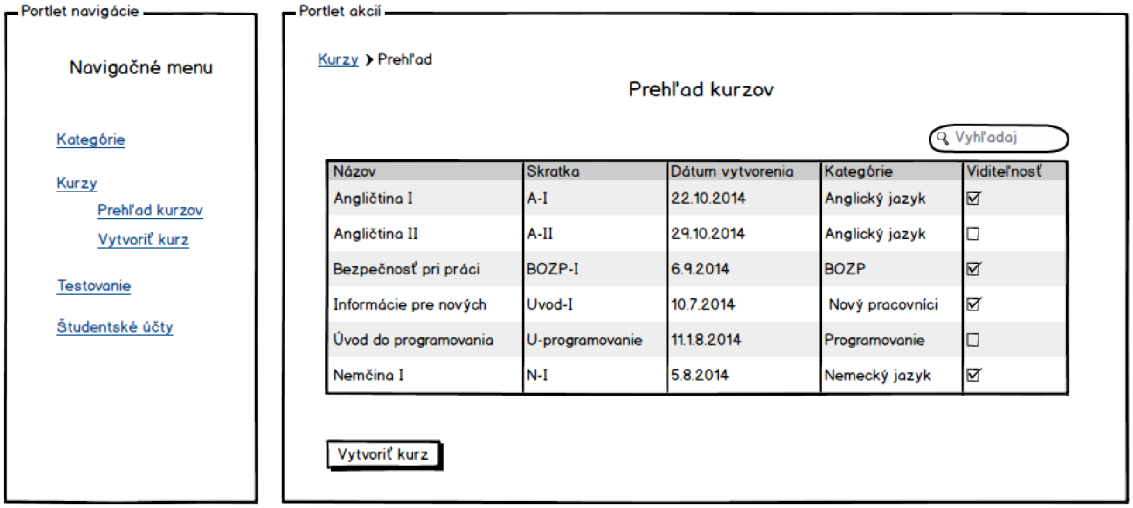
Obrázok č. 7: Návrh formulára pri vytváraní otázok určených pre testovanie študentov.

Medzi výhody spojené s tvorbou návrhov obrazoviek pokladáme skutočnosť, vďaka ktorej si môže vývojár ozrejmiť rôzne implementačné problémy, ktoré bude musieť riešiť. Osobne nám tvorba návrhov uľahčila vývoj tejto aplikácie a dopomohla nám k lepšej špecifikácii vytváranej aplikácie. Prikladáme obrázok č. 7 návrhu obrazovky, ktorý zobrazuje formulár pri vytváraní otázok určených pre testovanie študentov. Ostatné vytvorené návrhy čitateľ môže nájsť v prílohe na CD.

3.4 Rozdelenie portletov

Funkcionalita nami navrhovanej aplikácie pre riadenie výučby sa bude členiť do dvoch kategórií. Prvá kategória je určená pre administrátora, ktorý dohliada a zabezpečuje vzdelávací program vo firme. V druhej kategórii môžete nájsť portlety, ktoré sú určené pre zamestnancov firmy. Vložiť takúto funkcionalitu do jedného portletu by bolo po implementačnej stránke zbytočne zložité a preto z dôvodu lepšej prehľadnosti sme sa ich rozhodli takýmto spôsobom rozdeliť.

Portlety určené pre administrátora vzdelávania - portlety tohto charakteru budú pozostávať z administračnej časti vzdelávania vo firme. Prvý portlet (portlet navigácie) bude mať za úlohu zabezpečovať administrátorovi navigáciu pre prehľadnú prácu s druhým portletom. Druhému portletu (portlet akcií) bude umožnené vytváranie, úprava a mazanie všetkých základných entít.



Obrázok č. 8: Prototyp obrazovky zobrazujúci portlety navigácie a akcií pre administrátora vzdelávania.

Portlet navigácie: bude zobrazovať v stromovej štruktúre 4 základné záložky administračnej časti. Ide o kategórie kurzov, ďalšie kurzy, testy a študentské účty. Pri výbere niektorej zo záložiek sa zobrazia ďalšie záložky umožňujúce prácu a manipuláciu s entitami, ktoré sa budú zobrazovať v danom na to určenom portlete. Pre lepšie predstavy sme priložili obrázok č. 8 zobrazujúci portlet navigácie a protlet akcií.

Portlet aktivít: na základe portletu navigácie bude portlet aktivít určený pre zobrazovanie a správu:

> **kategórií kurzov** - portlet zobrazí kategórie kurzov vzdelávania. Pri výbere konkrétnej kategórie budú zobrazené všetky kurzy danej kategórie. Pri vytváraní a editácii kategórií budeme mať na výber zaškrtnutie políčka, ktoré umožní zaradenie kategórie so všetkými kurzami do sekcie voľne dostupných kurzov.
> **kurzov** - tie administrátor vzdelávania vytvára, upravuje a odstraňuje. Prideľuje im študujúcich z banky používateľských účtov pre e-learning. Po kliknutí na tlačidlo akcie príslušného kurzu sa nám zobrazia akcie, ktoré môžeme s daným kurzom vykonať. Buď pôjde o prácu so študijnými materiálmi, lekciami a testami alebo používateľmi. Za nevyhnutnosť považujeme tlačidlo pre aktiváciu a deaktiváciu kurzu do a zo vzdelávacieho procesu pre prípad jeho úpravy (editácie).

študijných materiálov - tie budú zobrazovať zoznam skupín dokumentov, ktoré pozostávajú z odkazov na jednotlivé študijné materiály uložené v centrálnom repozitári dokumentov a médií. Každému súboru bude pridelená patričná ikona pre ľahšiu identifikáciu.

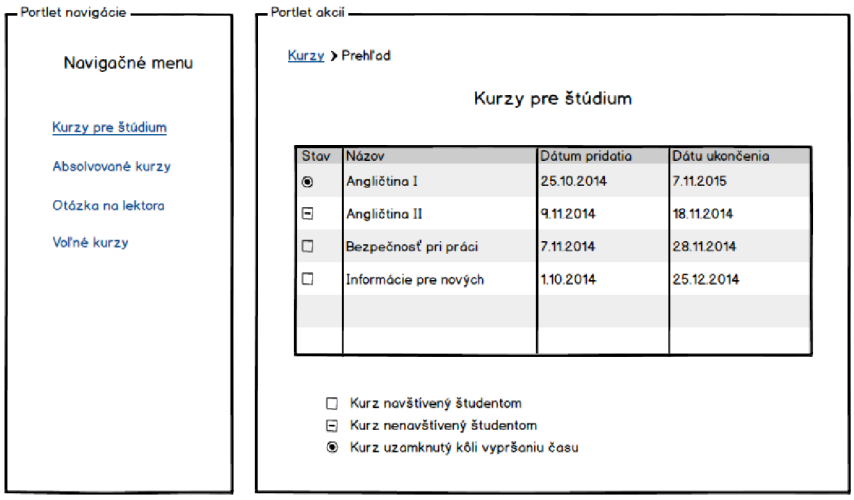
lekcií - v portlete sa bude nachádzať zoznam lekcií. Po kliknutí na príslušné tlačidlo sa zobrazí celkový náhľad lekcií vo forme stránok, ktoré môžeme listovať pomocou tlačidiel. Tieto stránky budú vytvárané pomocou wysiwyg editora.

testovania - v tejto sekcii budeme môcť vytvárať a spravovať otázky a odpovede určené pre absolvovanie kurzu. Tieto kontrolné otázky sa budú vkladať do bánk otázok, ktoré sa budú prideľovať kurzom. Ku každej otázke bude náhľad, ktorý zobrazí testovaciu otázku v takom štýle, v akom ju uvidí testovaný používateľ.

požívateľských účtov - budú slúžiť pre aktivovanie účtov pre používateľov vzdelávania. Portlet bude automaticky načítavať skupiny používateľov Liferay. Cez príslušné tlačidlo bude môcť byť načítaný obsah skupín používateľov. Každý takýto účet umožní zobrazenie stavu kurzov, ktoré konkrétny študujúci práve vykonáva. Taktiež budeme môcť zistiť či bol kurz úspešne absolvovaný.

Portlety určené pre študentov - Portlety určené pre používateľov e-learningu by mali byť rozdelené do minimálneho počtu pre splnenie podmienok jednoduchého používania a nasadenia do už bežiaceho podnikového portálu na platforme Liferay. Budú obsahovať dva portlety, z ktorých jeden bude mať funkciu navigácie a ďalší bude obsahovať celú funkcionalitu zobrazovania kurzov, testov a študijných materiálov.

Portlet navigácie: bude zobrazovať záložky medzi ktoré radíme kurzy k štúdiu, absolvované kurzy, otázky pre lektora a voľný výber kurzov. Po výbere jednotlivých záložiek sa zobrazia v portlete akcií dané entity, s ktorými môže vyškolený pracovník narábať. Pre lepšiu ilustráciu prikladáme obrázok č. 9 zobrazujúci portlet navigácie a protlet akcií.



Obrázok č. 9: Prototyp obrazovky zobrazujúci portlety navigácie a akcií.

Portlet aktivít: na základe portletu navigácie bude súžiť na zobrazovanie a prácu s:

kurzom určeným pre štúdium - študujúcemu sa budú zobrazovať v zozname kurzy pridané administrátorom vzdelávania. Pri označení konkrétneho kurzu budú k dispozícii informácie o kurze, všetky dostupné študijné materiály patriace danému kurzu, prehľad testov s možnosťou ukončenia kurzu a zobrazenie lekcií (stránok) určených pre štúdium.

absolvovaním kurzov - k ďalším položkám, ktoré sa budú zobrazovať patrí sekcia absolvovaných kurzov. Pôjde o zoznam kurzov, ktoré budú ponúkať nielen prehľad ukončených kurzov, ale taktiež materiály, s ktorými by sa mohol školiaci pracovník stretnúť. K dispozícii bude i zobrazenie počtu pokusov na úspešné ukončenie kurzu s najvyšším počtom získaných bodov.

posielaním správ - po kliknutí v navigačnom portlete na otázky pre lektora sa nám zobrazí formulár na odoslanie prípadnej otázky alebo potreby získania informácie od lektora.

výberom doplnkových kurzov - budú obsahovať kategórie kurzov, pod ktorými môže používateľ vzdelávania vyhľadávať voľne dostupné kurzy pre svoj osobný alebo kariérny rozvoj.

4. Realizácia a testovanie

4.1 Trojvrstvová architektúra

Celá aplikácia je riešená podľa trojvrstvovej architektúry. Jedná sa o návrhový vzor, ktorý je v dnešnej dobe zaužívaným princípom oddelenia jednotlivých vrstiev. Tieto vrstvy komunikujú medzi sebou prostredníctvom pevne zadefinovaných rozhraní.

Prezentačná vrstva - slúži k zobrazeniu informácií používateľom a to väčšinou prostredníctvom grafického rozhrania. Je nezávislá na platforme. Táto vrstva zahŕňa používateľskú interakciu s rôznymi zariadeniami a nástrojmi pre správu obsahu (CMS), s nástrojmi pre meranie správania používateľov, bezpečnostné prvky a opatrenia pre zabezpečenie fungovania projektu v prostredí internetu, rôzne marketingové on-line aktivity a rôzne rozhrania pre integráciu v štruktúrovanej podobe (rôzne XML výstupy, RSS[72], Google Sitemap[73], SOAP, REST, atd.). Cez túto vrstvu sú riešené kontroly vkladaných dát, ktoré vstupujú do aplikácie. Avšak tieto validačné kontroly nemôžu obsahovať priame spracovávanie dát. Na tejto vrstve sme validovali formuláre cez Java scriptový framework jQuery. Taktiež ho využívame i pri vyhľadávaní používateľov, pri funkcii pridania do kurzu a zmenu nastavenia dátumov riešenou cez datapicker. K zobrazovaniu správnych odpovedí pri výsledkoch testovania využívame natívny Java script. Spomenieme ešte základné štandardy ako HTML, CSS, JSP, ktoré majú na tejto vrstve svoje miesto. Celá vrstva bude typicky implementovaná podľa návrhového vzoru MVC[74]. Viac informácií o návrhovom vzore spomíname v realizácii 4.2 Použité technológie: Spring MVC.

Vrstva aplikačnej logiky - jedná sa o kľúčovú vrstvu z pohľadu prenosu dát medzi prezentačnou vrstvou a dátovou vrstvou. Vrstva aplikačnej logiky, inak nazvaná aj business logika, predstavuje dôležitú funkciu aj z hľadiska držania interného know-how. Uloženie tejto vrstvy mimo firmu by sa malo teda zminimalizovať. V našej aplikácii sa práve na tejto vrstve nachádza jasne zadefinované rozhranie, jadro celej aplikácie s jeho logikou, výpočtami a funkciami ser-

---

72 Rich Site Summary, https://www.rss.com/

73 Google SiteMap, https://www.xml-sitemaps.com/

74 Mode –View- Controller, http://msdn.microsoft.com/en-us/library/ff649643.aspx

visných tried, ktoré sú skrze svoje metódy volané prezentačnou vrstvou k získaniu informácií z databázového úložiska. Jedná sa o akéhosi prostredníka medzi týmito dvoma vrstvami. Nachádza sa tu taktiež úložisko zdieľaných informácií v rámci jednotlivých systémov získaných z prezentačnej a dátovej vrstvy.

Tak, ako v dátovej vrstve (viď nižšie) i tu sa používa aplikačný rámec Spring. Jedná sa o modulárny framework, ktorý nám umožňuje využiť len niektoré jeho časti, ktoré práve potrebujeme. Napr. modul implementujúci podporu pre aspektovo orientované programovanie, modul pre podporu testovania komponentov (JUnit[75], TestNG[76]), zabezpečenie Dependency Injection, modul prístupov (JDBC[77], ORM[78], JMS[79]) a veľa iných.

Dátová vrstva - zabezpečuje cieľové uloženie informácií a ich manipuláciu. Ide o perzistenčné úložisko dát, kde sa využívajú relačné databázy alebo databázy iného typu (NoSql[80]). V našej realizácii ide konkrétne o relačnú databázu PostgreSql. Ako môžeme vidieť na obrázku diagramu tried, každá táto entitná trieda je reprezentovaná v relačnej databáze tabuľkou a inštancia tejto triedy predstavuje jeden riadok tabuľky. Pri práci s objektmi nám ušetrila veľa práce technológia JPA implementovaná Hibernate, umožňujúca objektovo relačné mapovanie (viď 4.2 Použité technológie: Java Persistence API). Pre prácu s uloženými dátami v databáze používame DAO objekty. Prostredníctvom nich môžeme vkladať, upravovať, vyberať a zmazať entity v databáze. V aplikácii využívame generické DAO, čo znamená, že každá entita zdedí od triedy `GenericDao` na implementované metódy. Pre príklad bol uvedený obrázok č. 10, ktorý zobrazuje triedy z projektu (kurz, kategória, lekcia) implementujúce `GenericDao`. Všetky vzťahy medzi komponentmi sú sprostredkované pomocou aplikačného rámca Spring[81].

---

75 JUnit, http://junit.org/

76 TestNG, http://testng.org/doc/index.html

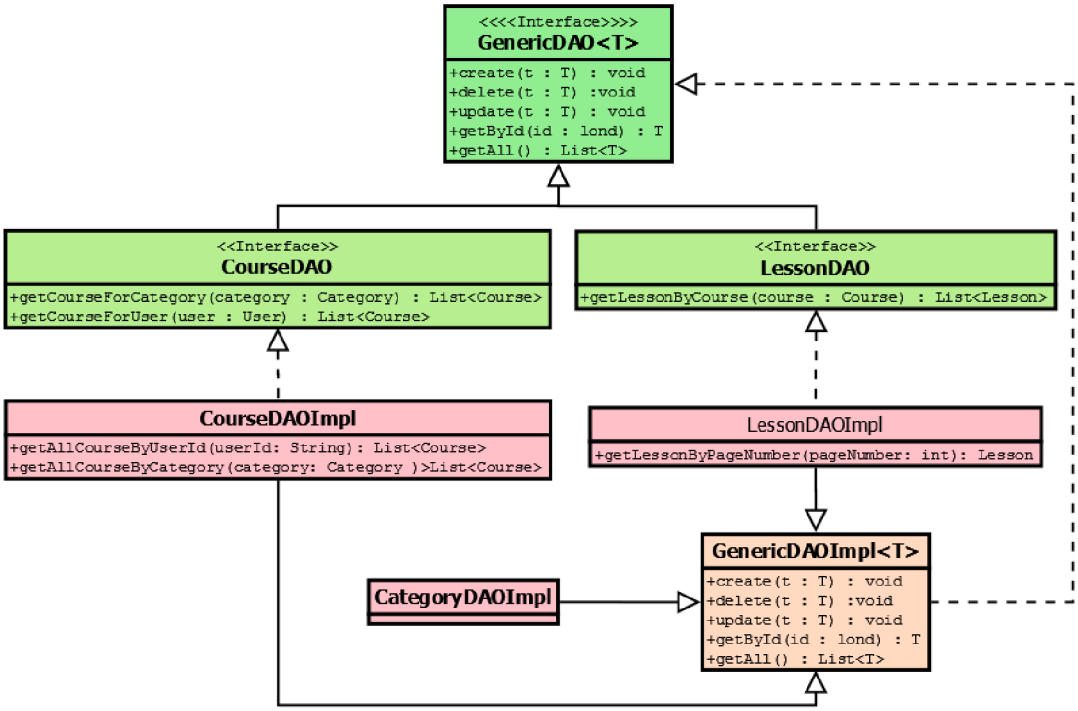
77 Java Database Connectivity, http://www.oracle.com/technetwork/java/javase/jdbc/index.html

78 Object Relational Mapping, http://hibernate.org/orm/what-is-an-orm/

79 Java Message Service, http://en.wikipedia.org/wiki/Java_Message_Service

80 NoSql, http://nosql-database.org/

81 Spring, http://spring.io/



Obrázok č. 10: Zobrazujúci generické DAO pre triedy category, course, lesson.

4.2 Použité technológie

JSP - technológia JSP umožňuje vloženie kúskov Java kódu (skripletov) do statických textových dokumentov. JSP stránka je textový dokument, ktorý obsahuje dva typy textu. A to statické údaje, ktoré sú vyjadrené v ľubovoľnom formáte ako HTML alebo XML. Druhým typom textu sú JSP elementy. Tie zabezpečujú dynamické vytváranie obsahu [29]. Oddelenie statického HTML od dynamického obsahu poskytuje viacero konkurenčných technológií ako je ASP[82], PHP, JS. Skriplety majú prístup k rovnakým automaticky definovaným premenným a výrazom (napr.: request, response, sesion a atď.). K štandardu JSP patria ďalšie dve technológie, JSTL[83] a EL[84] (Unified Expression Language Unified Expression Language). JSTL (Java Server Pages Standard Tag Library) je štandardná knižnica značiek zahrňujúca základné funkcionality spoločné pre mnoho JSP aplikácií. Pomocou nej môžeme manipulovať s XML dokumentmi, pracovať s premennými, cyklami, internacionalizáciou, poskytuje značky pre prís-

---

82 Active Server Pages, http://www.asp.net/

83 Standard Tag Library for JavaServer Pages, https://jstl.java.net/

84 Expresion Language, https://uel.java.net/

tup do databázy pomocou jazyka SQL[85] a iné. [30] Technológia JSP bola zahrnutá do aplikácií kvôli možnosti dynamického vkladania obsahu a širokej podpore poskytovaných funkcionalít a knižníc.

jQuery a JavaScript - JavaScript (JS) je dynamický skriptovací jazyk používaný programátormi pri vytváraní webových stránok. Jadro JS je podobné s programovacími konštrukciami ako je `if`, `while` a operátormi (`&&`, `||`), ktoré môžeme nájsť v jazykoch C[86], C++ [87] a Java. Táto syntaktická podobnosť tu končí, pretože JavaScript je voľne typový jazyk, čo znamená, že typ premennej nemusí byť určený. [31] Rozdiel medzi JS a JSP je nasledovný. Prvý jazyk beží na strane klienta, zatiaľ čo kódy JSP stránok sú spustené na serveri. JS môže mať vplyv len na konkrétnu stránku, na ktorej je vložený. [29] V súčasnej dobe je veľké množstvo implementácií JavaScriptu s proprietárnym alebo s otvoreným zdrojovým kódom. Medzi ne patrí aj jQuery. Je to JS knižnica, ktorá sa využíva na klientskej strane, kladúca dôraz na interakciu medzi HTML a JS. Knižnica je distribuovaná pod licenciami MIT[88] a GPL[89]. Obsahuje funkcie pre manipuláciu s CSS, spracovanie a obsluhu udalostí, mechanizmus pre prechádzanie a manipuláciu HTML dokumentov, efekty, animácie a interakciu pomocou AJAX technológie (Asynchronous JavaScript and XML). [32] Použitie jQuery a natívneho JavaScript-u v našej aplikácii našlo uplatnenie pri:

- validácii formulárov na klientovej strane,
- zadávaní dátumu pomocou kalendára,
- vyhľadávaní používateľov v select box-e,
- validácii a zobrazení správnych výsledkov pri zodpovedaní testovacích otázok (viď obrázok č. 11).

---

85 Structured Query Language, http://en.wikipedia.org/wiki/SQL

86 C, http://en.wikipedia.org/wiki/C_(programming_language)

87 C++, http://en.wikipedia.org/wiki/C%2B%2B

88 MIT, http://opensource.org/licenses/MIT

89 General Public License, http://www.gnu.org/copyleft/gpl.html

```
<script type="text/javascript">
    <%
        int param = (Integer) request.getAttribute(ATTR_SIZE_ITEMS);
        int paramMin = 0;
        if (param > 0) {
            paramMin = 1;
        }
    %>
    $(document).ready(function() {
        $("#fm").validate({
            rules: {
                name: {
                    required: true
                },
                inPercentDone: {
                    required: true,
                    min: 10,
                    max: 100,
                    digits: true
                },
                numberQForTest: {
                    required: true,
                    min: <%=paramMin%>,
                    max: <%=param%>,
                    digits: true
                }
            }, messages: {
                name: '<spring:message code="NotEmpty" javaScriptEscape="true" />',
                inPercentDone: {
                    required: '<spring:message code="NotEmpty" javaScriptEscape="true" />',
                    min: '<spring:message code="Min" arguments="${10}" javaScriptEscape="true" />',
                    max: '<spring:message code="Max" arguments="${100}" javaScriptEscape="true" />',
                    digits: '<spring:message code="Digits" javaScriptEscape="true" />'
                },
                numberQForTest: {
                    required: '<spring:message code="NotEmpty" javaScriptEscape="true" />',
                    min: '<spring:message code="Min" arguments="<%= paramMin%>" javaScriptEscape="true" />',
                    max: '<spring:message code="Max" arguments="${quizBankPto.numberAllQ}" javaScriptEscape="true" />',
                    digits: '<spring:message code="Digits" javaScriptEscape="true" />'
                }
            }
        });
    });
</script>
```

Obrázok č. 11: Ukážka validácie vytvoreného testu.

Spring MVC - Spring MVC predstavuje implementáciu, ktorá sa analogicky používa i v prostredí portálu a to za pomoci Spring Portlet MVC. Potrebná konfigurácia pri vývoji aplikácie je docielená väčšinou pomocou anotácií alebo v XML súboroch. Tento xml súbor zabezpečuje konfiguráciu aplikačného rámca. Môžeme v ňom nastaviť napríklad automatické vyhľadávanie anotovaných tried. Spring sa o tieto zaregistrované triedy bude starať, automaticky spravovať závislosti medzi nimi a spravovať beany (springom spravované inštancie tried). [33]

Tak, ako Spring MVC aj Spring Portlet MVC je založený na návrhovom vzore model-view-controller, ktorý sa snaží oddeliť dáta aplikačnej a prezentačnej vrstvy. [1]

Model - na tejto vrstve aplikácií môžeme nájsť dáta a pravidlá pre manipuláciu s dátami. Model spravuje dáta z aplikačnej vrstvy. Controller

upravuje dáta na základe vyvolaných aktivít a následne posiela informácie o výslednom stave vrstve view.

View - táto vrstva aplikácie poskytuje všetku logiku pre zobrazovanie dát používateľom poskytované vrstvou model. Môžeme tu manipulovať napríklad s poliami, prvkami formulárov ako skrytie alebo zobrazenie dát.

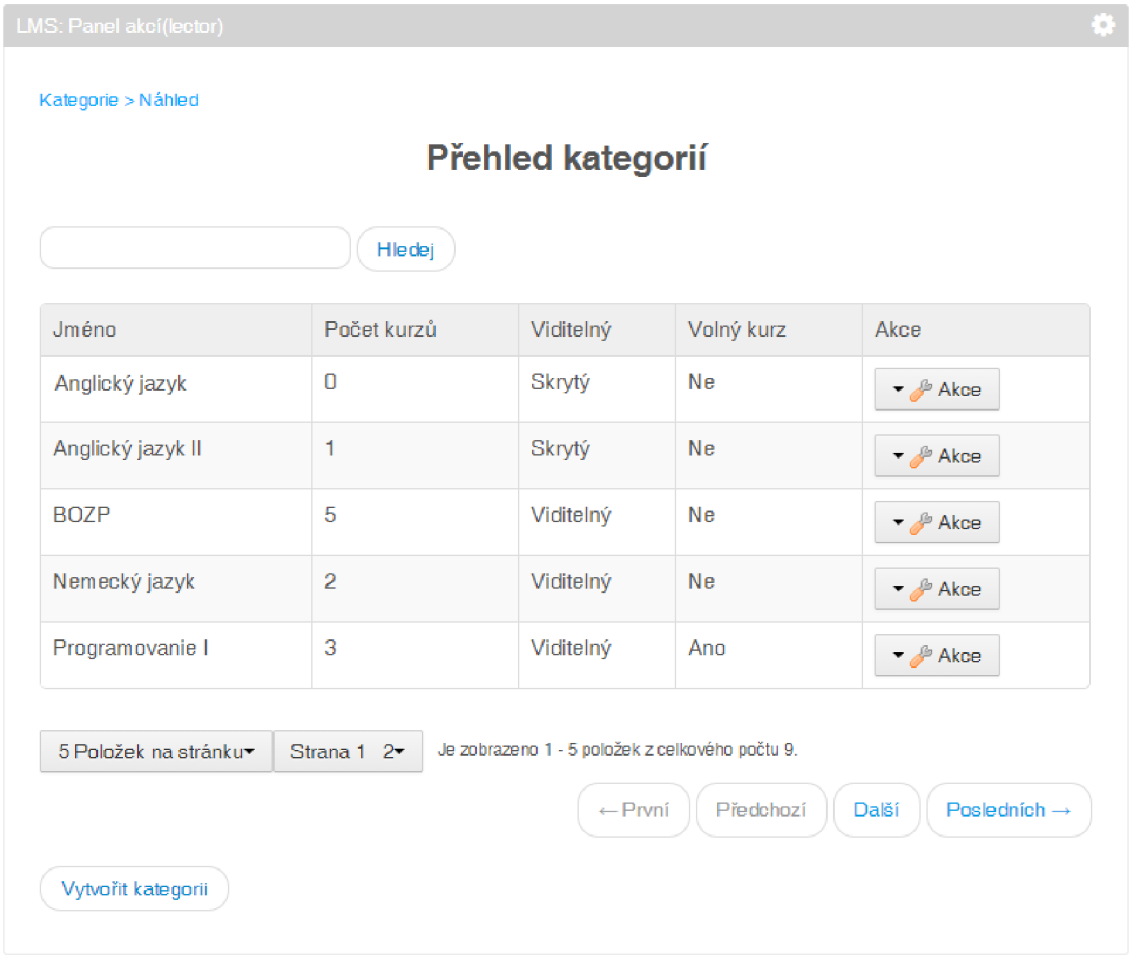
Controller - spracováva používateľov vstup a podľa potreby aktualizuje model a odosiela informácie do vrstvy view. Napríklad: controller je zodpovedný za určenie akcie používateľa, na ktoré tlačidlo klikol a tiež vykonanie správnej funkcie.

Spring Portlet MVC poskytuje i veľa ďalších funkcionalít, čím sa stal hlavným kadetom použitia. Medzi tie, ktoré boli použité i v praktickej časti patrí práca a validácia s formulármi, poskytovanie atribútov objektu za pomoci `@ModelAttribute`, vkladanie závislosti, automatické vyhľadávanie komponentov, podpora anotácií atď. V aplikácii na vrstve modelu sa v našom prípade nachádzajú ľubovoľné triedy s dátami, ako sú inštancie kurz alebo banka otázok. Na vrstve view môžeme nájsť stránky tvorené s pomocou JSP. Na poslednej vrstve Spring Portlet MVC sa nachádzajú triedy, ktoré majú za úlohu spracovanie http požiadaviek, rôzne triedy určené na zmenu a zobrazenia dát. Taktiež sa tu nachádza viacero controlerov, ktoré sa viažu na jednotlivé portlety. Pri práci s MCV sme do adresára WEB-INF umiestnili niekoľko súborov a taktiež definujúce rôzne nastavenia a inicializácie vytváraných komponentov nájdených v súbore {meno potletu}-`portlet.xml`. Súboru `portlet.xml` sme zadefinovali objekt `DispatcherPortlet` a konfiguráciu k súborom, ktoré majú na starosti podporu lokalizácie.

Liferay UI Tags - portál Liferay poskytuje bohaté množstvo UI značiek, ktoré šetria čas programátorovi potrebný pre vývoj. Značky obsahujú vlastné používateľské rozhranie, napríklad pre hlásenie chýb, kalendár, CAPTCHA, navigáciu, panel sociálnych aktivít, stránkovanie, záložky, ikony a iné. Využitie tejto knižnice nájdeme na prezentačnej vrstve a v JSP stránkach. Pri zobrazovaní údajov v tabuľkách je vhodné využiť funkcie tejto knižnice. Pri dodaní zoznamu objektov sa táto knižnica postará o zobrazenie dát v tabuľke, lokalizáciu, triedenie alebo stránkovanie. [9]

V aplikácii boli použité značky z knižnice Liferay UI práve pre takýto spôsob zobrazovania objektov v tabuľkách a taktiež sa využila i ponúkaná možnosť

stránkovania celého obsahu (viď obrázok č. 12). V navigačnom portlete pre menežovanie a správu kurzov boli použité značky typu liferay-ui:panel-container určené pre zobrazovanie a skrývanie údajov v spoločnom uzle.



Obrázok č. 12: Obrázok zobrazujúci stránkovanie položiek v kategórii kurzov.

Java Persistence API - dátové úložisko sa stalo neoddeliteľnou súčasťou ľubovoľnej aplikácie. Medzi bežné prístupy k týmto úložiskám patril štandard JDBC. Java vývojári museli písať zložité SQL dotazy, čo viedlo k ich slušnej znalosti a pri narastaní aplikácií sa stávali neudržiavateľnými. Síce programátori mali úplnú kontrolu nad celou databázou, ale bolo pomerne zložité vymýšľať spôsob transformácie objektov do relačných databáz. [34]

Java Persistence API je štandardom pre objektovo-relačné mapovanie (ORM). Jedná sa o špecifické rozhranie určené pre programovací jazyk Java, ktoré popi-

suje správu relačných dát v aplikácii využívajúcej Java SE [90] alebo Java EE. JPA umožňuje POJO (Plain Old Java Object) objektom uloženie bez akejkoľvek nutnosti implementácií rozhrania alebo metódy. Celé mapovanie je umožnené na základe štandardných anotácií alebo xml dokumentov, ktoré popisujú akým spôsobom sa majú mapovať objekty do relačných databáz. JPA taktiež definuje manažéra entít (EntityManager API) pomocou ktorého spracovávame dotazy a transakcie. Dotazy nie sú vo forme SQL jazyka, ale sú riešené pomocou JPQL[91] jazyka. JPQL je o objektovo orientovaný opytovací jazyk, ktorý je nezávislý na platforme. Síce je silno inšpirovaný SQL syntaxou, ale pracuje s JPA entitami a nie priamo s databázovými tabuľkami. [35]

V jednotlivých portletoch aplikácie sme použili zmienenú špecifikáciu, ktorú využíva nami použitý nástroj Hibernate[92]. V aplikácii sa nachádzajú triedy, ktoré sú označené anotáciou `@Entity`, čo znamená, že sa jedná o perzistentné objekty určené na načítavanie a ukladanie do databázy. Za inštančnú premenu, ktorá zodpovedá za identifikáciu sme označili anotáciou `@ID`. Celé JPA obsahuje len atribúty a metódy. Nemala by obsahovať žiadnu časť aplikačnej logiky. V týchto entitách definujeme i prepojenia objektov pomocou anotácií (`@OneToMany`, `@ManyToOne`, `@OneToOne`), generovanie primárneho kľúča (`@GeneratedValue`), názvy jednotlivých stĺpcov tabuľky (`@Column(name="bank_questions")`) a iné. Všetky prezistenčné triedy sa nachádzajú v balíku `lms.core.entity`.

Tak, ako sme už spomenuli v trojvrstvovej architektúre, (viď 4.1 Trojvrstvová architektúra: Dátová vrstva) DAO objekty slúžia k vykonávaniu základných operácií nad každou perzistenčnou entitou. Tieto operácie sú vykonávané za pomoci správcu entít (`EntityManager`) a JPQL jazyka.

Triedy, ktoré obsahujú tieto metódy nájdeme v balíku `lms.core.dao`. Celá konfigurácia prebieha v xml súbore `core-jpa.xml` nachádzajúca sa v adresári WEB-INF/spring-context. V ňom je vytvorený bean, ktorý je zodpovedný za

---

[90] Java Standard Edition, http://www.oracle.com/technetwork/java/javase/overview/index.html

[91] Java Persistence Query Language, http://en.wikipedia.org/wiki/Java_Persistence_Query_Language

[92] Hibernate, http://hibernate.org/

manažovanie a správcovstvo entít (`entity manager facotry`). Ten spravuje jednotlivé objekty, životný cyklus, udržiava spojenie s databázou atď.

Vyžitie služieb ponúkaných portálom - na základe zistených požiadaviek môžeme využiť portálové API pre prácu s používateľmi a skupinami používateľov. Nielenže sa nemusíme starať o ich registráciu a spravovanie, ale nemusíme tiež riešiť ich prístupové práva k jednotlivým funkcionalitám systému. Taktiež sme využili Document managment systém, ktorý je zabudovaný v rámci Liferay internej Documents a Media library. Tento integrovaný systém nám slúži pre správu dokumentov, obrázkov, videí alebo audia, ktorý budeme využívať pri napĺňaní kurzov študijnými materiálmi.

Za veľkú výhodu môžeme pokladať prípadné využitie existujúcich portletov, ktorými Liferay už vo verzii CE disponuje. Portál má v ponuke viac ako 60 portletov, medzi ktorými sa nachádzajú také, ktoré nájdu veľmi rýchlo uplatnenie pri vzdelávaní pracovníkov. Ide hlavne o diskusné fóra, blogy, chaty a wiki.

Nesmieme však zabudnúť na jednoduchosť pridávania a administrovania portletov na platformu. Administrátor portálu má v úplnej kompetencii prístupové práva a právo k usporiadaniu portletov na portáli. Taktiež sa na celú aplikáciu vzťahuje možnosť zmeny vzhľadu za pomoci tém (viď 1.4 Liferay: Rozšírenia portálu Liferay).

4.3 Testovanie

V dnešnej dobe existuje veľké množstvo publikovaných techník testovania. Niektoré sa síce viac-menej prekrývajú, alebo navzájom až vylučujú. Neexistuje jediná správna technika pri testovaní softvéru. Každá testovacia technika má svoje klady a zápory, a preto je dobre použiť viacero techník, kedy môže dôjsť k zlepšeniu pokrytia testovaného softvéru. Hlavnými úlohami testovacích tímov býva hľadanie chýb od najzávažnejších k menej závažnejším, či zhodnocovanie stavu vyvíjaného systému. Pri voľbe správnej stratégie testovania je nutné identifikovať ciele a výstupy testovacích procesov, určiť vhodný spôsob nasadenia dostupných zdrojov, definovať rozsah testovania, alebo určiť ako bude monitorovaný a vyhodnocovaný priebeh testovania. [36] [37]

V praktickej časti diplomovej práce sme využívali regresné testovacie stratégie, ktoré pozostávali z jednotkových testov (JUnit testov) pre zistenie správnosti

fungovania aplikačnej logiky. Neostali sme len pri tejto jednej technike, ale vypracovali sme aj testovacie scenáre pre overenie funkčnosti celej aplikácie. Pri formulároch bola dodatočne pridaná validácia pomocou knižnice jQuery. Táto validácia kontrolovala zadané hodnoty používateľom vo formulároch pred odoslaním požiadavky na server. Používateľ bol pri zadávaní nekorektných hodnôt ihneď oboznámený o podmienkach daného vkladaného políčka.

Jednotkové testovanie - ide o prvú fázu testovania. Táto technika prebieha na strane vývojárskeho tímu. Po dokončení časti systému vývojár svoju prácu otestuje. Ide prevažne o testy na overenie fungovania a korektnosti implementácie systému. Testy boli implementované v aplikácii pomocou JUnit rámca a spustené vždy pred nasadením na aplikačný server. Aj vďaka tejto technike boli odladené chyby, ktoré vznikli pri vyvíjaní aplikačnej logiky.

Integračné testy - testy aplikačného rozhrania boli vykonané manuálnym testovaním hneď po implementácii jednotlivých častí aplikácie. Pri ukončení vývojárskych prác bola aplikácia testovaná viacerými dobrovoľníkmi. Táto skupina ľudí testovala aplikáciu i podľa testovacích scenárov, ktoré boli navrhnuté tak, aby dokázali odhaliť chyby a overiť správnu funkčnosť vytvorenej aplikácie. Scenáre zahŕňali kľúčové procesy, ku ktorým dochádza pri vytváraní a napĺňaní kurzov na portáli. Zistené chyby boli v krátkej dobe odladené a vrátené do testovacieho procesu pre opätovné overenie. Tabuľka č.1 zobrazuje report chýb, ktoré boli spozorované dobrovoľníkmi pri testovaní. Chyby neboli iným spôsobom zdokumentované.

Nájdené chyby	**Počet chýb**
Validácia pri vkladaní dát do formulárov	4
Zlý výpočet bodov pri vyhodnotení testu	1
Chybné zobrazenie diakritiky pre jazykovú verziu CZ	1
Chyba pri odstránení objektov z DB	2
Chyba pri stránkovaní obsahu	1

Nesprávne zobrazenie navigácie	2
Chybné oznámenia o vykonanej akcii	5
Nefungujúce tlačidlo alebo odkaz	5
Neodstránenie adresára v DMS Liferay	1
Nesprávne zobrazenie stránky	3
Vykonanie inej akcie pri kliknutí na príslušný odkaz	6
Nesprávne pomenovanie nadpisov	4

Tabuľka č. 1: Chyby zistené testermi.

Testovacie scenáre (TS) použité pri zisťovaní chýb funkčnosti aplikácie:

TS-01 Vytvorenie kategórie kurzu

TS-02 Úprava kategórie kurzu

TS-03 Vytvorenie kurzu

TS-04 Úprava kurzu

TS-05 Vytvorenie banky dokumentov

TS-06 Vloženie súboru do banky dokumentov

TS-07 Vloženie odkazu (stránky, dokumentu, videa) do banky dokumentov

TS-08 Vytvorenie knihy

TS-09 Úprava strany v knihe

TS-10 Pridanie študenta do kurzu

TS-11 Vytvorenie otázky

TS-12 Úprava otázky

TS-13 Vytvorenie banky otázok

TS-14 Vloženie otázky do banky otázok

TS-15 Priradenie banky otázok kurzu

TS-16 Aktivovanie účtu pre štúdium

Prikladáme ako vzor TS-06 testovací scenár „vloženie súboru do banky dokumentov“ (viď tabuľka č. 2).

ID	06	Názov	Vloženie súboru do banky dokumentov
Vstupné podmienky			Aktér je v roli administrátora, prihlásený v portáli. Nachádza sa v portletoch administrácie firemného vzdelávania, v portlete zobrazujúcom vytvorené banky dokumentov. Predpokladá sa, že aktér vytvoril kategóriu (TS 01) , samotný kurz (TS 03) a vytvoril banku dokumentov (TS 05).
Výstupné podmienky			Vloženie dokumentu do banky dokumentov.

Krok	Akcia	Očakávaná akcia
1	Kliknutie na tlačidlo *„Akcie“* konkrétneho kurzu.	Systém zobrazí v oblasti tlačidla *„Akcie“* panel s ponukou akcií.
2	Kliknutie na ikonu *„Pridať súbor“*.	Systém zobrazí popis akcií, ktoré sa vykonajú pri stlačení príslušných tlačidiel a samotného tlačidla *„Vložiť odkaz“* a *„Nahrať súbor“*.
3	Kliknutie na tlačidlo *„Nahrať súbor“*.	Zobrazia sa tlačidlá *„Prehľadávať“* , *„Nahrať“* a *„Zrušiť“*.
4	Kliknutie na tlačidlo *„Prehľadávať“*.	Systém zobrazí pop-up okno pre vyhľadávanie súborov v našom počítači.
5	Aktér vyberie dokument a klikne na tlačidlo *„Otvoriť“*.	Systém vloží názov vybraného súboru pri tlačidle *„Prehľadávať“*
6	Kliknutie na tlačidlo *„Nahrať“*.	Systém uloží vkladaný súbor do repozitára Dokumenty a Média a zároveň vloží vygenerovaný odkaz z tohto dokumentu do *banky dokumentov*. Systém zobrazí detail danej banky dokumentov s príslušnými vloženými dokumentmi. Systém oznámi aktérovi

		úspešnosť jeho akcie.

Tabuľka č. 2: Príklad testovacieho scenára pre vkladanie dokumentov do banky dokumentov.

K vývoju celej aplikácie bolo použité vývojové prostredie Netbeans 8[93]. Pri testovaní bol použitý portál Liferay vo verzii 6.2 CE, ktorý bežal na aplikačnom serveri Tomcat 7. Aplikácia bola testovaná na prehliadačoch Mozilla verzie 33.1, Chrome 39. a Explorer 11.

4.4 Dokumentácia

Len málo ľudí nazrie pri kúpe softvéru aj do dokumentácie. Avšak to nie je dôvodom prečo ju nepísať a nezahrnúť do fáz vývoja produktu. Bežný používateľ po prvýkrát otvára dokumentáciu, keď hľadá pomoc pri inštalácii daného produktu alebo má problém, ktorý nevie vyriešiť. Dokumentáciu by mal obsahovať každý softvér, pretože odzrkadľuje kvalitu celého produktu a uľahčuje prácu s novým produktom, resp. technológiu, ktorú sa používateľ rozhodol používať. Pri písaní dokumentácie je na mieste, ak k nej pisateľ pristupuje inovatívne.

Používateľská dokumentácia - bežného používateľa aplikácie nezaujíma štruktúra a vnútorná implementácia daného softvéru. Nie je preňho dôležité akým spôsobom boli veci vytvorené a naprogramované, či aké jazyky a technológie boli použité. Avšak dokumentácia, ktorá obsahuje množstvo príkladov a odpovedí je na najčastejšie vyskytnuté problémy na mieste. Forma dokumentácie by mala byť písaná s vedomím, že ju bude čítať osoba, ktorá nerozumie programovaniu a v daných technológiách nie je utvrdená. Mali by sme pri písaní pristupovať defenzívne a vysvetľovať, i pre nás, triviálne funkcie a možnosti programu.

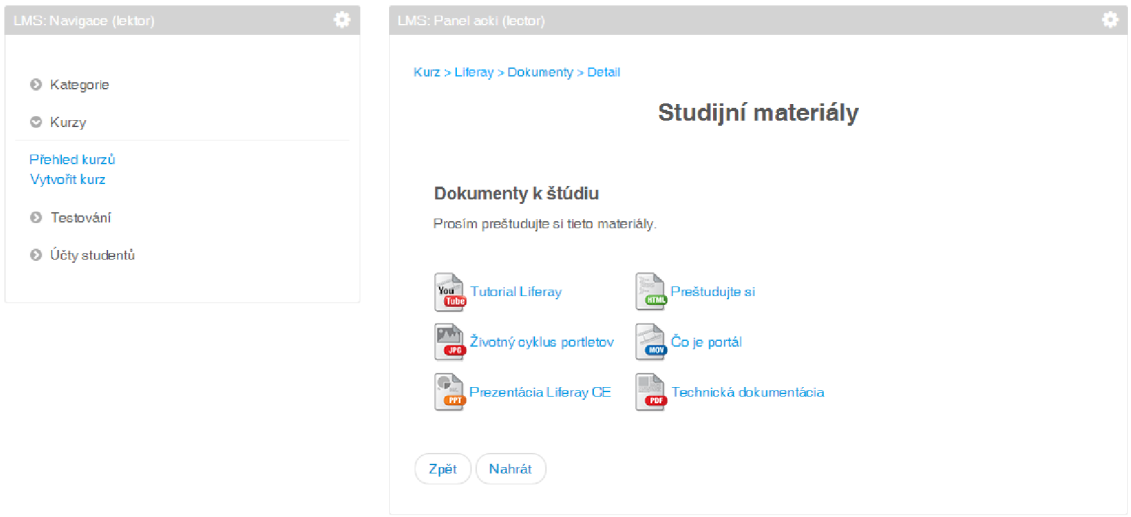
Pre prototyp aplikácie vzdelávania a riadenia výučby bola vytvorená dokumentácia, ktorá je určená administrátorom a lektorom firemného vzdelávania. Dokumentácia bola napísaná netradičným spôsobom, teda formou helpdesku, v ktorom môžeme nájsť rôzne odpovede na otázky tykajúce sa vytvárania a spravovania kurzov. Pre lepšiu orientáciu sa tieto otázky zozbierali do tematických skupín nasledovne:

---

[93] Netbeans, https://netbeans.org/

- prvotné nastavenia aplikácie,
- vytváranie, spravovanie kategórií, kurzov a lekcií,
- pridávanie a spravovanie materiálov v kurze,
- testovanie študentov,
- administrácia študentských účtov.

V každej spomenutej tematickej skupine nájdeme sadu otázok a odpovedí, ktoré sú doplnené screenshot-ami obrazovky pre lepšiu navigáciu a porozumenie danej akcie. Všetky otázky a odpovede si čitateľ môže prezrieť v prílohe na CD. Pre náhľad vytvorenej aplikácie bol priložený obrázok č. 13 zobrazujúci detail študijných materiálov v kurze.



Obrázok č. 13: Screenshot zobrazujúci detail študijných materiálov v kurze.

Vývojárska dokumentácia - Vývojársku dokumentáciu môžeme pri porovnaní s používateľskou nazvať technickou dokumentáciou, ktorá v sebe obsahuje práve tie špecifické záležitosti softvéru, ako je popis tried a ich atribútov, zoznam globálnych identifikátorov, zoznam funkcií a podobne. Vývojárska dokumentácia sa píše počas vývoja softvéru. Pri zmene implementácie, alebo štruktúry dochádza i k zmenám danej časti dokumentácie. K základným funkciám týchto dokumentácií patrí lepšia údržba a rozšíriteľnosť softvérového produktu.

Pomocou programu Netbenas 8 sme vygenerovali základnú technickú dokumentáciu Javadoc[94]. Táto dokumentácia je vo formáte HTML a je priemyselným štandardom pre dokumentáciu tried písaných v jazyku Java. Javadoc nám poskytuje informácie o parametroch metódy o tom, aké výnimky môže metóda vyhodiť, či informácie o návratových parametroch metódy. [38] V prílohe na CD môžeme nájsť vygenerovanú dokumentáciu.

[94] Javadoc, http://en.wikipedia.org/wiki/Javadoc

Záver

Diplomová práca sa venovala problematike portálových riešení, cez vývoj portletov a nadobudnutia teoretických znalostí, až po samotnú analýzu a implementáciu prototypu pre správu elektronických vzdelávacích materiálov a kurzov. Cieľom práce bolo riešenie dvoch častí.

V prvej, teoretickej časti sme sa zaoberali využitím portletov, kde sme si predstavili ich neodmysliteľnú komponentu - portlet. Zoznámili sme sa so životným cyklom, štruktúrou portálových aplikácií, špecifikáciou JRS-168 a JRS-286, s ktorou sa vývojári stretávajú pri tvorbe a vývoji portletov. Ďalej sa v práci zaoberáme samotnou platformou Liferay, jeho základnými funkciami a podporovanými technológiami. Pozreli sme sa na ponúkané edície a možnosti, s ktorými môžeme spomenutý portál rozšíriť. V ďalšej teoretickej časti sme sa zamerali na potreby firemného vzdelávania a princípy systému na riadenie výučby (LMS). Dozvedeli sme sa, že vzdelávanie je investícia do budúcnosti a podniky podporujúce vzdelávanie a rozvoj svojich zamestnancov získajú okrem zvyšovania svojej hodnoty a ekonomických výsledkov aj lepší imidž. Rozobrali sme hodnotu ľudského kapitálu ako jeden z determinantov dlhodobej prosperity firiem a organizácií. Popísali sme základné metódy a ciele firemného vzdelávania. Predstavili sme si fenomén 21. storočia pod pojmom e-learning, ktorý nemá staré korene v edukácii. Spomenuli sme si viacero aspektov, ktoré rieši elektronické vzdelávanie i to, že v plnej miere využíva informačno-komunikačné technológie. Využitie sme našli nielen u organizáciách, ktoré potrebujú vyškoliť, či rekvalifikovať veľký počet pracovníkov v krátkom čase. Vzhľadom k rôznorodosti e-learningových portálov sme spravili analýzu dostupných portálov používaných vo vzdelávaní.

V druhej, praktickej časti, sme sa zamerali na analýzu a návrh aplikácie, ktorá rieši správu a organizáciu kurzov v prostredí podnikového vzdelávania. Špecifikovali sme hlavné požiadavky na spomenutú aplikáciu, rozkreslili sme prípady použitia a dátový model. Na základe návrhu boli vybrané vhodné technológie a špecifikácie aplikácie. Diagram tried reprezentoval entitné triedy s atribútmi, ktoré predstavovali základné entity, ktoré sa pomocou technológie JPA inšpirovanej Hibernate mapovali do relačnej databázy. Pre vývoj portletov sa využívala databáza PostgreSQL. Celá aplikácia bola navrhnutá pomocou UML diagramov. Pre kreslenie diagramov sme použili modelovací nástroj

Dia[95]. Pre návrh obrazoviek bol použitý softvér na prototypovanie a mockup návrhy Balsamiq. Technológie použité pri implementácii sme popísali v kapitole zaoberajúcej sa realizáciou prototypu aplikácie. Vývoj prebiehal na portáli Liferay CE 6.2 a vytvorené portlety dodržiavajú štandard JRS-286. Samotná aplikácia dokáže pracovať s kurzami, členiť ich do skupín, napĺňať kurzy študijnými materiálmi, spravovať testy a študentov. Prototyp dokáže využiť komponenty DMS a CMS implementované platformou portálu. Aplikácia bola testovaná počas vývoja pomocou JUnit rámcov. Funkčné testovanie prebiehalo podľa testovacích scenárov po dokončení ucelených aplikačných častí. V poslednej podkapitole realizácie môžeme nájsť priloženú dokumentáciu určenú pre administrátorov vzdelávania. Táto aplikácia bola navrhnutá tak, aby sa dala jednoducho používať, rozšíriť a nasadiť do rôznych organizácií, ktoré používajú platformu podnikový portál Liferay. Výsledkom práce je funkčný prototyp aplikácie vytvorený pre potreby firemného vzdelávania.

[95] Dia, http://dia-installer.de/

Zoznam použitej literatúry

[1] SEZOV, Richard. *Liferay In Action (1st. ed.).* MEAP. Greenwich : Manning Publications, 2011. 5 s. ISBN 9781935182825.

[2] JSR 286: Portlet specification 2.0 [online]. 2008 [cit. 13.05.2014] Dostupné na internete: <https://jcp.org/aboutJava/communityprocess/final/jsr286/>.

[3] SARANG, Poornachandra. *Practical Liferay: Java–based Portal Applications Development (1st ed.).* New York, 2009. 3 s. ISBN 1-4302-1847-9.

[4] ASHISH, S. *Portlets in Action*. Manning Publications, 2011. ISBN 1935182544.

[5] NICKLOUS, Martin Scott. *JSR 286: Portlet Specification 2.0*, 2008 [cit. 18.05.2014]. Dostupne na internete: <https://jcp.org/en/jsr/detail?id=286>.

[6] ABDELNUR, A., HEPPER, S. *JSR 168: Portlet Specification* [online], 2003 [cit. 18.05.2014]. Dostupné na internete: <http://www.jcp.org/en/jsr/detail?id=168>.

[7] LINWOOD, J., MINTER, D. *Building Portals with the Java Portlet API,* New York, 2004. 33-38 s. ISBN 1-59059-284-0.

[8] Portlet Tutorial: *Portlet lifecycle* [online]. [cit. 30.05.2014]. Dostupné na internete: <http://jsr286tutorial.blogspot.cz/p/portlet -lifecycle.html>.

[9] YUAN Jonas X, CHEN X., YU Frank. *Liferay user interface development*. 2010 ISBN 978-1-849512-62-6.

[10] Understanding the Java Portlet Specification 2.0 (JSR 286) - Project Kenai [online]. [cit. 30.5.2014]. Dostupné na internete: <https://portlet-container.java.net/docs/jsr>.

[11] LIFERAY [online]. [cit. 5.6.2014]. Dostupné na internete: <http://www.liferay.com/downloads/liferay-portal/license>.

[12] Liferay. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikipedia Foundation, 2011 [cit. 5.6.2014]. Dostupné na internete: <http://en.wikipedia.org/wiki/Liferay>.

[13] MCKAY, C., FERRER, J. Liferay developer guide 6.0. 2011. 11s.

[14] ARMSTRONG, M. *Řízení lidských zdrojů*. 10 vyd. Praha: Grada Publishing, 2007. ISBN 978-80-247-1407-3.

[15] STN EN ISO 9004. *Systémy manažérstva kvality Návod na zlepšovanie výkonnosti. Bratislava* : Slovenský ústav technickej normalizácie [online], 2001 [cit. 1.8.2014]. 104 s. Dostupné na internete: <http://www.kvalitaprodukcie.info/wp-content/uploads/2011/06/STN-ISO-10006-2004.pdf>.

[16] HVIZDOVÁ, E. 2011. *Riadenie ľudských zdrojov v znalostnej ekonomike*. Bratislava: Vydavateľstvo EKONÓM, 2011. 95 s. ISBN 978-80-225-3174-0.

[17] SENGE, P. M. *Pátá disciplína: teorie a praxe učící se organizace*, Praha, 2007, ISBN 978-80-7261-162-1 (The Fifth Discipline: The Art and Practice of the Learning Organization, Doubleday, New York, 1990, ISBN 978-0-385-26095-4).

[18] KUCHARČÍKOVÁ, A. *Ako prispieť ku konkurenčnej schopnosti podniku*. Podniková ekonomika a manažment [online]. 2006, č. 3 [cit. 9.8.2014]. Dostupné na internete: <www: http://ke.uniza.sk/application_data/ke/uploads/Data/novinyke_306fin.pdf> 265-266s, ISSN 1336-5878.

[19] KUCHARČÍKOVÁ, A. *Podniková ekonomika a manažment*. 2006, 6 s.

[20] KOUBEK, J. *Řízení lidských zdrojů*. Praha: Management Press, 2007, 265-266 s., 1998. ISBN 80-85943-51-4.

[21] TURČÁNI, M. *Learning - nová forma práce metodikov*. Spracované z prednášky pre projekt Tvorba e-kurzov. [online]. 2005 [cit. 9.08.2014]. Dostupné na internete: <http://mcmb.mcmb.sk/ESF/tvorba_ekurzov.htm>.

[22] PRUCHA, J., WALTEROVÁ, E., MAREŠ, J. *Pedagogický slovník*. 6. vyd. Praha: Portál, 2009. 66 s. ISBN 978-80-7367-647-6.

[23] DLOUHÝ, J., JANČAŘÍK, A. *Metodika tvorby textů v otevřeném internetovém prostoru/Co je e-learning?/LMS prostředí*. [online]. 2011 [cit. 10.8.2014]. Dostupné na internete: <http://www.enviwiki.cz/w/index.php?title=Metodika_tvorby_text%C5%AF_v_otev%C5%99en%C3%A9m_internetov%C3%A9m_prostoru/Co_je_e-learning/LMS_prost%C5%99ed%C3%AD&oldid=11616>.

[24] PARR, J.M.; Fung, New Zealand Ministry of Education [online]. 2005 [cit. 15.9.2014] Dostupne na internete: <http://www.minedu.govt.nz/index.cfm?layout=document&documentid=5499&ind>.

[25] KERSCHENBAUM, Steven. *LMS Selection Best Practices* [online]. 2009 [cit. 15.9.2014] 1–15s. Dostupné na internete: <http://www.trainingindustry.com/media/2068137/lmsselection_full.pdf>.

[26] LMS Software [online]. 2014 [cit. 20.9.2014] Dostupne na internete: <http://www.capterra.com/learning-management-system-software/#infographic>.

[27] Kontis [online]. 20014 [cit. 20.9.2014]. Dostupné na internete: <http://www.kontis.sk/reseni_reference.asp?menu=reseni&pos=5>.

[28] ŠTÍPEK, Jiří. Learning management systems a budoucnost e-learningu. In: *Učitelský spomocník* [online]. 2005, 26.01.2005 [cit. 2013-05-04]. Dostupné na internete: <http://www.spomocnik.cz/index.php?id_document=129>.

[29] MUKHAR K., ZELENK. CH. *Beginning Java EE 5 from novice to Professional*. 2005. ISBN (pbk): 1-59059-470-3.

[30] JENDROCK, E., EVANS. I., Haase. K., Srivathsa, CH. GOLLAPUDI, D. NAVARO, R. C. *The java EE 6 tutorial, advanced topics*. Indiana 2013. ISBN-13: 978-0-13-708186-8.

[31] What Is JavaScript, media.wiley.com. [online]. [cit. 22.10.2014] Dostupné na internete: <http://media.wiley.com/product_data/excerpt/88/07645790/0764579088.pdf>.

[32] jQuery [online]. [cit. 22.10.2014]. Dostupné na internete: <http://jquery.com/>.

[33] JOHNSON, R. et al. *Spring Framework Reference Documentation 2.0.8*. 2010 [cit. 23.10.2014]. Dostupné na internete: <http://docs.spring.io/spring-framework/docs/2.f0.x/reference/portlet.html>.

[34] BAUER Christian, KING Gavin. *Java Persisence whit Hibernate*. 2007. Manning Publications, ISBN 1-932394-88-5.

[35] MARK Garz, NOVOTNÝ Róbert, *Java Persistence API v príkladoch z Hibernate*. 2009 [cit. 18.11.2014]. Dostupné na: <http://ics.upjs.sk/~novotnyr/java/java-persistence-api-tutorial/jpa.pdf>.

[36] ZELINKA Ľuboš. *Manažment kvality a testovanie softvéru*. [cit. 18.11.2014]. Dostupné na internete: <http://www2.fiit.stuba.sk/~bielik/courses/msi-slov/kniha/2008/essays/msipapersource55-zelinka.pdf>.

[37] ABRAN, A., MOORE, J.W. Guide to the Software Engineering Body of *Knowledge* [online]. 2005 [cit. 19.11.2014]. Dostupné na internete: <http://www.inf.ed.ac.uk/teaching/courses/seoc/2006_2007/resources/SWEBOK_Guide_2004.pdf>.

[38] Javadoc Tool [online], [cit. 18.11.2014]. dostupné na internete: <http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html>.

Prílohy

Príloha A

Obsah priloženého CD

Diplomova_praca.pdf – text práce
Helpdesk.pdf – súbor obsahuje používateľskú dokumentáciu k aplikácii
Javadoc – adresár obsahuje technickú dokumentáciu k projektu
Obrazovky – adresár obsahuje návrhy obrazoviek portletov
Testovacie_scenare.pdf – súbor obsahuje testovacie scenáre k aplikácií
Use_case – adresár obsahuje prípady použitia

Príloha B

Zoznam rozšírených produktov LMS systémov

Atutor - www.atutor.ca
Blackboard - www.blackboard.com
CertpointVLS - www.certpointsystems.com
Dacebo - www.dacebo.org
Desire2Learn - www.desire2learn.com
Dokeos - www.dokeos.com
dotLRN - www.dotlrn.org
eFront - www.efrontlearning.net
Ilias - www.ilias.de
iTutor - www.e-learn.cz
Joomla LMS - www.joomlalms.com
Moodle - www.moodle.org
Mzinga - www.mzinga.com
Oracle iLearning - ilearning.oracle.com
Plateau - www.plateau.com
Saba - www.saba.com
Sakai - www.sakaiproject.org
SharePointLMS - www.sharepointlms.com

Príloha C

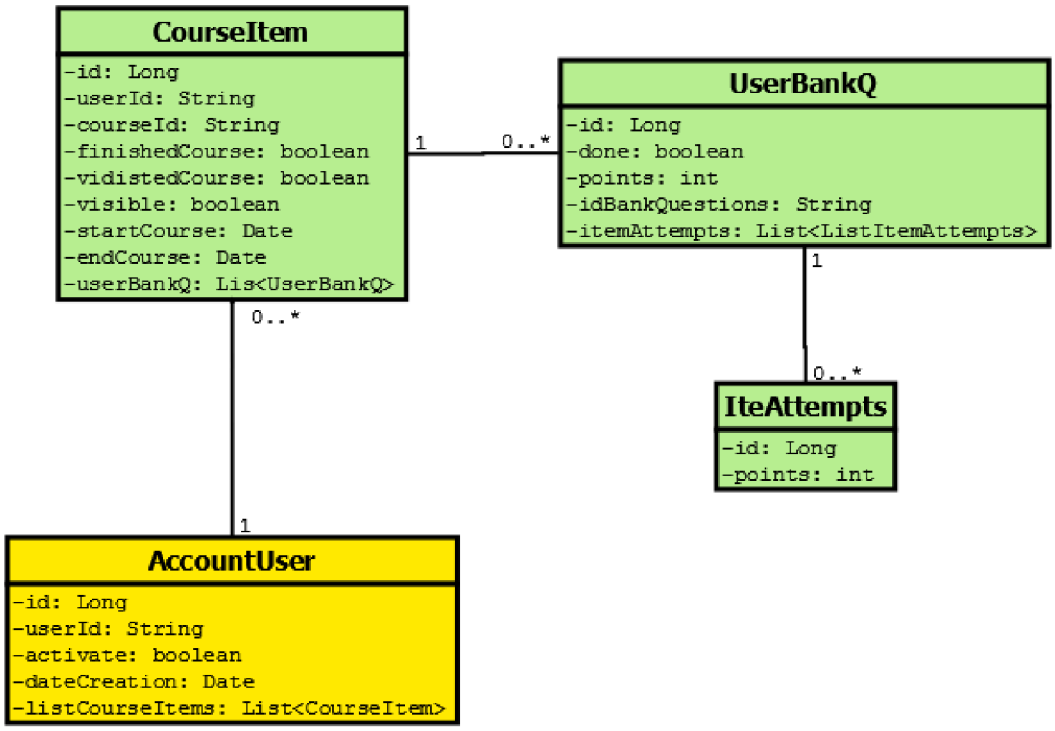



Diagram tried zobrazujúci entity určené k správe účtov