



Báza znalostí a zručností študentov

Vypracoval:	Tím č. 11	Bc. Slavomír Červeň
Vedúci tímu:	Ing. Ivan Kapustík	Bc. Andrej Fenik
Predmet:	Tvorba informačného systému v tíme I.	Bc. Martin Kováčik
Študijný odbor:	Informačné systémy	Bc. Juraj Malečka
Semester:	zimný	Bc. Marián Miština
Školský rok:	2005/2006	Bc. Martina Práznovská
Kontakt:	_elf_@googlegroups.com	Bc. Michal Sabo

Obsah dokumentácie

PROJEKTOVÁ DOKUMENTÁCIA

ZIMNÝ SEMESTER

Úvod	I
Analýza	II
Špecifikácia	III
Hrubý návrh riešenia	IV
Prototyp	V

LETNÝ SEMESTER

Zmeny špecifikácie	VI
Návrh systému	VII
Realizácia (implementácia)	VIII
Overenie (testovanie)	IX
Záver	X

PRÍLOHY

Dokumentácia k produktu	XI
-------------------------	----

DOKUMENTÁCIA K RIADENIU PROJEKTU

ZIMNÝ SEMESTER

Úvod	I
Úlohy členov tímu	II-z
Štandardy kódovania	III-z
Ponuka	IV-z
Plán projektu	V-z
Zápisnice	VI-z
Preberacie protokoly	VII-z
Posudky	VIII-z

LETNÝ SEMESTER

Úlohy členov tímu	II-1
Manažment verzií a štandardy	III-1
Plán projektu	V-1
Zápisnice	VI-1
Posudky	VII-1
Preberacie protokoly	VIII-1



PROJEKTOVÁ DOKUMENTÁCIA

ZIMNÝ SEMESTER

Báza znalostí a zručností študentov

I ÚVOD

1 ZADANIE PROJEKTU	1
2 PREDHOVOR.....	2
2.1 ÚČEL DOKUMENTU.....	2
2.2 PREHEAD DOKUMENTU.....	3
3 SLOVNÍK POJMOV	4
4 OPIS PROBLÉMOVEJ OBLASTI	5

II ANALÝZA

1 ANALÝZA PODOBNÝCH RIEŠENÍ	1
1.1 VLASTNOSTI SYSTÉMOV	1
1.2 TYPY INFORMÁCIÍ, INFORMAČNÝCH ZDROJOV A ICH ZÍSKAVANIE	1
1.3 POSKYTOVANIE INFORMÁCIÍ	2
1.4 TYPY POUŽÍVATEĽOV A SKUPÍN	2
1.5 BEZPEČNOSŤ SYSTÉMOV	2
1.6 TESTOVANIE A BUILDOVANIE SYSTÉMOV	3
1.7 ARCHITEKTÚRA SYSTÉMOV A POUŽITÉ TECHNOLOGIE	3
1.8 PRIEBEH VÝVOJA	3
1.9 ZÁVER Z ANALÝZY PODOBNÝCH RIEŠENÍ	4
2 ANALÝZA PROBLÉMU	5
2.1 ANALÝZA SÚČASNÉHO STAVU	5
2.2 ANALÝZA EXTERNÝCH SYSTÉMOV	5
2.2.1 <i>Informačný systém ŠTUDENT</i>	5
2.2.2 <i>EMA</i>	7
2.2.3 <i>Informačný systém Posudky:</i>	8
3 ANALÝZA TECHNOLOGIÍ	9
3.1 POTREBNÉ KOMPONENTY	9
3.1.1 <i>J2SDK</i>	9
3.1.2 <i>JAVA Aplikačný server</i>	9
3.1.3 <i>AXIS</i>	10
3.1.4 <i>JWSDP</i>	10
3.1.5 <i>Spring framework</i>	10
3.1.6 <i>Nástroj na zostavovanie JAVA aplikácií</i>	12
3.1.7 <i>IDE</i>	12
3.2 KOMPONENTY A NÁSTROJE, KTORÉ NAVRHUJEME POUŽIŤ	13

III ŠPECIFIKÁCIA

1 VŠEOBECNÝ OPIS SYSTÉMU.....	1
1.1 PERSPEKTÍVA SYSTÉMU.....	1
2 ŠPECIFICKÉ POŽIADAVKY	2
2.1 MODEL PRÍPADOV POUŽITIA	2
2.2 CHARAKTERISTIKY POUŽÍVATEĽOV	4
2.3 PRÍPADY POUŽITIA	5
2.4 VSTUPY A VÝSTUPY	15
2.4.1 <i>Vstupy</i>	15
2.4.2 <i>Výstupy</i>	16
2.4.3 <i>Proces naplňania dátami zo systému ŠTUDENT</i>	18
2.4.4 <i>Notifikácia neaktuálnosti dát</i>	19
2.4.5 <i>Vonkajšie rozhranie</i>	19
3 ĎALŠIE POŽIADAVKY	20
3.1 BEZPEČNOSŤ A OCHRANA INFORMÁCIÍ.....	20
3.2 POŽIADAVKY NA SOFTVÉROVÉ VYBAVENIE	21
3.3 POŽIADAVKY NA VÝKONNOSŤ	21

IV HRUBÝ NÁVRH

1 ARCHITEKTÚRA SYSTÉMU.....	1
1.1 PREZENTAČNÁ VRSTVA	1
1.2 POUŽÍVATEĽSKÉ ROZHRANIE	2
1.3 APLIKAČNÁ VRSTVA	2
1.4 APLIKAČNÉ SLUŽBY	2
1.5 OBJEKTIVOVO-RELAČNÉ MAPOVANIE.....	2
1.6 WEBOVÉ SLUŽBY	2
1.7 DÁTOVÁ VRSTVA	3
1.8 BEZPEČNOSŤ SYSTÉMU.....	3
2 MODEL ÚDAJOV	4
2.1 LOGICKÝ MODEL ÚDAJOV	4
2.2 ENTITY LOGICKÉHO MODELU ÚDAJOV	8
2.3 FYZICKÝ MODEL ÚDAJOV	9

I ÚVOD

1 Zadanie projektu

Študent už od začiatku štúdia na fakulte získava, viac či menej úspešne, množstvo odborných informácií a znalostí, ktoré mu môžu významne pomôcť uplatniť sa v oblasti svojho záujmu. Hoci využitie týchto znalostí sa predpokladá najmä v komerčnej sfére, aj na fakulte existujú predmety a projekty, kde študent môže preukázaním svojich znalostí získať určitú výhodu pri výbere konkrétneho projektu či začlenením sa do určitého riešiteľského kolektívu. V súčasnosti však na fakulte neexistuje žiadny systém, ktorý by toto "preukázanie znalostí" podporoval, tzn. v nejakej podobe znalosti a zručnosti študentov získaval, udržiaval či poskytoval.

Cieľom projektu je návrh a realizácia softvérového systému pre organizovanie bázy znalostí a zručností študentov našej fakulty. Informácie uložené v báze znalostí by bolo možné následne využiť napríklad v existujúcom softvérovom systéme na podporu riadenia projektov Yonban, kde by si študent mohol určiť priority vybraných projektov či kolegov, s ktorými by rád spolupracoval (v prípade projektov vyžadujúcich prácu v tíme). Vedúcemu projektu by mohli pri jeho pridelovaní naopak pomôcť informácie o doterajších skúsenostiach, schopnostiach či záujmoch študenta, ako napríklad mimofakultné projekty, sťažky, nepriama informácia o jeho študijných výsledkoch (napr. či sa nachádza v TOP 10 % najúspešnejších študentov z vybraného predmetu) a pod.

V rámci riešenia projektu sa treba zamerať najmä na tieto činnosti:

- získavanie informácií o študentoch rôznymi spôsobmi (priamym vstupom od študentov - záujmy, prax, certifikáty, ovládané technológie; poznámkami cvičiacich; analýzou ich doterajších študijných výsledkov a pod.),
- zabezpečenie pravidelného získavania informácií z rôznych zdrojov (napr. upozorňovaním pomocou e-mailov),
- poskytovanie uchovávaných znalostí pre iné systémy (napr. vo forme webových služieb),
- použitie systému minimálne pre potreby predmetov, v rámci ktorých sa rieši záverečný projekt bakalárskeho štúdia, diplomový projekt alebo projekt tvorby systému v tíme.

2 Predhovor

Fundamentálnou súčasťou každej organizácie sú ľudia a súhrn individualít týchto ľudí určuje individualitu organizácie. Plnenie poslania organizácie a jej úspešnosť v tomto úsilí nie je závislá na žiadnom zákone ani tlaku okolia, ale od schopností jednotlivcov a umení ich správneho použitia. I keď sa v tejto súvislosti používa slovo umenie, v súčasnosti sú k dispozícii metódy a recepty najmä z oblasti riadenia ľudských zdrojov a manažmentu znalostí, ako sa takýmto „umelcom“ stať. Fakulta informatiky a informačných technológií je tiež organizácia, ktorá má svoje poslanie, ľudí (zamestnancov a študentov) i špecifické metódy, ako čo najefektívnejšie sklbiť ľudské zdroje s úlohami, ktoré sa na pôde fakulty riešia. Neustále zlepšovanie týchto metód a zapájanie nových technológií je na našej fakulte samozrejmosťou.

Manažment ľudských zdrojov i manažment znalostí sa nezaobíde bez informácií, ktoré sú systematicky zbierané, ukladané, spracúvané a v prípade potreby vhodne prezentované. V súvislosti so študentmi sa fakulta (vyučujúci) zaujímajú najmä o ich znalosti a zručnosti. Získavanie takýchto informácií sa deje napríklad v procese prijímania na štúdium. O študentských znalostiach a zručnostiach svedčia vysvedčenia zo strednej školy, ocenenia zo súťaží, ale aj výsledky prijímacej skúšky. Ďalšími príkladmi sú pridelovanie záverečných projektov bakalárskeho štúdia, pridelovanie diplomových projektov či uchádzanie sa o zamestnanie. Vo všetkých prípadoch sú rozhodujúce znalosti a zručnosti jednotlivcov. Vznikla teda potreba sústrediť tieto informácie na jednom mieste a vytvoriť systém, ktorého primárnou funkciou bude poskytovanie súhrnných informácií o študentoch, ich agregácia a porovnávanie.

Tento dokument má za cieľ analyzovať problémovú oblasť, špecifikovať požiadavky, funkcie a ohraničenia systému bázy znalostí a zručností študentov a na základe hrubého návrhu zdokumentovať prototyp tohto systému. V ďalšej fáze bude projekt pokračovať etapami podrobného návrhu, implementácie a testovania systému.

2.1 Účel dokumentu

Riešenie zadania tímového projektu *Báza znalostí a zručností študentov* pozostáva z viacerých častí, o ktorých je potrebné zbierať informácie. Ide o fázu definovania požiadaviek, analýzy a návrhu systému, následnej implementácie, testovania a údržby. Dokument slúži na komunikáciu v rámci tímu ako aj na komunikáciu s vonkajším prostredím. Informuje budúcich používateľov, vedúceho tímu, iné tímy, prípadne ďalších záujemcov o výsledky projektovania. Čiže dokument je odzrkadlením priebežného a systematického zhromažďovania a spracúvania informácií za účelom ich uchovania, prenosu a ďalšieho využitia.



2.2 Prehľad dokumentu

Dokumentácia k projektu je členená na viacero kapitol. V prvej kapitole je uvedená analýza externých systémov, súčasného stavu a podobných riešení. Takisto je tu uvedená analýza použiteľných softvérových technológií. V druhej kapitole sa špecifikujú požiadavky na systém pomocou diagramov prípadov použitia. V poslednej kapitole je hrubý návrh spolu s architektúrou, logickým a fyzickým dátovým modelom.

3 Slovník pojmov

BŠP - bodový študijný priemer

ostre dáta - údaje, ktoré sa spracúvajú v bežnej prevádzke (nie testovacie dáta)

PGO - pedagogické oddelenie

ŠTUDENT - existujúci systém

VŠP - výsledný študijný priemer

YonBan - systém pre podporu riadenia projektov na FIIT STU

zahmlenie informácií - zámerné pretransformovanie informácií do podoby s inou (prevažne nižšou) informačnou hodnotou

ZNALOSTI - riešený systém, ktorý slúži na zhromažďovanie, uchovávanie a poskytovanie znalostí a zručností študentov FIIT STU

znalosť - informovanosť, ovládanie niečoho, vedomosti o niečom na základe štúdia a skúsenosti

4 Opis problémovej oblasti

Pojmy znalosť a zručnosť charakterizujú nejakú schopnosť, ktorú študent získal buď štúdiom (znalosť) alebo skúsenosťou (zručnosť). Znalosti študent nemusí získavať iba v škole, ale napríklad aj v práci alebo vo svojom voľnom čase. Zručnosti študent obvykle nadobúda pri práci na projektoch a vychádzajú aj z určitej miery teoretického zvládnutia problematiky. Môžeme teda povedať, že ak má študent zručnosť v istej oblasti, s veľkou pravdepodobnosťou bude mať v rovnakej oblasti aj znalosť.

Typickým meradlom znalostí je známka. Pri pridelovaní projektov (záverečných, diplomových, či iných) sa v prípade absencie iných údajov prihliada práve na známky, aké študent získal z predmetov súvisiacich s témou projektu. Podľa platnej legislatívy (11/2005) sa však nikto bez súhlasu študenta nemôže dozvedieť jeho známky. Tie sa môže dozvedieť iba študent sám. Kritériom na rozhodovanie medzi viacerými uchádzačmi o projekt by teda mala byť informácia, ktorá dáva možnosť porovnať študentov, ale neprezrádza známky. Znamky musia zostať „zahmlené“. Jednou z možností ako to dosiahnuť, je zoradovanie študentov na základe študijných výsledkov. Toto zoradenie však má tiež určité obmedzenia, pretože z percentuálneho rozdelenia známok daného predmetu v danom roku sa v určitých prípadoch dá presne odvodiť známka každého študenta. Existujú dva spôsoby, ako tomu predísť. Nedovoliť zoradovať všetkých študentov daného predmetu, ale iba dostatočne malú podmnožinu, alebo nezoradovať študentov exaktne, ale zaradovať ich do dostatočne veľkých tried, napríklad podľa toho, či patrili medzi 50 percent najlepších alebo 50 percent najhorších.

Pre uchádzača o projekt sú vždy veľkou výhodou skúsenosti v danej oblasti. Meradlom skúseností môže byť počet rokov praxe. Nastáva však problém s overením hodnovernosti študentovho tvrdenia. Študent môže na požiadanie priniesť certifikát, program, ukážku zdrojového kódu, referenciu od zamestnávateľa, prípadne iný dôkaz jeho zručností. Na FIIT zatiaľ neexistuje systém, ktorý by skúsenosti študentov zbieral.

Problémom iného druhu je otázka, ako motivovať študenta, aby svoje údaje pravidelne aktualizoval. Príkladom negatívnej motivácie je index spoľahlivosti študenta, ktorý klesá, ak študent neaktualizuje svoj profil. V očiach vyučujúceho to môže vzbudzovať dojem nezodpovednosti a študent je preto nútený svoj profil udržiavať v aktuálnom stave. Pozitívnu motiváciou by mohla byť odmena za to, že si študent profil aktualizuje, napríklad sprístupnenie informácií o predmetoch (úspešnosť, záujem o predmet v minulých rokoch, percentuálne rozloženie známok a pod.).

K informáciám, ktoré môžu pomôcť pri rozhodovaní v procese pridelovania projektu patria aj také, ktoré sa nedajú merať. Ide o certifikáty a slovné hodnotenia. Certifikát je dokument, ktorý môže vyjadrovať buď znalosť (napr. študent urobil test z Nemčiny na 96 percent) alebo zručnosť (napr. študent sa zúčastnil 4 týždňovej stáže v Nemecku). Slovné hodnotenie je hodnotenie najmä od



cvičiaceho, prípadne sebahodnotenie samotným študentom na spôsob motivačného listu. K certifikátom a slovným hodnoteniam je spravidla možné priradiť niekoľko kľúčových slov, ktoré vyjadrujú ich podstatu. S pomocou kľúčových slov je práca s týmito informáciami jednoduchšia.

Niektoré zo spomínaných informácií sa na Fakulte Informatiky a Informačných Technológií zbierajú už dnes, sú však roztrúsené v rôznych systémoch. Systém ŠTUDENT je určený na zber študijných výsledkov, systém EMA uchováva informácie o predmetoch (názvy, sylaby a pod.) a systém YonBan je určený na uchovávanie informácií o záverečných projektoch bakalárskeho štúdia a diplomových projektoch (názvy, zadania, abstrakty, posudky a pod.). Systém, ktorý by zbieral znalosti a zručnosti študentov zatiaľ neexistuje, bude však musieť spolupracovať s uvedenými existujúcimi systémami, pretože výmena údajov medzi nimi bude nevyhnutná.

II ANALÝZA

1 Analýza podobných riešení

Pri štúdiu a získavaní informácií o problémovej oblasti sa nám nepodarilo nájsť systém, ktorý by svojou špecifikáciou zodpovedal nášmu systému, preto sme sa zamerali na analýzu podobných systémov.

Prvý systém je z oblasti poisťovníctva (ozn. **S1**) a druhý je z oblasti logistiky (ozn. **S2**). Obidva systémy sú nasadené v prevádzke a spoľahlivo plnia svoj účel. Informácie o týchto systémoch sme získali na základe skúseností členov tímu pri vývoji týchto systémov.

1.1 Vlastnosti systémov

- obidva systémy sú webové aplikácie, ktoré sprístupňujú požadované informácie a umožňujú používateľom manipuláciu (prezeranie, modifikáciu, vyhľadávanie, vymazanie) s nimi podľa ich oprávnení
- vstupy do systémov môžu byť zo súboru, z iného systému, pomocou užívateľského formulára
- všetky informácie sú uložené v databáze
- pri **S1** je webová aplikácia rozhranie, ktoré sprístupňuje informácie z už existujúceho systému
- zabezpečenie pomocou autentifikácie, autorizácie a nastavenia práv pre jednotlivé objekty
- zabezpečenie komunikácie pomocou SSL
- trojvrstvová architektúra (prezentačná, aplikačná, dátová vrstva)

1.2 Typy informácií, informačných zdrojov a ich získavanie

Pri **S1** sú hlavnými údajmi, ktoré sú uchovávané informácie o zákazníkoch a o poisťkách. Tieto informácie vstupujú do systému prostredníctvom formulára na ich zadanie. Druhý spôsob vstupu informácií je vstup z existujúceho systému. V tomto prípade existuje v systéme modul, ktorý zabezpečí získanie údajov z produkčnej databázy a transformuje ich do XML súboru. Tento XML súbor s údajmi vstupuje do systému a ukladá údaje do databázy webovej aplikácie.

V **S2** sú hlavnými typmi údajov informácie o prepravách. Tieto informácie sa získavajú pomocou skenera priamo zo zásielok. Naskenované súbory sa pomocou programu transformujú do textového súboru s pevným formátom. Tento súbor je vstupom do aplikácie. Príslušný modul

v aplikácii zabezpečí parsovanie tohto súboru a vloženie potrebných údajov do databázy. Druhým spôsobom získavania vstupných informácií je vstup priamo od užívateľa pomocou formulára na zadanie informácií.

Postup pri vložení informácií pomocou súboru:

- import súboru na web server
- kontrola formátu súboru
- kontrola duplicity údajov (či už bol tento súbor importovaný)
- parsovanie dokumentu
- vloženie do databázy
- vymazanie súboru z web servera

1.3 Poskytovanie informácií

Informácie, ktoré sú uložené v systémoch sú prístupné pomocou formulára, kde je možné si ich prehliadať alebo je možné ich exportovať. V systéme **S2** je umožnený export do formátov csv, xls, pdf alebo xml. Tieto exportované dáta slúžia ako vstup pre iné systémy (napr. fakturačný systém).

Používateľ si môže exportovať iba údaje, ku ktorým má prístup.

1.4 Typy používateľov a skupín

V **S2** sú definované štyri skupiny používateľov, ktoré zodpovedajú štyrom oddeleniam firmy, ktorá systém využíva. Každé oddelenie pracuje s istým druhom údajov a ostatné údaje nie sú pre nich dôležité. Na základe toho boli definované práva pre jednotlivé skupiny. V rámci skupín sa vytvárajú používatelia, ktorými sú pracovníci jednotlivých oddelení. Títo používatelia implicitne dedia prístupové práva skupiny, do ktorej patria. V prípade potreby je možné ich zmeniť.

1.5 Bezpečnosť systémov

Obidva systémy využívajú na zabezpečenie údajov technológie autentifikácie a autorizácie.

Používateľ, ktorý chce pracovať so systémom sa musí najskôr prihlásiť pomocou prihlasovacieho formulára. Zadaním a potvrdením svojho mena a hesla sa spustí proces autentifikácie. Tieto údaje sa porovnávajú s údajmi, ktoré sú uložené v databáze. Pokiaľ sa nájde zhoda, užívateľovi je umožnené prihlásenie sa do systému. Meno a heslo sú pri tomto procese posielané v šifrovanej podobe.

Každý používateľ je zaradený do jednej alebo viacerých skupín (roly). Informácie o skupinách sú uložené v databáze. Pri prihlasovaní sa zároveň zistí do akej skupiny používateľ patrí a aké má

oprávnenia. Je možné definovať „access list“, v ktorom sú uvedení používatelia alebo skupiny, ktoré majú povolený alebo zakázaný prístup k jednotlivým stránkam.

V S1 existuje ďalší stupeň ochrany na úrovni nastavenia práv pre jednotlivé objekty v systéme. Tieto práva sú na úrovni viditeľnosti konkrétneho objektu (objekt je viditeľný alebo neviditeľný - zakázaný). Administrátor systému tak môže nastaviť pre konkrétneho používateľa prístup k ľubovoľnému objektu.

V S1 aj S2 je ďalej definované nastavenie prístupových práv pre zobrazované informácie. Tieto práva sú typu RO(ReadOnly), RW(ReadWrite) a NA(NotAvailable).

Komunikácia s webovým serverom je šifrovaná pomocou SSL a využíva protokol HTTPS. Na webovom serveri je potrebné mať nainštalovaný certifikát od dôveryhodnej certifikačnej autority.

1.6 Testovanie a buildovanie systémov

Pri testovaní funkčnosti systémov sme využívali unit testy. Využívali sme testovací framework NUnit, ktorý je určený pre testovanie .NET aplikácií. Tento framework je klon frameworku Junit, ktorý je určený pre technológiu Java. Pri testovaní sme vytvorili sadu testov, ktoré otestovali funkčnosť aplikácie. NUnit je automat, ktorý spúšťal tieto testy a zobrazoval štatistiky.

Ako buildovací nástroj sme využívali NAnt, ktorý je podobný nástroju Ant, ktorý sa používa na buildovanie aplikácií založených na Jave.

1.7 Architektúra systémov a použité technológie

Obidva systémy využívajú trojvrstvovú architektúru. Prezentačná vrstva je implementovaná pomocou webového prehliadača (štandard HTML 4.0).

Táto vrstva má na starosti interakciu s používateľom.

Aplikačná vrstva je implementovaná pomocou .NET frameworku a IIS(Internet Information services). Využíva štandard ASP.NET 1.0. Táto vrstva implementovala logiku aplikácie. V prípade poisťovacieho systému išlo o implementáciu poisťovacích pravidiel.

Dátová vrstva je založená na SQL 2000 serveri. Táto vrstva obsahovala hlavne technický kód a zabezpečovala načítanie údajov z databázy a ich spätné uloženie. Taktiež tu boli zaradené úložné procedúry, ktoré zabezpečili zrýchlený prístup k dátam.

1.8 Priebeh vývoja

Pri vývoji systémov sa vychádzalo z požiadaviek zákazníka na vytváraný systém. Na základe toho sa vytvorila podrobná špecifikácia požiadaviek. Z tejto špecifikácie sa vychádzalo pri návrhu architektúry a štruktúry systémov.

Pri návrhu architektúry sa celý systém rozvrhol do jednotlivých komponent, ktoré ho tvorili. Tieto komponenty mali definované presné rozhrania, aby bolo možné vytvárať kód využívajúci komponent, ktorý ešte nebol dokončený. Vývoj každého komponentu obsahoval fázy analýzy, programovania a testovania.

Návrh architektúry bol rozdelený do troch vrstiev (dátová, aplikačná, prezentačná). Pre implementáciu jednotlivých vrstiev boli určení vývojári, ktorí sa špecializovali na konkrétnu oblasť.

Následne sa navrhli a vybrali technológie, vývojové prostriedky a knižnice, ktoré boli použité pri vývoji a taktiež popis hardvérových a softvérových požiadaviek na systém (operačný systém, databázový a aplikačný server, komunikačné a sieťové komponenty).

V ďalšej časti sa navrhla štruktúra databázy, jednotlivých komponentov a zároveň testovanie týchto komponentov aj celej aplikácie.

Následovala fáza implementácie a testovania jednotlivých komponentov ako aj databázovej štruktúry.

1.9 Záver z analýzy podobných riešení

V tejto časti sa pokúsime zhrnúť akými vlastnosťami analyzovaných systémov by sme sa mohli inšpirovať a podobne ich realizovať v našom systéme. Aj keď opisované systémy nie sú totožné s našim systémom, majú niektoré spoločné vlastnosti.

Pri získavaní informácií môžeme využiť zadávanie informácií pomocou formulára alebo pomocou vstupného súboru ako to bolo uvedené vyššie.

V prípade, že náš systém bude získavať informácie z iného systému v reálnom čase a nie prostredníctvom dávkových súborov, môžeme navrhnúť procesný modul, ktorý bude tento proces zabezpečovať. Výstupom by mohol byť súbor s dátami vo formáte XML, ako to bolo uvedené vyššie.

Taktiež sa môžeme inšpirovať exportovaním údajov do rôznych typov výstupných súborov.

V otázke bezpečnosti by sme mohli využiť technológie autentifikácie a autorizácie ako aj zabezpečenia prístupu k jednotlivým objektom. Taktiež by sme mohli použiť šifrovanú komunikáciu pomocou SSL.

Tieto analyzované systémy neriešia otázku konfigurovateľnosti systému, pretože vzhľadom k oblasti ich použitia to ani nie je potrebné. Konfigurácia v týchto systémoch sa objavuje iba na úrovni vytvárania nových používateľov a skupín a nastavenia prístupových práv.

V našom systéme uvažujeme použiť konkurenčnú technológiu J2EE, a preto nebudeme môcť využiť znalosti technológie .NET. Vzhľadom k niektorým podobným črtám obidvoch technológií nám však môže uľahčiť prácu pri štúdiu novej technológie.

Pri testovaní a buildovaní aplikácií môžeme využiť znalosti z použitia nástrojov, ktoré sú principiálne podobné nástrojom, ktoré sa používajú pre technológiu J2EE.

2 Analýza problému

Tento projekt vznikol z potreby informácií o študentoch pri pridelovaní bakalárskych a diplomových projektov. Zadávateľ projektu dostane prostredníctvom systému YONBAN zoznam študentov, ktorí majú záujem o jeho tému. Zadávateľ nemá žiadne priame informácie o týchto študentoch, preto vzniká potreba informačného systému, ktorý by poskytoval informácie o znalostiach študentov.

2.1 Analýza súčasného stavu

V súčasnosti nie je možné získať zo študijného oddelenia ani BŠP študentov pre takéto potreby. Preto náš systém nesmie poskytovať priame BŠP ale iba napr. zoznam prvých X študentov najlepších v nejakom predmete.

Informácie pre náš systém sa dajú získať zo systému:

- ŠTUDENT – známky
- YONBAN – slovné hodnotenia bakalárskych a diplomových projektov
- Vstup od pedagógov – slovné hodnotenia pedagógov po skončení semestra

2.2 Analýza externých systémov

Tieto externé zdroje budú použité ako zdroje dát pre náš systém.

2.2.1 Informačný systém ŠTUDENT

Tento systém používajú všetky fakulty Slovenskej Technickej Univerzity. Používa sa na kompletnú evidenciu týkajúcu sa štúdiá na vysokej škole. Sú v ňom uložené základné informácie o vyučovaných predmetoch (názov, identifikačné číslo, počet kreditov, ...) a o študentoch (celé meno, osobné číslo, absolvované predmety s príslušným hodnotením študenta, ...)[1].

Študijné výsledky sa budú získať po skončení semestra ručným vygenerovaním obsahu databázy systému ŠTUDENT. Formát extraktu je v prílohe A, ktorá obsahuje iba cenzurované študijné výsledky. Výsledné hodnotenie, t.j. počty známok z predmetov sa môžu taktiež použiť, formát tohto extraktu je v prílohe B.

Systém ŠTUDENT bol vyvinutý AIS CVT STU v Bratislave v tesnej spolupráci s firmou SyComp Bratislava. Systém môžeme opísať nasledovne:

Funkcie systému

Program obsahuje tieto podsystemy a služby:

- základné údaje
- študijné výsledky
- voľba študijných odborov a zameraní
- štipendiá
- rodinné prídavky
- odmeny SVS a SPS
- ubytovanie v študentskom domove
- manipulácie a analýzy
- servisné činnosti
- archív a obnova databáz

Obsah vybraných podsystemov:

- Základné údaje - individuálne zmeny základných a ročných údajov, hromadné zmeny zaradenia študentov do skupín, tlač rôznych zoznamov, adres a štatistík.
- Študijné výsledky - individuálne a hromadné zmeny študijných výsledkov, tlač rôznych hárkov, výpisov, štatistík a analýz, výpočet pedagogických výkonov na fakulte, práca s číselníkmi evidencie SV (číselník predmetov, študijné plány a pod.).
- Štipendiá - individuálne zmeny štipendijných údajov, tlač návrhovej a výplatnej štipendijnej listiny, štatistika a ročný prehľad vyplatených štipendií, výpočet konkurzných štipendií.

Program obsahuje spracovanie rôznych detailových aj agregovaných tlačových výstupov, z ktorých niektoré sú k dispozícii ako samostatné služby programu, iné možno realizovať pomocou zabudovaných nástrojov univerzálneho charakteru.

Používatelia systému

Používatel'mi systému sú pracovníci pedagogického oddelenia (PGO). Študenti majú prístup k svojím študijným výsledkom cez webové rozhranie. Pomocou tohto rozhrania sa môžu informovať o známkach absolvovaných predmetov, o predmetoch, ktoré majú zapísané a tiež o počte získaných kreditov a BŠP. Údaje v systéme sú denne aktualizované a zálohované.

Platforma

Program je vypracovaný v databázovom prostredí FoxPro 2.6 a je určený na prevádzku na osobných počítačoch triedy min. 386, RAM 4 MB, v lokálnej počítačovej sieti typu Novell NetWare, MS Windows, Banyan Vines, LAN Manager a pod.[2].

Zhrnutie vlastností

IS "Študent vysokej školy" nám ponúka možnosť získať informácie o dosiahnutých študijných výsledkoch a o predmetoch, ktoré má študent zapísané .

2.2.2 EMA

Systém EMA slúži hlavne ako databáza informácií o predmetoch. Je prístupný cez webové rozhranie. Tento systém sme mali možnosť analyzovať len zo strany používateľa – študenta, preto analýza ponúka len čiastočné informácie.

Funkcie systému

Systém eviduje informácie o predmetoch vyučovaných na FEI STU. Predmety sú rozdelené do 3 skupín: predmety bakalárskeho štúdia, predmety inžinierskeho štúdia a fakultné voliteľné predmety. Pri každom predmete sa eviduje:

- názov a číslo predmetu
- semester
- počet kreditov
- ukončenie predmetu
- počet hodín prednášok za týždeň
- počet hodín cvičení za týždeň a typ cvičenia
- kľúčové slová
- anotácia
- sylaby
- harmonogram
- garant
- nadväznosť predmetu
- podmienky absolvovania
- literatúra
- poznámka

Zhrnutie vlastností

Pre náš systém predstavuje EMA zdroj údajov o predmetoch. Tento systém sa nepoužíva na FIIT. Alternatíva je získavať údaje o predmetoch ako kľúčové slová, zhrnutie a sylaby zo študijného plánu na stránke fakulty. Prípadne z iného priameho zdroja, napr. od Ing. Petra LACKA, ktorý podľa informácií z PGO poskytuje zoznamy predmetov na spracovanie [3].

2.2.3 Informačný systém Posudky:

Informačný systém Posudky obsahuje hodnotenia bakalárskych a diplomových projektov a taktiež abstrakt a kľúčové slová riešených projektov. Z nich by sme mohli získavať informácie o znalostiach študentov.

Administrátor systému - Roman Filkorn - sa vyjadril, že jediný prístup k databáze Posudky je priamy prístup: "*Ideálne by bolo, keby ste pristupovali k tej istej DB a z nej "online" získavali údaje*".

3 Analýza technológií

Náš systém chceme vytvoriť ako webovú službu, preto je potrebné zistiť aké rôzne prístupy sa využívajú pri tvorbe webových služieb.

3.1 Potrebne komponenty

Na vývoj a prevádzku JAVA webových služieb sú potrebné nasledovné súčasti:

- J2SE SDK (J2SDK) - Java 2 Standard Edition Software Development Kit
- Aplikačný server
- Pri použití aplikačného servera, ktorý nemá implementovanú úplnú J2EE špecifikáciu (napr. Tomcat) treba tiež komponenty na vývoj webových služieb
- Vhodné je tiež použiť nejaký nástroj na zostavovanie JAVA aplikácií (ant, maven, make,...)
- Vhodné je použiť niektoré z voľne dostupných IDE (Integrated Development Environment)

3.1.1 J2SDK

Balíček na vývoj JAVA aplikácií je možné zdarma stiahnuť zo stránok SUN microsystems. Tento balík je už nainštalovaný v softvérovom štúdiu vo verzii 1.4, čo nám vyhovuje a nie je potrebné robiť žiadne zmeny.

3.1.2 JAVA Aplikačný server

Tu sa ponúka niekoľko riešení. Budeme sa zaoberať iba riešeniami, ktoré sú voľne dostupné, najlepšie však open-source riešenia.

Apache Tomcat

Apache Tomcat je open-source aplikačný server. Nie je to úplná implementácia J2EE špecifikácie, je to servlet kontajner pre Java Servlet a JavaServer Pages technológie. Tento aplikačný server je nainštalovaný v softvérovom štúdiu. Pri použití tohto aplikačného servera by však bolo potrebné doinštalovať nástroje na vývoj webových služieb (SOAP, XML), nakoľko ako už bolo povedané Tomcat neimplementuje úplnú J2EE špecifikáciu, ktorá tiež zahŕňa API pre vývoj webových služieb. Ponúka sa tu niekoľko riešení. Najznámejšie sú JWSDP (Java Web Services Development Pack) od SUNu a AXIS od Apache.

JBoss

JBoss je asi najpoužívanejší open-source J2EE aplikačný server. JBoss implementuje úplnú J2EE špecifikáciu. Poskytuje teda okrem iného API pre vývoj webových služieb. Ďalšou vlastnosťou je podpora clusteringu a cachovania. JBoss nie je nainštalovaný v softvérovom štúdiu.

Sun Java system application server

Aplikačný server od SUNu, nie je open-source, ale je dostupný zdarma. Implementuje úplnú J2EE špecifikáciu. Teda podobne ako JBoss poskytuje teda okrem iného API pre vývoj webových služieb.

3.1.3 AXIS

Apache AXIS je open-source implementácia SOAP protokolu. AXIS je hojne používaná báza pre webové služby. Má aktívnu používateľskú základňu a používa ho veľa spoločností ako základ pre svoje webové služby. Výhodou je výborná dokumentácia a veľa názorných príkladov ako AXIS používať. AXIS ku správne fungovaniu potrebuje knižnicu Apache Xerces, ktorá obsahuje funkcie na parsovanie XML dokumentov. Knižnica Xerces je nainštalovaná v softvérovom štúdiu, AXIS nie je.

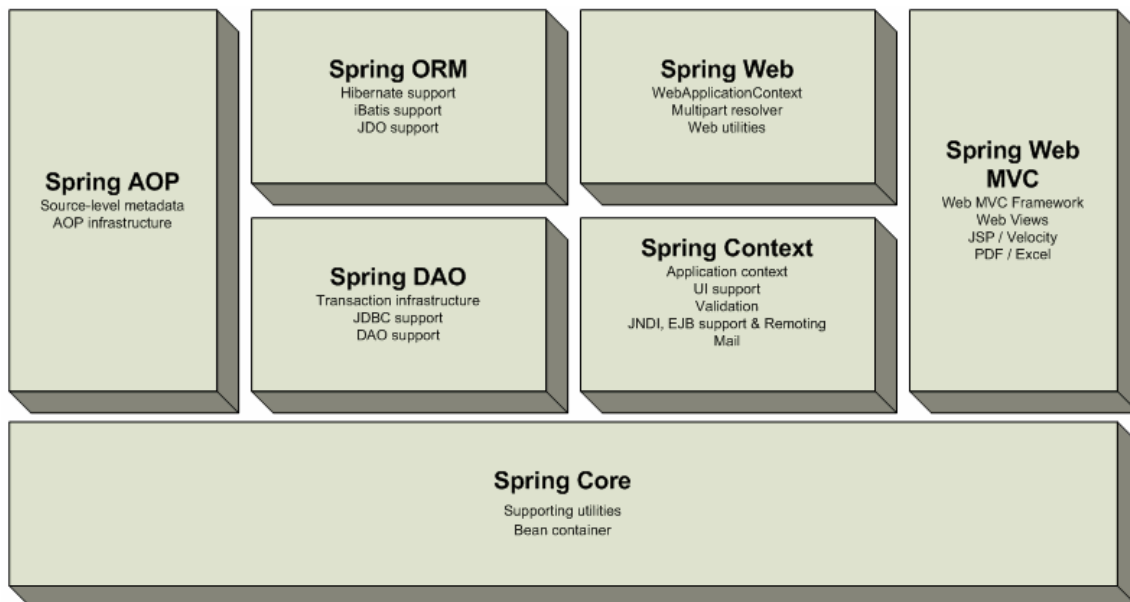
3.1.4 JWSDP

JWSDP je balíček od SUNu na vývoj webových aplikácií. Tento balíček obsahuje všetky súčasti, ktoré je potreba na vývoj webových služieb. Nemá takú širokú základňu ako AXIS, pravdepodobne preto, že je to balíček určený hlavne pre vývoj webových aplikácií, nie však už pre ich prevádzku. Tento balíček je založený prevažne na softvéri od spoločnosti Apache (Ant, Xerces, Xindice)

3.1.5 Spring framework

Architektúra a vlastnosti

Spring Framework je systém pre podporu vývoja aplikácií. Jeho snahou je uľahčiť prácu s J2EE, resp. byť lepšou alternatívou k Enterprise JavaBeans. Základná architektúra tohto frameworku je na nasledujúcom obrázku [Obrázok II - 1].



Obrázok II – 1: Základné moduly frameworku Spring

Jednotlivé moduly sa využívajú v rámci kontajneru a jeho inicializácia je otázkou jedného riadku kódu. Kontajner je prostredie, v ktorom sa odohráva život všetkých objektov, ktoré pomocou Springu spravujeme.

Práca s databázou

Modul DAO frameworku Spring poskytuje abstraktnú vrstvu pre prácu s JDBC API. Jeho hlavné výhody sú, že odstraňuje veľa zbytočného kódu (získanie connection, zrušenie connection, iterovanie cez result set). Ďalšou výhodou je správa výnimiek, ktorá je prevedená z `java.sql.SQLException` do inteligentnej hierarchie runtime výnimiek o ktoré sa nemusí starať programátor. Snahou je, aby sa programátor zamerlal len na prácu s SQL a extrahovanie výsledkov.

Výhody použitia

- jednoduchý na používanie
- nie je náročný na učenie
- pomerne triviálna inicializácia
- jednoduché použitie pre unit testovanie (JUnit)
- informácie o prepojení sú uložené v XML konfiguračných súboroch
- integráciu môže robiť jeden človek (programátori sa nemusia objektovému rámcu prispôsobovať)

- umožňujú nezávislú implementáciu komponentov
- podpora objektovo-relačných mapovačov (Hibernate, iBatis,...)
- učí používať best-practises (overené veci), nemusíme sa učiť na vlastných chybách
- menej kódu
- zjednodušenie používania ďalších častí J2EE (JavaMail, JDBC, ...)
- podpora pre webové aplikácie

Nevýhody použitia

- nie je to štandard
- nevhodný pre distribuované aplikácie, keďže priamo nepodporuje volania vzdialených objektov (skôr určený na lokálne "pozliepanie" modulov)

3.1.6 Nástroj na zostavovanie JAVA aplikácií

Tu sa s výhodou môžu použiť nástroje Apache Ant alebo Apache Maven.

Nástroj Ant je s obľubou používaný medzi vývojármi JAVA aplikácií, má však niekoľko nedostatkov. Hlavný nedostatok je potreba vytvoriť pomerne zložitý XML skript build.xml, ktorý hovorí nástroju Ant, čo má vykonať. Ďalším nedostatkom je nemožnosť znovupoužitelnosti skriptu build.xml a z tohto plynúce ďalšie nedostatky.

Nástroj Maven sa snaží odstrániť nedostatky nástroja Ant. Namiesto zložitého XML skriptu build.xml stačí vytvoriť XML súbor project.xml, v ktorom popíšeme štruktúru projektu. O všetko ostatné sa postará nástroj Maven.

V našom projekte by sme chceli využívať nástroj Maven, ktorý však nie je nainštalovaný v softvérovom štúdiu.

3.1.7 IDE

Kvôli uľahčeniu práce pri vývoji nášho systému by sme radi využili možnosti niektorého z integrovaných vývojových prostredí (IDE). Asi najznámejšie IDE sú Eclipse a Netbeans. Obe prostredia sa vyznačujú množstvom funkcií, ktoré ponúkajú. Obe prostredia sú open-source, obe taktiež majú veľkú používateľskú základňu. S vlastnej skúsenosti však môžeme povedať, že prostredie Netbeans pôsobí o niečo menej prehľadne ako Eclipse. Netbeans má tiež o niečo väčšie hardvérové nároky (RAM) ako Eclipse. Pre Eclipse tiež existuje veľké množstvo prídavných modulov (pluginov), ktoré ešte viac rozširujú jeho možnosti. Prostredie Eclipse je dostupné v softvérovom štúdiu.



3.2 Komponenty a nástroje, ktoré navrhujeme použiť

Ako aplikačný server navrhujeme použiť systém Apache Tomcat. Jednak aj kvôli tomu, že je nainštalovaný v softvérovom štúdiu, jednak aj pre to že je to rozšírená architektúra a v prípade problémov existuje mnoho riešení zo strany komunity používateľov. Keďže bude treba ešte knižnice na vývoj webových služieb navrhujem použiť Apache AXIS spoločne s knižnicou Apache Xerces.

Ako nástroj na zostavovanie aplikácií navrhujeme použiť Apache Maven. Maven sa javí ako inteligentný nástupca Apache Ant, ktorý rieši niektoré nedostatky Antu.

Ako integrované vývojové prostredie navrhujeme použiť Eclipse. Jednak ide o rozšírené a odskúšané prostredie, jednak je k dispozícii v softvérovom štúdiu.

III ŠPECIFIKÁCIA

V tejto špecifikácii je zhrnutý súbor požiadaviek zákazníka, ktorý bude základným odrazovým bodom pri fáze návrhu a implementácie systému.

Produkt sa vyvíja najmä za účelom uľahčenia rozhodovania vyučujúcich pri pridelení projektov ako aj na zhromažďovanie znalostí študentov Fakulty informatiky a informačných technológií STU. V nasledujúcich podkapitolách sú uvedené funkcionálne a nefunkcionálne požiadavky na systém, jeho vstupy a výstupy, používatelia a okrem iného aj bezpečnostné požiadavky.

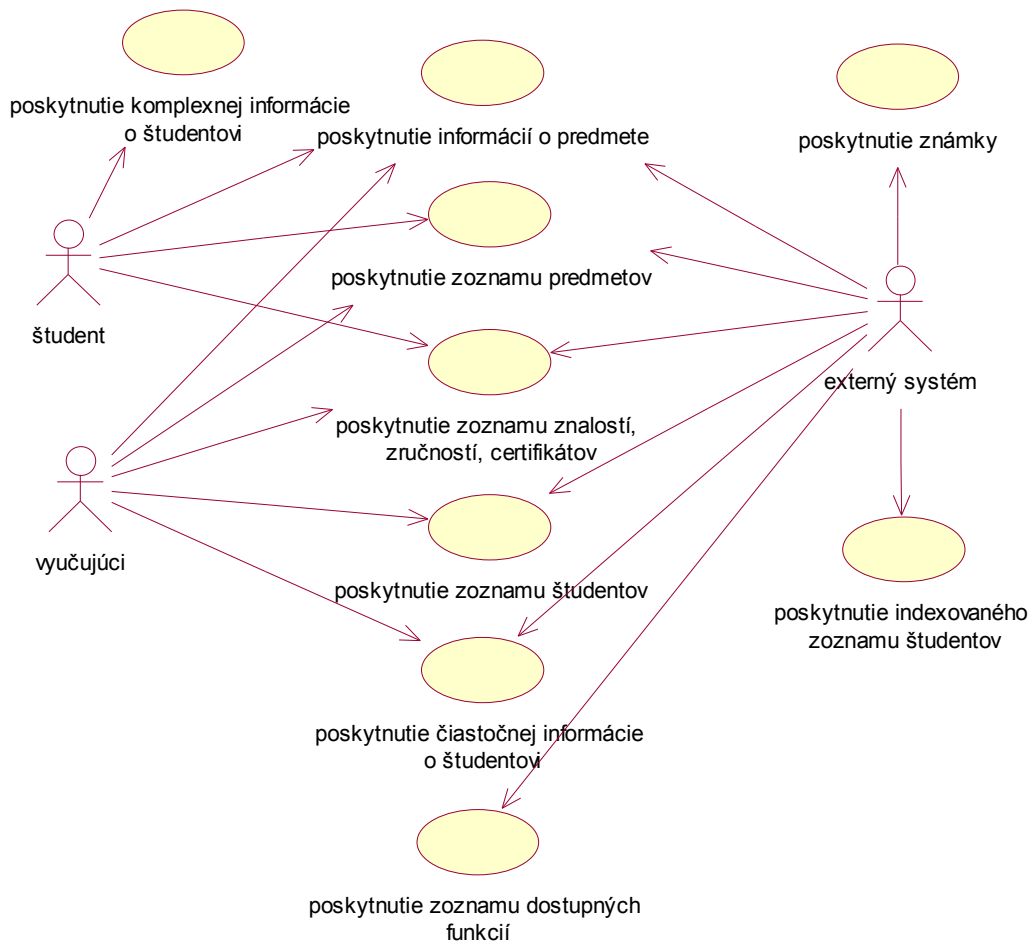
1 Všeobecný opis systému

Realizovaný systém ZNALOSTI je softvérový systém, ktorý bude využiteľný na akademickej pôde; jeho hlavnou úlohou bude organizovanie bázy znalostí a zručností študentov našej fakulty. Informácie uložené v systéme bude možné získať pre ďalšie spracovávanie v externých systémoch (napr. v systéme YonBan).

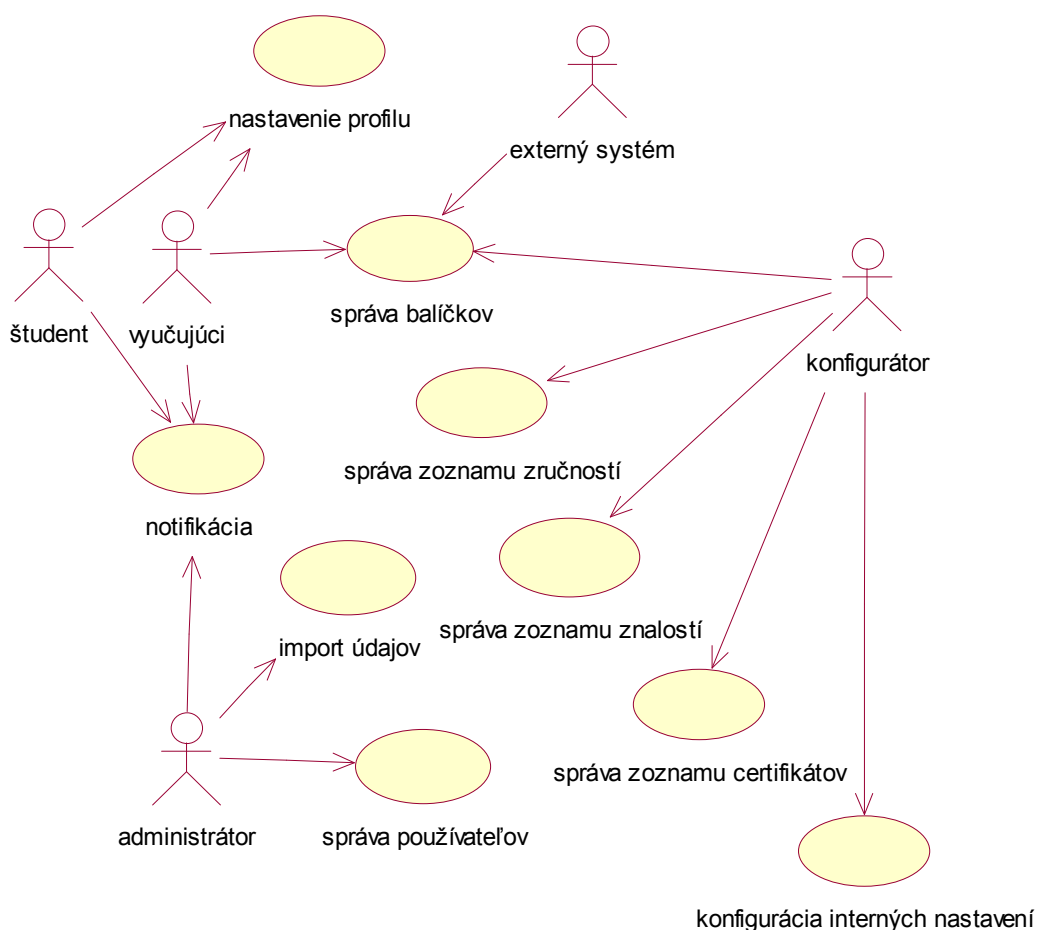
Používateľ systému (vyučujúci alebo externý systém) si bude môcť určiť kritériá a priority, podľa ktorých si zoradí vybraných študentov – vhodných kandidátov na projekt a pod. Vyučujúcim budú teda pomáhať informácie o doterajších skúsenostiach, schopnostiach a záujmoch študenta. Systém však nebude poskytovať priame informácie o študijných výsledkoch ani osobné číslo spolu s menom študenta kvôli zákonu o ochrane osobných údajov (Zákon č. 363/2005 Z.z. o ochrane osobných údajov v znení neskorších predpisov).

1.1 Perspektíva systému

Systém ZNALOSTI nebude izolovaný systém, ale môže byť súčasťou väčšieho celku tým, že bude spolupracovať s inými systémami – či už pomocou exportov a importov alebo priamou komunikáciou. Realizovaný systém nebude komunikovať priamo so systémom ŠTUDENT, nakoľko ŠTUDENT túto funkciu neposkytuje ako aj kvôli citlivosti a bezpečnosti spracovávaných údajov. Údaje sa budú zo systému ŠTUDENT importovať. Jeho výhodou bude konfigurovateľnosť a modulárnosť, jeho štandardné rozhranie a možnosť rozšírenia.



Obrázok III - 2: Diagram prípadov použitia č.2



Obrázok III - 3: Diagram prípadov použitia č.3

2.2 Charakteristiky používateľov

Z hľadiska rolí používateľov systému sú identifikovaní nasledovní používatelia:

Študent	Zadáva do systému informácie o vlastných znalostiach a zručnostiach.
Vyučujúci	Zadáva do systému informácie o študentoch. Požaduje od systému znalosti o študentoch.
Externý systém	Využíva znalosti poskytnuté navrhovaným systémom na ich ďalšie spracovanie alebo priamu prezentáciu, alebo poskytuje informácie o študentovi (ako sú známky, VŠP a pod.)... Získava informácie o znalostiach študenta.



Administrátor	Spravuje systém a má na starosti jeho správny chod počas rutinnej prevádzky.
Konfigurátor	Konfiguruje základné parametre systému, určuje, aké typy informácií budú prístupné používateľom.

2.3 Prípady použitia

Funkcie systému (popísané prípadmi použitia) sa dajú rozdeliť na dve základné skupiny z:

1. funkcie súvisiace so získavaním informácií [Obrázok III - 1]
2. funkcie súvisiace s poskytovaním informácií [Obrázok III - 2]
3. funkcie súvisiace so správou a konfiguráciou systému [Obrázok III - 3]

Nasleduje opis jednotlivých prípadov použitia:

Poznámka: Pre všetky prípady použitia týkajúce sa vkladania údajov existuje aj obdobný prípad použitia editácie týchto údajov. Pre jednoduchosť však tieto funkcie nie sú explicitne uvedené, avšak systém ich musí obsahovať.

<i>Pridanie certifikátu</i>	
<i>Vstupy:</i>	certifikát alebo ocenenie v súťaži, popis napr. linka
<i>Výstupy:</i>	všetky vstupy (uložia sa do databázy)
<i>Používatelia:</i>	študent, externý systém
<i>Opis:</i>	Vloženie získaného certifikátu alebo ocenenia v súťaži, ďalej je možné vložiť dôkaz teda hypertextový odkaz potvrdzujúci toto ocenenie alebo certifikát. Táto funkcia je súčasťou vyplnenia profilu študenta [Obrázok III - 5].
<i>Pridanie zručnosti</i>	
<i>Vstupy:</i>	zručnosť teda praktická skúsenosť, stupeň ovládania, popis napr. linka
<i>Výstupy:</i>	všetky vstupy (uložia sa do databázy)
<i>Používatelia:</i>	študent
<i>Opis:</i>	Vloženie stupňa skúsenosti viažuceho sa na určitú technológiu/oblasť. Je možné vložiť aj popis alebo hypertextový odkaz potvrdzujúci danú zručnosť. Táto funkcia je tiež súčasťou vyplnenia profilu študenta [Obrázok III - 5].

**Pridanie znalosti**

<i>Vstupy:</i>	vedomosť, úroveň ovládania, popis
<i>Výstupy:</i>	všetky vstupy (uložia sa do databázy)
<i>Používatelia:</i>	študent
<i>Opis:</i>	Vloženie stupňa vedomosti viažuceho sa na určitú technológiu resp. oblasť. Je možné vložiť aj popis, ktorý upresňuje alebo potvrdzuje danú znalosť. Táto funkcia je súčasťou vyplnenia profilu študenta [Obrázok III - 5].

Pridanie textu o sebe

<i>Vstupy:</i>	text hodnotenia samého seba, kľúčové slová
<i>Výstupy:</i>	všetky vstupy (uložia sa do databázy)
<i>Používatelia:</i>	študent
<i>Opis:</i>	Vloženie hodnotenia študenta samého, zamýšľanej profilácie, v ktorej môže vyzdvihnúť svoje schopnosti také, ktoré sa nedajú charakterizovať znalosťami, zručnosťami a certifikátmi. Je možné vložiť ľubovoľný text, teda aj hypertextový odkaz. Táto funkcia je tiež súčasťou vyplnenia profilu študenta [Obrázok III - 5].

Pridanie hodnotenia

<i>Vstupy:</i>	text hodnotenia študenta pedagógom, klasifikácia, kľúčové slová
<i>Výstupy:</i>	všetky vstupy (uložia sa do databázy)
<i>Používatelia:</i>	vyučujúci, externý systém
<i>Opis:</i>	Vloženie slovného aj číselného hodnotenia, zadanie kľúčových slov k danému hodnoteniu.

Pridanie kľúčových slov

<i>Vstupy:</i>	kľúčové slová
<i>Výstupy:</i>	všetky vstupy (uložia sa do databázy)
<i>Používatelia:</i>	vyučujúci, študent
<i>Opis:</i>	Vloženie kľúčových slov k hodnoteniam vyučujúceho a študenta. Kľúčové slová by mali vystihovať podstatu vkladaneho hodnotenia a uľahčujú vyhľadávanie.

Tabuľka III - 1: Opis prípadov použitia súvisiacich so získavaním informácií

Poskytnutie indexovaného zoznamu študentov	
<i>Vstupy:</i>	zoznam študentov určených na zoradenie
<i>Výstupy:</i>	zoznam študentov zoradený podľa vypočítaného indexu, index pre každého študenta
<i>Používatelia:</i>	vyučujúci, externý systém
<i>Opis:</i>	Slúži na zoradenie zoznamu študentov na základe definovaných váh. Váhy sa vzťahujú na predmety, znalosti a zručnosti. Váha predmetu určuje, aká významnosť je priradená známke z daného predmetu. Podobne váha znalosti alebo zručnosti určuje významnosť príslušného hodnotenia. Diagram aktivít platný pre vyučujúceho (prístup cez web rozhranie) opisujúci tento prípad použitia je na obrázku [Obrázok III - 4]. Pre externý systém je výstup dostupný pomocou web služieb ako XML dokument.
Poskytnutie komplexnej informácie o študentovi	
<i>Vstupy:</i>	meno alebo osobné číslo študenta
<i>Výstupy:</i>	osobné údaje, znalosti a zručnosti študenta, certifikáty respektíve ocenenia, hodnotenia, všetky textové informácie zadané študentom, percentuálne vyjadrenie úspešnosti študenta v určitých skupinách predmetov (vypočítané na základe známok študenta)
<i>Používatelia:</i>	študent
<i>Opis:</i>	Poskytuje kompletný profil študenta aj s osobnými údajmi. Sprístupní zobrazenie všetkých informácií, teda aj tých, ktoré študent nezažadoval, ako sú hodnotenia vyučujúcich a podobne. Slúži na informovanie študenta o všetkom, čo je o ňom v systéme uložené. Je prístupný cez web formulár ako aj cez web-services pre externý systém.
Poskytnutie čiastočnej informácie o študentovi	
<i>Vstupy:</i>	meno alebo osobné číslo študenta
<i>Výstupy:</i>	meno alebo osobné číslo študenta, znalosti a zručnosti študenta, certifikáty respektíve ocenenia, hodnotenia, všetky textové informácie zadané študentom, percentuálne vyjadrenie úspešnosti študenta v určitých skupinách predmetov (vypočítané na základe známok študenta) zaradené do percentuálnych tried (napríklad v 10-20% najlepších študentov)

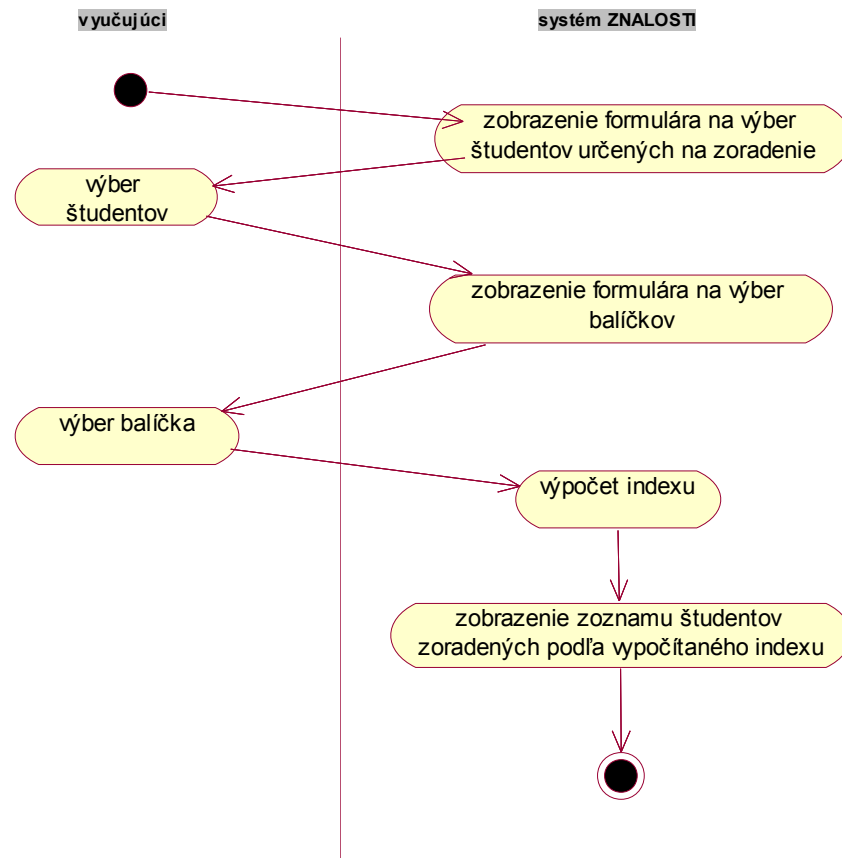


	z programovacích predmetov)
<i>Používatelia:</i>	vyučujúci, externý systém
<i>Opis:</i>	Poskytuje profil študenta, ktorý slúži na informovanie vyučujúceho o znalostiach, zručnostiach, schopnostiach študenta všeobecne bez zamerania sa na konkrétnu oblasť. Je prístupný cez web formulár ako aj cez web-services pre externý systém.
<i>Poskytnutie informácií o predmete</i>	
<i>Vstupy:</i>	identifikátor predmetu, zvolené obdobie
<i>Výstupy:</i>	názov, číslo a anotácia predmetu, garanti, prednášajúci, počet zapísaných študentov [priemerný ako aj pre každý semester zvlášť], percentuálne rozloženie známok (napr. A-10%, B-20%, C-30%,D-20%,E-25%,F-5%) [priemerné ako aj pre každý semester a termín zvlášť]
<i>Používatelia:</i>	vyučujúci, externý systém, študent
<i>Opis:</i>	Poskytne základné a agregované informácie o predmete. Má za úlohu najmä informovať študenta o žiadanosti predmetu a o rozložení známok v jednotlivých rokoch, ako aj o priemernej žiadanosti a priemerných známkach. Funkcia bude študentovi sprístupnená iba po vyplnení profilu študentom, a teda má aj motivačnú funkciu.
<i>Poskytnutie zoznamu predmetov</i>	
<i>Vstupy:</i>	žiadny vstup, kľúčové slová, meno alebo priezvisko učiteľa, počet kreditov, časť názvu predmetu
<i>Výstupy:</i>	názov, číslo a kredity predmetu, meno a priezvisko učiteľa predmetu
<i>Používatelia:</i>	študent, vyučujúci, externý systém
<i>Opis:</i>	Poskytne zoznam predmetov na základe zvolených parametrov. Má úlohu sprístupniť zoznam predmetov buďto len pre informatívne účely alebo ako zoznam používaný pri správe balíčkov. Je prístupný cez web formulár ako aj cez web-services pre externý systém.
<i>Poskytnutie zoznamu znalostí, zručností, certifikátov</i>	
<i>Vstupy:</i>	žiadny vstup, časť názvu znalosti, zručnosti alebo certifikátu, typ úrovne
<i>Výstupy:</i>	názov znalosti, zručnosti alebo certifikátu, typ úrovne
<i>Používatelia:</i>	študent, vyučujúci, externý systém

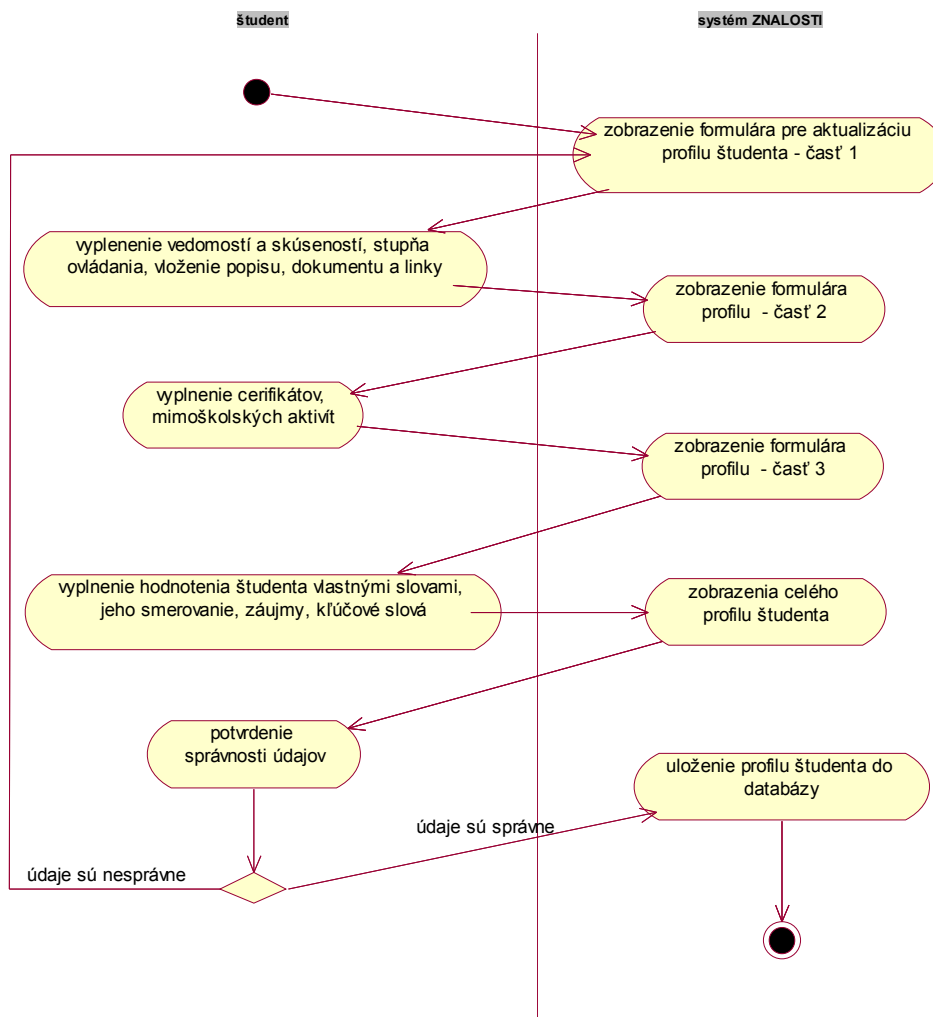


<i>Opis:</i>	Poskytne zoznam znalostí, zručností alebo certifikátov. Má úlohu sprístupniť tento zoznam buďto len pre informatívne účely alebo ako zoznam používaný pri správe balíčkov. Je prístupný cez web formulár ako aj cez web-services pre externý systém.
<i>Poskytnutie známky</i>	
<i>Vstupy:</i>	meno študenta alebo jeho osobné číslo, názov alebo číslo predmetu
<i>Výstupy:</i>	známka
<i>Používatelia:</i>	externý systém
<i>Opis:</i>	Slúži na poskytovanie známok študentov pre externé systémy, ktoré budú mať k tomu príslušné oprávnenie. Takýmito externými systémami budú výhradne fakultné informačné systémy, ktoré môžu využiť systém na získavanie takýchto informácií, keďže sú v systéme uchovávané. Je prístupný cez web-services pre externý systém.
<i>Poskytnutie zoznamu študentov</i>	
<i>Vstupy:</i>	žiadny vstup, časť mena alebo priezviska študenta
<i>Výstupy:</i>	meno a priezvisko študenta, email
<i>Používatelia:</i>	vyučujúci, externý systém
<i>Opis:</i>	Poskytne zoznam študentov na základe zvolených parametrov. Má úlohu sprístupniť tento zoznam buďto len pre informatívne účely alebo ako zoznam používaný pri prípade použitia ‚Poskytnutie indexovaného zoznamu študentov‘. Je prístupný cez web formulár ako aj cez web-services pre externý systém.
<i>Poskytnutie zoznamu funkcií</i>	
<i>Vstupy:</i>	žiadny vstup
<i>Výstupy:</i>	zoznam funkcií s popisom
<i>Používatelia:</i>	externý systém
<i>Opis:</i>	Poskytne zoznam funkcií a ich popis. Slúži pre externý systém, aby mohol využívať funkcie systému cez web-services.

Tabuľka III - 2: Popis prípadov použitia súvisiacich s poskytovaním informácií



Obrázok III - 4: Diagram aktivít pre prípad použitia Poskytnutie zoznamu študentov podľa výsledného indexu



Obrázok III - 5: Vyplnenie profilu

Import údajov	
<i>Vstupy:</i>	súbor obsahujúci dáta zo systému ŠTUDENT
<i>Výstupy:</i>	vstupy pretransformované na internú reprezentáciu (uložia sa do databázy)
<i>Používatelia:</i>	administrátor
<i>Opis:</i>	Umožňuje importovanie údajov zo systému ŠTUDENT pomocou načítania z exportovaného súboru. Prebieha po skončení každého semestra a inicializuje ho administrátor manuálne. Systém ho však notifikuje, ak import posledného semestra ešte neprebehol, alebo ak nastala pri importovaní chyba.
Notifikácia	



<i>Vstupy:</i>	oznámenie o udalosti, pri ktorej sa má notifikácia vykonať
<i>Výstupy:</i>	oznam na web stránke alebo emailová správa
<i>Používatelia:</i>	administrátor, vyučujúci, študent
<i>Opis:</i>	Oznamuje používateľovi výskyt istej udalosti, alebo ho žiada o vykonanie nejakej akcie. Môže byť načasovaná (výskyt udalosti generuje časovač), napríklad pravidelné zaslanie emailu so žiadosťou o aktualizáciu profilu študenta. Rovnako môže byť vyvolaná udalosťou, napríklad oznámenie o neúspešnom importovaní údajov zo systému ŠTUDENT. Spôsob notifikácie (oznam/email) si používateľ môže nastaviť pomocou <i>nastavenia profilu</i>
<i>Nastavenie profilu</i>	
<i>Vstupy:</i>	požadovaný spôsob notifikácie, iné osobné nastavenia
<i>Výstupy:</i>	rovnaké ako vstupy (uložia sa do databázy)
<i>Používatelia:</i>	vyučujúci, študent
<i>Opis:</i>	Umožňuje používateľovi nastaviť osobné preferencie pri používaní systému, napríklad spôsob notifikácie.
<i>Konfigurácia interných nastavení</i>	
<i>Vstupy:</i>	minimálny počet študentov na porovnanie, minimálny počet predmetov vo váhovaní, nastavenie sprístupnenia osobných údajov, definovanie jemnosti percentuálnych tried
<i>Výstupy:</i>	rovnaké ako vstupy (uložia sa do databázy)
<i>Používatelia:</i>	konfigurátor
<i>Opis:</i>	Slúži na nastavenie „minimálneho zahmlenia informácií“. Ide teda o definovanie úrovne hrubosti alebo jemnosti zobrazených informácií, ktoré sú získané na základe známok študentov. Určené hranice nebude môcť vyučujúci prekročiť, a teda sa nebude môcť dozvedieť napríklad usporiadanie študentov podľa známok z konkrétneho predmetu.
<i>Správa balíčkov</i>	
<i>Vstupy:</i>	predmety, znalosti, zručnosti a certifikáty s im prislúchajúcou váhou
<i>Výstupy:</i>	balíček s definovanými váhami (uloží sa do databázy)
<i>Používatelia:</i>	konfigurátor, vyučujúci, externý systém

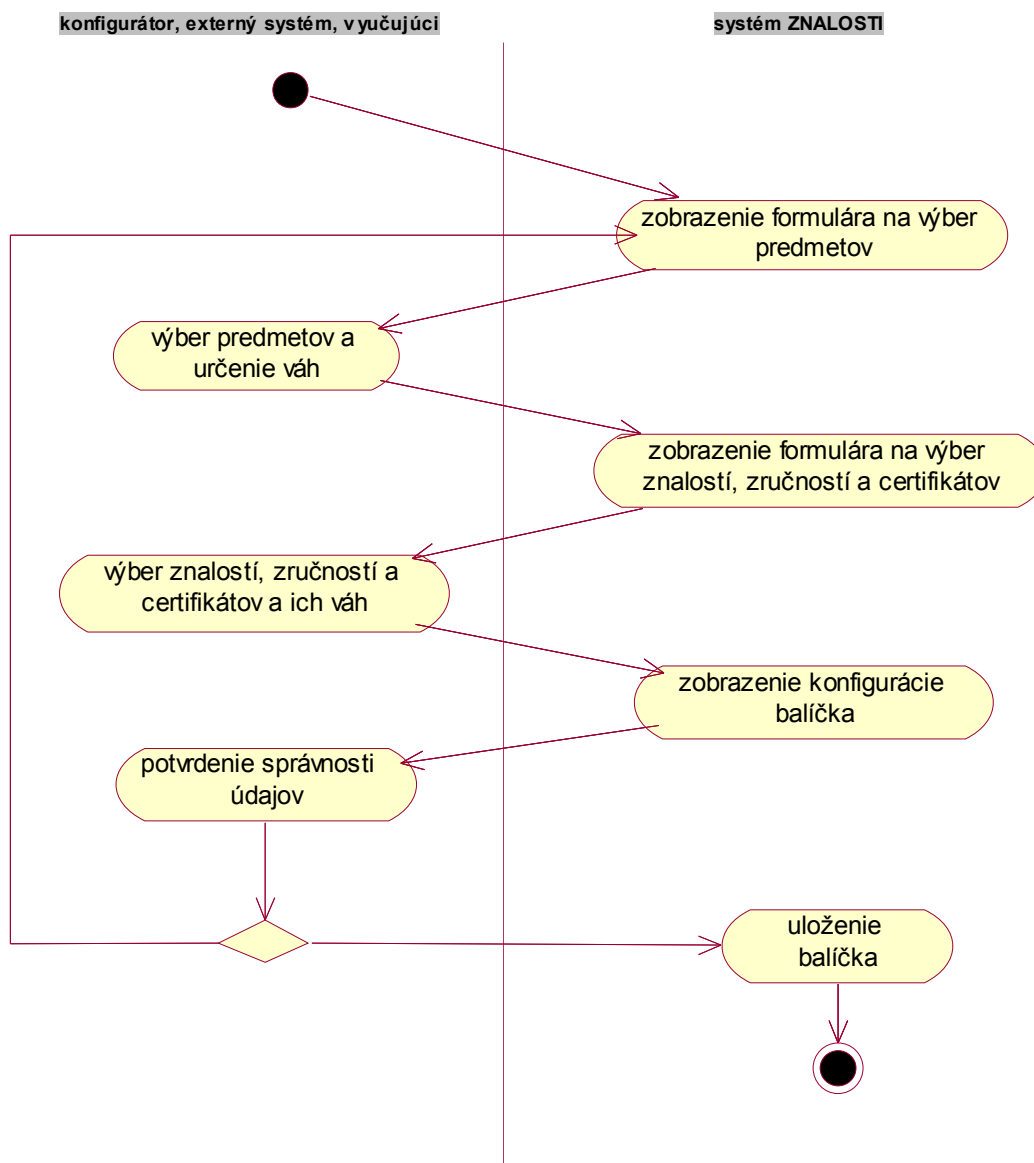


<i>Opis:</i>	<p>Balíček je množina dvojíc (položka, váha) kde položka môže byť predmet, znalosť, zručnosť alebo certifikát. Slúži na definovanie preferencií používateľa pri hľadaní vhodného kandidáta. Správa balíčkov umožňuje vytvárať a meniť balíčky na úrovni konfiguračnej (tieto balíčky budú prístupné každému) a na úrovni používateľskej (balíčky prístupné konkrétnemu používateľovi) [Obrázok III - 6]</p> <p>Takisto slúži na definovanie skupín predmetov konfigurátorom. Pre tieto skupiny sa bude počítať percentuálne vyjadrenie úspešnosti študenta (vypočítané na základe jeho známok) aj s určenou váhou. Príkladom takejto skupiny je skupina <i>Programovacie predmety</i> = {<i>Programovanie v C (váha 5)</i>, <i>Programovacie techniky (váha 3)</i>, <i>Algoritmizácia a programovanie (váha 1)</i>, <i>Stavba operačných systémov (váha 1)</i>}</p>
<i>Správa zoznamu zručností</i>	
<i>Vstupy:</i>	nové zručnosti
<i>Výstupy:</i>	obnovený zoznam zručností (uloženie do databázy)
<i>Používatelia:</i>	konfigurátor
<i>Opis:</i>	Poskytuje možnosť pridávania zručností do zoznamu. V prípade vzniku novej zručnosti bude konfigurátor požiadaný o rozšírenie tohto zoznamu a aj definovanie typu úrovne zručnosti.
<i>Správa zoznamu certifikátov</i>	
<i>Vstupy:</i>	nové certifikáty
<i>Výstupy:</i>	obnovený zoznam certifikátov (uloženie do databázy)
<i>Používatelia:</i>	konfigurátor
<i>Opis:</i>	Slúži na pridávanie nových druhov certifikátov do systému ZNALOSTI. Ak študent získa certifikát, ktorý sa v systéme nenachádza, požiada konfigurátora o začlenenie tohto typu certifikátu do zoznamu.
<i>Správa zoznamu znalostí</i>	
<i>Vstupy:</i>	nové znalosti
<i>Výstupy:</i>	obnovený zoznam znalostí (uloženie do databázy)
<i>Používatelia:</i>	konfigurátor
<i>Opis:</i>	Umožňuje začlenenie nových druhov znalostí do systému. Konfigurátor pridá novú znalosť spolu s definovaním typu úrovne hodnotenia (napr.



	klasifikácia alebo stupeň – aktívne, pasívne)
<i>Správa používateľov</i>	
<i>Vstupy:</i>	meno, priezvisko používateľa, email, [telefón], prihlasovacie meno, heslo, privilégia
<i>Výstupy:</i>	všetky vstupy (uložia sa do databázy)
<i>Používatelia:</i>	administrátor
<i>Opis:</i>	Slúži na pridávanie alebo odoberanie používateľov systému, prípadne zmenu ich atribútov. Administrátor pridáva používateľov systému pričom definuje vektor ich privilégií. Všetci študenti, všetci učitelia a všetci konfigurátori budú mať obvykle rovnaké privilégia vyplývajúce priamo z diagramu prípadov použitia. Externé systémy budú mať naopak väčšinou rôzne privilégia.

Tabuľka III - 3: *Opis prípadov použitia súvisiacich s konfiguráciou*



Obrázok III - 6: Diagram aktivít pre prípad použitia Správa balíčkov

2.4 Vstupy a výstupy

2.4.1 Vstupy

Existuje viacero typov informácií o študentovi, ktoré systém spracováva. Ide o kvalitatívne ako aj kvantitatívne dáta. Je snaha väčšinu informácií popisujúcich kvalitu previesť na kvantifikovateľné (teda strojovo spracovateľné) číselné hodnoty. To však nie je možné vždy. Najčastejšie budú spracovávané informácie obsahovať číselné, ako aj textové hodnotenie a doplnkové informácie.

Základné typy informácií o študentovi sú:

- **známky** – vzťahujú sa buď na konkrétny predmet, alebo ide o VŠP
- **znalosti** – reprezentujú nadobudnuté vedomosti študenta
- **zručnosti** – reprezentujú schopnosti študenta nadobudnuté na základe skúsenosti
- **certifikáty** – osvedčenie o znalosti problematiky
- **hodnotenia** – predstavujú jeho vlastnú mienku o sebe, slovné hodnotenie bakalárskeho a diplomového projektu (zo systému YonBan), slovné hodnotenie pedagógov na konci semestra

Iné vstupy sú:

- **Informácie o predmetoch** – zahŕňajú názov predmetu, anotáciu, garanta, prednášajúceho, počet zapísaných študentov, dosiahnuté známky, vzťahujúce sa na semester.

Príklady spomenutých typov informácií sú uvedené v tabuľke [Tabuľka III - 4: Príklady typov spracovávaných informácií

]. Rovnako je uvedené, či ide o číselné (kvantitatívne) alebo o popisné (kvalitatívne) informácie, teda či môžu byť reprezentované číslom alebo len textom.

Typ informácie	Príklad informácie	
známky	známka z predmetu Počítačové siete celkový doterajší VŠP	číselné
znalosti	znalosť XML znalosť C++	číselné / popisné
certifikáty	certifikát CISCO certifikát z anglického jazyka ocenenie v súťaži ACM	číselné / popisné
zručnosti	ovládanie C++ tvorba WEB prezentácií komunikácia so zákazníkom	číselné / popisné
hodnotenia	životopis študenta hodnotenie vedúceho bakalárskeho projektu správa študenta o zamýšľanej profilácii	popisné (/ číselné)

Tabuľka III - 4: Príklady typov spracovávaných informácií

2.4.2 Výstupy

Je identifikovaných viacero základných druhov výstupov. Ide o výstupy primárne určené pre vyučujúceho alebo študenta. Informácie tvoriace výstup sú uvedené v tabuľke [Tabuľka III - 5: **Základné druhy výstupov**

].

Výstup	Popis	Používateľ
Zoznam študentov podľa výsledného indexu	Je to zoznam vybraných študentov usporiadaných podľa definovaného váhovania kvantifikovateľných údajov. Služi vyučujúcemu na porovnanie vybraných študentov.	Vyučujúci
Komplexná informácia o študentovi	Je to poskytnutie všetkých informácií o študentovi. Príklad takéhoto výstupu je v tabuľke [Tabuľka III - 6: Príklad súhrnnej informácie o študentovi].	Vyučujúci Študent
Informácie o predmete	Charakteristika predmetu, priemerná dosahovaná známka, úspešnosť študentov, obľúbenosť (počet zapísaných študentov), percentuálne rozdelenie známok...	Študent

Tabuľka III - 5: Základné druhy výstupov

Komplexná informácia o študentovi

Prístupná po zadaní mena alebo osobného čísla študenta bude obsahovať napr.:

Meno študenta
Hodnotenie študenta vlastnými slovami , jeho smerovanie
Bc./Ing. štúdium ukončil s vyznamenaním
Téma záverečného projektu
Abstrakt
Hodnotenie vedúceho
Téma diplomového projektu
Abstrakt
Hodnotenie vedúceho
Študijné výsledky v predmetoch tvoriacich určitú skupinu (skupiny môžu byť konfigurovateľné), napríklad: Študijné výsledky v predmetoch zameraných na programovanie (Programovanie v C, Programovacie techniky, Operačné systémy, Stavba operačných systémov, Špecifikačné a opisné jazyky, Strojovo orientované jazyky, ...) Študijné výsledky v predmetoch zameraných na vývoj informačných systémov (Princípy softvérového inžinierstva, Databázové systémy 1)
vyjadrenie v tvare „študent sa nachádza medzi prvých 10 % najúspešnejších“
Súťaž: Programátorská súťaž ACM, regionálne kolo (rok, umiestnenie), CSIDC, ... + prípadné odkazy na stránky venované súťaži



Ocenenia: pochvalné uznanie dekana za výborné študijné výsledky, záverečný projekt a pod.
Certifikáty: Cisco Networking Academy a pod.
Mimoškolské aktivity: zamestnanie, účasť na projektoch ... (vyplňa študent)
Zručnosti - ovládané technológie spolu so stupňom ich ovládania + odkazy na dosiahnuté výsledky
Záujmy študenta – tieto informácie zadáva študent pri vyplňaní svojho profilu
Životopis

Tabuľka III - 6: Príklad súhrnnej informácie o študentovi

Zoznam študentov podľa výsledného indexu

Ďalším možným výstupom je zoznam študentov podľa výsledného indexu. Vyučujúci si vo formulári môže zvoliť rôzne kritériá na porovnanie viacerých študentov. K dispozícii sú prospech, znalosť a zručnosť pričom si môže zvoliť stupeň dôležitosti (váhy) jednotlivých kategórií.

- Formulár pozostáva z viacerých častí – sekcií – Prospech, Znalosť a Zručnosť. V sekcii Prospech si môže vyučujúci zvoliť predmet, príp. okruh predmetov, podľa ktorých sa budú študenti porovnávať. Výslední študenti môžu byť zoradení podľa výsledkov, ale aj náhodne (záleží od konfigurácie).
- Sekcia znalosť – ponúkne zoznam znalostí, ktoré systém uchováva a používateľ si tiež môže vybrať úroveň znalosti.
- Sekcia zručnosť – opäť ponúkne zoznam zručností, úroveň, prístupnosť referencie – odkazu, prílohy
- Sekcia hodnotenia – vyučujúci si môže zvoliť požadované klasifikačné hodnotenie.

Informácie o predmete

Pre študentov budú prístupné informácie o predmetoch. Zobrazovať sa budú informácie o predmete – charakteristika predmetu, priemerná dosahovaná známka, úspešnosť študentov, obľúbenosť (počet zapísaných študentov), prípadne percentuálne rozdelenie známok.

Tento výstup pre študenta bude prístupný až po vyplnení profilu, čo je pozitívna motivácia pre napĺňanie systému ZNALOSTI.

V prípade, že študent súhlasí so zverejnením, informácie o študentovi budú zverejnené na internete pre potreby firiem - budúcich potenciálnych zamestnávateľov.

2.4.3 Proces naplňania dátami zo systému ŠTUDENT

Po skončení semestra, pracovník PGO dostane notifikačný mail, aby vygeneroval obsah databázy "Študenta", potom ho musí poslať administrátorovi systému, ktorý ho importuje.

2.4.4 Notifikácia neaktuálnosti dát

Ak sú dáta v systéme neaktuálne, t.j. nastala chyba v procese naplňania dátami zo systému "ŠTUDENT", tak sa na stránke systému objaví notifikácia pre používateľa o neaktuálnosti.

2.4.5 Vonkajšie rozhranie

Systém bude s okolím komunikovať pomocou troch typov rozhraní:

- Web-formulár – používa ho študent a vyučujúci
- Web-services – akákoľvek komunikácia s ostatnými systémami
- Import údajov – získavanie údajov zo systému ŠTUDENT

Webová služba bude dostupná nepretržite prostredníctvom internetu.

3 Ďalšie požiadavky

3.1 Bezpečnosť a ochrana informácií

Vzhľadom na to, že systém spracováva a uchováva údaje chránené zákonom č. [363/2005 Z.z.](#) o ochrane osobných údajov v znení neskorších predpisov, musia byť tieto chránené pred náhodným ako aj nezákonným poškodením a zničením, náhodnou stratou, zmenou, nedovoleným prístupom a sprístupnením ako aj pred akýmkoľvek inými neprípustnými formami spracúvania.

Systém musí preto obsahovať nasledovné štandardné spôsoby ochrany:

- Autentifikácia
- Autorizácia

Používateľ, ktorý chce pracovať so systémom, sa musí najskôr prihlásiť pomocou prihlasovacieho formulára. Zadaním a potvrdením mena a hesla sa spustí proces autentifikácie. Zadané údaje sa porovnávajú s údajmi, ktoré sú uložené v databáze. Pokiaľ sa nájde zhoda, používateľovi je umožnené prihlásenie sa do systému. Meno a heslo sú pri tomto procese posielané v šifrovanej podobe.

Každý používateľ je zaradený do jednej alebo viacerých skupín definujúcich jeho práva na používanie jednotlivých funkcií systému. Na základe členstva v skupinách sa mu sprístupnia dané funkcie.

Ochrana osobných údajov musí byť zabezpečená aj v prípade autorizovaných používateľov. To znamená, že hoci systém takéto údaje spracováva, neposkytuje ich priamo svojim používateľom. Výstup systému tvoria znalosti vyššej úrovne, ktoré už nie sú identifikovateľnými osobnými údajmi.

Keďže zákony sa neustále upravujú a aktualizujú, je vhodné, aby bol systém konfigurovateľný z hľadiska poskytnutia informácií používateľom. Teda aby bola nastaviteľná minimálna úroveň zahmlenia informácií, ktoré sú osobnými údajmi.

Túto konfiguráciu bude vykonávať autorizovaná osoba, ktorá nemá prístup k výstupom systému, ktorých formát konfiguruje.

Na import surových dát zo systému ŠTUDENT do nášho systému sa vytvorí proces, v ktorom do styku s ostrými dátami prídu iba poverené osoby, t.j. pracovník PGO a administrátor systému.

3.2 Požiadavky na softvérové vybavenie

Náš systém je určený do prostredia vysokej školy, teda kladie sa dôraz aj na cenu výsledného produktu. Za týmto účelom bude informačný systém založený na voľne dostupných serverovo orientovaných programoch. Výhodou navrhovaného riešenia sú minimálne nároky na klientský počítač.

3.3 Požiadavky na výkonnosť

Systém musí byť koncipovaný minimálne pre:

- 10 000 študentov
- 500 pedagógov

Výkon musí byť dostatočný na obsluhu ¼ všetkých používateľov naraz, pretože sa predpokladá aj náborové zaťaženie systému.

Maximálna konfigurácia stroja, na ktorom bude systém bežať je vzhľadom k možnostiam fakulty obmedzená zhora nasledovne:

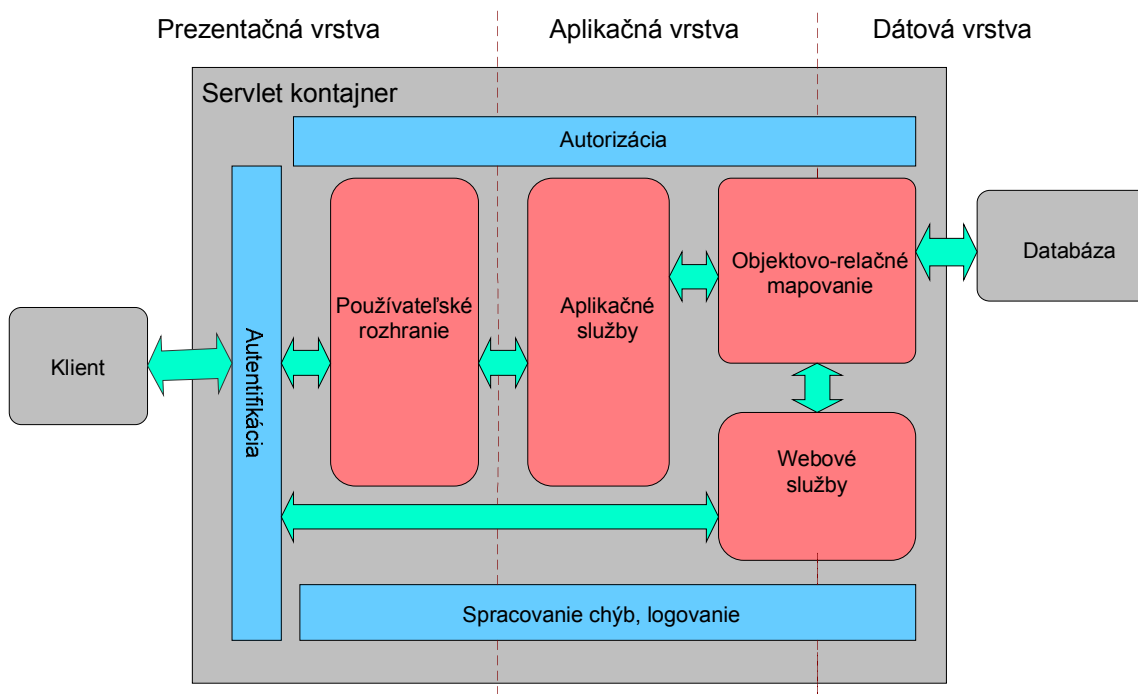
- Pentium III 1GHz
- 256 MB RAM
- 10GB diskového priestoru.

IV HRUBÝ NÁVRH

Táto kapitola opisuje hrubý návrh systému a je rozdelená na dve časti. V prvej je uvedená navrhovaná architektúra systému a v druhej je opísaný dátový model systému.

1 Architektúra systému

Navrhnutá architektúra vychádza zo špecifikácie požiadaviek. Rozhodli sme sa pre trojvrstvovú architektúru [Obrázok IV - 1], v ktorej jednotlivé vrstvy medzi sebou navzájom komunikujú a vymieňajú si potrebné údaje. Dátové toky sú na obrázku znázornené zelenými šípkami a ich význam bude postupne opísaný ďalej v texte. Systém bude využívať služby aplikačného servera (Servlet kontajner).



Obrázok IV - 1: Architektúra systému

1.1 Prezentačná vrstva

Prezentačná vrstva zabezpečuje priamy styk klienta so systémom. Klient môže byť fyzická osoba (študent, vyučujúci, administrátor, konfigurátor) alebo externý systém (systémy Študent,

YONBAN). Úlohou tejto vrstvy je taktiež autentifikovať klienta. Prístup k službám, ktoré náš systém poskytuje budú mať iba autentifikovaní klienti.

1.2 Používateľské rozhranie

Klient po úspešnej autentifikácii môže využívať služby nášho systému pomocou vizuálneho používateľského rozhrania. Toto rozhranie môže byť realizované buď pomocou webových stránok (JSP stránky) alebo pomocou aplikácie s grafickým rozhraním. V prípade použitia webových stránok je na strane klienta potrebný webový prehliadač a komunikácia so systémom prebieha pomocou bezpečného protokolu HTTPS.

V prípade, že klient je externý systém, tak tento využíva služby nášho systému pomocou webových služieb.

1.3 Aplikačná vrstva

Súčasťou aplikačnej vrstvy sú algoritmy, ktoré realizujú logiku systému. Táto vrstva zároveň predstavuje rozhranie pre prezentačnú vrstvu, ktorej poskytuje svoje služby. Do tejto vrstvy patria aj webové služby.

1.4 Aplikačné služby

Tieto služby definujú aplikačnú funkcionality a sú priamo volané z prezentačnej vrstvy, t.j. predstavujú rozhranie na aplikačnú logiku. Realizujú požiadavky, ktoré boli definované v časti III 2 - Špecifické požiadavky (napr. poskytnutie zoznamu študentov, vkladanie údajov, konfigurácia systému, administrácia systému,...). Podporujú získavanie informácií o dostupnosti danej služby, od čoho sa odvíja správanie prezentačnej vrstvy.

Pomocou objektovo-relačného mapovania komunikujú s dátovou vrstvou.

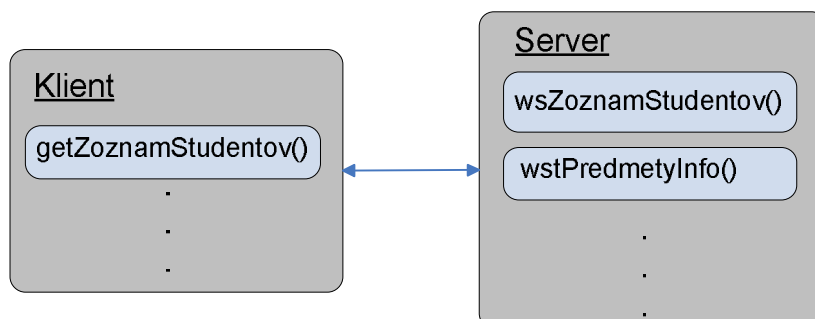
1.5 Objektovo-relačné mapovanie

Objektovo relačné mapovanie mapuje dátové objekty priamo na dáta v tabuľkách databázy. Toto zabezpečuje framework pre mapovanie dát. Pre takéto riešenie sme sa rozhodli hlavne kvôli podstatnému zjednodušeniu práce s databázou. Výhodou tohto frameworku je priame mapovanie dátových objektov na SQL príkazy, pričom framework zabezpečí typovú správnosť jednotlivých atribútov tried.

1.6 Webové služby

Ďalší spôsob komunikácie klienta so systémom je pomocou webových služieb [Obrázok IV - 2]. Tieto pozostávajú z dvoch častí. Serverová časť slúži ako poskytovateľ webových služieb pre

klienta. Systém bude poskytovať služby, ktoré sú uvedené v špecifikácii požiadaviek (napr. zoznam študentov, informácie o predmetoch, ...). Klientovi budú poskytnuté iba služby, na ktoré má oprávnenie.



Obrázok IV - 2: Webové služby

1.7 Dátová vrstva

Dátová vrstva zaisťuje hlavne načítanie dát z databázy a ich uloženie naspäť. V tejto vrstve sa tiež vyskytujú úložné procedúry, ktoré budeme využívať pri prístupe k dátam. Úložné procedúry využijeme hlavne kvôli ich výhodám (izolujú aplikáciu od zmien v databáze, zabezpečujú vyššiu výkonnosť pri zložitejších dotazoch). Databázová vrstva komunikuje s aplikačnou vrstvou a jej službami pomocou objektovo-relačného mapovania.

1.8 Bezpečnosť systému

Ako vidno z obrázku IV-1, so systémom môže pracovať iba autentifikovaný používateľ. V prípade použitia webového prehliadača ako používateľského rozhrania bude prístup k nášmu systému realizovaný pomocou bezpečného protokolu HTTPS.

Autorizácia zabezpečí, že rôzni používatelia systému majú prístup iba k tým službám, na ktoré majú oprávnenie.

Systém umožňuje logovanie rôznych aktivít, ktoré sa v ňom odohrávajú (úspešné/neúspešné prihlásenia, chyby systému,...).

2 Model údajov

V tejto kapitole je popísaná koncepcia návrhu modelu uchovávaných údajov na logickej a fyzickej úrovni. Model údajov umožňuje identifikovanie údajov, ktoré systém prijíma, s ktorými systém pracuje a ktoré produkuje. Zároveň vyjadruje vzťahy medzi identifikovanými údajmi a ich časťami a určuje obsah údajov.

Kapitola pozostáva zo štyroch častí. V kapitole 2.1 je uvedený pohľad na uchovávané údaje ako celok, identifikácia entít a vzťahy medzi nimi. Tieto entity sú podrobnejšie popísané v kapitole 2.2. Kapitola 2.3 predstavuje fyzický model údajov popísaných v kapitolách 2.1 a 2.2. V tejto kapitole sa taktiež nachádza opis transformácie logickej úrovne na fyzickú.

2.1 Logický model údajov

Na obrázku IV - 3 je uvedený diagram zobrazujúci logický model údajov. Mnohé entity, ktoré boli v systéme identifikované vyplynuli priamo zo zadania. Boli to entity pomenúvajúce základné stavebné prvky, ako je *student*, *učiteľ*, *používateľ*, *predmet*, *známka*, *zručnosť*, *znalosť*, *certifikát*, *konfigurácia*. Po hlbšej analýze boli identifikované entity *dôkaz*, *váha*, *ovláda*, *úroveň*, *typ úrovne*, *balíček*, *hodnotenie*, *student o sebe*, *klúčové slovo*.

Medzi entitami *známka* a *student* je vzťah *obsahuje*, ktorý vyjadruje že každá entita *známka* musí obsahovať *studenta*, ktorý danú známku dostal. Vzťah tiež hovorí, že *student* môže mať viacero *známok*.

Medzi entitami *známka* a *predmet* existuje vzťah *obsahuje*, čo hovorí podobne ako v predošlom prípade, že entita *známka* musí obsahovať *predmet*. Z konkrétneho *predmetu* naopak existuje viacero *známok*.

Vzťah *učí* medzi entitami *učiteľ* a *predmet* vyjadruje, že *učiteľ* môže učiť viacero *predmetov* a súčasne *predmet* môže byť vyučovaný viacerými *učiteľmi*.

Vzťah *vytvára* medzi entitami *učiteľ* a *hodnotenie* hovorí, že *učiteľ* môže vytvoriť *hodnotenie*, avšak *hodnotenie* je vytvorené práve jedným *učiteľom*.

Medzi entitami *student* a *hodnotenie* existuje vzťah *hodnotí*, ktorý hovorí, že pre *studenta* existuje viacero *hodnotení*, ale *hodnotenie* sa vzťahuje na konkrétneho *studenta*.

Medzi entitami *hodnotenie* a *klúčové slovo* je vzťah *obsahuje*, ktorý vyjadruje, že *hodnotenie* obsahuje *klúčové slová* a súčasne *klúčové slová* sa môžu nachádzať vo viacerých *hodnoteniach*.

Vzťah *píše* medzi entitami *študent* a *študent o sebe* hovorí, že *študent* o sebe môže napísať viacero článkov (*študent o sebe*), pričom článok (*študent o sebe*) sa vždy vzťahuje na konkrétneho študenta.

Vzťah *obsahuje* medzi entitami *študent o sebe* a *klúčové slovo* hovorí, že článok o študentovi (*študent o sebe*) obsahuje *klúčové slová*. *Ključové slová* sa tiež môžu nachádzať vo viacerých článkoch (*študent o sebe*).

Vzťah *obsahuje* medzi entitami *predmet* a *klúčové slovo* vyjadruje, že predmet môže obsahovať *klúčové slová* a naopak *klúčové slová* môžu byť obsiahnuté vo viacerých *predmetoch*.

Vzťah *obsahuje* medzi entitami *študent* a *ovláda* hovorí, že každá entita *ovláda* obsahuje jedného *študenta*. *Študentovi* môže patriť viacero entít *ovláda*.

Vzťah *má* medzi entitami *študent* a *dôkaz* hovorí, že pre každú entitu *dôkaz* existuje jeden *študent*. Pre *študenta* môže existovať viacero *dôkazov*.

Medzi entitami *certifikát* a *dôkaz* je vzťah *dokazuje*, ktorý hovorí že *certifikát* môže byť dokázaný viacerými *dôkazmi*, ale *dôkaz* sa vždy vzťahuje na konkrétny *certifikát*.

Medzi entitami *ovláda* a *zručnosť* je vzťah *obsahuje*, ktorý hovorí o tom, že pre každú entitu *ovláda* existuje jedna entita *zručnosť* a naopak pre jednu entitu *zručnosť* existuje viacero entít *ovláda*.

Medzi entitami *ovláda* a *znalosť* je vzťah *obsahuje*, ktorý hovorí o tom, že pre každú entitu *ovláda* existuje jedna entita *znalosť* a naopak pre jednu entitu *znalosť* existuje viacero entít *ovláda*.

Medzi entitami *ovláda* a *úroveň* je vzťah *obsahuje*, ktorý hovorí o tom, že pre každú entitu *ovláda* existuje jedna entita *úroveň* a naopak pre jednu entitu *úroveň* existuje viacero entít *ovláda*.

Vzťah *pozostáva* medzi entitami *úroveň* a *typ úrovne* vyjadruje, že entita *typ úrovne* pozostáva z viacerých entít *úroveň*, ale *úroveň* pozostáva z jedného *typu úrovne*.

Vzťah *má* medzi entitami *zručnosť* a *typ úrovne* vyjadruje, že každá *zručnosť* má určitý *typ úrovne*, pričom *typ úrovne* môže prislúchať viacerým entitám *zručnosť*.

Vzťah *má* medzi entitami *znalosť* a *typ úrovne* vyjadruje, že každá *znalosť* má určitý *typ úrovne*, pričom *typ úrovne* môže prislúchať viacerým entitám *znalosť*.

Medzi entitami *používateľ* a *balíček* je vzťah *vlastní*, čo vyjadruje, že *používateľ* môže vlastniť viacero *balíčkov* ale konkrétny *balíček* môže byť vlastnený nanajvýš jedným *používateľom*.

Medzi entitami *balíček* a *váha* je vzťah *pozostáva* vyjadrujúci, že *balíček* pozostáva z niekoľkých *váh*, ale *váha* patrí konkrétnemu *balíčku*.



Vzťah *má* medzi entitami *váha* a *predmet* vyjadruje, že pre entitu *váha* existuje práve jeden predmet, ale *predmet* môže existovať vo viacerých *váhach*.

Vzťah *má* medzi entitami *váha* a *certifikát* vyjadruje, že pre entitu *váha* existuje práve jeden *certifikát*, ale *certifikát* môže existovať vo viacerých *váhach*.

Vzťah *má* medzi entitami *váha* a *zručnosť* vyjadruje, že pre entitu *váha* existuje práve jedna *zručnosť*, ale *zručnosť* môže existovať vo viacerých *váhach*.

Vzťah *má* medzi entitami *váha* a *znalosť* vyjadruje, že pre entitu *váha* existuje práve jedna *znalosť*, ale *znalosť* môže existovať vo viacerých *váhach*.



2.2 Entity logického modelu údajov

Študent

Táto entita združuje základné informácie o študentovi, ako je meno, priezvisko, e-mail a podobne.

Učiteľ

Táto entita podobne ako entita študent združuje základné informácie o učiteľovi.

Používateľ

Ide o používateľa systému. Pre každého používateľa sú definované prístupové práva, prihlasovacie meno, heslo a podobne. Používateľom môže byť študent, učiteľ, konfigurátor, administrátor, alebo externý systém.

Predmet

Táto entita združuje informácie o predmete.

Známka

Entita obsahuje informáciu o známke konkrétneho študenta z konkrétneho predmetu.

Zručnosť

Definuje typy zručností študenta, ktoré sú v systéme uchovávané.

Znalosť

Definuje typy znalostí študenta, ktoré sú v systéme uchovávané.

Certifikát

Definuje typy certifikátov študenta, ktoré sú v systéme uchovávané.

Konfigurácia

Táto entita uchováva konfiguráciu systému.

Dôkaz

Entita obsahuje linku (URI), ktorá obsahuje dôkaz, že študent nadobudol certifikát.

Váha

Entita váha určuje akou váhou sa bude podieľať na výslednom hodnotení študenta známka z predmetu, certifikátu, znalosti alebo zručnosti.

Ovláda

Entita ovláda združuje informácie o znalosti alebo zručnosti, ktorú študent ovláda, spolu s úrovňou ovládania tejto znalosti alebo zručnosti.

Úroveň

Entita uchováva názvy úrovni (napr. výborne, dostatočne, nedostatočne), ktoré prislúchajú jednotlivým typom úrovni.

Typ úrovne

Typ úrovne uchováva názvy typov úrovni. Môže ísť napríklad o typ úrovne jazyk (ten môže mať potom úrovne napr. plynule, písmom, materinský), binárna úroveň (áno, nie), alebo typ známka (1, 2, 3, 4, 5) a podobne.

Balíček

Balíček združuje prednastavené hodnoty o váhach jednotlivých entít, ktoré má zmysel váhovať (známka z predmetu, znalosť, zručnosť, certifikát).

Hodnotenie

Entita uchováva hodnotenie študenta od učiteľa.

Študent o sebe

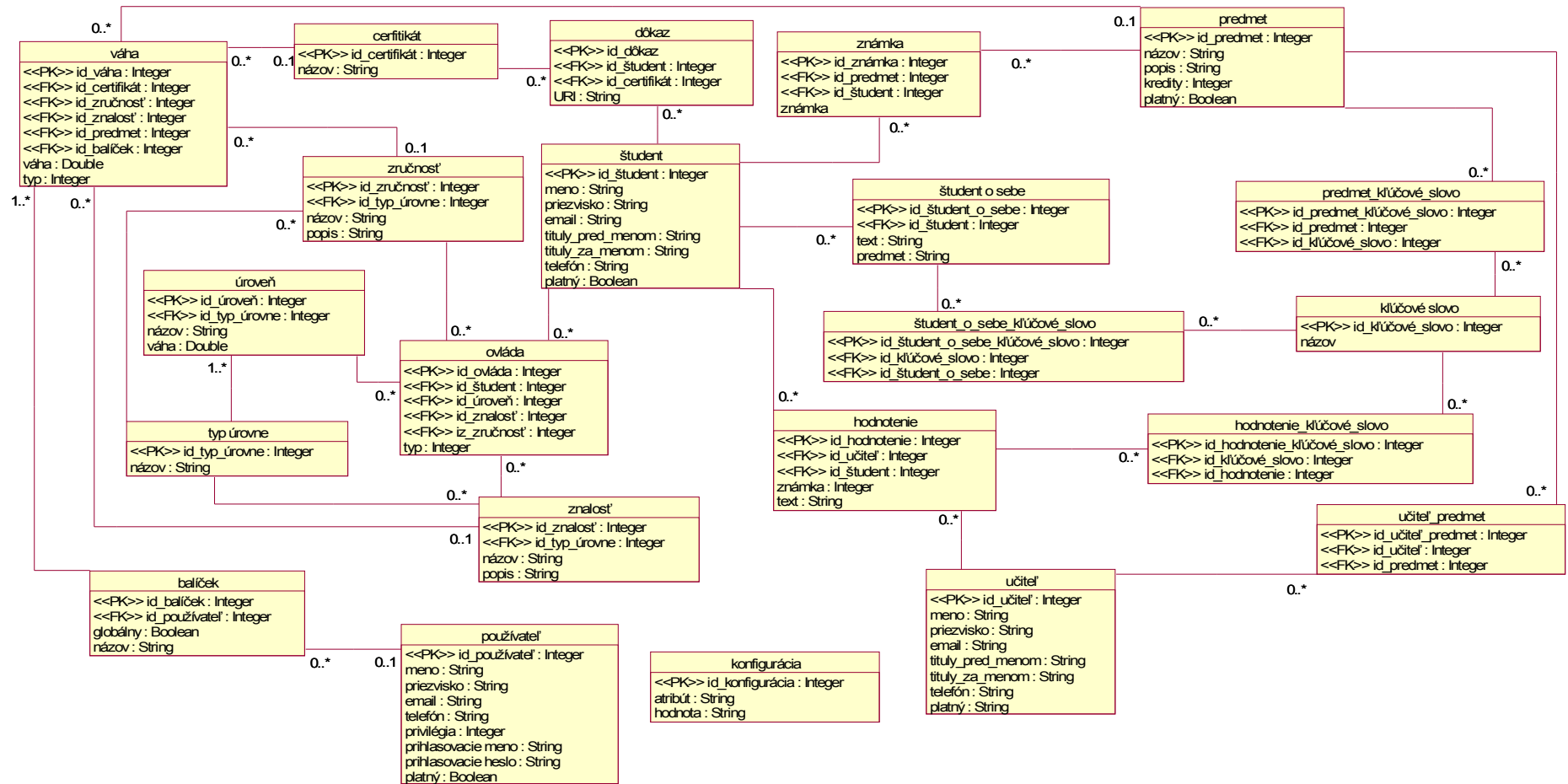
Uchováva študentovo hodnotenie samého seba. Môže ísť napríklad o životopis, popis pracovných skúseností a podobne.

Kľúčové slovo

Vo viacerých prípadoch systém presnejšie definuje text pomocou kľúčových slov, táto entita preto uchováva zoznam kľúčových slov.

2.3 Fyzický model údajov

Na Obrázok IV - 4 sa nachádza diagram zobrazujúci fyzický model údajov. Ide v podstate o priamočiaru úpravu logického modelu. Ako je možné pozorovať z dole uvedeného obrázku, k atribútom jednotlivých entít pribudli ďalšie atribúty. Ide o atribúty reprezentujúce primárne, respektíve cudzie kľúče jednotlivých entít. Vzťahy *M ku N* boli dekomponované a sú reprezentované väzobnými entitami. Taktiež boli definované typy atribútov jednotlivých entít.



Obrázok IV - 4: Fyzický model údajov