



Báza znalostí a zručností študentov

Vypracoval: Tím č. 11
Vedúci tímu: Ing. Ivan Kapustík
Predmet: Tvorba informačného systému v tíme I.
Študijný odbor: Informačné systémy
Semester: zimný
Školský rok: 2005/2006
Kontakt: _elf_@googlegroups.com

Bc. Slavomír Červeň
Bc. Andrej Fenik
Bc. Martin Kováčik
Bc. Juraj Malečka
Bc. Marián Miština
Bc. Martina Práznovská
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

Použitá literatúra

PRÍLOHY

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

DOKUMENTÁCIA K RIADENIU PROJEKTU

ZIMNÝ SEMESTER

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

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 PREHĽAD 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 napĺňania 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 OBJEKTOVO-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

V PROTOTYP

1 CIELE PROTOTYPOVANIA	1
1.1 SOFTVÉROVÝ PROTOTYP	1
1.2 IMPLEMENTAČNÉ PROSTREDIE	2
1.2.1 Databáza	2
1.2.2 Web-services	2
1.2.3 Parser	2
2 TECHNICKÁ DOKUMENTÁCIA.....	3
2.1 DATABÁZA	3
2.1.1 Tabuľky a relácie	3
2.1.2 Formulár pre študenta.....	6
2.1.3 Formulár výberu študentov a „balíčka“	6
2.1.4 Formulár na konfiguráciu „balíčkov“	7
2.1.5 Formulár na konfiguráciu skupín predmetov.....	7
2.1.6 Funkcie na výpočet výsledného indexu na základe „balíčkov“	8
2.1.7 Prezentácia výsledkov výpočtu výsledného indexu na základe „balíčkov“	10
2.2 WEB-SERVICES	10
2.3 PARSER	11
3 POUŽÍVATEĽSKÁ PRÍRUČKA	12
3.1 DATABÁZA	12
3.1.1 Inštalácia a spustenie	12
3.1.2 Návod na používanie	13
3.2 WEB-SERVICES	19
3.2.1 Inštalácia a spustenie	19
3.2.2 Návod na používanie	20
3.3 PARSER	21
3.3.1 Inštalácia a spustenie	21
3.3.2 Návod na používanie	22
3.3.3 Riešenie prípadných problémov	24
4 ZHODNOTENIE PROTOTYPU	25

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 zužitkovanie 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 "preukazovanie 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, sťažie, 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 skĺbiť 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 študentových 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 zaradiť 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 štúdia 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, adries 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 svojim 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.

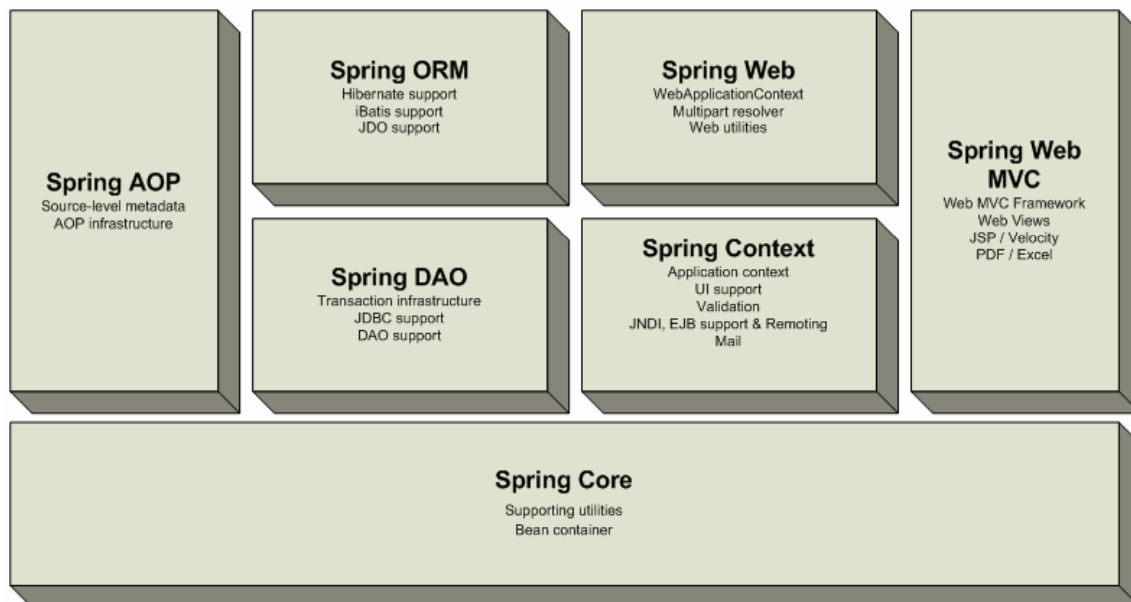
3.1.4 JWSDP

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 zameril 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 pridelovaní 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.

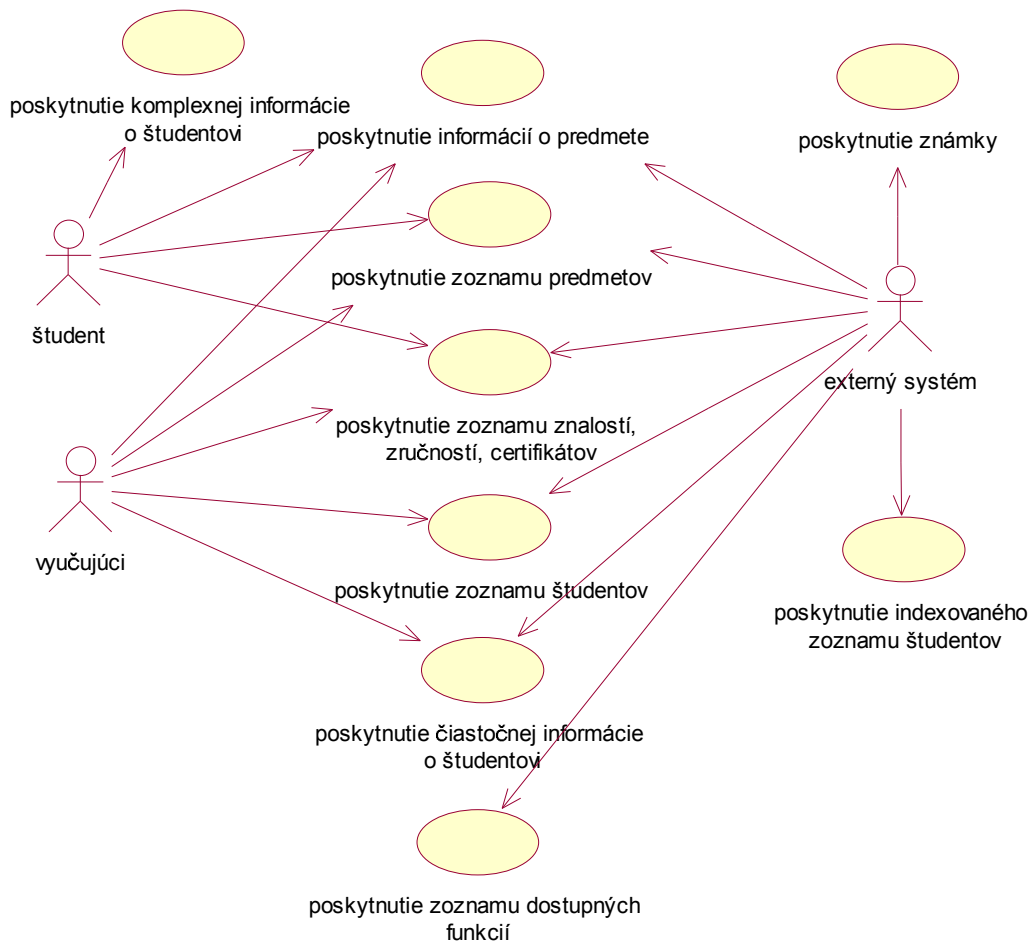
1 Všeobecný opis systému

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).

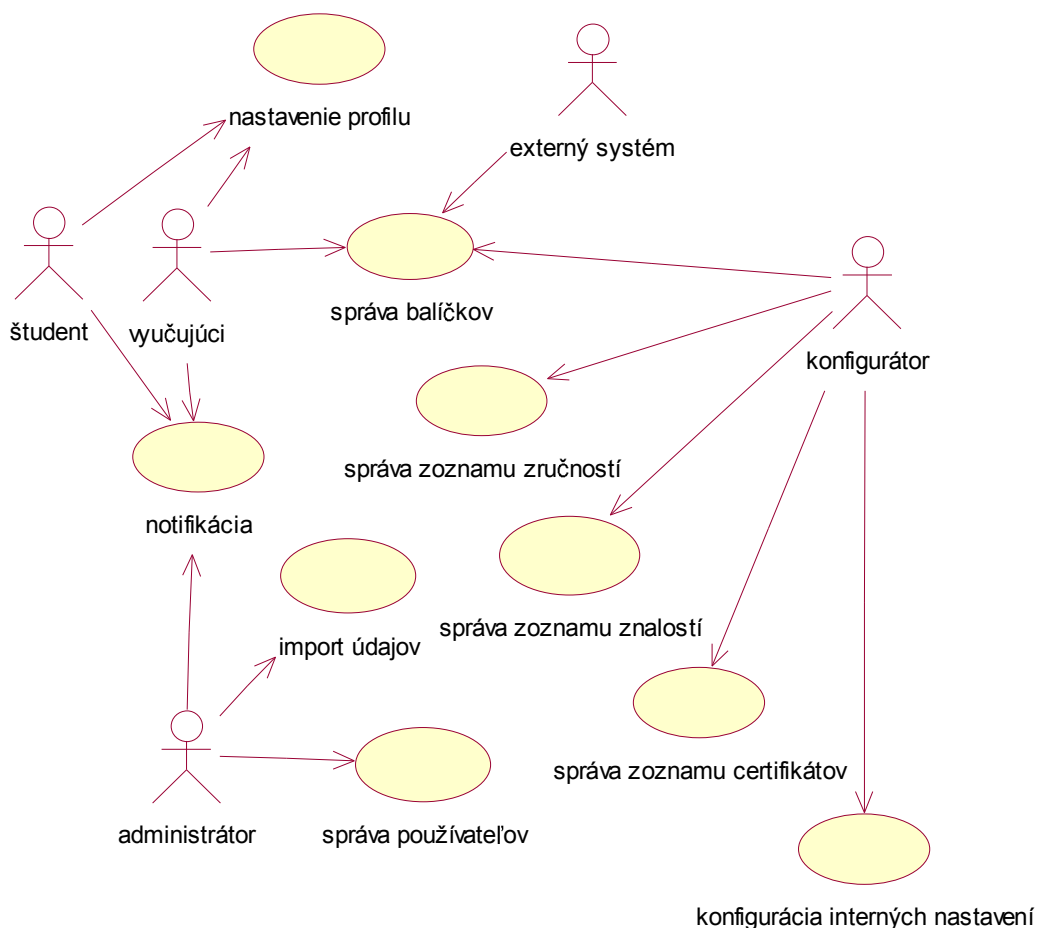
Používateľ systému (vyučujúci alebo externý systém) si bude môcť určiť kritériá a priority, podľa ktorých si zoradí vybraných študentov – vhodných kandidátov na projekt a pod. Vyučujúcim budú teda pomáhať informácie o doterajších skúsenostiach, schopnostiach a záujmoch študenta. Systém však nebude poskytovať 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

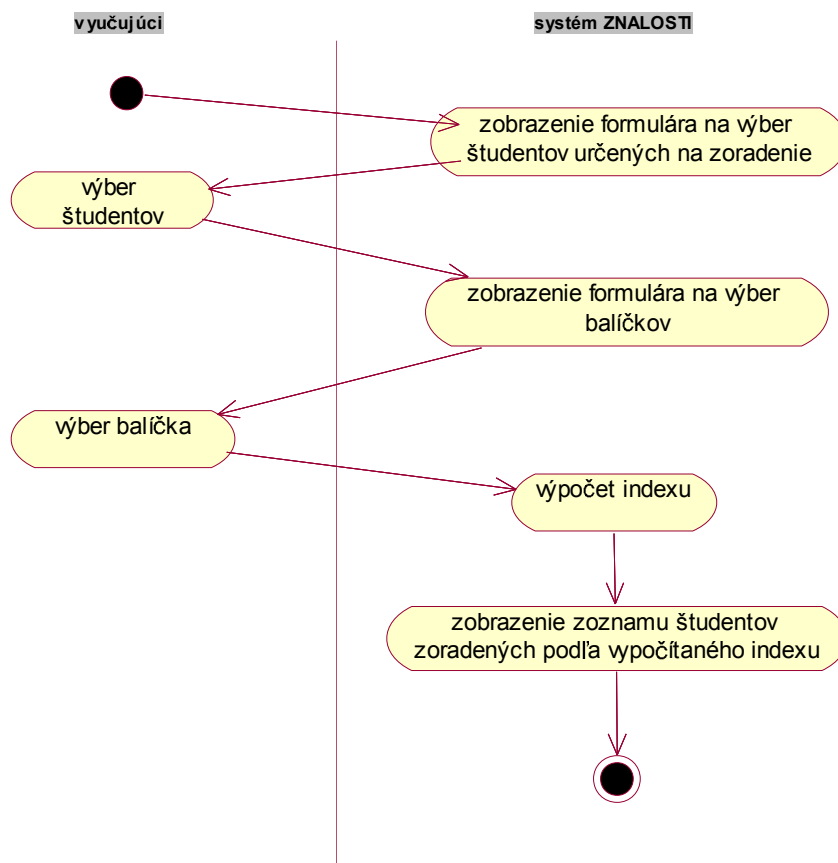
Poskytnutie indexovaného zoznamu študentov	
<i>Vstupy:</i>	zoznam študentov určených na zoradenie
<i>Výstupy:</i>	zoznam študentov zoradený podľa vypočítaného indexu, index pre každého študenta
<i>Používatelia:</i>	vyučujúci, externý systém
<i>Opis:</i>	Slúži na zoradenie zoznamu študentov na základe 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)



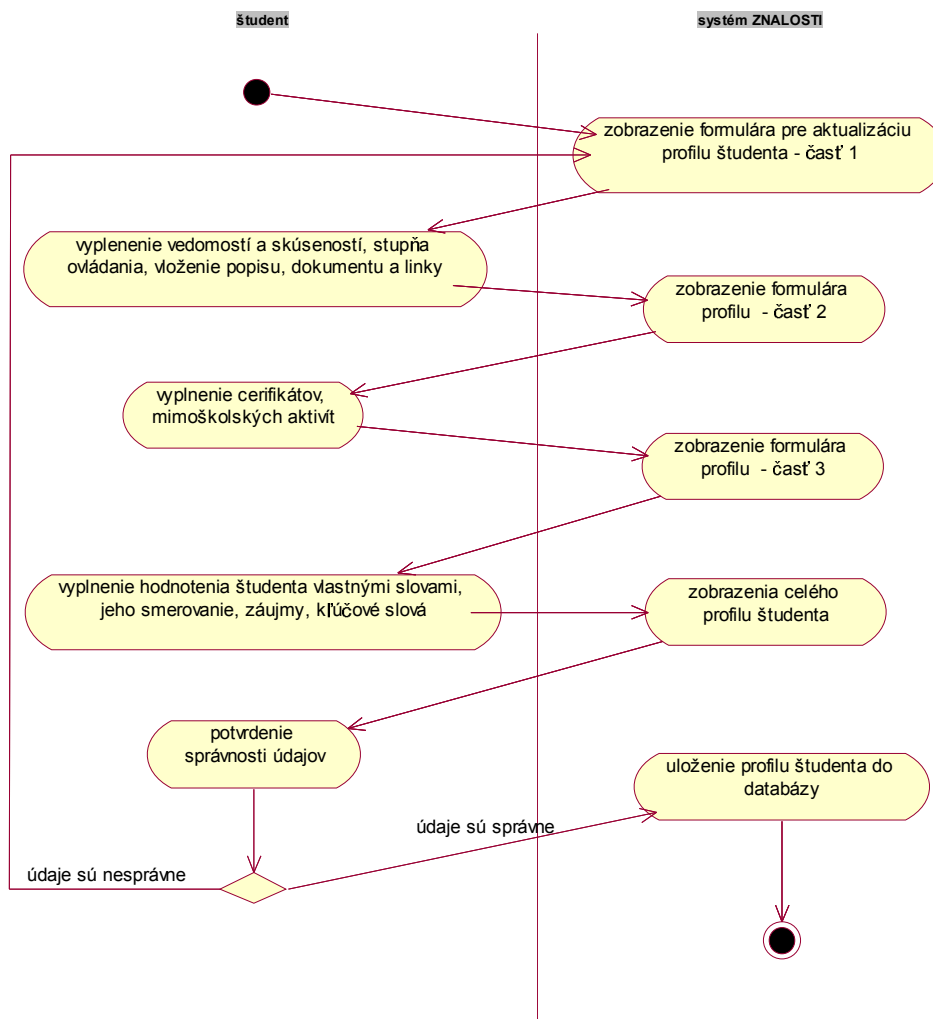
	z programovacích predmetov)
<i>Používatelia:</i>	vyučujúci, externý systém
<i>Opis:</i>	Poskytuje profil študenta, ktorý slúži na informovanie vyučujúceho o znalostiach, zručnostiach, schopnostiach študenta všeobecne bez zamerania sa na konkrétnu oblasť. Je prístupný cez web formulár ako aj cez web-services 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	



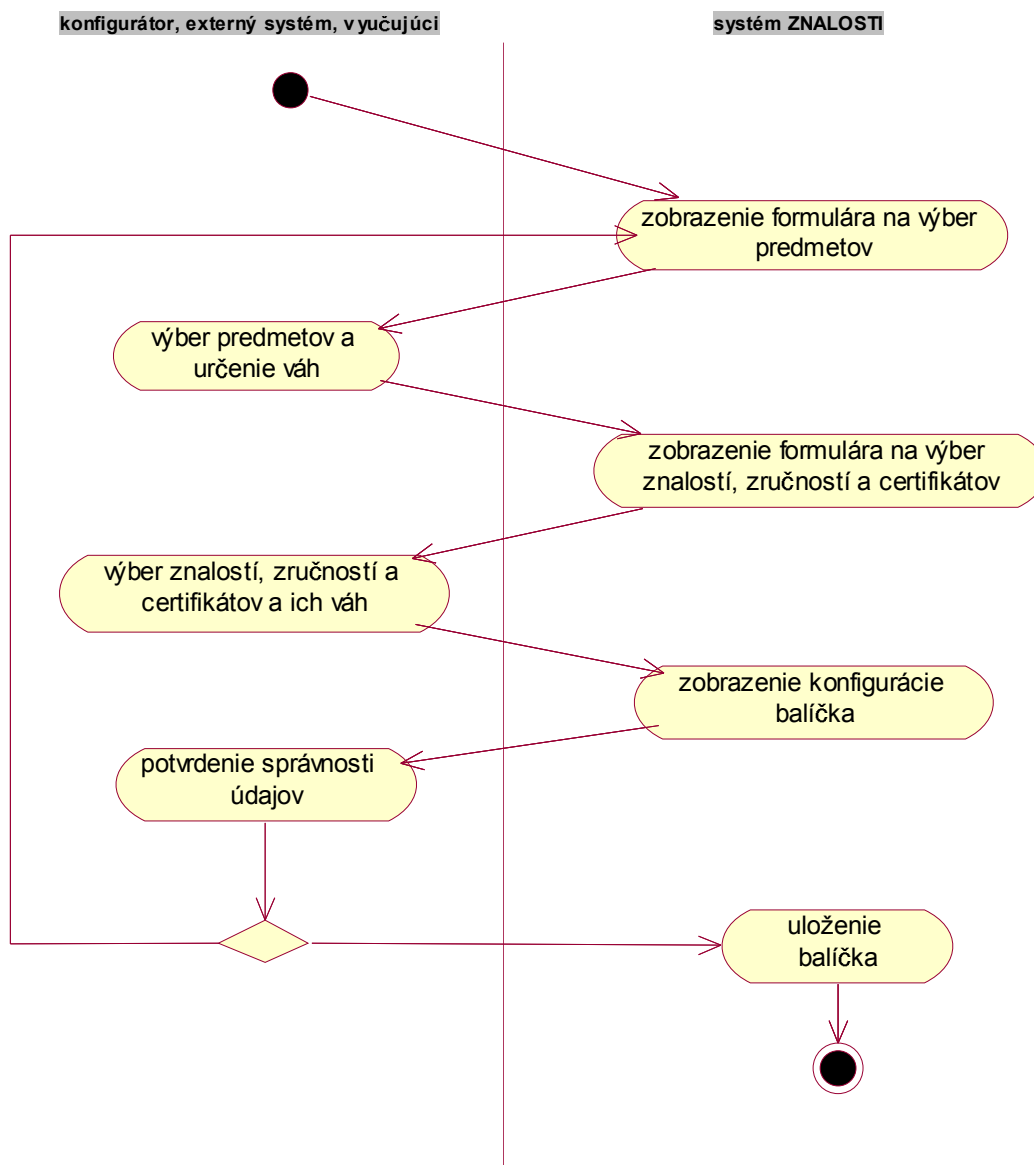
<i>Vstupy:</i>	oznámenie o udalosti, pri ktorej sa má notifikácia vykonať
<i>Výstupy:</i>	oznam na web stránke alebo emailová správa
<i>Používatelia:</i>	administrátor, vyučujúci, študent
<i>Opis:</i>	Oznamuje používateľovi výskyt istej udalosti, alebo ho žiada o vykonanie nejakej akcie. Môže byť načasovaná (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. Slúži 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
Zájmy 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 napĺňania 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 obslúženie $\frac{1}{4}$ 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:

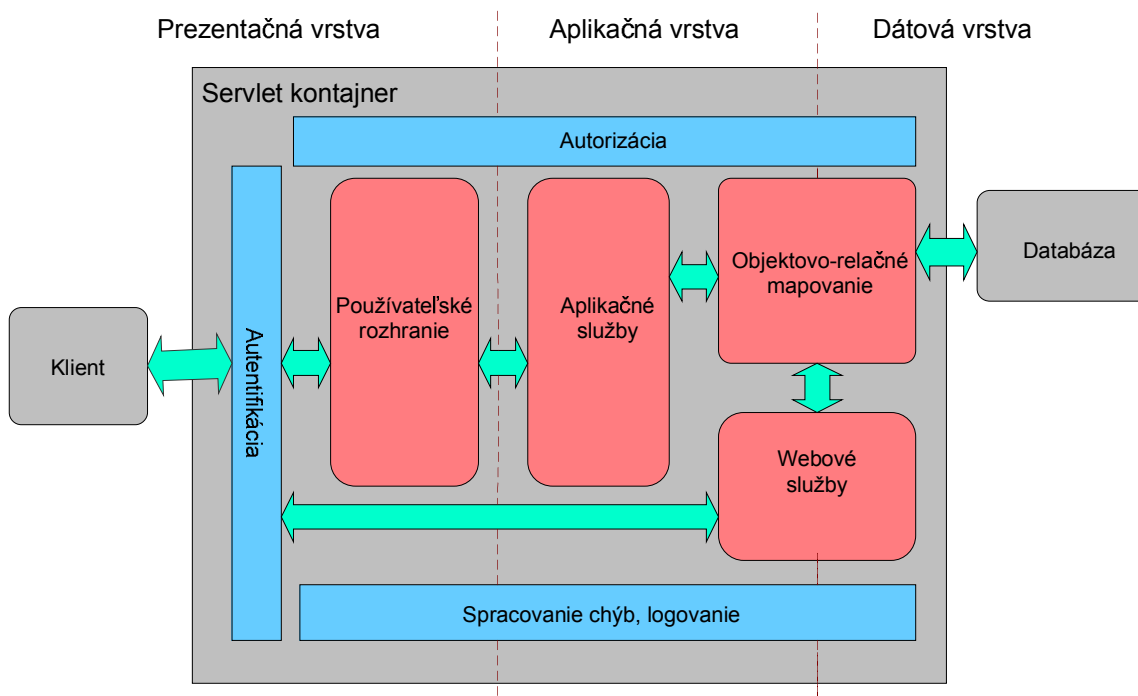
- Pentium III 1GHz
- 256 MB RAM
- 10GB diskového priestoru.

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 funkcionálnosť 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.

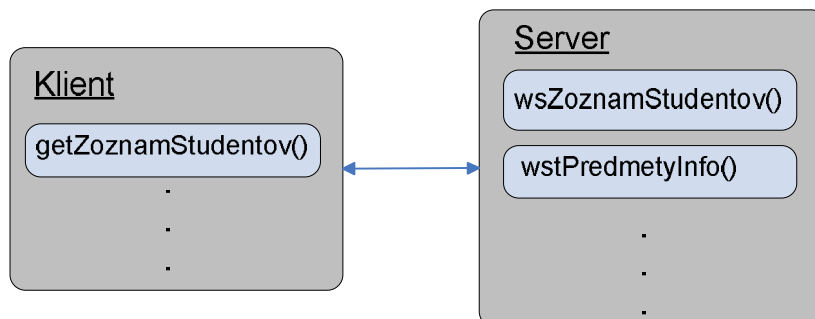
1.5 Objektovo-relačné mapovanie

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 *šstudent* a *šstudent o sebe* hovorí, že *šstudent* o sebe môže napísať viacero článkov (*šstudent o sebe*), pričom článok (*šstudent o sebe*) sa vždy vzťahuje na konkrétneho študenta.

Vzťah *obsahuje* medzi entitami *šstudent o sebe* a *klúčové slovo* hovorí, že článok o študentovi (*šstudent o sebe*) obsahuje *klúčové slová*. *Ključové slová* sa tiež môžu nachádzať vo viacerých článkoch (*šstudent 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 *šstudent* a *ovláda* hovorí, že každá entita *ovláda* obsahuje jedného *šstudenta*. *Šstudentovi* môže patriť viacero entít *ovláda*.

Vzťah *má* medzi entitami *šstudent* a *dôkaz* hovorí, že pre každú entitu *dôkaz* existuje jeden *šstudent*. Pre *šstudenta* 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.

V PROTOTYP

V nasledujúcej časti dokumentácie sú popísané postupy, ciele a výsledky prototypovania systému Baza znalostí a zručností študentov. V závere sú uvedené prínosy a závery, ktoré vyplynuli z tohto procesu.

V prvej podkapitole sa venujeme prehľadu a cieľom prototypovania konkrétnych častí systému a zdôvodňujeme výber implementačného prostredia pre prototyp. V ďalších podkapitolách sa zaoberáme konkrétnymi časťami prototypu, teda opisujeme kroky implementácie, prípadné problémy, ich riešenie a zmeny vplývajúce na dátový model príp. iné časti hrubého návrhu. Na záver každej podkapitoly je stručný prehľad ponaučení a výsledkov, ktoré budú nápomocné pri realizácii systému v nasledujúcom období.

1 Ciele prototypovania

1.1 Softvérový prototyp

Softvérový prototyp je čiastočná implementácia systému, ktorej cieľom je dozvedieť sa niečo viac o riešenom probléme alebo o možnom riešení problému; zvyčajne sa orientuje na špecifikáciu požiadaviek [4]. V tomto prípade sa vytvára prototyp najmä kvôli tomu, že členovia tímu sa zhodujú na funkcionalite, avšak nikto v skutočnosti nevie ako by mal systém „vyzerat“ a zároveň môže vzniknúť veľa ďalších zatiaľ neodhalených požiadaviek na realizovaný systém.

Realizácia prototypu je rozdelená na tri nezávislé logické časti, ktoré sú kritické pre návrh systému a jeho budúcu realizáciu:

- Databáza
- Web-services – komunikácia s databázou
- Parser pre získavanie údajov zo systému ŠTUDENT

Cieľom prototypovania databázy je lepšie porozumieť požiadavkám na systém a odhaliť nedostatky v údajovom modeli systému. Ide teda o prototyp na zahodenie [4].

Komunikácia s databázou prostredníctvom web-services je prístup, s ktorým členovia tímu nemajú praktické skúsenosti. Prototypovanie tejto časti má za cieľ oboznámiť sa s technológiami (webové služby, databáza, JSP stránky), ktoré chceme využívať v našom projekte a vytvorenie praktických príkladov použitím týchto technológií, pričom požiadavky sú jasne stanovené (prieskumné prototypovanie) [4].



Pred nasadením systému ZNALOSTI do prevádzky bude potrebné naplniť databázu veľkým množstvom údajov, ktoré spolu súvisia. Toto predstavuje netriviálny problém. Požiadavky na transformáciu dokumentov do jednotného tvaru vhodného na import sú však jasné (prieskumné prototypovanie). Prototyp parsera má za úlohu overiť poloautomatizovaný proces konverzie údajov od iných systémov do správneho tvaru a ich import do databázy, ale tiež odhaliť možné štrukturálne problémy v databáze. Parser je použiteľný aj v ďalších etapách implementácie.

1.2 Implementačné prostredie

1.2.1 Databáza

Implementačné prostredie pre časť prototypu – databáza sme zvolili MS Access kvôli tomu, že každý člen tímu sa s týmto prostredím stretol a pre prototyp, ktorého účelom je overiť správnosť návrhu, je postačujúce.

1.2.2 Web-services

Aplikačnú logiku tejto časti prototypu sme implementovali v jazyku Java (vývojové prostredie Eclipse), ktorý sme si zvolili ako hlavný implementačný jazyk pre náš projekt. Vzhľadom k platformovej nezávislosti tohto jazyka je možné spustenie prototypu tak pod operačným systémom Windows, ako aj pod operačným systémom Linux. Na využívanie webových služieb sme použili balík AXIS, ktorý obsahuje implementáciu protokolu SOAP.

Na úrovni dátovej vrstvy sme použili databázu PostgreSQL, ktorú sme sa tiež rozhodli využívať v našom projekte.

Na implementáciu prezentačnej úrovne sme použili JSP stránky.

1.2.3 Parser

Časť prototypu venovaná konverzii dát zo systému *Študent* je implementovaná pomocou súboru nástrojov narábajúcich s textom. Jednak sú to klasické UNIX programy obsiahnuté v každej implementácii shell-u (ako *cat*, *cut* a pod.), ako aj ďalšie komplexnejšie nástroje (*perl*, *sed*, *mawk*).

2 Technická dokumentácia

2.1 Databáza

2.1.1 Tabuľky a relácie

Realizácia a opis funkčnosti

Tabuľky a relácie boli vytvorené na základe dátového modelu uvedeného v časti IV Hrubý návrh. Oproti pôvodnému návrhu neboli v prototyp implementované hodnotenia študenta a učiteľa. Taktiež vznikla nová tabuľka *tmpStudentIndex* slúžiaca na dočasné ukladanie výsledkov výpočtu ohodnotenia študentov.

Výsledky

Pri implementácii sme zistili určité nedostatky v návrhu. Oproti pôvodnému fyzickému modelu nastali len menšie zmeny a týkali sa tabuliek *predmet*, *študent* a *známka*. Taktiež bola pridaná nová tabuľka *odbor*, ktorá udržiava názvy vyučovaných odborov. Zmenený model údajov je znázornený na obrázku [Obrázok V - 1]. Zmeny sú vyznačené červenou farbou.

Tabuľka *študent* bola obohatená o niekoľko atribútov:

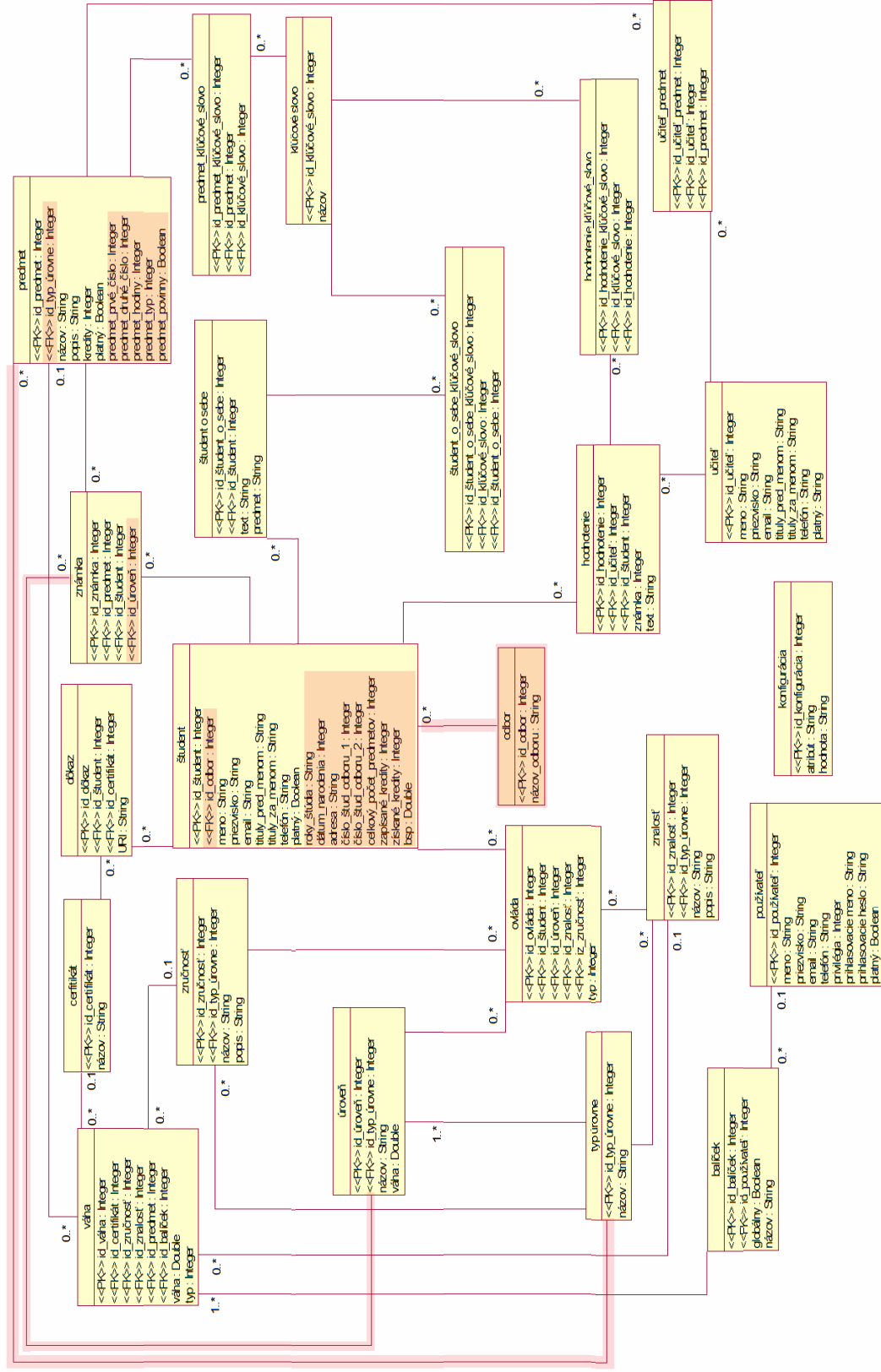
- *id_odbor* – identifikuje zaradenie študenta na určitý odbor. Medzi tabuľkami *študent* a *odbor* tak vzniká relácia *študuje odbor*.
- *roky_štúdia* – doterajší rozsah štúdia študenta (teda napríklad 2001-2004)
- *dátum_narodenia* – dátum narodenia študenta
- *adresa* – adresa bydliska študenta
- *číslo_štud_odboru_1* a *číslo_štud_odboru_2* – jednociferné a štvorciferné číslo odboru
- *celkový_počet_predmetov* – celkový počet doteraz absolvovaných predmetov
- *zapísané_kredity* – suma doteraz zapísaných kreditov
- *získané_kredity* – suma doteraz získaných kreditov
- *bsp* – aktuálny bodový študijný priemer študenta

V tabuľke *predmet* pribudli nasledovné atribúty:



- *id_typ_úrovne* – identifikuje typ úrovne použitý pri známkovaní predmetu (napríklad môže ísť o slovné známkovanie alebo o písmenkové známkovanie). Medzi tabuľkami *predmet* a *typ_úrovne* tak vzniká nová relácia *má typ úrovne*.
- *predmet_prvé_číslo* – predčíslie čísla predmetu
- *predmet_druhé_číslo* – štvorčíslie predmetu
- *predmet_hodiny* – počet vyučovacích hodín predmetu v jednom týždni

Tabuľka *známka* sa oproti pôvodnej zmenila. Pôvodný atribút *známka* bol nahradený atribútom *id_úroveň*, čím sa vytvoril vzťah s tabuľkou *úroveň*. Úrovne (hodnoty) známok sú teraz uchovávané v tabuľke *úroveň*, na ktoré sa každá inštancia tabuľky *známka* odkazuje.



Obrázok V - 1: Modifikovaný fyzický model údajov.

2.1.2 Formulár pre študenta

Realizácia a opis funkčnosti

Študent si v prototypu dokáže pridávať, editovať a odoberať znalosti, zručnosti a certifikáty spolu s ich ohodnotením. Každá znalosť a zručnosť môže mať rôznu stupnicu na hodnotenie, napr. slovnú alebo stupňami. V prípade certifikátov je hodnotenie nahradené odkazom na miesto, kde je zverejnená informácia o jeho vydaní.

Formulár bol vytvorený v prostredí MS Access. Jeho účelom je zabezpečiť základné funkcie informačného systému (CRUD – Create, Retrieve, Update, Delete), aby bolo možné jednoducho modifikovať databázu a pozorovať najmä správanie algoritmu na výpočet výsledného indexu na základe balíčkov.

Výsledky

Implementácia prototypu formuláru pre študenta potvrdila správnosť návrhu.

2.1.3 Formulár výberu študentov a „balíčka“

Realizácia a opis funkčnosti

Tento formulár sprístupňuje základnú funkčnosť systému, ktorou je práve vyhodnocovanie študentov na základe vybraného balíčka. Používateľ si môže na základe určitých kritérií vybrať študentov, ktorých chce hodnotiť a následne vybrať „balíček“, na základe ktorého ich chce hodnotiť. Základné kritériá pre výber študentov sú ročník a odbor ich štúdia, avšak kritérium pre odbor nebolo v prototypu implementované. Používateľ môže balíček vybrať buď z globálnych balíčkov v systéme alebo z balíčkov, ktoré si sám v priebehu používania systému nadefinoval. Implementácia prototypu však nepracuje s používateľmi systému, preto je mu umožnené vybrať zo všetkých balíčkov v systéme.

Formulár v MS ACCESS pristupuje k tabuľkám *student* a *balicek*. Na základe výberu kritéria pre študentov upravuje databázový dotaz a zobrazuje len požadovaných študentov. Z tých je potom možné ešte presnejšie vybrať len úzku skupinu, ktorá bude ohodnotená na základe zvoleného balíčka. Vyhodnotenie je uskutočnené pomocou funkcie na výpočet výsledného indexu na základe „balíčkov“.

Výsledky

Implementácia prototypu formuláru výberu študentov a balíčka potvrdila správnosť návrhu.

2.1.4 Formulár na konfiguráciu „balíčkov“

Balíček je skupina znalostí, zručností, certifikátov a predmetov, ktoré majú používateľom priradené váhy v intervale $<0, 1>$. Slúži na definovanie kritérií, ktoré sú pre používateľa (vyučujúceho) dôležité pre výpočet výsledného indexu.

Balíček môže byť buď globálny, teda viditeľný pre každého používateľa alebo personalizovaný, teda viditeľný iba pre jedného konkrétneho používateľa, ktorý ho vytvoril.

Realizácia a opis funkčnosti

V prostredí MS ACCESS bol vytvorený formulár pracujúci nad tabuľkami *balicek* a *vaha*, ktorý umožňuje vytvárať balíčky a definovať jednotlivé položky s priradením váh. Jednotlivé položky je možné editovať, pričom sa automaticky zmení typ. Napríklad ak používateľ zmení položku *znalost' C++* na položku *predmet MAS*. Váhy je možné zadávať ako číslo z intervalu $<0, 1>$, pričom 0 znamená že položka nebude dočasne braná do úvahy pri výpočte výsledného indexu. Každá položka môže byť trvale odstránená z balíčka.

Výsledky a odporúčania pre zlepšenie návrhu

Pri návrhu sme zistili, že tabuľka balíček sa viaže na používateľa, avšak nie na vyučujúceho. Chýba definícia vzťahu vyučujúci – používateľ. Odporúčame vytvoriť vzťah *1-k-1* medzi tabuľkami *ucitel* a *pouzivatel*. Takisto podobný vzťah by mal byť definovaný medzi používateľom a študentom.

2.1.5 Formulár na konfiguráciu skupín predmetov

Skupina predmetov definuje množinu predmetov, z ktorých každému je priradená váha z intervalu $<0, 1>$. Príkladom môže byť skupina s názvom *Programovacie predmety*, ktorá obsahuje napríklad aj predmet *Modelovanie a simulácia*, hoci tento nie je priamo zameraný na programovanie (študenti však pri jeho absolvovaní v hojnej miere využijú svoje programátorské zručnosti). [Tabuľka V-1] Skupiny slúžia buď ako globálne balíčky, ale rovnako sú použité aj v súhrnnej informácii o študentovi, kde môže byť uvedené, že z programovacích predmetov patrí študent medzi 20% najlepších. Skupiny predmetov vytvára konfigurátor.

Predmet	Váha
Programovanie v C	1
Objektovo orientované programovanie	1
Modelovanie a simulácia	0.3
Stavba operačných systémov	0.4
...	...

Tabuľka V-1: Príklad skupiny predmetov



Realizácia a opis funkčnosti

Skupina predmetov je balíček, ktorý ako položky obsahuje iba predmety. Preto bol vytvorený podobný formulár ako ten, ktorý slúži na správu balíčkov, avšak jeho funkčnosť je obmedzená tým, že ako položku dovoľuje definovať iba predmet a priradiť mu váhu.

Výsledky a odporúčania pre zlepšenie návrhu

Bolo by vhodné do tabuľky *balicek* pridať *bool* atribút *jeSkupinaPredmetov*, ktorý by definoval, či je daný balíček štandardný, alebo reprezentuje skupinu predmetov. Zatiaľ bola na toto rozlíšenie použitá prítomnosť alebo absencia *id_pouzivatel*.

2.1.6 Funkcie na výpočet výsledného indexu na základe „balíčkov“

Výpočet výsledného hodnotenia študenta na základe konkrétneho balíčka prebieha nasledovne. Zistí sa obsah balíčka a následne sa pre každú položku balíčka (ako už vieme balíček môže obsahovať ľubovoľný počet znalostí, zručností, predmetov a certifikátov) počíta bodový zisk, ktorý študent za konkrétnu položku balíčka získa. Postupne sa tieto bodové hodnoty nasčítavajú cez všetky položky balíčka, čím sa získa výsledné hodnotenie študenta. Ak tento postup aplikujeme na množinu študentov, získame tým zoznam výsledných ohodnotení študentov. Získaný zoznam sa následne usporiada podľa výsledného hodnotenia, čím sa získa poradie študentov od najlepšieho po najhoršieho.

- Pretože používateľ by nemal vidieť usporiadaný zoznam študentov, tak ako bol vygenerovaný v predošlom kroku (od najlepšieho študenta po najhoršieho) je potreba, aby bol tento zoznam určitým spôsobom „zahmlený“, tak aby sme vedeli rozlíšiť iba skupiny študentov (napr. skupina výborných študentov, skupina priemerných študentov, ...). Funkcia, ktorá zaraďuje študentov do tried by teda mala mať nasledovné vlastnosti:
- musí vedieť rozlíšiť dobrých a horších študentov, teda požaduje sa, aby študenti s lepším hodnotením boli v lepšej triede ako študenti z horším hodnotením
- v prípade, že všetci hodnotení študenti majú veľmi podobné výsledky, požaduje sa, aby aj napriek tomu bolo možné študentov zaradiť do rozličných tried a tak ich navzájom rozlíšiť
- jednoduchá, prípadne žiadna konfigurácia

Je teda zrejmé, že počet tried, do ktorých sa študenti rozdelia nemôže byť dopredu určený, ale musí byť vypočítaný na základe konkrétnych hodnotení študentov. Ak by bol počet tried určený napevno, mohlo by sa stať, že všetci hodnotení študenti by ležali v jednej triede, čím by nám systém neposkytol žiadnu informáciu.

Ak má byť počet tried vypočítaný na základe konkrétnych hodnotení študentov, zrejme bude vhodné využiť niektorú štatistickú funkciu. Ako vhodné sa javí použiť smerodajnú odchýlku. Tá má nasledovné vlastnosti: [5], [6]

- hovorí o tom, ako sa odlišujú jednotlivé realizácie náhodných premenných (pozorovania)
- je tým väčšia, čím viac sú jednotlivé pozorovania ďalej od priemeru, čiže ak sú všetky blízko seba, smerodajná odchýlka je malá
- ak sú pozorovania rozdelené podľa normálneho rozdelenia, vieme, že vo vzdialenosti jednej smerodajnej odchýlky (+/- 1*StdDev) leží 68% pozorovaní, vo vzdialenosti dvoch 95% a vo vzdialenosti troch 99.7% pozorovaní
- vypočítava sa z nezáporných dát

Kvôli predošlým vlastnostiam sme sa rozhodli použiť na získanie počtu tried smerodajnú odchýlku. Najskôr sa vypočíta smerodajná odchýlka z normalizovaného zoznamu výsledkov študentov (normalizovaný na interval <0,1>). Následne sa počet tried T vypočíta ako

$$T = \frac{1}{StdDev(výsledky_študentov)}$$

Nakoniec sú študenti zadelení do tried podľa toho, do ktorej triedy spadajú ich výsledné hodnotenia.

Realizácia a opis funkčnosti

Pretože sme sa rozhodli prototyp realizovať v prostredí MS ACCESS, funkcia na výpočet výsledných hodnotení a následné zaradenie do tried je napísaná v jazyku MS Visual Basic.

Pretože funkcia je pomerne zložitá, nie je (v MS ACCESS) možné napísať jeden SQL dopyt, ktorý by realizoval všetky hore uvedené funkcie. Z tohto dôvodu bolo treba jednotlivé časti robiť postupne a výsledky postupne ukladať do pomocnej tabuľky (bolo treba vytvoriť novú tabuľku lebo MS ACCESS nepodporuje vytváranie dočasných tabuliek). Do tejto tabuľky sa v priebehu výpočtu ukladajú výsledné indexy hodnotenia jednotlivých študentov. Okrem výsledného indexu hodnotenia sa do tabuľky vkladajú aj informácie, ako na výsledný index hodnotenia vplyvajú znalosti, zručnosti, certifikáty a známky z predmetov. Následne sa do tabuľky pridajú normalizované indexy hodnotenia (normalizuje sa na interval <0,1>) pre každého študenta. Ako bolo popísané vyššie, vypočíta sa smerodajná odchýlka z normalizovaných indexov a následne počet tried a študenti sú rozdelení do tried na základe výsledku. Okrem triedy sa do tabuľky pridá aj rozsah triedy. Napríklad trieda môže mať rozsah (v závislosti od toho koľko tried existuje) 100% - 90%.

2.1.7 Prezentácia výsledkov výpočtu výsledného indexu na základe „balíčkov“

Realizácia a opis funkčnosti

Prezentácia výsledkov nadväzuje na formulár výberu študentov a „balíčka“. Výsledky výpočtu funkcie na výpočet výsledného indexu na základe „balíčkov“ sú uložené do tabuľky *tmpStudentIndex*, do ktorej sa následne pristupuje pomocou databázovej požiadavky *qStudentiTriedy*. Táto požiadavka usporiada študentov podľa jednotlivých tried a ich mena a následne ich zobrazí používateľovi v podobe správy. Správa obsahuje okrem mena študenta a zadelenia do triedy aj percentuálne príspevky zručností, znalostí, certifikátov a predmetov k celkovému ohodnoteniu študenta.

Výsledky

Výberom vhodných kombinácií balíčkov a študentov (t.j. zručností, znalostí, certifikátov a predmetov), sme pozorovali správanie funkcie na výpočet výsledného indexu na základe balíčkov. Toto zodpovedalo očakávanému správaniu sa, jediným nedostatkom bolo, že najhorší študenti boli v rovnakej triede ako tí, ktorých sa daná konfigurácia balíčka vôbec netýkala, teda neovládali znalosť, zručnosť, nevlastnili certifikát alebo nemali známku z predmetu takú, ako bola nastavená v zvolenom balíčku.

2.2 Web-services

Realizácia a opis funkčnosti

Prototyp webových služieb, ktorý sme implementovali pozostáva z dvoch častí (klient a server). Serverová časť je umiestnená na strane aplikačného servera (Tomcat) a jej úlohou je poskytovať služby (ktorými disponuje) klientom, ktorí o ne žiadajú. V rámci prototypu sme implementovali tieto služby: poskytnutie zoznamu predmetov, znalostí, zručností, certifikátov, študentov a poskytnutie informácií o študentovi a o predmete na základe identifikačného čísla predmetu, resp. študenta.

Kód serverovej časti je umiestnený v súbore s príponou *jws*. Je to v podstate kód napísaný v jazyku Java, ktorý obsahuje metódu pre inicializáciu databázového spojenia a metódy pre konkrétne webové služby. Každá metóda pristupuje k databáze (sql dotaz) a získava z nej požadované údaje. Tieto údaje potom vhodným spôsobom vracia. Pri zavolaní webovej služby sa kód skompiluje (vytvorí sa class súbor) a spustí.

Klientská časť bola implementovaná použitím jsp stránok. Hlavná štruktúra kódu klientskej časti je nasledovná:

- nastavenie adresy serverovej časti webovej služby
- vytvorenie objektov typu Service a Call
- nastavenie konkrétnej metódy webovej služby, ktorú chceme volať
- volanie webovej služby
- spracovanie vrátených výsledkov

Výsledky

Implementácia tejto časti prototypu splnila zamýšľaný účel. Oboznámili sme sa z technológiou webových služieb a prakticky sme si to vyskúšali implementáciou konkrétnych služieb v jazyku Java. Vyskúšali sme si prácu s databázou PostgreSQL a jsp stránkami.

2.3 Parser

Parser slúži na prispôbenie vstupných dát pochádzajúcich zo systému ŠTUDENT, ktorý obsahuje absolvované predmety a výsledky študentov z týchto predmetov. Ďalší vstupný súbor je zoznam predmetov, pochádzajúci tiež zo systému ŠTUDENT a obsahuje popis predmetov. Výstupom parsera sú súbory vhodné na ľahké importovanie do tabuliek databázy.

Ide o súčasť prieskumového prototypu, pretože sa prototypuje jasná požiadavka naplňania databázy údajmi zo systému ŠTUDENT. Znalosti a zdrojový kód získané pri prototypovaní budú použiteľné aj vo finálnej verzii aplikácie.

Realizácia a opis funkčnosti

Za účelom konverzie súborov, ktoré tvoria výstup systému ŠTUDENT do podoby vhodnej na import dát do tabuliek MS ACCESS bola vytvorená sada *bash* skriptov. Súbory sú prekonvertované z textovej podoby s kódovaním cp852 alebo utf-8 do textového súboru s kódovaním utf-8, ktorý je priamo importovateľný do tabuliek MS ACCESS. Extrahovanie dát prebieha v nasledovných krokoch:

1. Vyťaženie známok absolvovaných predmetov pre študentov zo systému ŠTUDENT pomocou skriptu *parse_znamky.sh*.

2. Vyťaženie predmetov a odborov použitím skriptu *parse_predmety_odbory.sh*.

3. Import dát do databázy – Ručné importovanie dát z textových súborov do tabuliek MS ACCESS.

4. Spustenie pomocných selektov – Správne naplnenie tabuľky *znamka* pomocou príkazu SELECT.



Podrobnejší popis spomenutých krokov sa dá nájsť v používateľskej príručke.

Výsledky

Výstupy parsera boli importované do databázy a dáta boli použité pri prototypu. Parser bude s výhodou použitý aj v ďalšom vývoji produktu.

3 Používateľská príručka

Táto používateľská príručka má slúžiť ako návod na inštaláciu a používanie prototypu. Je rozdelená do troch základných častí. A to časti týkajúcej sa funkčnosti zabezpečenej pomocou databázy, časti venujúcej sa webovým službám a tiež časti zaoberajúcej sa parserom.

3.1 Databáza

3.1.1 Inštalácia a spustenie

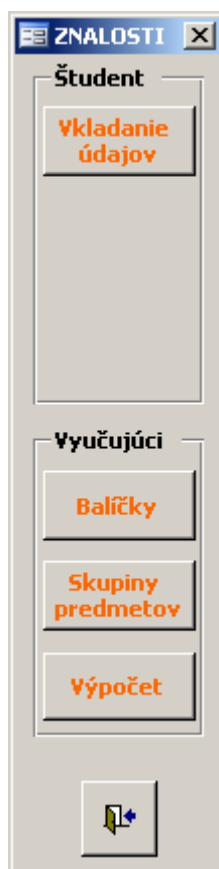
Prototyp na overenie funkčnosti zabezpečenej pomocou databázy je vytvorený v prostredí MS ACCESS. Inštalácia aj spustenie prototypu sú preto jednoduché:

1. Nainštalujte MS ACCESS z balíka MS OFFICE (verzia 2000 a vyššia, anglická jazyková mutácia)
2. Skopírujte súbor **prototyp.mdb** na lokálny disk.
3. Štandardným spôsobom spustíte nakopírovaný súbor.

Ak nastal problém s neregistrovanými súčastami MS ACCESS, skontrolujte verziu, ktorú používate a skúste súbor spustiť ešte raz. Ak problém pretrváva, postupujte nasledovne: V menu zvolíte View > Code a potom Insert > Module. Zatvorte aplikáciu a presvedčte sa o správnej funkčnosti. V prípade nefunkčnosti zvolíte View > Code a potom Tools > References. V zobrazenom formulári zakliknite potrebnú referenciu a znovu reštartujte aplikáciu.

3.1.2 Návod na používanie

Úvodný formulár



Obrázok V - 2: Úvodný formulár

Na obrázku [Obrázok V - 2] je zobrazený úvodný formulár na výber jednotlivých formulárov prototypu pomocou tlačidiel. V časti označenej študent sa nachádza tlačidlo *Vkladanie údajov*, ktoré slúži na otvorenie formuláru pre študenta. Časť pre vyučujúceho obsahuje 3 tlačidlá – tlačidlo *Baličky* otvára formulár konfigurácie balíčkov, tlačidlo *Skupiny predmetov* otvára formulár konfigurácie týchto skupín a tlačidlo *Výpočet* otvára formulár slúžiaci na vyhodnocovanie študentov na základe balíčka.

Formulár pre študenta

Obrázok [Obrázok V - 3] znázorňuje formulár pre študenta. Je navrhnutý pre potreby prototypu, teda tak, aby sa s ním čo najpohodľnejšie pracovalo a plnil účel pridávania, odoberania a editovania znalostí, zručností a certifikátov jednotlivým študentom.

Po vybratí študenta v oblasti 1, sa v oblastiach 2, 3 a 4 zobrazia jeho znalosti, zručnosti a certifikáty. Pomocou tlačidiel v oblasti 5 je možné pridávať nové, editovať vybrané alebo odstraňovať vybrané položky.

The screenshot shows a window titled "Student" with the following structure:

- 1**: A dropdown menu for "Študent:" containing the name "Baranec Michal".
- 2**: A table titled "Znalosť" (Knowledge) with columns "znalosť.nazov" and "uroven.nazov".

znalosť.nazov	uroven.nazov
C++	0
C--	5
C#	3
- 3**: A table titled "Zručnosť" (Skill) with columns "zrucnosť.nazov" and "uroven.nazov".

zrucnosť.nazov	uroven.nazov
OOP	2
Databazy	menej ako veľmi
Komunikacia	viac ako najlepšie
- 4**: A table titled "Certifikát" (Certificate) with columns "nazov" and "URI".

nazov	URI
acm	http://dokaz
- 5**: A set of three control buttons for each table: "Pridaj" (Add), "Edituj" (Edit), and "Vymaž" (Delete).

Obrázok V - 3: Formulár pre študenta na pridávanie znalostí, zručností a certifikátov

Formulár na pridávanie nových znalostí [Obrázok V - 4] obsahuje dve rozbaľovacie ponuky. Po vybraní znalosti (ľavá ponuka) sa sprístupní rozbaľovacia ponuka pre výber úrovne znalosti. Stlačením tlačidla OK sa nová znalosť pridá do databázy a zobrazí sa v oblasti 2 formulára pre študenta [Obrázok V - 3]. Formulár na pridávanie znalostí sa po stlačení OK nezatvorí, takže používateľ môže jednoducho pridávať ďalšie znalosti. Nie je však dovolené pridať znalosť, ktorá už v raz bola študentovi pridaná.

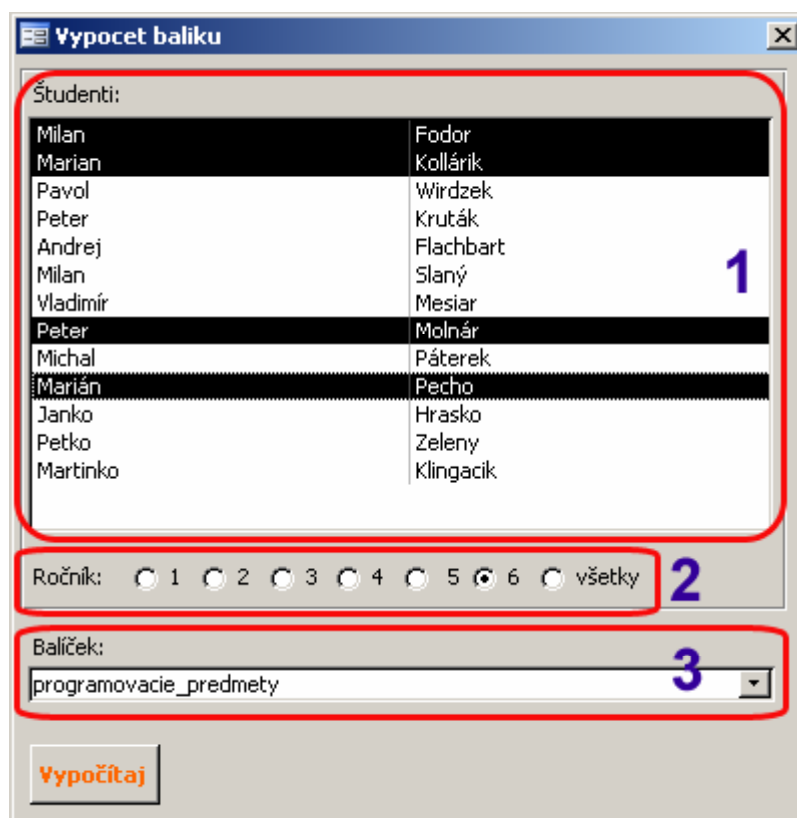
The "Pridaj znalosť" dialog box contains:

- A dropdown menu for "Znalosť:" with "C--" selected.
- A dropdown menu for "Uroveň:" with a list of levels: 0, 1, 2 (highlighted), 3, 4.
- A "Spat" (Back) button.

Obrázok V - 4: Formulár na pridávanie znalostí

Formulár výberu študentov a „balička“

Na základe tohto formulára [Obrázok V - 5] je možné vyhodnocovať študentov podľa zvoleného balička. V časti 1 je zobrazený zoznam študentov. Tento zoznam je filtrovaný na základe kritérií v oblasti 2. Týmto kritériom je aktuálny ročník štúdia študentov, ktorý je možné v tejto oblasti zvoliť. Výber študentov je možné robiť klasickým štýlom, teda súčasných držaním tlačidla ctrl alebo shift a označením študenta ľavým tlačidlom myši. Oblasť 3 slúži na výber balička, na základe ktorého chceme študentov hodnotiť. Samotné vyhodnotenie je uskutočnené po stlačení tlačidla *Vypočítaj* a zobrazený je formulár s výsledkami [Obrázok V - 6].



Vypocet baliku

Študenti:

Milan	Fodor
Marian	Kollárik
Pavol	Wirdzek
Peter	Kruták
Andrej	Flachbart
Milan	Slaný
Vladimír	Mesiar
Peter	Molnár
Michal	Páterek
Marián	Pecho
Janko	Hrasko
Petko	Zeleny
Martinko	Klingacik

Ročník: 1 2 3 4 5 6 všetky

Baliček:

Vypočítaj

Obrázok V - 5: Formulár na výpočet balička

Prezentácia výsledkov výpočtu výsledného indexu na základe „baličkov“

Formulár s výsledkami [Obrázok V - 6] sa zobrazí na základe požiadavky na výpočet zadanej vo formulári výberu študentov a „balička“. Formulár zobrazuje zaradenie študentov do jednotlivých tried (text oranžovej farby), pričom je možné vidieť aj jednotlivé percentuálne príspevky znalostí/zručností/certifikátov/predmetov na celkovom hodnotení študenta.

Rozdelenie študentov do tried

Trieda	Meno študenta	% znalosť	% zručnosť	% certifikát	% predmet
100% - 75%					
	Bobovský Martin	100	0	0	0
50% - 25%					
	Baranec Michal	100	0	0	0
25% - 0%					
	Budinský Tomáš	0	0	0	0
	Celler Lukáš	0	0	0	0
	Flešár Libor	100	0	0	0
	Fodor Milan	0	0	0	0
	Gabriel Miroslav	0	0	0	0
	Gregor Miroslav	0	0	0	0
	Chamilla Andrej	0	0	0	0
	Jánošík Ladislav	0	0	0	0
	Jelačič Boris	0	0	0	0

Obrázok V - 6: Zobrazenie výsledku výpočtov na základe balíčka

Formulár na konfiguráciu „balíčkov“

Uvedený formulár slúži na definovanie a konfiguráciu balíčkov. Na obrázku [Obrázok V - 7] je uvedený príklad balíčka, ktorý si vytvoril vyučujúci na výber študenta pre projekt OOP pri matematických výpočtoch.

Opis prvkov formulára:

- Názov balíčka – (nachádza sa v oblasti 1)

- Globálny – (nachádza sa v oblasti 2) pri zakliknutí je balíček globálny, teda bude viditeľný pre všetkých používateľov.
- Pre používateľa – (nachádza sa v oblasti 2) Obsahuje referenciu na používateľa, ktorému balíček patrí
- Váhy – (oblasť 3) Táto časť obsahuje definíciu položiek s priradením váhy. V každom riadku je jedna položka, ktorá môže byť zmazaná (pomocou kliknutia na ikonu odpadkového koša). Zmeny položky môžu byť „vzaté späť“ pomocou kliknutia na ikonu s modrou šípkou. Z predmetu, znalosti, zručnosti alebo certifikátu je vždy aktívny iba jeden, čo určuje typ položky.

The screenshot shows a window titled "Balíček" with a close button in the top right corner. The window contains the following elements:

- Field 1:** "Názov balíčka" (Package Name) with the value "Pokusny balicek".
- Field 2:** "Pre používateľa:" (For user) with a dropdown menu showing "Kabel".
- Checkbox:** "Globalny" (Global) is checked.
- Table:** A table with columns: "Predmet" (Subject), "Znalosť" (Knowledge), "Zrucnosť" (Skill), "Certifikát" (Certificate), and "Vaha" (Weight). The table contains five rows of data, each with a weight value and a set of navigation icons (blue arrow, trash, and a blue star).

The table data is as follows:

Predmet	Znalosť	Zrucnosť	Certifikát	Vaha
	C#			0,2
		Databazy		0,1
			CCNA	0,3
Analýza a zložitosť algorit				0,4

At the bottom of the window, there is a "Záznam:" (Record) section with navigation buttons and the text "1 z 3".

Obrázok V - 7: Formulár na konfiguráciu balíčkov

Ovládanie formulára je intuitívne so štandardnými ovládacími prvkami MS ACCESS. Medzi jednotlivými balíčkami sa dá listovať pomocou navigačného panelu vľavo dole. Pridanie balíčka sa realizuje prechodom na poslednú položku (kliknutím na ikonu so šípkou a hviezdíčkou v navigačnom paneli).

Ku každej položke v balíčku je priradená váha z intervalu $<0, 1>$, ktorá odráža významnosť danej položky v hodnotení. Váha 0 znamená žiadnu významnosť a váha 1 maximálnu.

Formulár na konfiguráciu skupín predmetov

Uvedený formulár slúži na definovanie a konfiguráciu skupín predmetov. Na obrázku [Obrázok V - 8] je uvedený príklad skupiny programovacích predmetov. Tento je použiteľný buď ako globálny balíček, alebo môže byť zobrazený v komplexnej informácii o študentovi s uvedením jeho úspešnosti v tejto skupine. Prototyp obsahuje iba prvú možnosť. Váhy majú rovnaký význam ako v balíčku.

Opis prvkov formulára:

- Názov skupiny – (oblasť 1)
- Váhy predmetov – (oblasť 2) Táto časť obsahuje definíciu položiek s priradením váhy. V každom riadku je jeden predmet s definovaním váhy, ktorý môže byť zmazaný (pomocou kliknutia na ikonu odpadkového koša). Zmeny môžu byť „vzaté späť“ pomocou kliknutia na ikonu s modrou šípkou. Na pridanie predmetu slúži posledný prázdny riadok.

Predmet	Váha
programovanie v c++	0,3
programovanie v java	0,3
	0

Obrázok V - 8: Formulár pre definovanie skupiny predmetov

Ovládanie formulára je intuitívne so štandardnými ovládacími prvkami MS ACCESS. Medzi jednotlivými skupinami sa dá listovať pomocou navigačného panelu vľavo dole. Pridanie skupiny



predmetov sa realizuje prechodom na poslednú položku (kliknutím na ikonu so šípkou a hviezdíčkou v navigačnom paneli).

3.2 Web-services

3.2.1 Inštalácia a spustenie

V tejto časti uvádzame potrebné produkty, kroky inštalácie a konfigurácie potrebné k správne chodu tejto časti prototypu. Uvádzame verzie produktov, na ktorých bol prototyp vyvíjaný. Jedná sa o produkty pre operačný systém Windows. Všetky potrebné súčasti sú k dispozícii na priloženom CD nosiči.

Databáza

Je potrebné mať nainštalovanú databázu Postgresql (verzia 8.1). Po inštalácii je treba vytvoriť databázu Znalosti a naplniť ju potrebnými údajmi. Odporúčame nasledovný postup:

- vytvorenie prázdnej databázy Znalosti (vlastník postgres) pomocou nástroja pgAdmin, ktorý je súčasťou produktu
- urobiť restore databázy, ktorá je na priloženom CD a obsahuje potrebné dáta
- treba nastaviť správne heslo (patanka) pre prístup k databáze alebo zmeniť toto heslo v zdrojovom kóde webovej služby, pri inicializácii databázového spojenia

Aplikačný server

Je potrebné mať nainštalovaný aplikačný server Tomcat (Jakarta-tomcat-4.1.18).

Java

Treba mať nainštalovanú Javu (verzia j2sdk 1.4.2.10).

Odporúčaný postup inštalácie a konfigurácie

Najskôr treba nainštalovať javu, následne aplikačný server Tomcat. Databáza sa môže inštalovať nezávisle.

Následne treba skopírovať z adresára **axis-1_3** podadresár **webapps\axis** do adresár **Apache Group\Tomcat 4.1\webapps**

Ďalej treba skopírovať do adresára **Apache Group\Tomcat 4.1\webapps\axis\WEB-INF\lib** súbory **activation.jar**, **xercesImpl.jar** a **xml-apis.jar**, **postgresql.jar** z priloženého CD nosiča. Súbory **postgresql.jar** a **activation.jar** treba nakopírovať do adresára **C:\j2sdk1.4.2_10\lib**.



Posledným krokom je nastavenie systémových premenných nasledovne (uvádzame prednastavené adresáre po inštalácii jednotlivých produktov):

```
JAVA_HOME = C:\j2sdk1.4.2_10  
AXIS_HOME = C:\Program Files\Apache Group\Tomcat 4.1\webapps\axis  
AXIS_LIB = %AXIS_HOME%\lib  
AXISCLASSPATH=%AXIS_LIB%\axis.jar;%AXIS_LIB%\commons-discovery-  
0.2.jar;%AXIS_LIB%\commons-logging-1.0.4.jar; %AXIS_LIB%\jaxrpc.jar;  
%AXIS_LIB%\saaaj.jar; %AXIS_LIB%\log4j-1.2.8.jar; %AXIS_LIB%\xml-apis.jar;  
%AXIS_LIB%\xercesImpl.jar; %AXIS_LIB%\postgresql.jar  
CLASSPATH = %AXIS_LIB%\axis.jar; %AXIS_LIB%\xml-apis.jar;  
%AXIS_LIB%\jaxrpc.jar; C:\j2sdk1.4.2_10\lib\tools.jar ;.;  
C:\j2sdk1.4.2_10\lib\postgresql.jar; C:\j2sdk1.4.2_10\lib\activation.jar
```

Po zmene nastavení treba reštartovať server Tomcat.

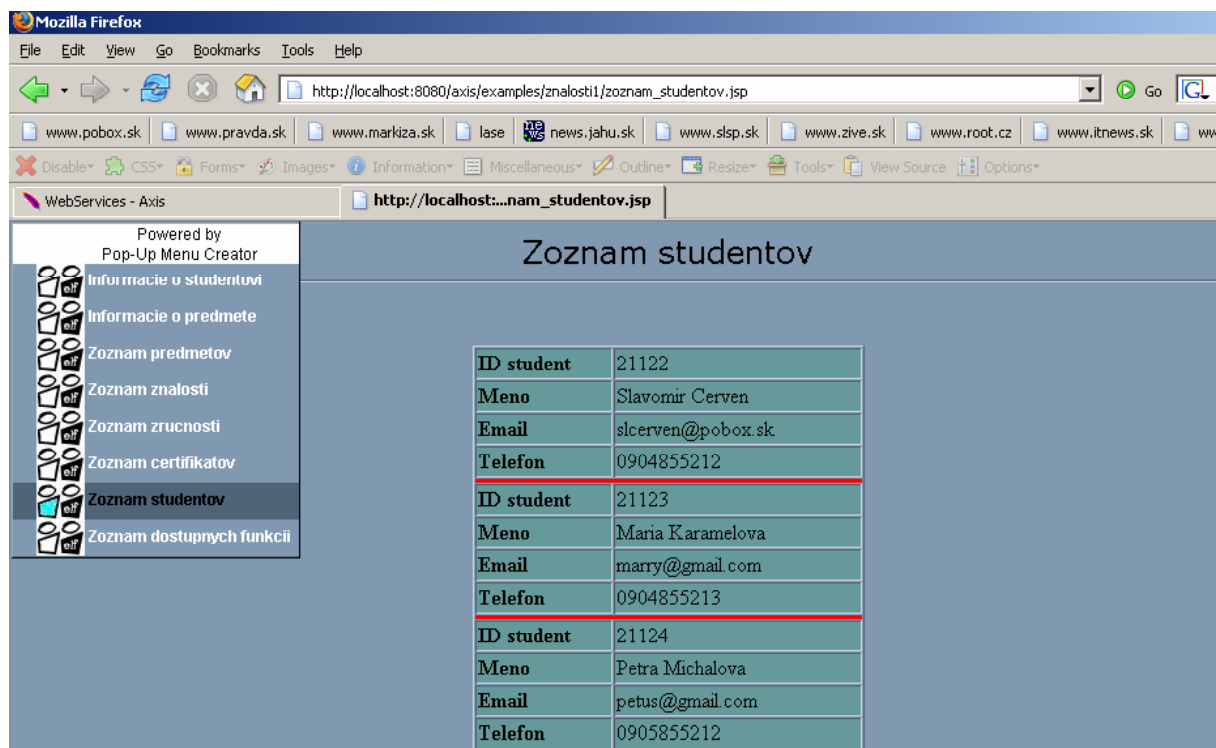
Inštalácia zdrojových súborov

Z priloženého CD treba skopírovať súbor SelectZnalosti.jws do adresára C:\Program Files\Apache Group\Tomcat 4.1\webapps\axis, ďalej treba skopírovať adresár znalosti1 do adresára C:\Program Files\Apache Group\Tomcat 4.1\webapps\axis\examples.

3.2.2 Návod na používanie

Prezentačná časť tohto prototypu je vytvorená pomocou jsp stránok. Preto používateľovi stačí zadať v prehliadači cestu k úvodnej stránke (podľa vyššie uvedeného postupu to bude localhost:8080/axis/examples/znalosti1/index.jsp).

Úvodná stránka klientskej časti [Obrázok V - 9] obsahuje menu, v ktorom je zoznam služieb, ktoré sú k dispozícii. Po kliknutí na konkrétnu položku z menu sa spustí jsp stránka, ktorá zavolá webovú službu a zobrazí vrátené výsledky.



Obrázok V - 9: Prezentačné rozhranie pre webové služby pomocou jsp stránok

AXIS zahŕňa nástroj Java2WSDL, ktorý generuje wsdl súbor (popisuje webovú službu) zo zdrojového kódu v Jave. Tento súbor je možné zobrazit' priamo v prehliadači zadaním adresy webovej služby a pridaním postfixu ?wsdl (napr. localhost:8080/axis/Service1.jws?wsdl).

3.3 Parser

Parser slúži na prispôsobenie vstupných dát pochádzajúcich zo systému ŠTUDENT, ktorý obsahuje absolvované predmety a ich výsledky študentov. Ďalší vstupný súbor je zoznam predmetov, pochádzajúci tiež zo systému ŠTUDENT a obsahuje opis predmetov. Výstupom parsera sú súbory vhodné na ľahké importovanie do tabuliek databázy.

3.3.1 Inštalácia a spustenie

Pre správny chod parsovacích skriptov je potrebné mať nainštalované nasledovné verzie programov:

- Linux labss2 2.4.18-1-686-smp
- iconv (GNU libc) 2.3.2
- perl, v5.8.4 built for i386-linux-thread-multi
- mawk 1.3.3 Nov 1996



- sed version 4.1.2
- bash, version 2.05a.0(1)-release (i386-pc-linux-gnu)

Súčasná konfigurácia servera labss2 umožňuje spustenie skriptov.

3.3.2 Návod na používanie

Výstupný formát parsovaných dát, vhodných na import do databázy je v utf-8, stĺpce sú oddelené bodkočiarkou a prvý riadok obsahuje názvy stĺpcov.

Kroky pri konverzii a importe dát:

1. **Vytáženie známok** absolvovaných predmetov pre študentov zo systému ŠTUDENT pomocou skriptu *parse_znamky.sh*.

```
fenik@labss2:~/parser_v8$ ./parse_znamky.sh

Usage ./parse_znamky.sh [-convert|-noconvert] input_file student_info_file
predmet_info_file
```

Prepínač *-convert* spôsobí konverziu kódovania vstupného súboru, ktoré je zvyčajne cp852 na kódovanie utf-8. V prípade, že vstupný súbor je prekódovaný na utf-8 nejakou inou aplikáciou napr. textovým editorom, použije sa prepínač *-noconvert*.

input_file je vstupný súbor (napríklad *_svahnfe.txt*). Výstupné súbory sú *student_info_file* (odporúčaný názov je *student_info.txt*), kde sa uložia informácie o študentoch a *predmet_info_file* (odporúčaný názov je *predmet_info.txt*), kde sa uložia informácie o známkach daných študentov, prepojených osobným číslom študenta.

Pre prípadné zahmlenie testovacích dát, ktoré obsahujú ostré údaje, je potrebné editovať a použiť skript *./cenzure_student.sh > student_info_cenz.txt* a *./cenzure_predmety.sh > predmet_info_cenz.txt*. Ďalej je potrebné skontrolovať počty riadkov pomocou *wc*. Tieto skripty vymenia a prepíšu iba osobné čísla.

Potom treba vybrať iba isté stĺpce z hodnotenia použitím *cat predmet_info_cenz.txt | cut -f1,4,10 -d";" > student_predmet_znamka_cenz.txt*.

Bežné spustenie:

```
fenik@labss2:~/parser_v8$ ./parse_znamky.sh -convert _svahnfe.txt
student_info.txt predmet_info.txt

fenik@labss2:~/parser_v8$ ./cenzure_student.sh > student_info_cenz.txt
```




```
fenik@labss2:~/parser_v8$ ./cenzure_predmety.sh > predmet_info_cenz.txt
fenik@labss2:~/parser_v8$ cat predmet_info_cenz.txt |cut -f1,4,10 -d";" >
student_predmet_znamka_cenz.txt
```

2. **Vytáženie predmetov a odborov** použitím skriptu *parse_predmety_odbory.sh*. Vstupné súbory sú zo systému ŠTUDENT.

```
fenik@labss2:~/parser_v8$ ./parse_predmety_odbory.sh

Usage ./parse_predmety_odbory.sh [-convert|-noconvert] [-parse_student|-
no_parse_student] input_file odbor_file predmet_file predmet_info_file

predmet_info.txt file must exist, must contain student, predmet and znamky
and should come from ./parse_znamky.sh!

Pri pouziti prepinaca -parse_student odparsuje aj predmety zo suboru
pochadzajuceho so Studenta, kde sa daju zistit rozne predmety
```

V prípade, že sú k dispozícii dva súbory s predmetmi pre bakalárske a inžinierske štúdium, je dobré ich zjednotiť pomocou *cat _svcspbc.txt _svcspin.txt > _svcsp_all.txt*.

Prepínač *-convert* a *-noconvert* majú podobný účel ako v bode **Error! Reference source not found.** Prepínač *-parse_student* spôsobí, že skript vyhľadá predmety aj v súbore s výsledkami, keďže najnovší zoznam predmetov neobsahoval predmety pred vznikom fakulty. Skript tieto predmety zoradí a spraví jednoznačnými, tzn. bez duplicit. V prípade, že sa nájdu dva predmety s rovnakým číslom a takmer podobným názvom, napr. pridaná pomlčka, treba názvy ručne zjednotiť. Predtým ako sa použije prepínač *-parse_student*, musí byť úspešne spustený skript *parse_znamky.sh* a výstupný súbor z neho musí byť *predmet_info.txt*.

Bežné spustenie:

```
fenik@labss2:~/parser_v8$ cat _svcspbc.txt _svcspin.txt > _svcsp_all.txt
fenik@labss2:~/parser_v8$ ./parse_predmety_odbory.sh -convert -
parse_student _svcsp_all.txt odbor_full.txt predmet_full.txt
predmet_info.txt
```

3. Import dát do databázy

Import použitím MS ACCESS v rámci tabuliek prototypu produkovaného systému je obdobný aj pre iné databázy. *File -> get external data*, súbor *student_info_cenz.txt*. Označiť *Delimited*, v *Advanced* nastaviť kódovanie *utf-8*. Ďalej zaznačiť *First row contains field names*, importovať do existujúcej tabuľky *student*.



Obdobne import súboru *student_predmet_znamka_cenz.txt* do tabuľky *Student_predmet_znamka_cenz*.

Obdobne import súboru *predmet_full.txt* do tabuľky *predmet*.

Obdobne import súboru *odbor_full.txt* do tabuľky *Odbor_full*

4. Spustenie pomocných selektov

Nakoniec treba spustiť tento selekt, ktorý správne naplní tabuľku *znamka*, napríklad dopísaním *INSERT INTO znamka* pred *SELECT* do sqlview v query *Student_predmet_znamka_cenz*
Query:

```
INSERT INTO znamka SELECT predmet.id_predmet AS id_predmet,  
student.id_student AS id_student, uroven.id_uroven AS id_uroven  
  
FROM (student INNER JOIN (predmet INNER JOIN Student_predmet_znamka_cenz ON  
predmet.id_predmet=Student_predmet_znamka_cenz.predmet_druhe_cislo) ON  
student.id_student=Student_predmet_znamka_cenz.os_cislo) INNER JOIN uroven  
ON Student_predmet_znamka_cenz.znamka=uroven.nazov;
```

3.3.3 Riešenie prípadných problémov

Pri zlom zobrazení znakov je potrebné nastaviť encoding v putty na utf-8 nasledovne: *change settings -> window -> translation*.

V prípade, že sa nájdu dva predmety s rovnakým číslom a takmer podobným názvom, je potrebná ručná oprava .

Pri importe môže nastať *violation*, potrebné vymazať riadky, ktoré obsahujú kolidujúce dáta.

Všetky zdrojové texty a súbory potrebné k úspešnému spusteniu aplikácií prototypu sa nachádzajú v prílohe D – Elektronický nosič – CD-ROM.



Zhodnotenie prototypu

Prototyp systémy ZNALOSTI sa skladá z troch častí: databáza, webové služby a parser. Databáza je prototypom na zahodenie, pomocou ktorého sme overovali správnosť návrhu. Podarilo sa nám na niekoľkých miestach zovšeobecniť model údajov, najmä umožnením používania rôznych klasifikačných stupníc pre predmety. Ďalšou dôležitou úlohou bolo overiť mechanizmus balíčkov a algoritmus na zaradovanie študentov do tried, ktorý používa smerodajnú odchýlku. Toto sa tiež podarilo splniť, i keď drobné modifikácie algoritmu do budúcnosti nevylučujeme.

Oboznámenie sa s implementáciou webových služieb počas prototypovania prinieslo niekoľko výhod. Okrem praktického vyskúšania vývojového prostredia Eclipse, jazyka Java a balíka AXIS, sa podarilo pochopiť spôsob implementácie webových služieb a preklenúť „úsalia“ začiatku práce s touto technológiou.

Prototyp parseru je dôležitý z hľadiska zaistenia kompatibility pri importovaní údajov od iných systémov. Už v tejto fáze je parser otestovaný na reálnych údajoch zo systému *Študent*. Keďže spĺňa všetky požiadavky na vysokej úrovni počítame s jeho použitím vo výslednom systéme.

POUŽITÁ LITERATÚRA

- [1] Túlavý Tuleň: Ponuka,
<http://www2.dcs.elf.stuba.sk/TeamProject/2003/team01/dokumenty/ponuka.doc>, október 2003
- [2] Mylbachrová D., Sýkora S. 1996. Informačná podpora pre kreditný systém štúdia na VŠ - príspevok k UNINFOS '96 - Univerzitné informačné systémy, dostupné na internete
http://www.fem.uniag.sk/konferencie_a_seminare/uninfos/1996/prispevky/vozarik2.htm
- [3] Horda: Podpora plánovania štúdia – Projektová dokumentácia, 2004, Dokumentácia k softvérovému systému, finálna verzia,
http://www2.dcs.elf.stuba.sk/TeamProject/2003/team03/horda.php?page_id=5
- [4] Bieliková M.: Softvérové inžinierstvo. Princípy a manažment. Bratislava: Vydavateľstvo STU, 2000. 220 s. ISBN 80-227-1322-8.
- [5] Kornblith, G.: How do I organize Data into Categories? [online]. American Social History Productions, Inc., 2005. [cit. 2005-12-03]. Dostupné na Internete:
<http://historymatters.gmu.edu/mse/numbers/question6.html>
- [6] Campbell, M.: Statistics at Square One. [online]. BMJ Publishing Group Ltd., 2005. [cit. 2005-12-03]. Dostupné na Internete: <http://bmj.bmjournals.com/collections/statsbk/2.shtml>



DOKUMENTÁCIA K RIADENIU PROJEKTU

ZIMNÝ SEMESTER

Báza znalostí a zručností študentov

I ÚVOD

1 ÚVOD.....	1
1.1 ÚČEL DOKUMENTU.....	1
1.2 PREHEAD DOKUMENTU.....	1

II ÚLOHY ČLENŮ TÍMU

1 TÍM _ELF_	1
1.1 VÝBER TÉMY	1
1.2 VÝBER NÁZVU PRE TÍM.....	1
1.3 ROZDELENIE ÚLOH.....	1
1.4 AUTORSTVO ČASTÍ DOKUMENTÁCIE.....	2

III ŠTANDARDY KÓDOVANIA

1 POUŽITÉ ŠTANDARDY	1
2 MANAŽMENT VERZIÍ DOKUMENTOV	1
3 CONCURRENT VERSION SYSTEM	2
3.1 KLIENTI	2
3.2 POUŽÍVANIE	2

IV PONUKA

1 BÁZA ZNALOSTÍ A ZRUČNOSTÍ ŠTUDENTOV - PONUKA.....	1
---	---

V PLÁN PROJEKTU

1 ČASOVÝ PLÁN PROJEKTU	1
1.1 PRIEBEŽNÉ ÚLOHY	1
1.2 PODROBNÝ PLÁN NA JEDNOTLIVÉ TÝŽDNE	1
1.3 HRUBÝ PLÁN PROJEKTU.....	3

VI ZÁPISNICE

ZÁPIS Č. 1 ZO STRETNUTIA TÍMU	1
ZÁPIS Č. 2 ZO STRETNUTIA TÍMU	4
ZÁPIS Č. 3 ZO STRETNUTIA TÍMU	10
ZÁPIS Č. 4 ZO STRETNUTIA TÍMU	16
ZÁPIS Č. 5 ZO STRETNUTIA TÍMU	20
ZÁPIS Č. 6 ZO STRETNUTIA TÍMU	24
ZÁPIS Č. 7 ZO STRETNUTIA TÍMU	28
ZÁPIS Č. 8 ZO STRETNUTIA TÍMU	32

VII PREBERACIE PROTOKOLY

PREBERACÍ PROTOKOL K DOKUMENTÁCI Č. 1.....	1
PREBERACÍ PROTOKOL K DOKUMENTÁCI Č. 2.....	2
PREBERACÍ PROTOKOL K DOKUMENTÁCI Č. 3.....	3
PREBERACÍ PROTOKOL K DOKUMENTÁCI Č. 4.....	4



I ÚVOD

„Schádzať sa spolu je začiatok.

Udržať sa spolu je pokrok.

Pracovať spolu je úspech.“

Henry Ford

1 Úvod

1.1 Účel dokumentu

Kvalitná dokumentácia je dôležitá časť manažmentu pracovných procesov. Predkladaná dokumentácia riadenia je akýmsi prehľadom činnosti tímu, obsahuje dokumenty, ktoré postupne vznikajú v čase riešenia projektu s cieľom efektívneho riadenia tímovej práce. Slúži na komunikáciu v rámci tímu ako aj s vonkajším prostredím, poskytuje prehľad vykonávaných činností rozdelených medzi jednotlivých členov tímu. Dokumentácia všetkých procesov v projekte vyžaduje komunikáciu v rámci tímu, preto je výzvou pre náš tím, aby sa dokumentácia udržiavala aktuálna, a aby bola k dispozícii všetkým zainteresovaným osobám. Dokumentácia je tiež dôležitá pre každý projekt kvality, napr. pri zavádzaní ISO 9000/14000.

1.2 Prehľad dokumentu

Dokumentácia k riadeniu je členená na viacero kapitol. V druhej kapitole predstavíme tím, pôvod vzniku mena a rozdelenie úloh členov tímu. V tretej kapitole uvedieme používané štandardy kódovania a manažment verzií. V ďalšej sa odkážeme na ponuku, ktorá bola vypracovaná na tému *Báza znalostí a zručností študentov*. Na základe nej sme tento projekt aj získali. V piatej kapitole je uvedený hrubý i podrobný plán projektu. V šiestej kapitole sú zahrnuté zápisy zo stretnutí tímu v chronologickom poradí a následne v ďalšej kapitole sú preberacie protokoly.



II ÚLOHY ČLENOV TÍMU

V nasledujúcej kapitole sú zhrnuté dôležité informácie o tíme, úlohy jednotlivých členov tímu a podiel na spolupráci pri tvorbe dokumentácie k projektu.

1 Tím *_elf_*

1.1 Výber témy

Ako už bolo spomenuté v ponuke, výber témy nášho tímu bol jednoznačný. Rozhodnutie pre výber témy *Báza znalostí a zručností študentov* padlo najmä vďaka jej dostatočnému potenciálu rozvinúť doterajšie znalosti v odbore štúdia všetkých členov tímu. Je pre nás výzvou vytvoriť produkt, ktorý bude slúžiť potrebám fakulty, ale zároveň bude použiteľný aj v komerčnej sfére.

Tím je založený na pevných základoch, jeho členovia sa dobre poznajú a sú si vedomí svojich schopností, pretože už v minulosti spolupracovali na rôznych školských aj mimoškolských aktivitách.

1.2 Výber názvu pre tím

elf ako meno tímu vzniklo po dohode. Iné námety boli taktiež spojené s číslom 11 a jeho jazykovými mutáciami. Nakoniec však všetci súhlasili s týmto názvom. Členovia tímu potom navrhovali logá, ktoré sa zhodnotili a vybral sa najlepší námet. Námet v logu predstavuje číslo 11, ale dá sa naň pozerat' aj ako na postavy dvoch ľudí, čo má evokovat' tímovú prácu.

1.3 Rozdelenie úloh

V štvrtom týždni zimného semestra boli rozdelené úlohy medzi členov tímu, ktoré vyplynuli už počas tvorby ponuky.

- Vedúci tímu: Juraj Malečka
- Správca webstránky: Marián Miština
- Dokumentarista: Martina Práznovská

Ďalšie roly ako analytik, návrhár a pod. neboli pridelené jednotlivcom, ale sú zdieľané viacerými členmi tímu.

1.4 Autorstvo častí dokumentácie a podiel práce na projekte

Na vytvorení jednotlivých kapitol dokumentácie vždy spolupracovali všetci členovia tímu. Ich percentuálny podiel vyjadruje tabuľka II - 1. Použité sú iniciály mien.

Podiel členov tímu na tvorbe dokumentácie (%)								
	Kapitola	AF	JM	MK	MM	MP	MS	SC
1.	Ponuka	10	20	20	10	10	10	20
2.	Úvod		100					
3.	Slovník pojmov	34			33	33		
4.	Analýza	25	6	17	6	6	15	25
5.	Špecifikácia	6	15	15	23	18	18	5
6.	Hrubý návrh	5	15	20	15	15	15	15
7.	Prototyp	15	15	22	17	8	8	15

Tabuľka II – 1: Podiel členov tímu na tvorbe dokumentácie v %

Okrem úloh spojených s tvorbou dokumentácie mali jednotliví členovia tímu aj iné úlohy, napr. tvorba zápisníc, tvorba webu, tvorba plánov projektu, integrácia dokumentácie a pod. Tabuľka II - 1 preto nevyjadruje skutočný podiel práce jednotlivých členov tímu na projekte. Celkový prehľad podielu práce členov tímu v zimnom semestri 2005 je zobrazený v tabuľke II – 2.

Celkový podiel práce členov tímu na projekte ZNALOSTI za zimný semester 2005 (%)							
AF	JM	MK	MM	MP	MS	SC	
10	17	20	15	10	13	15	

Tabuľka II – 2: Celkový podiel práce členov tímu na projekte ZNALOSTI za zimný semester 2005



III ŠTANDARDY KÓDOVANIA

Pri tvorbe projektu v tíme je podstatné používanie štandardov a manažmentu verzií, tejto problematike sa venujú nasledovné kapitoly.

1 Použité štandardy

Na stránke <http://geosoft.no/development/javastyle.html#introduction1> sa nachádza podrobný opis štandardov, ktoré sa používajú pri programovaní v jazyku Java, vo forme prehľadného rýchlokurzu s množstvom užitočných príkladov vychádzajúcich z praxe. Týchto štandardov sa budeme pridŕžať pri implementácii. Odkaz je umiestnený na web-stránke tímu, aby bol v prípade potreby jednoducho prístupný všetkým členom.

2 Manažment verzií dokumentov

Procesy použité v manažmente verzií sa delia na správu dokumentov a správu zdrojového kódu.

Spočiatku sa na uchovávanie dokumentov používal „online project management“ systém NetOffice. Po získaní prístupu na školský server *labs2*, sa všetky dokumenty uchovávajú v adresári tímu, v podadresári *dokumentácia*. Dokumenty sa ukladajú vždy do adresára označeného dátumom ďalšieho kontrolného bodu odovzdania dokumentácie, pričom dokumenty spojené s riadením projektu sú oddelené od dokumentov spojených s informačným systémom. Pre označovanie verzií dokumentov bola vytvorená metodika, ktorá stanovuje názov súboru a jeho umiestnenie. Kvôli kompatibilite všetci členovia tímu na tvorbu dokumentov používajú balík MS Office (MS Word) so zapnutým sledovaním zmien. Informovanie ostatných členov tímu o novej verzii dokumentov prebieha prostredníctvom internetovej diskusnej skupiny, kde sa ku každej správe o novej verzii dokumentu tento dokument pripojí.



3 Concurrent Version System

Na správu zdrojových kódov a súborov nášho projektu používame od 8. týždňa zimného semestra softvér Concurrent Version System (CVS).

3.1 KLIENTI

Najznámejšie klientské programy pre systém CVS pod operačným systémom windows sú `wincvs` (<http://www.wincvs.org/download.html>) a `tortoise cvs` (<http://www.tortoise cvs.org/download.shtml>).

Ďalšia možnosť je použiť `cvs` z príkazového riadku. Väčšina vývojových prostredí podporuje `cvs`, vrátane Eclipse, ktorý budeme používať pri implementácii projektu.

3.2 POUŽÍVANIE

`Wincvs` má svoje vlastné grafické používateľské rozhranie. `Tortoise cvs` je však používateľsky prívetivejší a preto sme sa ho rozhodli používať. Po inštalácii sa automaticky pridá do kontextového menu a je možné ho používať priamo z explorera pravým kliknutím myšou v ľubovoľnom adresári.

Keď chceme stiahnuť prvý krát projekt zo servera, tak klikneme pravým tlačidlom v adresári, kde chceme vytvoriť projekt a zvolíme **checkout**. Zobrazí sa obrazovka s parametrami pripojenia, ktorú treba vyplniť nasledovne:

- Protocol: `ext`
- Server: `labss2.dcs.elf.stuba.sk`
- Repository folder: `/home/users/team11/team11/cvsroot`
- Username: `username_na_labss2`
- Module: zadáme názov_modulu alebo vyberieme možnosť `fetch list` a on by mal vylistovať existujúce moduly (projekty).
- Port: nie je nutné zadávať

Po správnom vyplnení sa vytvorí v uvedenom adresári nový projekt a prenesú sa doň všetky súbory aj s podadresármi zo servera.

Keď chceme pridať nový súbor, tak ho nakopírujeme do adresára, v ktorom je stiahnutý projekt. Umiestnime ho do príslušného podadresára a pravým tlačidlom zvolíme **CVS ADD**. Súbor sa pridá do projektu. Potom ešte treba zvoliť **CVS Commit** a ten ho následne uloží na server. Môžeme



„commitovať“ aj celý adresár. Keď chceme zmazať súbor, tak použijeme **CVS remove**. Keď si chceme stiahnuť najnovšiu verziu projektu, tak použijeme **CVS update**.

Pri commitovaní súboru môžeme **pridať komentár**, kde napíšeme, čo sme zmenili v súbore. Po každej zmene súboru a „commitnutí“ sa vytvorí nová verzia súboru. V **histórii** je uvedené kto, čo a kedy pridal.

Keď dvaja používatelia súčasne editujú ten istý súbor, prvý z nich ho „commitne“ bez problémov. Druhému vyhlási chybu, že súbor je „out of date“, keďže medzitým prišla novšia verzia. Vtedy treba stiahnuť novú verziu (**update**), pridať do nej zmeny, ktoré sme urobili a dať znovu commit.



IV PONUKA

1 Báza znalostí a zručností študentov - Ponuka

Kvôli celistvosti dokumentu je ponuka k projektu *Báza znalostí a zručností študentov* vložená v Prílohe C. Uložená je v pôvodnom znení ako bola odovzdaná na posúdenie. Celkovo bola ponuka ohodnotená známkou výborne, pričom boli vyzdvihnuté zaujímavé nápady (časť 4.1.1.1) a dobre premyslený návrh.



V PLÁN PROJEKTU

1 Časový plán projektu

1.1 Priebežné úlohy

- Tvorba projektového denníka
- Pripomienkovanie dokumentácie systému a dokumentov k riadeniu projektu
- Príprava na pravidelné stretnutia v softvérovom štúdiu (čím som sa zaoberal/a, prezentácia riešení, otázky)
- Štúdium technológií, ktoré budú použité vo fáze implementácie
- Sledovanie konkurenčných tímov
- Aktualizovanie webstránky

1.2 Podrobný plán na jednotlivé týždne

Termíny pre oficiálne odovzdávania častí – kontrolné body sú zvýraznené **červeným** písmom, interné termíny – stanovené v rámci tímu sú zvýraznené **sivozeleným** písmom.

4. týždeň: 17. 10. 2005 - 23. 10. 2005

- Pridelenie projektu
- Vytvorenie organizačnej štruktúry tímu
- Začiatok pravidelných stretnutí v softvérovom štúdiu
- Začiatok fázy analýzy
- Analýza pripomienok z hodnotenia ponuky
- Štúdium existujúcich systémov s podobným poslaním

5. týždeň: 24. 10. 2005 - 30. 10. 2005

- Špecifikácia požiadaviek na systém, funkcie systému (brainstorming)
- Rozdelenie úloh v rámci analýzy kontextu systému
 - Analýza potrebných technológií a technológií dostupných s softvérovom štúdiu
 - Analýza okolia systému
 - Analýza podobných systémov
- Vytvorenie plánu projektu
- Návrh šablóny dokumentov (dokumentácia a zápisy zo stretnutí)

6. týždeň: 31. 10. 2005 - 06. 11. 2005



- Ujasnenie špecifikácie (vstupy, výstupy, bezpečnosť, spolupráca s okolím, ...)
- Stanovenie priorít pre návrh a implementáciu
- Špecifikácia údajov v systéme, špecifikácia správania systému (brainstorming)
- Rozdelenie úloh v rámci špecifikácie funkcií, údajov a správania sa systému
- 02. 11. 2005 (streda) – Odovzdanie čiastkových analýz dokumentaristovi
- Korekcia šablón dokumentov
- Pripomienkovanie a zverejnenie plánu projektu

7. týždeň: 07. 11. 2005 - 13. 11. 2005

- 09. 11. 2005 (streda) – Odovzdanie čiastkových špecifikácií dokumentaristovi
- Hrubý návrh riešenia (brainstorming)
- Identifikácia dodatočných požiadaviek a ohraničení (brainstorming)
- Zvolenie metódy implementácie prototypu
- Vytvorenie pravidiel písania zdrojového kódu

8. týždeň: 14. 11. 2005 - 20. 11. 2005

- 15. 11. 2005 (utorok) – Odovzdanie hrubého návrhu dokumentaristovi
- Korekcia a kompletizácia fázy analýzy
- Vypracovanie preberacieho protokolu
- 18. 11. 2005 (piatok), 14:00 – odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek riešenia spolu s hrubým návrhom
- Zabezpečenie potrebných technológií na implementáciu prototypu

9. týždeň: 21. 11. 2005 - 27. 11. 2005

- Štúdium dokumentácie iného tímu
- Identifikácia slabých a silných stránok riešenia iného tímu (brainstorming)
- Vypracovanie posudku analýzy, špecifikácie a hrubého návrhu iného tímu
- Určenie častí systému, ktoré sa budú prototypovať (brainstorming)
- Návrh prototypu
- Rozdelenie úloh na implementácii prototypu
- Začiatok implementácie prototypu
- 25. 11. 2005 (piatok), 14:00 – odovzdanie posudku analýzy, špecifikácie a hrubého návrhu iného tímu

10. týždeň: 28. 11. 2005 - 04. 12. 2005

- Dopracovanie nedostatkov identifikovaných iným tímom
- Implementácia prototypu
- Návrh testovania prototypu a testovacích údajov

**11. týždeň: 05. 12. 2005 - 11. 12. 2005**

- Implementácia a integrácia a testovanie prototypu
- príprava dokumentácie k prototypu

12. týždeň: 12. 12. 2005 - 18. 12. 2005

- 15. 12. 2005 (štvrtok) - Dokončenie implementácie prototypu integrácia a spustenie prototypu v softvérovom štúdiu
- Testovanie a korekcia prototypu
- Korekcia a kompletizácia dokumentácie
- Vytvorenie používateľskej príručky
- Vytvorenie prezentácie prototypu

13. týždeň: 19. 12. 2005 - 21. 12. 2005

- 19. 12. 2005 (pondelok), 14:00 – odovzdanie prototypu vybraných častí systému spolu s dokumentáciou
- 19. 12. 2005 - 21.12. 2005 (podľa dohody tímov) – používateľská prezentácia prototypu

Úloha na prázdniny

- 02. 02. 2006 (štvrtok), do 14:00 – odovzdanie posudku prototypu iného tímu

1.3 Hrubý plán projektu

Zimný semester	
Týždeň	Predpokladaná činnosť
3 – 4	<ul style="list-style-type: none">▪ odovzdanie a prezentácia ponuky, pridelenie témy zadania, vytvorenie plánu projektu
4 – 8	<ul style="list-style-type: none">▪ analýza problému, špecifikácia požiadaviek a hrubý návrh riešenia▪ vytvorenie dokumentácie analýzy problému, špecifikácie požiadaviek riešenia spolu s hrubým návrhom▪ priebežná tvorba dokumentácie analýzy problému, špecifikácie požiadaviek riešenia spolu s hrubým návrhom
8 – 9	<ul style="list-style-type: none">▪ odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek riešenia spolu s hrubým návrhom▪ vytvorenie posudku analýzy, špecifikácie a hrubého návrhu iného tímu▪ odovzdanie posudku analýzy, špecifikácie a hrubého návrhu iného tímu
9 – 12	<ul style="list-style-type: none">▪ dopracovanie zistených nedostatkov▪ návrh a implementácia prototypu vybraných častí systému▪ vytvorenie dokumentácie a používateľskej prezentácie prototypu
12	<ul style="list-style-type: none">▪ odovzdanie prototypu vybraných častí systému spolu s dokumentáciou a používateľská prezentácia prototypu



Letný semester	
Týždeň	Predpokladaná činnosť
1	<ul style="list-style-type: none">▪ odovzdanie posudku prototypu iného tímu,▪ zhodnotenie výsledkov ZS, doplnenie a dopracovanie zistených nedostatkov,
2 – 3	<ul style="list-style-type: none">▪ zakomponovanie zmien do dokumentácie ZS, podrobný návrh, plán integrácie, plán overenia výsledku
3	<ul style="list-style-type: none">▪ dokončenie podrobného návrhu, implementácia
4	<ul style="list-style-type: none">▪ implementácia, postupná integrácia a overovanie výsledku, tvorba dokumentácie
5	<ul style="list-style-type: none">▪ implementácia, postupná integrácia a overovanie výsledku, tvorba dokumentácie
6	<ul style="list-style-type: none">▪ implementácia, postupná integrácia a overovanie výsledku, tvorba dokumentácie
7	<ul style="list-style-type: none">▪ integrácia produktu a overovanie, tvorba dokumentácie k produktu
8	<ul style="list-style-type: none">▪ integrácia produktu a overovanie, tvorba dokumentácie k produktu
9	<ul style="list-style-type: none">▪ integrácia produktu a overovanie, tvorba dokumentácie k produktu
10	<ul style="list-style-type: none">▪ odovzdanie produktu▪ a dokumentácie k produktu (potrebnej pre používanie produktu)
11	<ul style="list-style-type: none">▪ používanie produktu, údržba, kompletizácia dokumentácie
12	<ul style="list-style-type: none">▪ odovzdanie celkového výsledku projektu (produkt so zmenami v rámci údržby, dokumentácia)

VI ZÁPISNICE

ZÁPIS Č. 1 ZO STRETNUTIA TÍMU	1
ZÁPIS Č. 2 ZO STRETNUTIA TÍMU	4
ZÁPIS Č. 3 ZO STRETNUTIA TÍMU	10
ZÁPIS Č. 4 ZO STRETNUTIA TÍMU	16
ZÁPIS Č. 5 ZO STRETNUTIA TÍMU	20
ZÁPIS Č. 6 ZO STRETNUTIA TÍMU	24
ZÁPIS Č. 7 ZO STRETNUTIA TÍMU	28
ZÁPIS Č. 8 ZO STRETNUTIA TÍMU	32



VI ZÁPISNICE

Zápis č. 1 zo stretnutia tímu

elf



Dátum:	27.10.2005 10:10 – 12:30	Miesto:	Softvérové štúdio
Vedúci tímu:	Ing. Ivan Kapustík	Host':	Ing. Vladimír Grlický
Prítomní:	Bc. Slavomír Červeň (SC), Bc. Andrej Fenik (AF), Bc. Martin Kováčik (MK), Bc. Juraj Malečka (JM), Bc. Marián Miština (MM), Bc. Martina Práznovská (MP), Bc. Michal Sabo (MS)		
Vypracoval:	Martina Práznovská		

Téma stretnutia

Diskusia ohľadom špecifikácie s Ing. Grlickým, rozdelenie úloh

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí:

Stretnutie vo štvrtok 20.10.2005 sa nekonalo. P. Ing. Kapustík bol služobne odcestovaný, úlohy však určil e-mailom.

ID	Zodpovedný	Popis úlohy	Termín zadania	Termín ukončenia	Stav úlohy
1.1	Každý člen tímu	Preštudovať danú problematiku - existujúce podobné riešenia	19/10/05	27/10/05	Riešená
1.2	Celý tím	Rozdelenie rolí v tíme	19/10/05	27/10/05	Ukončená

Rozdelenie úloh v tíme:

Vedúci tímu: Juraj Malečka
Správca webu: Marián Miština
Dokumentácia: Martina Práznovská

Ostatné úlohy sa budú rozdeľovať postupne na stretnutiach podľa potreby.

Andrej Fenik naštudoval výslednú dokumentáciu systému Yonban. Slavomír Červeň má skúsenosti so systémom, ktorý pracuje s osobnými údajmi na podobnom princípe ako uvažujeme riešiť systém ZNALOSTI.

Opis stretnutia:

1. Odporúčania Ing. Grlického na analýzu a špecifikáciu systému:

- pripraviť si úlohy, na ktoré sa chceme zamerať a následne ich zoradiť podľa priority
- už pri návrhu treba myslieť na rozhranie (prístupy k iným systémom)
- pri špecifikácii môžeme vychádzať z ponuky a pridávať nové myšlienky

2. Prebiehala **diskusia o vstupoch** do systému. Ako vstupy sme spomínali systém ŠTUDENT, hodnotenie študenta samého, hodnotenie študenta iným študentom, hodnotenie vyučujúcich resp. cvičiacich; prístupy napr. webový formulár. Záchytné body z diskusie:

- 1 VG: študent si bude vyplňať zručnosti pravdivo vo vlastnom záujme, keďže budú dostupné aj potenciálnym zamestnávateľom, ktorí si ich môžu overiť. Firmy sa nebudú registrovať za účelom pridávania komentárov k zručnostiam študentov.
- 2 VG: je potrebná pozitívna motivácia študentov, aby vkladali pravdivé informácie do systému. Treba hľadať možnosti a spôsoby ako to zakomponovať. Študent môže zvoliť, či chce, aby boli jeho „znalosti“ zverejnené na internete – napr. pre firmy
- 3 JM: študent by mohol zadávať referencie o sebe a dokladovať ich prostredníctvom odkazu prípadne v prílohe resp. by ho mohol doručiť na požiadanie. Odkazy na projekty v danej zručnosti.
- 4 IK: Zdrojom overenia zručnosti bude hlavne práca na bak. a dipl. projektoch, "ak študent vypracoval bakalársky projekt v Jave, tak ju ovláda".
- 5 SC: K dátam zo systému ŠTUDENT môžeme pristupovať nasledovne. Získavanie dát real-time alebo dávkovo t.j. každý semester, keďže sa známky nemenia počas semestra AF: Schodnejšia cesta z dôvodu ochrany systému by bola bola extrakcia dát dávkovo zo systému ŠTUDENT každý semester a import do nášho systému.
- 6 IK: import dát zo systému ŠTUDENT by mal byť inteligentný a nastavovateľný, aby sa integrita dát nepoškodila (zabránenie zdvojeniu a vymazávaniu dát) + pozor na zmeny v systéme ŠTUDENT
- 7 VG: Z hľadiska bezpečnosti nebude BŠP prístupné priamo, ale pomocou usporiadania podľa študentov podľa BŠP. Usporiadanie podľa BŠP je už menej citlivé na ochranu študijných výsledkov. IK: ochrana osobných údajov aspoň sťažením prístupu.
- 8 VG: Zákony sa menia, treba teda klásť dôraz na konfigurovateľnosť. Preto treba mať aj surové dáta, aj zahmlené. Niektoré informácie vieme zoradiť a ohodnotiť, iné nie – napr. textový vstup. Index výsledného hodnotenia je veľmi dobrá myšlienka, bolo by užitočné ju implementovať
- 9 Ďalší vstup do systému bude hodnotenie vyučujúcich – resp. cvičiacich, ktorí prichádzajú do kontaktu so študentom
- 10 IK: používať systém donútime užívateľa, ak to bude na jedno alebo čo najmenší počet kliknutí. Cvičiaci môžu, nemusia vyplňovať hodnotenie, ohodnotia najlepších a najhorších študentov – ostatní sú priemerní.
- 11 VG: bolo by zaujímavé hodnotiť aj účasť študentov na cvičení.
- 12 Vstup do systému cez prepojenie so systémom Yonban, napr. získanie posudkov k bakalárskej práci

3. Výstupy systému:

- 13 JM: výstup zo systému môže byť vlastne všetko, čo systém obsahuje, tento výstup je konfigurovateľný. Definujeme vlastnosti, definujeme okruhy predmetov. Užívateľ môže mať svoj osobný filter najpoužívanejších funkcií (personalizácia)
- 14 MK: používateľ by mal byť informovaný čo systém dokáže, čo môže od neho očakávať, aké informácie môže získať



- 15 VG: je potrebné zamerať sa na roly - t.j. kto, aké dáta môže vidieť - do návrhu zahrnúť maticu autorizácie.
- 16 Pre systém Yonban poskytovať informácie(vstup do systému môže byť ID študentov a výstup zo systému ZNALOSTI bude napr. ich zoradenie) toto sa poskytne vyučujúcemu, ktorý sa podľa zoradenia môže rozhodnúť o pridelení projektu študentovi.
- 17 Treba vyriešiť otázku akou formou bude používateľ pristupovať k systému ZNALOSTI (webová služba, aplikácia) MK: navrhol výstup ako webservises pre iné systémy a tiež prezentačný grafický interfejs pre užívateľa.

4. Jadro systému:

- 18 MM: navrhol riešenie pomocou vysoko abstraktného, všeobecného a univerzálneho systému. VG: namietal, lebo takéto riešenie je časovo náročné na implementáciu IK: špecializované riešenie je názornejšie.
- 19 Jadro systému konfiguruje iba administrátor
- 20 VG: pri definovaní špecifikácie sa popíšu funkcionality systému, následne ich treba usporiadať podľa priority. Nie všetky funkcionality bude potrebné implementovať alebo prototypovať t.j. zameriame sa na vybranú oblasť a tú implementujeme.

Úlohy pre členov tímu:

ID	Zodpovedný	Popis úlohy	Termín ukončenia
2.1	Slavomír Červeň	<ul style="list-style-type: none">Popísať naštudovanú problematiku – systém, s ktorým má skúsenosti.Dokument posluží ako príspevok do analýzy.Popísať vlastnosti systému, jeho výhody, nevýhody, ako bol realizovaný vývoj, čo je možné prevziať prípadne použiť pri riešení systému ZNALOSTI.	03. 10. 2005
2.2	Andrej Fenik	<ul style="list-style-type: none">Popísať naštudovanú problematiku - systém Yonban – budúci príspevok do analýzy.Popísať vlastnosti systému, jeho výhody, nevýhody, čo je možné prevziať prípadne použiť pri riešení systému ZNALOSTI prípadne ako by bolo možné systémy prepojiť.	03. 10. 2005
2.3	Michal Sabo	<ul style="list-style-type: none">Popísať vlastnosti jednotlivých prostredí a technológií (do analýzy).	03. 10. 2005
2.4	Martin Kováčik	<ul style="list-style-type: none">Zistiť čo môže byť nainštalované v softvérovom štúdiu a dohodnúť prídanie komponentov, ktoré potrebujeme na vývoj s p. Lackom.Popísať alternatívy prostredí (také, ktoré by mohli byť, ale nie sú k dispozícii v softvérovom štúdiu).	03. 10. 2005
2.5	Juraj Malečka	<ul style="list-style-type: none">Vypracovať podrobný plán činnosti tímu na zimný semester. Rozpísať úlohy a termíny.	03. 10. 2005
2.6	Marián Miština Martina Práznovská	<ul style="list-style-type: none">Pripraviť dokument návrhu špecifikácie, rozpísať body, o ktorých sa diskutovalo s Ing. Grlickým.	03. 10. 2005
2.7	Martina Práznovská	<ul style="list-style-type: none">Pripraviť šablónu dokumentácie.	03. 10. 2005

**Zápis č. 2 zo stretnutia tímu****_elf_**

Dátum:	3.11.2005 10:10 – 12:30	Miesto:	<i>Softvérové štúdio</i>
Vedúci tímu:	Ing. Ivan Kapustík		
Prítomní:	<i>Bc. Slavomír Červeň (SC), Bc. Andrej Fenik (AF), Bc. Martin Kováčik (MK), Bc. Juraj Malečka (JM), Bc. Marián Miština (MM), Bc. Martina Práznovská (MP), Bc. Michal Sabo (MS)</i>		
Vypracoval:	<i>Marián Miština</i>		

Téma stretnutia

Kontrola splnenia úloh, diskusia a komentáre k ich vypracovaniu, pridelenie nových úloh.

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí:

Tabuľka uvádza stav riešenia úloh zadaných počas stretnutia dňa 27.10.2005 a skôr. Staršie ukončené úlohy nie sú v tabuľke zahrnuté.

ID	Zodpovedný	Popis úlohy	Termín ukončenia	Stav úlohy
2.1	Každý člen tímu	<ul style="list-style-type: none">• Preštudovať danú problematiku - existujúce podobné riešenia	03. 10. 2005	Ukončená
2.1	Slavomír Červeň	<ul style="list-style-type: none">• Popísať naštudovanú problematiku – systém, s ktorým má skúsenosti.• Dokument poslúži ako príspevok do analýzy.• Popísať vlastnosti systému, jeho výhody, nevýhody, ako bol realizovaný vývoj, čo je možné prevziať prípadne použiť pri riešení systému ZNALOSTI.	03. 10. 2005	Ukončená



2.2	Andrej Fenik	<ul style="list-style-type: none">• Popísať naštudovanú problematiku - systém Yonban – budúci príspevok do analýzy.• Popísať vlastnosti systému, jeho výhody, nevýhody, čo je možné prevziať prípadne použiť pri riešení systému ZNALOSTI prípadne ako by bolo možné systémy prepojiť.	03. 10. 2005	Ukončená
2.3	Michal Sabo	<ul style="list-style-type: none">• Popísať vlastnosti jednotlivých prostredí a technológií (do analýzy).	03. 10. 2005	Ukončená
2.4	Martin Kováčik	<ul style="list-style-type: none">• Zistiť čo môže byť nainštalované v softvérovom štúdiu a dohodnúť prídanie komponentov, ktoré potrebujeme na vývoj s p. Lackom.• Popísať alternatívy prostredí (také, ktoré by mohli byť, ale nie sú k dispozícii v softvérovom štúdiu).	03. 10. 2005	Ukončená
2.5	Juraj Malečka	<ul style="list-style-type: none">• Vypracovať podrobný plán činnosti tímu na zimný semester. Rozpísať úlohy a termíny.	03. 10. 2005	Ukončená
2.6	Marián Miština Martina Práznovská	<ul style="list-style-type: none">• Pripraviť dokument návrhu špecifikácie, rozpísať body, o ktorých sa diskutovalo s Ing. Grlickým.	03. 10. 2005	Ukončená
2.7	Martina Práznovská	<ul style="list-style-type: none">• Pripraviť šablónu dokumentácie.	03. 10. 2005	Riešená

Opis stretnutia:

Priebeh stretnutia sa odvíjal najmä od prezentácie splnených úloh a následnej diskusie.

1. Úvodné informácie

- AF: Informoval o zaslaní emailu adresovaného p. Urbaníkovej a p. Bielekovej so žiadosťou o poskytnutie základných informácií o systéme ŠTUDENT.

2. Prezentácia riešenia úlohy 2.1 (zodpovedný: SC)

- SC: Prezentoval analýzu dvoch systémov, s ktorými má skúsenosti. Ide o web systémy.
 - poisťovací systém
 - logistický systém

Všimol si ako a odkiaľ sa čerpajú informácie:

1. zadávanie priamo cez web formulár
2. import zo súboru s ochranou proti duplicitnému vloženiu (IK vyzdvihol prítomnosť takejto ochrany)
3. pomocou samostatného modulu, ktorý sa pripojí k databáze iného systému a vytvorí XML súbor, ktorý je vstupom pre daný systém

Oboznámil so spôsobom výstupu týchto systémov

1. formulár
2. súbory (XML, PSD, ...)

Predstavil definovanie používateľov systému a spôsob autentifikácie. Každý používateľ má priradenú rolu, ktorá sa spája s oprávneniami vykonávať jednotlivé akcie a pristupovať k jednotlivým



objektom.

IK: Existujú v podstate úrovne a to úroveň bezpečnosti a zobrazenia objektov.

SC popísal postup tvorby IS v ich firme. Najprv prebehne špecifikácia požiadaviek, po nej analýza a návrh. Ďalej sa systém rozdelí na komponenty, ktoré sa pridelia tímom na riešenie a implementáciu, pričom sa presne definujú rozhrania.

Predviedol 3 základné vrstvy spomínaných systémov:

1. prezentačná
2. aplikačná
3. dátová

Spomenul, že boli použité podobné nástroje ako v našom prípade.

3. Prezentácia riešenia úlohy 2.2 (zodpovedný: AF)

- AF: Venoval sa štúdiu predchádzajúcich súvisiacich tímových projektov. Predstavil postupne informácie relevantné pre náš systém z jednotlivých projektov:
- **Projekt 1** – nerealizovaný, pretože nemohol priamo zapisovať údaje do systému študent.
 - Znalosti získané z dokumentácie projektu týkajúce sa systému
 - ŠTUDENT - poskytuje informácie o predmetoch
 - poskytuje základné informácie o študentoch
- **Projekt 2** (tvorba OŠP)– AF prezentoval, že dokumentácia obsahuje zmienku o systéme ŠTUDENT:
 - poskytuje študijné výsledky
 - používatelia sú najmä pracovníci PGP a samotní študenti
 - denne sa aktualizuje a zálohuje
 - založený na technológii FoxPro 6.2
- Takisto obsahuje zmienku o systéme EMA
 - databáza informácií o predmetoch
 - informácie ako názov, číslo, počet hodín cvičení,...

AF predstavil spôsob riešenia prístupu k dátam zo ŠTUDENTA – existujú miestne kópie tabuliek ŠTUDENTA. V budúcnosti možnosť rozšírenia pomocou ODBC.

MM: spomenul nemožnosť riešenia pomocou ODBC.

MK: vidí ODBC ako reálnu možnosť.

IK: ťažko zavrhnúť akékoľvek riešenie.

AF: navrhol úlohu zistiť formát súboru exportovaného zo systému ŠTUDENT.

AF: je správne, ak je v systéme zadané, že má študent skúsenosť napríklad s jazykom Java, ale pritom sa ďalej nechce podrobne venovať danej oblasti?

IK: študent sa môže zamerať na jedno, alebo skúšať nové veci, rozširovať si možnosti.

MK: mohol by prezentovať svoje zámery, ako by sa chcel orientovať.



MS: navrhol rozlíšenie dvoch oblastí

1. čo študent vie
2. čo študent chce

MK+JM: uviedli príklad k niektorej technológii

1. ako ju študent ovláda (čo mám za sebou)
2. ako sa o ňu zaujíma (čo by som chcel urobiť)

- **Projekt 3** (YonBan) – AF spomenul zmienku o ambícii systému stať sa v budúcnosti bázou znalostí

IK: YonBan v súčasnosti umožňuje ručné prezeranie všetkých (aj starých) záznamov.

AF: zo systému YonBan by sa mohli použiť textové ako aj číselné dáta. Možnosť použiť kľúčové slová.

Takisto možnosť doplniť do YonBan-u zoznam použitých technológií.

JM: aj v systéme ZNALOSTI by sa dali použiť kľúčové slová na viacerých miestach.

- **Projekt 4** (elektronická prihláška) – AF oboznámil, že v dokumentácii sa spomína systém ŠTUDENT, ktorý sa používa na celej STU. Na získanie dát sa musí použiť funkcia import/export.

IK: navrhol úlohu pre AF: konzultovať systém ŠTUDENT s p. Bielekovou a PGO. Zistiť, či sa bude aj naďalej používať a ako funguje export/import.

- **Projekt 5** (SOPORIAP) – spolupráca s YonBan-om. Zodpovedný: Matej Makula alebo Roman Filkorn – zistiť.

- **Projekt 6** (hodnotenie projektov) – nefunguje

AF: vraj existuje zoznam študentov, ktorí nechodia na cvičenia.

IK: vie len o individuálnom prístupe (pedagóg -> prodekan)

- **Projekt 7** (tvorba rozvrhov) – plán využiť systém ŠTUDENT ako zdroj dát, ale ďalej sa systém nespomína.

MK: odkiaľ potom údaje čerpali?

IK: často bývajú systémy robené komerčne, teda ťažko rozšíriteľné.

4. Prezentácia riešenia úlohy 2.3 a 2.4 (zodpovední: MK, MS)

- MK: bola vôľa pracovať v Jave, použije sa teda SDK, ktorý je nainštalovaný v softvérovom štúdiu (SŠ)
- MK: možné rozšírenia pre prácu s web services:
 1. AXIS od APACHE – treba doinštalovať, vhodné na SOAP, prináša zjednodušenie, **odporučený**
 2. XERCES - XML parser vyžadovaný AXIS, je nainštalovaný
- MK: aplikačný server by mohol byť:
 1. TomCat – nainštalovaný v SŠ, akurát vyhovuje na projekt podobných rozmerov, **odporučený**
 2. JBoss – bolo by treba nainštalovať
 3. SUN application server – bolo by treba nainštalovať



- MK: možné nástroje pre buildovanie:
 1. ANT – je nainštalovaný, veľmi rozšírený, horšia znovupoužiteľnosť
 2. MAVEN – treba nainštalovať, netreba veľa krát prekonfigurovať, **odporučený**
- MK: Čo sa týka vývojových prostredí, je voľba menej jednoznačná:
 1. NetBeans
 2. Eclipse - nainštalovaný, rýchlejší, rozšíriteľný, **odporučené**
- MK: je teda potrebné nainštalovať MAVEN a AXIS.
- SC: aká databáza bude použitá?
- MK: Postgre – je nainštalovaná, vyspelejšia ako MySQL, obsahuje procedury aj triggery
- MS: Oboznámil s možnosťami frameworkov, ktoré by sa dali použiť [SPRING, HiveMind]. Hoci uvedené riešenia ponúkajú lepšiu podporu pre zásuvné moduly a ich výsledkom môže byť menej kódu, neodporúča ich použitie, pretože nemáme dostatočné skúsenosti a spomínané riešenia sa hodia na veľké enterprise aplikácie. , čiže by mohli iba spomaliť náš postup.
- IK: Súhlasil s nepridávaním zbytočnej práce. Prínosom by mohlo byť aj nepoužitie frameworku a poučenie sa z toho.
- MK: Ak je chyba v konfigurácii frameworku, kompilátor ju neodhalí.
- SC: Spomenul, že použitie frameworku môže mať značné výhody. Framework poskytuje už čiastočne hotové riešenia- uľahčenie implementácie.
- JM: Bolo by sa treba poradiť s niekým, kto má skúsenosti v tejto oblasti.
- MS: Použitie frameworku by bolo možno zaujímavé pre budúcnosť, ale asi nie efektívne pre súčasnosť.
- IK: Podotkol, že by sa mali použiť technológie, ktoré by zvládli dostatočne rýchlo naštudovať všetci členovia tímu.
- SC: Prisľúbil, že sa opýta Ing. Glického na výhody a nevýhody spomínaného riešenia.

5. Prezentácia riešenia úlohy 2.5 (zodpovední: JM)

- JM: Oboznámil s plánom projektu na zimný semester.
- Všetci súhlasili. IK navrhol zmenu na presun niektorých úloh z 9 týždňa do 7 a 8 týždňa.
- Bol schválený upravený plán.

6. Prezentácia riešenia úlohy 2.6 (zodpovední: MM a MP)

- MM: Oboznámil s návrhom špecifikácie. Informácie vstupujúce do systému by mali byť:
 - známky
 - znalosti
 - zručnosti
 - hodnotenia
- AF: navrhol pridať certifikáty.
- IK: súhlasil s návrhom. Na potvrdenie zručností by mohol byť odkaz na certifikáty.

- MP: Predstavila navrhované výstupy systému. Tvorí ich buď súhrnná informácia o študentovi, alebo zoznam študentov vyhovujúci kritériám. Pre študenta môže byť výstupom informácia o predmete (napríklad percentuálne rozdelenie známok)
- AF: navrhol zoradiť študentov vo výstupnom zozname podľa známok.
- JM: nesúhlasil, pretože to popiera filozofiu zatriedenia študentov do istých tried. Ďalej navrhol, aby pri výstupe bolo k dispozícii váhovanie jednotlivých informácií a zoradenie podľa výsledného indexu.
- MK: Mala by existovať funkcia, ktorá konfiguruje jednotlivé parametre po jednom.
- MS: dáta by však mali byť jeden XML súbor, ktorý sa vymieňa medzi systémami.
- AF: podotkol, že procesing by nemal prebiehať na strane klienta, iba servera.
- MS: navrhol, aby bol import údajov zo systému ŠTUDENT riešený pomocou klienta, ktorý načíta informácie zo ŠTUDENT-a a prevedie ich na formát podporovaný systémom ZNALOSTI.

7. Úlohy pre členov tímu



ID	Zodpovedný	Popis úlohy	Termín ukončenia
3.1	Slavomír Červeň	<ul style="list-style-type: none">Analyzovať SPRING framework, zistiť výhody a nevýhody jeho použitia (v spolupráci s Ing. Grlickým)	10.11. 2005
3.2	Andrej Fenik	<ul style="list-style-type: none">V spolupráci s tímom 6 (The Lama Team) zistiť informácie o systéme ŠTUDENTInformovať sa (takisto v spolupráci s tímom 6) o funkciách a plánoch so systémom ŠTUDENT (či sa aj naďalej bude používať) u p. Urbaníkovej a p. Bielekovej.	10.11. 2005
3.3	Juraj Malečka	<ul style="list-style-type: none">Napísať kapitolu Úvod a Zhrnutie analýzy do dokumentácie	10.11. 2005
3.4	Juraj Malečka	<ul style="list-style-type: none">Zvoliť metódu implementácie	10.11. 2005
3.5	Martin Kováčik	<ul style="list-style-type: none">Vytvoriť pravidlá písania zdrojového textu	10.11. 2005
3.6	Martin Kováčik Slavomír Červeň	<ul style="list-style-type: none">Na základe analýzy si utvoriť predstavu o možnom hrubom návrhu systému (netreba dokumentovať)	10.11. 2005
3.7	Marián Miština Michal Sabo Martina Práznovská	<ul style="list-style-type: none">Špecifikácia systému	
2.7	Martina Práznovská	<ul style="list-style-type: none">Pripraviť šablónu dokumentácie.	10.11. 2005

**Zápis č. 3 zo stretnutia tímu****_elf_**

Dátum:	<i>10.11.2005 10:10 – 12:30</i>	Miesto:	<i>Softvérové štúdio</i>
Vedúci tímu:	<i>Ing. Ivan Kapustík</i>		
Prítomní:	<i>Bc. Slavomír Červeň (SC), Bc. Andrej Fenik (AF), Bc. Martin Kováčik (MK), Bc. Juraj Malečka (JM), Bc. Marián Miština (MM), Bc. Martina Práznovská (MP), Bc. Michal Sabo (MS)</i>		
Vypracoval:	<i>Juraj Malečka</i>		

Téma stretnutia

Finalizovanie špecifikácie systému, kontrola splnených úloh, rozdelenie úloh na ďalší týždeň.

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí:

Tabuľka uvádza stav riešenia úloh Stretnutie vo štvrtok 20.10.2005 sa nekonalo. P. Ing. Kapustík bol služobne odcestovaný, úlohy však určil e-mailom.

ID	Zodpovedný	Popis úlohy	Termín ukončenia	Stav úlohy
2.7	Martina Práznovská	<ul style="list-style-type: none">• Pripraviť šablónu dokumentácie.	10. 11. 2005	Ukončená
3.1	Slavomír Červeň	<ul style="list-style-type: none">• Analyzovať SPRING framework, zistiť výhody a nevýhody jeho použitia (v spolupráci s Ing. Grlickým)	10. 11. 2005	Ukončená
3.2	Andrej Fenik	<ul style="list-style-type: none">• V spolupráci s tímom 6 (The Lama Team) zistiť informácie o systéme ŠTUDENT• Informovať sa (takisto v spolupráci s tímom 6) o funkciách a plánoch so systémom ŠTUDENT (či sa aj naďalej bude používať) u p. Urbaníkovej a p. Bielekovej	16. 11. 2005	Riešená
3.3	Juraj Malečka	<ul style="list-style-type: none">• Napísať kapitolu Úvod a Zhrnutie analýzy do dokumentácie	16. 11. 2005	Riešená
3.4	Juraj Malečka	<ul style="list-style-type: none">• Zvoliť metódu implementácie	10. 11. 2005	Ukončená



3.5	Martin Kováčik	<ul style="list-style-type: none">Vytvoriť pravidlá písania zdrojového textu	10. 11. 2005	Ukončená
3.6	Martin Kováčik Slavomír Červeň	<ul style="list-style-type: none">Na základe analýzy si utvoriť predstavu o možnom hrubom návrhu systému (netreba dokumentovať)	10. 11. 2005	Ukončená
3.7	Marián Miština Michal Sabo Martina Práznovská	<ul style="list-style-type: none">Špecifikácia systému	16. 11. 2005	Riešená

Opis stretnutia:

1. Informácie o systémoch EMA a ŠTUDENT od Ing. Kapustíka

IK získal od p.Bielekovej nasledovné informácie:

- IK: Systém EMA nie je pod správou FIIT, ale zostal pod správou FEI. FIIT si údaje o predmetoch musí vždy vyžiadať.
- IK: Systém ŠTUDENT bol vytvorený a vyvíja sa v CVS STU (Námestie Slobody) pod vedením prof. Horvátha. Dvaja študenti z FIIT, ktorý sa na tom podieľali sú teraz v Nemecku. Informácie z tohto systému by sa mali dať načítať, možno by sa dali aj zapísať.

2. Špecifikácia systému (zodpovední MM, MS a MP, informoval MM)

- MM: Vstupy do systému boli ponechané tak, ako boli prezentované na minulom stretnutí, výstupy sa mierne zmenili, existujú tri druhy výstupov:
 - Zoznam študentov podľa indexu vypočítaného váhovaním
MM upozornil, že zmena je v tom, že sa vráti kompletný zoznam študentov a nie iba tí študenti, ktorí sa umiestnili v prvých X percentách v určených predmetoch. Vyučujúci si bude môcť určiť podmnožinu študentov, ktorých chce zoradiť, podrobné informácie sa mu zobrazia až pri konkrétnom študentovi.
 - Komplexná informácia o študentovi - tu by sa mohla nachádzať informácia, ako úspešný bol študent v jednotlivých skupinách predmetov.
 - Informácie o predmete - tieto informácie si prezerá študent, bude tam aj percentuálne rozloženie známok a koľko ľudí predmet zvládlo/nezvládlo a koľkí si ho zapísali. MM podotkol, že bez údajov zo systému EMA by nebolo možné podobné štatistiky udržiavať.
 - IK: Možno by bolo dobré simulovať EMU našim systémom, na dekanáte by navyše potrebovali nejaké výstupy do MS Excel (postačilo by aj CSV).
 - **MS: Všimol si, že medzi vstupy patria aj informácie o predmetoch.**
MM vymenoval typické prípady použitia a rozdelil ich na dve skupiny:



- bežná prevádzka
- konfigurácia a administrácia
- **MS: Prístupové práva a správa používateľov môže byť jeden prípad použitia**
- IK navrhol použiť vzťah <include>
- MM: Externý systém z pohľadu používateľa dostáva iné informácie, dali by sa doplniť ďalšie prípady použitia.
- MM: Medzi prípadmi použitia nie je **nastavenie profilu (ukladanie prednastaveného váhovania)**.
- MS: V prípade **notifikácie** si každý používateľ prednastaví tie veci, ktoré ho zaujímajú, a nemusí to stále nastavovať.

IK upozornil, že prednastavenia by sa dali aj zdieľať.

JM: Prednastavenia určené na zdieľanie by mal mať na starosti administrátor, aby ich nevzniklo príliš veľa.

MK oponoval, že by všetci chodili za administrátorom.

MM ďalej komentoval prípad použitia „správa typov informácií“:

- MS: Ku každému predmetu budú kľúčové slová, na základe ktorých sa bude určovať, do akej skupiny daný predmet patrí.
- JM a MK nesúhlasili, lepšie je mať skupiny ako tabuľku v databáze

MM vysvetľoval spôsob práce so znalosťami a zručnosťami v systéme:

- MM: Znalosti predstavujú teoretické zvládnutie, zručnosti predstavujú praktické zvládnutie niečoho (aj toho istého). Znalosti a zručnosti sú kvantifikovateľné, certifikáty a hodnotenia sú nekvantifikovateľné. Ku kvantifikovateľným patria aj známky, tie sa však budú spracovávať osobitne. Nekvantifikovateľné informácie budú vyjadrené ako text a ich význam budú určovať metadáta (kľúčové slová), príp. k nim môže byť pripojený dokument alebo linka.

MM pripomenul, že vyučujúci si na začiatku vyberie **podmnožinu študentov**, určí na základe čoho ich chce porovnávať (predmety alebo skupiny predmetov), priradí jednotlivým kritériám váhy systém mu vráti študentov zoradených podľa vypočítaného indexu. Nebude sa nastavovať nič také ako „patrí medzi 10% najlepších“, to bude súčasťou informácie o jednom študentovi.

- JM: Ak bude študent veľmi dobrý v jednom a slabší (pod úrovňou, ktorú by vyučujúci ešte akceptoval) v druhom, môže to celkovo viesť k jeho vysokému hodnoteniu.
- IK a MM: Aj tak bude nižšie ako ten, ktorý je dobrý vo všetkom, závisí to aj od nastavenia váh.
- MS: Zoradenie podľa známok by tam tiež mohlo byť, ak sa známky neuvedú.
- JM Upozornil, že potom sa dá pozrieť percentuálne rozdelenie známok daného predmetu a presne zistiť, kto mal akú známku.
- MM: Dalo by sa skontrolovať, či je rozdelenie známok normálne.
- IK spresnil, že rozdelenie známok sa tak striktné nekontroluje, maximálne sa sleduje, či je dostatočný počet A-čiek.

MM predstavil rolu konfigurátora, ktorý by zaraďoval predmety do skupín a prispôboval výstupy, aby boli v súlade s platnou legislatívou (nastavenie základných parametrov systému).

Následne prebiehala diskusia o role externého systému.



- MM sa spýtal, ako sa komunikuje cez webové služby a ako sa dá zistiť, čo systém dokáže.
- JM a MK Vysvetlili, že každá webová služba má svoj popis v jazyku WSDL, ktorý sa získava štandardným spôsobom.
- MK: Všetko je vlastne externý systém – študenti, vyučujúci, konfigurátor aj administrátor. Zatiaľ máme zvýraznené, kto čo môže, ešte treba kto čo je (všetci sú externý systém).
- MM: Mala by existovať funkcia, ktorá vráti všetkých študentov?
- MS: Študenti sú trieda, mohla by byť funkcia vráť informácie o triede. Externý systém je zbytočné definovať.
- JM: Externými systémami sa myslia tie systémy, ktoré využívajú náš, ale nie sú to konkrétni študenti ani vyučujúci.
- **MS zhrnul, že by sa mali lepšie definovať externé systémy.**
- IK: Je potrebné formálne zadefinovať používateľa, či ide o osobu alebo o externý systém, o aký externý systém ide a prečo tam je.
- JM: Externé systémy sa môžu líšiť v tom, aké majú práva. Študenti ani vyučujúci sa v tom nelíšia. Malo by byť teda viac konkrétnych externých systémov.
- MK navrhol, že budeme mať nejaké funkcie a určíme práva, komu ich poskytneme.
- IK: Zatiaľ nemusíme definovať rozhrania na tejto úrovni. Ale externý systém nemôžeme definovať ako náhradu za iného používateľa.
- MM: Sú dve rozhrania: prístup cez web, ktorý využíva jadro a prístup priamo cez jadro.
- MK a MS prízvukovali, že všetko ide cez jadro, sú to skôr tie práva, ktoré treba vyriešiť.
- MK: Sprístupníme funkciu a tá pracuje s dátami.
- MS: Radšej definujme práva na nižšej úrovni – prístupové práva k objektom.
- MK: Prípád použitia síce vyzerá rovnako, ale napr. niektorému systému poskytne niečo iné ako inému, sú to vlastne dve funkcie.
- MS: Začína to byť príliš všeobecné.
- MM: Môžu sa urobiť špecifické funkcie pre každý systém so špecifickými právami.
- MK navrhol definovať prístupové práva k funkciám a navyše k jednotlivým dátam. Ku každému údaju budú pridané ID používateľov, ktorí k nim môžu pristupovať.
- IK: Matica používateľ a objekt bude asi zložitá. Existujú vlastne iba dva druhy objektov: konkrétny študent (má svoje ID, môže patriť do skupiny) a predmet (môže patriť do skupiny). Stačilo by aj niečo takéhoto, matica by mohla byť nerealizovateľná.
- MK: Nebolo by to až také zložitá, skupiny a práva sa definujú hlavne kvôli externému systému.
- JM: Typ študent by mal na tejto úrovni iba jeden ID, ktorý by reprezentoval všetkých študentov.
- IK súhlasil s prezentovaným riešením.
 - MK zhrnul výsledky diskusie o definovaní prístupových práv:
- MK: Každý používateľ bude mať pridelené funkcie a každý objekt, pri ktorom to má význam, bude mať definované práva, kto k nemu môže pristupovať. V špecifikácii to treba nakresliť aj z tohto pohľadu.
 - JM a MS: Nakreslené to bude až v návrhu, nie v špecifikácii.

3. Diskusia o používaní SPRING frameworku (zodpovedný SC)

SC informoval, že sa stretol s Ing. Grlickým, ktorý mu povedal, že použitie Spring-u jednoznačne uľahčuje vývoj.

- MK: Oponoval, že Ing. Grlický to zatiaľ tiež iba skúša.
- SC: Na projekte NAZOU sú k tomu vypracované analýzy, takže Ing. Grlický pozná všetky výhody aj nevýhody.
- MK: Projekt NAZOU je veľmi veľký v porovnaní s naším, v tom prípade sa použitie frameworku oplatí.
- SC: Nie všetci tam používajú Spring.

SC objasňoval spôsob práce pri použití Spring frameworku a na ukážke jednoduchého zdrojového kódu vysvetlil celý princíp.

- SC: Spring pozostáva z niekoľkých modulov, nemusíme sa ale všetky učiť. Core modul sa stará o vytváranie a rušenie objektov. Na začiatku sa naprogramuje rozhranie, jeho implementácia a vytvorí sa XML súbor, ktorý popisuje vlastnosti rozhrania. Zjednodušuje sa tiež práca s databázou.
- JM: Ak treba volať funkciu s dynamickým parametrom, jediná výhoda spočíva v správe objektov (automatické vytváranie a rušenie).
- MK: Nám bude stačiť programovať bez použitia Spring-u.
- IK: Spring možno začať používať v letnom semestri, prototyp sa môže spraviť bez neho.

4. Správa o postupe v získavaní informácií o systéme ŠTUDENT (zodpovedný AF)

AF informoval, že e-mailom oslovil riešiteľov projektov z minulých rokov, ktorých systémy používali systém ŠTUDENT. Na ich odpoveď sa stále čaká. Tiež si dohodol stretnutie s p. Urbaníkovou, ktorého sa zúčastní aj zástupca konkurenčného tímu. Toto stretnutie sa uskutoční v piatok 11. 11. 2005 na pedagogickom oddelení FIIT STU, AF o ňom podá správu na najbližšom stretnutí tímu.

5. Prečítanie úvodu (zodpovedný JM)

Vzhľadom na krátkosť času JM nečítal celý úvod. Ak niekto chce, môže si ho prečítať individuálne a vyjadriť sa neskôr.

6. Diskusia o metóde implementácie (zodpovedný JM)

JM vysvetľoval potrebu zjednotenia postupu pri implementácii prototypu.

MS oponoval, že všetko sa dôkladne navrhne a rozdelí na časti, ktoré potom implementujú jednotlivci.

JM zdôraznil, že úpravám v návrhu sa nedá vyhnúť a preto je nutné, aby sa všetci o zmenách dozvedeli.

Nakoniec sa všetci zhodli, že by sa mal používať systém na správu verzií. NetOffice, ktorý



tím používa v súčasnosti však nie je dostatočne spoľahlivý, v softvérovom štúdiu je k dispozícii systém Subversion, ktorý bude treba správne nakonfigurovať a zistiť, do akej miery sme ho schopní využiť pre naše potreby.

7. Určenie termínu ďalšieho stretnutia

Keďže v čase pravidelných stretnutí (17. 11. 2005) bude štátny sviatok, ďalšie stretnutie tímu sa uskutoční v stredu 16. 11. 2005 o 17:00 v softvérovom štúdiu.

V pondelok 14. 11. 2005 o 14:00 sa v miestnosti d-209 uskutoční neformálne stretnutie s cieľom navrhnuť model údajov systému.

Úlohy pre členov tímu:

ID	Zodpovedný	Opis úlohy	Termín ukončenia
4.1	Martina Práznovská	<ul style="list-style-type: none">• Vypracovať preberací protokol.	16. 11. 2005
4.2	Slavomír Červeň	<ul style="list-style-type: none">• Vypracovať predbežnú správu o CVS dostupnom v softvérovom štúdiu.	16. 11. 2005
4.3	Andrej Fenik	<ul style="list-style-type: none">• Dokončenie kapitoly Analýza (okolie, požiadavky, ...)	15. 11. 2005
4.4	Martin Kováčik	<ul style="list-style-type: none">• Pripraviť dátový model	14. 11. 2005
4.5	Slavomír Červeň	<ul style="list-style-type: none">• Pripraviť kapitolu Architektúra systému (vrstvy, moduly, interfejsy – môže byť aj slovne).	16. 11. 2005
4.6	Marián Miština	<ul style="list-style-type: none">• Pripraviť metodiku ukladania dokumentov, vytvoriť adresárovú štruktúru na serveri v softvérovom štúdiu.	16. 11. 2005
4.7	Juraj Malečka	<ul style="list-style-type: none">• Stanoviť poradie písania zápisníc	16. 11. 2005
4.8	Juraj Malečka	<ul style="list-style-type: none">• Vypracovať percentuálny rozpis podielov jednotlivých členov tímu na tvorbe dokumentácie.	16. 11. 2005
3.2	Andrej Fenik	<ul style="list-style-type: none">• V spolupráci s tímom 6 (The Lama Team) zistiť informácie o systéme ŠTUDENT• Informovať sa (takisto v spolupráci s tímom 6) o funkciách a plánoch so systémom ŠTUDENT (či sa aj naďalej bude používať) u p.Urbaníkovej a p. Bielekovej	16. 11. 2005
3.3.2	Juraj Malečka	<ul style="list-style-type: none">• Napísať kapitolu Zhrnutie analýzy do dokumentácie	16. 11. 2005
3.7	Marián Miština Michal Sabo Martina Práznovská	<ul style="list-style-type: none">• Špecifikácia systému - dokončenie	16. 11. 2005



Zápis č. 4 zo stretnutia tímu

elf

Dátum:	16.11.2005 17:00 – 19:00	Miesto:	<i>Softvérové štúdio</i>
Vedúci tímu:	Ing. Ivan Kapustík		
Prítomní:	<i>Bc. Slavomír Červeň (SC), Bc. Andrej Fenik (AF), Bc. Martin Kováčik (MK), Bc. Juraj Malečka (JM), Bc. Marián Miština (MM), Bc. Martina Práznovská (MP), Bc. Michal Sabo (MS)</i>		
Vypracoval:	<i>Andrej Fenik ,Juraj Malečka</i>		

Téma stretnutia

Diskusia ohľadom špecifikácie, detailizácia špecifikácie a návrhu, rozdelenie úloh.

Pokračovanie návrhov dátového modelu z neformálneho stretnutia v pondelok 14.11.05.

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí:

ID	Zodpovedný	Opis úlohy	Termín ukončenia	Stav úlohy
4.1	Martina Práznovská	<ul style="list-style-type: none">• Vypracovať preberací protokol.	16. 11. 2005	Ukončená
4.2	Slavomír Červeň	<ul style="list-style-type: none">• Vypracovať predbežnú správu o CVS dostupnom v softvérovom štúdiu.	16. 11. 2005	Ukončená
4.3	Andrej Fenik	<ul style="list-style-type: none">• Dokončenie kapitoly Analýza (okolie, požiadavky, ...)	15. 11. 2005	Ukončená
4.4	Martin Kováčik	<ul style="list-style-type: none">• Pripraviť dátový model	14. 11. 2005	Ukončená
4.5	Slavomír Červeň	<ul style="list-style-type: none">• Pripraviť kapitolu Architektúra systému (vrstvy, moduly, interfejsy – môže byť aj slovne).	16. 11. 2005	Ukončená
4.6	Marián Miština	<ul style="list-style-type: none">• Pripraviť metodiku ukladania dokumentov, vytvoriť adresárovú štruktúru na serveri v softvérovom štúdiu.	16. 11. 2005	Ukončená
4.7	Juraj Malečka	<ul style="list-style-type: none">• Stanoviť poradie písania zápisníc	16. 11. 2005	Ukončená



4.8	Juraj Malečka	<ul style="list-style-type: none">• Vypracovať percentuálny rozpis podielov jednotlivých členov tímu na tvorbe dokumentácie.	16. 11. 2005	Ukončená
3.2	Andrej Fenik	<ul style="list-style-type: none">▪ V spolupráci s tímom 6 (The Lama Team) zistiť informácie o systéme ŠTUDENT▪ Informovať sa (takisto v spolupráci s tímom 6) o funkciách a plánoch so systémom ŠTUDENT (či sa aj naďalej bude používať) u p.Urbaníkovej a p. Bielekovej	16. 11. 2005	Ukončená
3.3 .2	Juraj Malečka	<ul style="list-style-type: none">▪ Napísať kapitolu Zhrnutie analýzy do dokumentácie	16. 11. 2005	Ukončená
3.7	Marián Miština Michal Sabo Martina Práznovská	<ul style="list-style-type: none">▪ Špecifikácia systému - dokončenie	16. 11. 2005	Ukončená

Opis stretnutia:

MS: navrhol, že treba všetko robiť cez Webservices

MK: webslužby sú určené pre externé systémy

MS: externý systém nie je prioritou pri implementácii, možno neurobíme ani model externého systému aby sme to otestovali

MK: bežný používateľ to nepotrebuje

MS: tak to môžeme rovno vyčiarknuť

IK: navrhol aby sme sa pozerali na dátový model

MK: základné entity vyplynuli zo špecifikácie : študent, vyučujúci, predmet, zručnosť, znalosť, certifikát a hodnotenie. Zaujímavé sú váhy a balíčky, v balíčku môžu byť kombinácie znalostí, zručností ... prideliť sa im váha a vypočíta sa výsledný index. Definovali sme typy úrovní, ktoré sa líšia napríklad pre jazyky (ovláda plynule, jazyk je materinský). Entita hodnotenie - môže byť textová, známka, môže obsahovať kľúčové slová. Entita študent o sebe - píše napríklad životopis, motivačný list - tiež kľúčové slová. Entita predmet tiež kľúčové slová.

SC: navrhoval dať do MB aj logovacie súbory. JM: dá sa aj do súborov.

MS a IK: z hľadiska bezpečnosti lepšie v súboroch

MK: prezentoval aj fyzický model

Případy použitia:

MM: je tam pridávanie informácií, vrátenie informácií, konfigurácia

MS: prečo externý systém môže pridávať iba certifikáty? Dá sa to nastaviť pomocou práv.

MK: To už môžeme všetko urobiť cez webové služby

MS: teraz si to navrhujeme a potom v letnom semestri sa môžeme rozhodnúť, či to budeme implementovať

MM: navrhol som to podľa toho ako sme sa dohodli na stretnutí

JM: ak by sme to navrhli a potom implementovali len časť, dalo by sa to ?



IK: návrh by mal byť úplný

MM: aj ja som si pôvodne myslel, že to urobíme všeobecnejšie

MS: prečo externý systém môže pridať práve certifikát ?

MM: ak niekto napríklad skončí CISCO - iný systém tam automaticky môže pridať certifikát. Dám teda šípky aj k znalosti aj zručnosti a uvidíme, či sa bude implementovať. Konfigurácia : koľko minimálne ľudí a koľko minimálne predmetov do váhovania.

MS: kto pridáva predmety a známky - use case import

MM: robí to iba administrátor raz za pol roka

MS: konfigurátor by mohol meniť niečo v predmetoch ak sa tam niečo zmení.

MM: môže potom vzniknúť nekonzistencia, lebo u nás sa niečo zmení a na "Študentovi" to zostáva

MK: aby sa nestalo, že urobíme kópiu systému "Študent"

Diskusia o tom, či zaraďovať aj do tried aj podľa výstupného indexu

IK: v návrhu by sa mali vyskytovať obe alternatívy, ak by tam bolo viac predmetov, stávalo by sa, že by sme mali menej tried. Čo bude najlepšie sa ukáže už pri praktickom odskúšaní. Treba to vedieť usporiadať, na tom sa už dajú definovať triedy (absolútne aj relatívne).

MM: správa skupín predmetov je v podstate správa balíčkov ale z funkčného hľadiska je to iné, čo je v balíčku sa dá váhovať

MS a MK: treba vyhodiť správu skupín predmetov a v popise k správe balíčkov napísať aj o správe skupín predmetov

MK: externý systém by mal vedieť pridávať iba dôkaz a nie certifikát

MM: čo všetko sa dá notifikovať treba ešte dorobiť externé systémy

AF: formát extraktu "Študenta", odtiaľ vieme získať známky, treba ešte extraktor. Peter Lacko má v elektronickej podobe informácie o predmetoch, systém študent má aj nejaké webové rozhranie ? Nie, resp. je to náročné zisťovať a to čo máme je postačujúce, t.j. každý polrok.

MK: lepšie by bolo vybaviť si konto a získať nejaké pozmenené dáta aby sa to nemuselo vždy len importovať

IK: treba ich kontaktovať a zistiť, či sa nedá získať iná forma exportu, je to pravdepodobné, treba sa dohodnúť čo budeme prototypovať

MK: prototypovať môžeme v ľubovoľnom jazyku ?

IK: môže to byť prototyp na zahodenie , na overenie alebo sa bude ďalej rozvíjať.

percentá aj na ponuku

Úlohy pre členov tímu:

D	Zodpovedný	Opis úlohy	Termín ukončenia
2.1	Slavomír Červeň	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	24. 11. 2005
2.2	Andrej Fenik	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	24. 11. 2005
2.3	Michal Sabo	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	24. 11. 2005
2.4	Martin Kováčik	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	24. 11. 2005



2.5	Juraj Malečka	<ul style="list-style-type: none">• posudzovanie príbuznej kapitoly	24. 11. 2005
2.6	Marián Miština	<ul style="list-style-type: none">• posudzovanie príbuznej kapitoly	24. 11. 2005
2.7	Martina Práznovská	<ul style="list-style-type: none">• posudzovanie príbuznej kapitoly	24. 11. 2005

**Zápis č. 5 zo stretnutia tímu****_elf_**

Dátum:	24.11.2005 10:10 – 12:30	Miesto:	<i>Softvérové štúdio</i>
Vedúci tímu:	Ing. Ivan Kapustík		
Prítomní:	<i>Bc. Slavomír Červeň (SC), Bc. Andrej Fenik (AF), Bc. Martin Kováčik (MK), Bc. Juraj Malečka (JM), Bc. Marián Miština (MM), Bc. Martina Práznovská (MP), Bc. Michal Sabo (MS)</i>		
Vypracoval:	<i>Martin Kováčik</i>		

Téma stretnutia

Diskusia k hrubému návrhu konkurenčného tímu, rozhodnutie čo dáme do posudku, rozdelenie úloh na ďalšie dva týždne.

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

Tabuľka uvádza stav riešenia úloh zadaných počas stretnutia dňa 16.11.2005 a skôr. Staršie ukončené úlohy nie sú v tabuľke zahrnuté.

ID	Zodpovedný	Popis úlohy	Termín ukončenia	Stav úlohy
5.1	Slavomír Červeň	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	16.11. 2005	ukončená
5.2	Andrej Fenik	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	16.11. 2005	ukončená
5.3	Michal Sabo	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	16.11. 2005	ukončená
5.4	Martin Kováčik	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	16.11. 2005	ukončená
5.5	Juraj Malečka	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	16.11. 2005	ukončená
5.6	Marián Miština	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	16.11. 2005	ukončená
5.7	Martina Práznovská	<ul style="list-style-type: none">posudzovanie príbuznej kapitoly	16.11. 2005	ukončená

Opis stretnutia

Priebeh stretnutia sa odvíjal najmä od prezentácie splnených úloh a následnej diskusie.

1. Diskusia k hrubému návrhu konkurenčného tímu

- IK: Dal nám rady, čo do posudku napísať: Treba napísať, čo majú dobre, čo nie, ale treba si vziať aj ponaučenie.
- AF: Nepáčilo sa mu že konkurenčný tím umožňuje pedagógovi vidieť výsledky študenta. Referencie nie sú podľa ISO 690. Tím tiež hovorí, že bude riešiť bezpečnosť a odozvu, ale odozva sa ďalej už nerieši. V dokumentácii je opísaných 5 používateľských rolí, ale neskôr nie sú všetky popísané a ďalej v dokumente je ich dokonca viac.
- MM: Povedal, že v konkurenčnej dokumentácii nie je vysvetlený vzťah medzi používateľmi.
- AF: Poznamenal, že vyplnenie testu nie je dôveryhodné, a že študent nie je motivovaný k vypĺňaniu testov.
- JM: Testy sa mu až tak veľmi nepáčia – nemá to prínos, nevnáša to tam objektivitu.
- MM: Nesúhlasil, podľa neho sú testy OK – zapadá to do ich filozofie mať všetko podložené.
- AF: Povedal, že systém by mal mať administrátorské rozhranie a administrátor by sa nemal starať o bezpečnosť systému.
- MM: Súhlasil, bezpečnosť by mala byť už v systéme a nemala by byť vecou administrátora.
- MM: Oboznámil nás, že konkurencia má v jednej kapitole namiesto opisu aké údaje vstupujú do systému, opis ako údaje vstupujú do systému.
- MM: Podľa neho je konkurenčný systém navrhnutý veľmi komplexne a nie je použiteľný, pretože ho neustále treba naplňať údajmi a to nie len na začiatku ale aj za behu. Učiteľ napríklad musí po absolvovaní predmetu vyplniť množstvo údajov.
- AF: Poznamenal, že im chýba výber technológií.
- MS: Podľa neho je dôležité do posudku napísať, že konkurencia nemá strop pri určovaní úrovne znalosti predmetu.
- JM: Poznamenal, že neexistuje človek, ktorý by dokázal objektívne určiť akú úroveň znalostí poskytuje daný predmet.
- JM: Podľa neho by mal používateľ mať možnosť určiť si ako dôležitý je predmet z jeho pohľadu.
- IK: Súhlasil, pretože aj on by chcel mať takú možnosť. Miera (váha) sa robí na základe aplikácie, preddefinované hodnoty sú OK, ale existujú ľudia, ktorí používajú vlastné váhy.
- MS: Vysvetlil IK ako funguje výpočet znalostí v tíme Llama
- IK: Povedal, že naše balíčky dokážu spraviť to, čo ich „vec“, okrem toho v ich systéme to môže vyplúť až nezaujímavé hodnoty. Tiež povedal, že naša reprezentácia je síce plytkejšia ale v praxi asi použiteľnejšia.
- JM: Poznamenal, že v konkurenčnom systéme je navyše oproti nášmu, aký predmet vplyva akou znalosťou, môže to mať však neželané pôsobenie.
- AF: Nepáčila sa mu mapovacia funkcia, nevie prečo si tím vybral práve takú.
- MK: Povedal, že funkcia je OK, účel, ktorý s ňou zamýšľali plní.

- IK: Podľa neho si tím zvolil interval $<0, \infty)$ nešťastne, skôr si mali zvoliť interval typu $<0,10>$
- MM: Povedal, že popis ontológie, ktorý má konkurencia v návrhu patrí skôr do analýzy.
- MK: Povedal, že treba konkurencii vytknúť, že nemajú kompletný dátový model.
- AF: Spomenul, že konkurencia nemá špecifikáciu údajov v systéme.
- AF: Pochválil definíciu znalosti zo slovníka, prepracované vnímanie znalostí.
- MM: Predstavil nám svoje postrehy z dokumentácie konkurenčného tímu, ale pre nedostatok času len veľmi stručne (ale zrozumiteľne, použiteľne a k veci).

2. Voľba implementačného prostredia pre prototyp

- IK: Oznámil, že v ďalšom sa treba zamerať na tvorbu prototypu; do dvoch týždňov by mala existovať dáka verzia prototypu.
- MK: Povedal, že sa treba dohodnúť v akom prostredí/nástroji sa bude prototyp implementovať. Navrhol MS ACCESS, pretože s ním máme všetci už skúsenosti. Tiež spomenul nevýhody výberu tohto prostredia, pretože neposkytuje možnosť paralelného vývoja prototypu viacerými členmi tímu. Ako ďalšiu možnosť navrhuje PHP+MySQL, s ktorým má väčšina členov tímu tiež skúsenosti. Výhodou tohto riešenia je možnosť paralelnej práce na implementácii.
- MS: Oponoval, povedal, že keď to máme spraviť v PHP, tak to rovno môžeme spraviť v JAVE, s tou však nemáme skúsenosti, preto navrhol použiť MS ACCESS s tým, že si úlohy rozdelíme, tak ako to bude možné. Ostatní členovia tímu súhlasili.

Úlohy pre členov tímu

ID	Zodpovedný	Popis úlohy	Termín ukončenia
6.1	Slavomír Červeň	<ul style="list-style-type: none">• naštudovať web services s použitím navrhnutých technológií• vytvoriť názorný príklad (najlepšie niečo, čo sa bude dať využiť neskôr) použitia web services	1. 12. 2005
6.2	Andrej Fenik	<ul style="list-style-type: none">• vytvoriť parser na transformáciu výstupného súboru zo systému ŠTUDENT do formátu, ktorý sa dá importovať do MS ACCESS (modelové súbory poskytnete MP)	1. 12. 2005
6.3	Michal Sabo	<ul style="list-style-type: none">• vytvoriť databázu (tabuľky a relácie medzi nimi)• naplniť ju údajmi (údaje v správnom formáte poskytnete AF)• vytvoriť formulár výberu „balíčku“ prezentovanie výsledkov výpočtu výsledného indexu na základe „balíčkov“	8. 12. 2005



6.4	Martin Kováčik	<ul style="list-style-type: none">• napísať funkcie na výpočet výsledného indexu na základe „balíčkov“	8. 12. 2005
6.5	Juraj Malečka	<ul style="list-style-type: none">• vytvoriť formuláre pre učiteľa a študenta• získať sedem licencií MS ACCESS	8. 12. 2005
6.6	Marián Miština	<ul style="list-style-type: none">• vytvoriť formuláre na konfiguráciu balíčkov• vytvoriť formuláre na konfiguráciu skupiny predmetov	8. 12. 2005
6.7	Martina Práznovská	<ul style="list-style-type: none">• vytvoriť údaje (dostatočne veľa), ktoré budú reprezentovať výstup zo systému ŠTUDENT	1. 12. 2005

**Zápis č. 6 zo stretnutia tímu****_elf_**

Dátum:	<i>1.12.2005 10:30 – 12:30</i>	Miesto:	<i>Softvérové štúdio</i>
Vedúci tímu:	<i>Ing. Ivan Kapustík</i>		
Prítomní:	<i>Bc. Slavomír Červeň (SC), Bc. Andrej Fenik (AF), Bc. Martin Kováčik (MK), Bc. Juraj Malečka (JM), Bc. Marián Miština (MM), Bc. Martina Práznovská (MP), Bc. Michal Sabo (MS)</i>		
Vypracoval:	<i>Michal Sabo</i>		

Téma stretnutia

Vyjadrenie k posudku od tímu č. 6 a konzultácia doterajšej práce na prototyp

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

Tabuľka uvádza stav riešenia úloh zadaných počas stretnutia dňa 24.11.2005 a skôr. Staršie ukončené úlohy nie sú v tabuľke zahrnuté.

ID	Zodpovedný	Popis úlohy	Termín ukončenia	Stav úlohy
6.1	Slavomír Červeň	<ul style="list-style-type: none">naštudovať web services s použitím navrhnutých technológiívytvoriť názorný príklad (najlepšie niečo, čo sa bude dať využiť neskôr) použitia web services	01. 12. 2005	čiastočne ukončená
6.2	Andrej Fenik	<ul style="list-style-type: none">vytvoriť parser na transformáciu výstupného súboru zo systému ŠTUDENT do formátu, ktorý sa dá importovať do MS ACCESS (modelové súbory poskytne MP)	01. 12. 2005	čiastočne ukončená
6.3	Michal Sabo	<ul style="list-style-type: none">vytvoriť databázu (tabuľky a relácie medzi nimi) naplniť ju údajmi (údaje v správnom formáte poskytne AF)vytvoriť formulár výberu „balíčku“ prezentovanie výsledkov výpočtu výsledného indexu na základe „balíčkov“	01. 12. 2005	riešená



6.4	Martin Kováčik	<ul style="list-style-type: none">• napísať funkcie na výpočet výsledného indexu na základe „balíčkov“	08. 12. 2005	riešená
6.5	Juraj Malečka	<ul style="list-style-type: none">• vytvoriť formuláre pre učiteľa a študenta• získať sedem licencií MS ACCESS	08. 12. 2005	riešená
6.6	Marián Miština	<ul style="list-style-type: none">• vytvoriť formuláre na konfiguráciu balíčkov• vytvoriť formuláre na konfiguráciu skupiny predmetov	08. 12. 2005	riešená
6.7	Martina Práznovská	<ul style="list-style-type: none">• vytvoriť údaje (dostatočne veľa), ktoré budú reprezentovať výstup zo systému ŠTUDENT	01. 12. 2005	čiastočne ukončená

Opis stretnutia

Priebeh stretnutia sa odvíjal najmä od prezentácie splnených úloh a následnej diskusie.

1. Vyjadrenia k posudku od konkurenčného tímu

- MK: povedali nám, že to máme dobré, teda sme sa vlastne nič nedozvedeli
- JM: našli nám nejaké gramatické chyby
- MM: boli proti našej základnej filozofii
- JM: vytýkali nám, že sme sa nezaoberali vzťahmi medzi znalosťami
- MK: vytýkali nám, že nemáme vzorec na výpočet indexu študenta
- MM: v podstate nás celkom pochválili
- SC: zistil, že tím č. 6 sú trochu nahnevaní, že sme ich skritizovali a pritom oni nás vychválili.
Zhodnotil, že je dobré kritizovať, pretože sa opravujú chyby
- IK: povedal, že treba aj pochváliť, lebo to motivuje k ďalšej práci

2. Prezentácia riešenia úlohy 6.5 (zodpovedný: JM)

- JM: napísal Ing. Bernátovi o licenciách. Je potrebné sa dohodnúť, kedy si osobne prídeme pre CD. Zatiaľ však nie je Access, ale dá sa však zohnať aj ten.
- JM: čo sa týka formulárov, tie zatiaľ nestihol vytvoriť
- MK: povedal JM, že spraví formulár pre vybratie znalostí a zručností a certifikátov a integruje funkciu usporiadania a výberu balíčka a študentov

3. Prezentácia riešenia úlohy 6.6 (zodpovedný: MM)

- MM: prezentoval, čo doteraz spravil, ale ešte to nie je dokončené. Nevedel aký je interval pre váhy.
- MK: váhy sú z intervalu $<0;1>$
- MK: nepáčilo sa mu, že skupina predmetov je samostatný formulár
- VŠETCI: váhy dáme od 0 do 1; prebiehala debata o formulári a čo vylepšiť
- IK: známky treba dodefinovať lebo A má výrazne kratší interval ako ostatné – treba to ešte

doriešiť

- MM: navrhol, aby sme známku dali cez tabuľku úroveň a nie tabuľku známka
- MS: navrhol, aby sa vytvorilo prepojenie *študent* – *ovláda* – *predmet* cez úroveň by bolo možné potom určiť známku a následne by sa mohla zrušiť tabuľka známka
- VŠETCI: odmietli návrh MS, ale dohodli sa, že sa vytvorí prepojenie tabuliek známka a úroveň
- AF: zastal si MS ☹
- MP: nezastala si MS
- MS: hnevá sa ☹
- MS: navrhol premenovať atribút váha úrovne na stupeň úrovne
- VŠETCI: navrhli MM dorobiť balíčky a potom to s JM dať dokopy

4. Prezentácia riešenia úlohy 6.1 (zodpovedný: SC)

- SC: našťudoval Web Services. Nemohol však prezentovať, pretože v SW štúdiu zatiaľ chýba AXIS. Oznámil, že vlastne sa píše len nejaké classy na strane servera a na strane klienta sa pripojíme na server a voláme funkciu. Chce vyskúšať zadávať niečo cez web-services (asi známky).
- VŠETCI: navrhli SC spraviť príklad na webslužby, ktoré budú pracovať aj s našou databázou

5. Prezentácia riešenia úlohy 6.2 (zodpovedný: AF)

- AF: predstavil parser v jeho terajšej podobe
- VŠETCI: navrhli aby sme dali osobné číslo ako ID študenta dajme
- MS: navrhol, či by nebolo by lepšie mať známku 1, 1.5, 2, ...
- IK: lepšie by bolo interne známky reprezentovať rovnako
- VŠETCI: bavíme sa o známkach a ich reprezentácii a kódovaní vstupných súborov – je to ASCII
- JM: navrhol, nech parser rozhodí výstup do viacerých súborov, aby bolo možné dáta ľahšie importovať do jednotlivých tabuliek
- VŠETCI: navrhli AF vyparsovať ešte aj predmety a dorobiť formát výstupu CSV

6. Prezentácia riešenia úlohy 6.7 (zodpovedný: MP)

- MP: do databázy naplnila študentov, navrhla rozšíriť atribút názov v tabuľke *predmet*, doplniť do tabuľky stĺpec *skratka predmetu*, *ID_predmet* by mal byť štvormiestny kód predmetu. Tabuľka *študent* by sa mala doplniť o *adresu*, *dátum narodenia*, aktuálny *BŠP*, *odbor* a *ročník*. Tabuľka *známka* by mohla byť rozšírená o *dátum zapísania známky*. Treba sa zamyslieť aj nad interpretáciou známky „započítané“.

7. Prezentácia riešenia úlohy 6.7 (zodpovedný: MK)

- MK: prezentuje výpočtovú funkciu a prebieha diskusia k výpočtu výsledného indexu
- MS: zamyslime sa nad tým obmedzením výpočtu známky
- MK: urobia sa ešte nejaké triedy
- JM: dá sa spraviť balíček z balíčkov?
- VŠETCI: diskutujeme.. spraví sa asi vyššia úroveň hierarchie – akési nadbalíčky



- IK: najlepšie cez jeden interface a vyberať si nadbalíček, či podbalíček, treba to však ešte zvážiť.
- MS: spravil výber študentov z formulára, ktorý vráti priamo databázový dotaz
- MK: navrhol doplniť výber aj na základe ročníka, odboru
- MS: spraví sa to potom, nie v prototype

Úlohy pre členov tímu

Zostávajú úlohy z minulého týždňa, ktoré treba upraviť na základe predchádzajúcej diskusie.

ID	Zodpovedný	Popis úlohy	Termín ukončenia
7.1	Martina Práznovská	<ul style="list-style-type: none">• pripraví používateľskú dokumentáciu	15.12.2005

**Zápis č. 7 zo stretnutia tímu****_elf_**

Dátum:	8.12.2005 10:15 – 11:30	Miesto:	<i>Softvérové štúdio</i>
Vedúci tímu:	Ing. Ivan Kapustík		
Prítomní:	<i>Bc. Slavomír Červeň (SC), Bc. Andrej Fenik (AF), Bc. Martin Kováčik (MK), Bc. Juraj Malečka (JM), Bc. Marián Miština (MM), Bc. Martina Práznovská (MP), Bc. Michal Sabo (MS)</i>		
Vypracoval:	<i>Slavomír Červeň</i>		

Téma stretnutia

Prezentovanie doterajších výsledkov prototypu a s tým súvisiaca diskusia.

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

Tabuľka uvádza stav riešenia úloh zadaných počas stretnutia dňa 1.12.2005 a skôr. Staršie ukončené úlohy nie sú v tabuľke zahrnuté.

ID	Zodpovedný	Popis úlohy	Termín ukončenia	Stav úlohy
6.1	Slavomír Červeň	<ul style="list-style-type: none"> naštudovať web services s použitím navrhnutých technológií vytvoriť názorný príklad (najlepšie niečo, čo sa bude dať využiť neskôr) použitia web services 	8. 12. 2005	čiastočne ukončená
6.2	Andrej Fenik	<ul style="list-style-type: none"> vytvoriť parser na transformáciu výstupného súboru zo systému ŠTUDENT do formátu, ktorý sa dá importovať do MS ACCESS (modelové súbory poskytne MP) 	8. 12. 2005	ukončená
6.3	Michal Sabo	<ul style="list-style-type: none"> vytvoriť formulár výberu „balíčku“ prezentovanie výsledkov výpočtu výsledného indexu na základe „balíčkov“ 	8. 12. 2005	riešená
6.4	Martin Kováčik	<ul style="list-style-type: none"> napísať funkcie na výpočet výsledného indexu na základe „balíčkov“ 	8. 12. 2005	ukončená
6.5	Juraj Malečka	<ul style="list-style-type: none"> vytvoriť formuláre pre učiteľa a študenta 	8. 12. 2005	riešená



6.6	Marián Miština	<ul style="list-style-type: none">vytvoriť formuláre na konfiguráciu balíčkovvytvoriť formuláre na konfiguráciu skupiny predmetov	8. 12. 2005	ukončená
6.7	Martina Práznovská	<ul style="list-style-type: none">pripraví používateľskú dokumentáciu	1. 12. 2005	ukončená

Opis stretnutia

Priebeh stretnutia sa odvíjal najmä od prezentácie splnených úloh a následnej diskusie.

1. Úvodná diskusia

- IK: priniesol dokumentáciu. Oznámil, že by sme si mali dohodnúť stretnutie s tímom 6 kôli prezentovaniu prototypu a že v stredu (21.12) sa učí ako vo štvrtok.

2. Prezentácia riešenia úlohy 6.1 (zodpovedný: SC)

- SC: vytvoril jsp stránku, na ktorej bolo znázornené použitie webových služieb (pre získavanie informácií o študentovi, predmetoch, znalostiach, zručnostiach...). Na labss2 ešte nie je AXIS, takže to prezentoval na notebooku. Žiaľ ani tam to nechcelo korektné fungovať hoci deň predtým to išlo. Tak nabudúce ukáže konečnú verziu. Vytvoril databázu v postgresql.

3. Prezentácia riešenia úlohy 7.1 (zodpovedná: MP)

- MP: vytvorila dokumentáciu k prototypu. Opisuje vytvorenú dokumentáciu (rozdelenie na 3 časti: Ciele, Implementácia, Zhodnotenie)
- MS: kde je používateľská príručka?
- MM: bude to úplne zvlášť?
- IK: bude to v rámci dokumentácie k prototypu. Zvykne to bývať aj ako príloha.
- MP: opisuje podrobne dokumentáciu a kto má čo napísať do jednotlivých častí (Realizácia a popis funkčnosti, Výsledky).
- IK: vo výsledkoch bude to čo sa nám podarilo a či to funguje alebo nie

4. Prezentácia riešenia úlohy 6.2 (zodpovedný: AF)

- AF: Prezentuje parser. Prvý riadok sú názvy stĺpcov. Jednotlivé položky sú oddelené čiarkou (predtým bola pomlčka). Vyextrahoval všetko čo sa dalo. Na labss2 je stará verzia shellu, ktorá nepozná match, preto nevidno poriadne niektoré znaky.

5. Prezentácia riešenia úlohy 6.3 (zodpovedný: MS)

- MS: má čiastočne hotové formuláre na výber balíčkov (nestíhal dorobiť). Databázu už má hotovú.

6. Prezentácia riešenia úlohy 6.4 (zodpovedný: MK)

- MK: dorobil triedy. Pridal, ako percentuálne prispievajú k výslednému indexu. Normalizoval



výsledný index. Napísal funkciu, ktorá rozdelí študentov do tried. Počítal to pomocou smerodajnej odchýlky. Prišiel na to empiricky, nie je to matematicky dokázané. Využil štatistické funkcie MS ACCESS.

- AF: čo bude vidieť používateľ?
- MK: používateľovi nastavíš, čo sa mu má zobrazit'. Nemalo by sa mu zobrazit' percentuálne rozdelenie.
- MS: chce niekto dokázať tú funkciu?
- IK: netreba, smerodajná odchýlka sa môže použiť.
- IK: správa sa to podobne pre veľké aj pre malé čísla
- JM: čo sa stane, keď bude veľa študentov?
- IK: treba vyskúšať ☺
- MK: ja to robiť nebudem ☺

7. Prezentácia riešenia úlohy 6.5 (zodpovedný: JM)

- JM: vybavil licencie. Máme si prísť osobne pre ne a podpísať to. Formuláre zatiaľ nie sú.

8. Prezentácia riešenia úlohy 6.6 (zodpovedný: MM)

- MM: upravil váhy od 0 do 1. Prezentuje formulár na konfiguráciu balíčkov.
- MS: páči sa mu to ako aj ostatným. „Čo tak to nechať v ACESSE?“ ☺
- MK: a pristupoval by k tomu iba jeden používateľ a vytvárala by sa fronta... ☺
- MM: k balíčkom sú priradené váhy
- MS: čo budeme robiť budúci semester? ☺
- MM: objavil funkciu get external data v ACCESS na importovanie objektov do databázy
- MM: pridal linku predmet-typ_urovne v dátovom modeli. Poznámene, že v databáze ešte nie sú informácie o predmete.
- MP: treba ešte pridať atribúty do tabuľky študent

9. Záverečná diskusia

- IK: posudzovať sa bude prototyp a prezentácia
- AF: ktorá databáza je aktuálna?
- JM: dajme tam číslo verzie
- MM: cvs dáva automaticky
- JM: podujme sa na dôkaz funkcie, ktorú MK navrhol.
- AF: vyrobí konečnú databázu

10. Záverečné poznámky

- MS a MK sa dnes výnimočne nehádali. ☺
- MS dnes normálne bežal do školy (alebo utekal pred MP) ☺
- MM skonštatoval, že vedúci zabudol, že dnes je stretnutie a ešte pred chvíľou bol v pyžame. ☺
- Všetci sa tešíme na vianoce a chceme, aby bol tento semester už za nami.



- Sme šťastní, že dnes sme tak skoro skončili

Úlohy pre členov tímu

ID	Zodpovedný	Popis úlohy	Termín ukončenia
8.1	Slavomír Červeň	<ul style="list-style-type: none">• dokončiť príklady na webové služby	15.12.2005
8.2	Andrej Fenik	<ul style="list-style-type: none">• vytvoriť finálnu databázu	15.12.2005
8.3	Michal Sabo	<ul style="list-style-type: none">• vytvoriť formulár pre výber balíčka	15.12.2005
8.4	Martin Kováčik	<ul style="list-style-type: none">• napíše dokumentáciu k svojej časti prototypu	15.12.2005
8.5	Juraj Malečka	<ul style="list-style-type: none">• dôkaz matematickej funkcie• dokončiť formuláre pre študenta a učiteľa	15.12.2005
8.6	Marián Miština	<ul style="list-style-type: none">• napíše dokumentáciu k svojej časti prototypu	15.12.2005
8.7	Martina Práznovská	<ul style="list-style-type: none">• pripraví používateľskú príručku	15.12.2005
8.8	Michal Sabo, Martin Kováčik	<ul style="list-style-type: none">• zkompletizujú vytvorené časti prototypu v MS ACCESS do funkčnej podoby	15.12.2005

**Zápis č. 8 zo stretnutia tímu****_elf_**

Dátum:	<i>15.12.2005 10:15 – 11:30</i>	Miesto:	<i>Softvérové štúdio</i>
Vedúci tímu:	<i>Ing. Ivan Kapustík</i>		
Prítomní:	<i>Bc. Slavomír Červeň (SC), Bc. Andrej Fenik (AF), Bc. Martin Kováčik (MK), Bc. Juraj Malečka (JM), Bc. Marián Miština (MM), Bc. Martina Práznovská (MP), Bc. Michal Sabo (MS)</i>		
Vypracoval:	<i>Martina Práznovská</i>		

Téma stretnutia

Prezentovanie ďalších prác na prototypu a diskusia

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

ID	Zodpovedný	Popis úlohy	Termín ukončenia	Stav úlohy
8.1	Slavomír Červeň	<ul style="list-style-type: none">dokončiť príklady na webové služby	15.12.2005	ukončená
8.2	Andrej Fenik	<ul style="list-style-type: none">vytvoriť finálnu databázu	15.12.2005	riešená
8.3	Michal Sabo	<ul style="list-style-type: none">vytvoriť formulár pre výber balíčka	15.12.2005	riešená
8.4	Martin Kováčik	<ul style="list-style-type: none">napíše dokumentáciu k svojej časti prototypu	15.12.2005	riešená
8.5	Juraj Malečka	<ul style="list-style-type: none">dôkaz matematickej funkciedokončiť formuláre pre študenta a učiteľa	15.12.2005	čiastočne ukončená
8.6	Marián Miština	<ul style="list-style-type: none">napíše dokumentáciu k svojej časti prototypu	15.12.2005	riešená
8.7	Martina Práznovská	<ul style="list-style-type: none">pripraví používateľskú príručku	15.12.2005	čiastočne ukončená
8.8	Michal Sabo, Martin Kováčik	<ul style="list-style-type: none">skompletizujú vytvorené časti prototypu v MS ACCESS do funkčnej podoby	15.12.2005	riešená



Opis stretnutia

1. Prezentácia riešenia úlohy 8.1 (zodpovedný: SC)

Web-services

- Na minulom stretnutí nebežal správne Axis, pretože k svojej činnosti potrebuje prístup na internet a ten vtedy nebol k dispozícii. SC predvádza web-aplikáciu ZNALOSTI, ktorá sa pomocou webových služieb pripája na databázu a odtiaľ vyberá informácie o študentovi, informácie o predmetoch, zoznam znalostí, zručností, certifikátov, študentov. V sekcii informácie o študentovi sa zadá ID študenta a odošle sa požiadavka. Vráti sa informácie o študentovi, ktoré sú uložené v databáze. Všetko to beží cez webové služby.
- MS: každé pole je jedna funkcia? SC: nie, vráti sa to ako jeden string.
- MS: xml musíš sám navrhnuť.
- SC popisuje zdrojový kód, vysvetľuje jednotlivé časti, pripojenie k databáze, metódy, result set, ktorý sa vracia na základe query, nakoniec return vráti to, čo sa získalo zo selectu. Endpoint definuje, kde beží webová služba. Každá položka v menu znamená volanie jednej webovej služby, ktorá vo svojom klientskom kóde definuje meno metódy, ktorá sa má zavolať, predajú sa jej parametre zadané napr. v editboxe.
- Ďalej stručne popísal prostredie a postup inštalácie. Je potrebné nainštalovať a nakonfigurovať JAVA, TOMCAT, axis, PostgreSQL, (activation.jar).
- MM: páči sa mi to! SC: je to jednoduché ☺

2. Prezentácia riešenia úlohy 8.5 (zodpovedný: JM)

Teória o smerodajnej odchýlke

- JM: Smerodajná odchýlka popisuje ako veľmi sa líšia hodnoty od „stred“, za „stred“ sa môže brať buď medián alebo stredná hodnota. Ak je veľmi veľa hodnôt sústredených okolo „stred“, smerodajná odchýlka bude malá. MK: V našom prípade je tento spôsob interpretácie pomocou operácií so smerodajnou odchýlkou vhodný, pretože rozlíšia dobrí študenti od zlých a zadelia sa do tried.

Formulár pre študenta

- JM: Problém s query „úroveň“ vo formulári (vylisovanie obsahu, ktorý tam je, aby sa mohol následne zmeniť). Také query sa nedá vymyslieť.
- MK: nie je to kvôli query, je to preto, že Access to robí tak ako to robí.
- MM: iba vylisovať obsah, ktorý tam je bez možnosti zápisu a pridávanie riešiť ďalším formulárom (ten by obsahoval: v prvom comboboxe predmet, znalosť, zručnosť alebo certifikát, v druhom comboboxe úroveň)
- MK: nemusí to byť pekné, hlavne že to bude fungovať
- JM: toto som chcel urobiť na vylisovanie toho, čo tam je. Keď už to tam je, tak by sa to potom mohlo dať aj zmeniť.



- MK: predviedol ako má vyzerať formulár (editbox a dve tlačidlá)
- MK: treba nazývať premenné správne, aby sa to dalo importovať
- JM: ukázal formulár informácie o študentovi obsahuje aj tlačidlo hodnotenie.
- MS: hodnotenie netreba, nie sú na to tabuľky.

3. Diskusia o termíne prezentácie

- Prezentáciu prototypu sme dohodli na pondelok o 17:00, všetci súhlasili. Keby sa niečo zmenilo, budeme všetkých informovať. Miesto sa tiež dohodne dodatočne. Nie sme závislí na mieste konania, lebo budeme prezentovať na laptope.

4. Prezentácia vylepšenia zobrazenia tabuľky výpočtu hodnotení (MK)

- MK: prerobil dizajn výstupu – listbox - výsledná tabuľka hodnotení, rozdelenie študentov do tried je prezentované percentuálnym rozsahom triedy (100-98%). Percentuálne prispievanie jednotlivých znalostí do hodnotenia.

5. Prezentácia riešenia úlohy 8.7 (zodpovedná: MP)

Používateľská príručka

- Diskusia o dokumentácii a zaradení používateľskej príručky. Dohodli sme sa na členení kapitoly prototyp na 3 časti: Technická dokumentácia, Používateľská príručka, Zhodnotenie. Pričom technická dokumentácia aj používateľská príručka budú pozostávať z dvoch častí – prvá pre databázu a druhá časť pre web-services.

6. Záverečná diskusia

- IK: budúcu stredu 21.12. sa nekoná tímové stretnutie
- IK: dokumentácia aj prototyp by mali byť dostupné na web-stránke tímu a na CD nosiči pripojenej k vytlačenej dokumentácii
- MP: termín na dodanie dokumentácie k prototypu je sobota 17.12 večer.
- Sme šťastní, že dnes sme tak skoro skončili

Úlohy pre členov tímu

Tabuľka obsahuje iba nové úlohy pre členov tímu. Staršie neukončené úlohy nie sú v tabuľke zahrnuté.

ID	Zodpovedný	Popis úlohy	Termín ukončenia
9.1	Slavomír Červeň	<ul style="list-style-type: none">• Napáliť CD, ktoré bude obsahovať všetky inštalačné programy k prototypu, samotný prototyp, dokumentácie a napísať postup inštalácie• Upraviť návod k CVS, ktorý sa zverejní v dokumentácii k riadeniu	19.12.2005



9.2	Andrej Fenik	<ul style="list-style-type: none">• napísať dokumentáciu k svojej časti prototypu	17.12.2005
9.3	Michal Sabo	<ul style="list-style-type: none">• napísať dokumentáciu k svojej časti prototypu	17.12.2005
9.4	Juraj Malečka	<ul style="list-style-type: none">• napísať dokumentáciu k svojej časti prototypu	17.12.2005
9.5	Marián Miština	<ul style="list-style-type: none">• napísať dokumentáciu k svojej časti prototypu	17.12.2005
9.6	Martina Práznovská	<ul style="list-style-type: none">• kompletizácia dokumentácie k prototypu• kompletizácia dokumentácie k riadeniu• vypracovať protokol o prevzatí dokumentácie• opraviť chyby – krížové odkazy v dokumentácii	19.12.2005



VII PREBERACIE PROTOKOLY

Preberací protokol k dokumentácii č. 1

Tím č. 11_elf_



Autor:	<i>Tím č. 11_elf_</i>
Členovia tímu:	<i>Bc. Slavomír Červeň, Bc. Andrej Fenik, Bc. Martin Kováčik, Bc. Juraj Malečka, Bc. Marián Miština, Bc. Martina Práznovská, Bc. Michal Sabo</i>
Posudzovateľ:	<i>Tím č. 6 The Llama Team</i>
Členovia tímu:	<i>Bc. Martin Adam, Bc. Imrich Balko, Bc. Rudolf Dačo, Bc. Michal Habala, Bc. Ondrej Hluchý, Bc. Tomáš Klempa</i>

Predmet preberania: ***Dokumentácia projektu:***
Projektová dokumentácia – Úvod, Analýza, Špecifikácia, Hrubý návrh, Prílohy A, B, C
Dokumentácia k riadeniu projektu ku kontrolnému bodu 18.11.2005

Týmto potvrdzujem, že som od autora prebral dokumentáciu v uvedenom rozsahu.

Podpis posudzovateľa:

.....

za The Llama Team

Podpis autora:

.....

za tím_elf_

V Bratislave dňa 18. 11. 2005

**Preberací protokol k dokumentácii č. 2****Tím č. 11_elf_**

Autor:	<i>Tím č. 11_elf_</i>
Členovia tímu:	<i>Bc. Slavomír Červeň, Bc. Andrej Fenik, Bc. Martin Kováčik, Bc. Juraj Malečka, Bc. Marián Miština, Bc. Martina Práznovská, Bc. Michal Sabo</i>
Posudzovateľ:	<i>vedúci tímu – Ing. Ivan Kapustík</i>

Predmet preberania:	<i>Dokumentácia projektu: Projektová dokumentácia – Úvod, Analýza, Špecifikácia, Hrubý návrh, Prílohy A, B, C Dokumentácia k riadeniu projektu ku kontrolnému bodu 18.11.2005</i>
---------------------	--

Týmto potvrdzujem, že som od autora prebral dokumentáciu v uvedenom rozsahu.

Podpis posudzovateľa:

.....

Ing. Ivan Kapustík

Podpis autora:

.....

za tím_elf_

V Bratislave dňa 18. 11. 2005

**Preberací protokol k dokumentácii č. 3****Tím č. 11_elf_**

Autor:	<i>Tím č. 11_elf_</i>
Členovia tímu:	<i>Bc. Slavomír Červeň, Bc. Andrej Fenik, Bc. Martin Kováčik, Bc. Juraj Malečka, Bc. Marián Miština, Bc. Martina Práznovská, Bc. Michal Sabo</i>
Posudzovateľ:	<i>Tím č. 6 The Llama Team</i>
Členovia tímu:	<i>Bc. Martin Adam, Bc. Imrich Balko, Bc. Rudolf Dačo, Bc. Michal Habala, Bc. Ondrej Hluchý, Bc. Tomáš Klempa</i>

Predmet preberania:	<i>Dokumentácia projektu: Projektová dokumentácia – Úvod, Analýza, Špecifikácia, Hrubý návrh, Prototyp, Prílohy A, B, C Dokumentácia k riadeniu projektu ku kontrolnému bodu 19.12.2005</i>
---------------------	--

Týmto potvrdzujem, že som od autora prebral dokumentáciu v uvedenom rozsahu.

Podpis posudzovateľa:

.....

za The Llama Team

Podpis autora:

.....

za tím_elf_

V Bratislave dňa 19. 12. 2005

**Preberací protokol k dokumentácii č. 4****Tím č. 11_elf_**

Autor:	<i>Tím č. 11_elf_</i>
Členovia tímu:	<i>Bc. Slavomír Červeň, Bc. Andrej Fenik, Bc. Martin Kováčik, Bc. Juraj Malečka, Bc. Marián Miština, Bc. Martina Práznovská, Bc. Michal Sabo</i>
Posudzovateľ:	<i>vedúci tímu – Ing. Ivan Kapustík</i>

Predmet preberania:	<i>Dokumentácia projektu: Projektová dokumentácia – Úvod, Analýza, Špecifikácia, Hrubý návrh, Prototyp, Prílohy A, B, C Dokumentácia k riadeniu projektu ku kontrolnému bodu 19.12.2005</i>
---------------------	--

Týmto potvrdzujem, že som od autora prebral dokumentáciu v uvedenom rozsahu.

Podpis posudzovateľa:

.....

Ing. Ivan Kapustík

Podpis autora:

.....

za tím_elf_

V Bratislave dňa 19. 12. 2005



PRÍLOHY

ZOZNAM PRÍLOH

Príloha A: Formát extraktu dokladu o vykonaných skúškach a zápočtoch

Príloha B: Príloha B: Formát extraktu sumárneho hodnotenia študenta

Príloha C: Ponuka

Príloha D: Elektronický nosič – CD-ROM

\dokumentacia

\dokumentacia_KB3.pdf

\install

\axis

\jaf-1.0.2

\Jakarta-tomcat-4.1.18

\java

\PostgreSQL

\xerces-2_7_1

\prototyp

\Acess

\Parser

\Web_services



PRÍLOHA A

FIIT STU V BRATISLAVE
11.11.2005 Str. 1

DOKLAD O VYKONANÝCH SKÚŠKACH A ZÁPOČTOCH

/2001 - 2004/

Os.čís: XXXXX Meno: Priezvisko Meno
Dátum nar.: DD.MM.RRRR

Adresa: Ulica číslo, PSC Mesto

Štud. odbor: 4 9400 Informatika

Predmet	Typ	Rozsah	Kred.	Výsledok	Dátum

[hod/týž]

1. ročník

01 3-1403	Bezpečnosť a ochrana zdravia I.	Z/Z	1	0	0 + xxxx	28.11.2001
01 3-1404	Lab. cvičenia BOZ	Z/Z	0	2	0 + xxxx	15.11.2001
01 3-1501	Fyzika I.	S/Z	3	3	7 + xxxx	20.12.2001
01 3-1701	Lineárna algebra	S/Z	3	2	6 + xxxx	03.01.2002
01 3-1702	Matematická analýza I.	S/Z	4	2	7 + xxxx	16.01.2002
01 3-2201	Počítače	S/Z	3	1	5 + xxxx	10.01.2002
01 3-3400	Telesná kultúra - z	Z/Z	0	2	0 + xxxx	10.01.2002
01 3-3303	Anglický jazyk	Z/Z	0	2	0 + xxxx	14.12.2001
01 3-1706	Seminár z lineárnej algebry	Z/Z	0	2	0 + xxxx	18.01.2002
01 3-1502	Fyzika II.	S/L	3	3	7 + xxxx	17.05.2002
01 3-1703	Matematická analýza II.	S/L	4	4	9 + xxxx	14.06.2002
01 3-1704	Diskrétna matematika	S/L	2	2	5 + xxxx	03.06.2002
01 3-2200	Programovanie v jazyku "C"	S/L	3	2	6 + xxxx	27.05.2002
01 3-3304	Anglický jazyk /1.r./	Z/L	0	2	0 + xxxx	13.05.2002
01 3-3401	Telesná kultúra - 1	Z/L	0	2	0 + xxxx	21.05.2002
01 3-3527	Ekonómia	S/L	3	2	6 + xxxx	19.06.2002
01 3-2503	Elektrické obvody I.	S/L	2	2	5 + xxxx	07.06.2002

2. ročník

02 3-1119	Teória systémov	S/Z	3	2	6 + xxxx	08.01.2003
02 3-1714	Numerické metódy	S/Z	3	2	6 + xxxx	21.01.2003
02 3-2211	Programovacie techniky	S/Z	3	2	6 + xxxx	19.12.2002
02 3-2212	Logické systémy	S/Z	3	2	6 + xxxx	28.01.2003
02 3-2519	Elektrické obvody II/i	S/Z	3	3	7 + xxxx	14.01.2003
02 3-3313	Anglický jazyk /záp.2.r/	Z/Z	0	2	0 + xxxx	09.12.2002
02 3-3410	Telesná kultúra - z	Z/Z	0	2	0 + xxxx	09.12.2002
02 3-3516	História	S/Z	1	2	4 + xxxx	30.01.2003
02 3-1717	Pravdepodobnosť a štatistika	S/L	3	2	6 + xxxx	09.06.2003
02 3-2112	Elektronické systémy	S/L	3	2	6 + xxxx	26.05.2003
02 3-2213	Operačné systémy I	S/L	3	2	6 + xxxx	30.05.2003
02 3-2215	Teoretické základy informatiky	S/L	3	2	6 + xxxx	16.06.2003
02 3-2411	Digitálne komunikácie	S/L	3	2	6 + xxxx	20.06.2003
02 3-3314	Anglický jazyk /zap.2.r/	S/L	0	2	4 + xxxx	15.05.2003
02 3-1410	Bezpečnosť a ochrana zdr.práci II/2.r./Z/L	1	0	0 + xxxx	06.05.2003	
02 3-1411	Laboratorne cvičenia BOZP II	Z/L	0	4	0 + xxxx	30.04.2003
02 3-3411	Telesná kultúra - 1	Z/L	0	2	0 + xxxx	19.05.2003



3. ročník

03 3-1926	Meranie /i,t	S/Z	2	3	6 + xxxx	22.12.2003
03 3-2221	Architektúra poč. systémov I	S/Z	3	2	6 + xxxx	12.01.2004
03 3-2229	Strojovo orientované jazyky	S/Z	2	2	5 + xxxx	16.12.2003
03 3-3420	Telesná kultúra	S/Z	0	2	0 + xxxx	10.12.2003

FIIT STU V BRATISLAVE
11.11.2005 Str. 2

DOKLAD O VYKONANÝCH SKÚŠKACH A ZÁPOČTOCH

/2001 - 2004/

Os.čís: -||-
||-

Meno: -||-

Dátum nar.: -

Adresa: -||-

Štud. odbor: 4 9400 Informatika

Predmet	Typ	Rozsah	Kred.	Výsledok	Dátum
	Sem	Pr.Cv.	body		
	[hod/týž]				
03 3-1727	Analýza a zložitosť algoritmov	S/Z	3	2	6 + xxxx 22.01.2004
03 3-2228	Funkcionálne a logické prog.	S/Z	3	2	6 + xxxx 07.01.2004
03 3-2220	Modelovanie a simulácia	S/L	3	2	6 + xxxx 08.06.2004
03 3-2222	Stavba operačných systémov	S/L	3	2	6 + xxxx 16.06.2004
03 3-3421	Telesná kultúra	Z/L	0	2	0 + xxxx 14.05.2004
03 3-2224	Metódy a prostr. Špecifikácie	S/L	3	2	6 + xxxx 19.05.2004
03 3-2239	Objektovo - orientované prog.	S/L	3	2	6 + xxxx 15.06.2004
03 3-2236	Princípy softvér. inžinierstva	S/L	3	3	7 + xxxx 02.06.2004

4. ročník

04 7-1005	Počítačové siete	S/Z	3	2	6 + xxxx	22.12.2004
04 7-3074	Databázové systémy	S/Z	3	3	7 + xxxx	31.01.2005
04 7-2080	Záverečný projekt I	K/Z	0	8	6 + xxxx	14.01.2005
04 7-2078	Opis a preklad prog. jazykov	S/Z	3	2	6 + xxxx	10.01.2005
04 7-2085	Manažment kvality	S/Z	2	2	5 + xxxx	13.01.2005
04 7-3073	Počítačová grafika	S/Z	3	2	6 + xxxx	27.01.2005
04 7-1069	Integrované služby dig. Sietí	S/L	3	2	6 + xxxx	23.05.2005
04 7-2081	Záverečný projekt II	S/L	0	8	6 + xxxx	06.06.2005
04 7-1080	Vnorené systémy	S/L	2	3	6 + xxxx	25.05.2005
04 7-2079	Umelá inteligencia	S/L	3	2	6 + xxxx	31.05.2005

Získané známky:	1	1-	2	2-	3	3-	4	Celkový počet ukon.
predmetov:	56							
	3	6	12	8	1	2	0	Zapísané kredity: 252.0
Celkový bodový študijný priemer:	2.13							Získané kredity: 252.0

Poznámka:

Do sumárnych výsledkov (BŠP, ...) sú zahrnuté len predmety ukončené známkou alebo "Z"



+ Kredity predmetu sa študentovi započítajú
V "nekreditných" ročníkoch sa všetky predmety započítavajú s váhou "1"

Hodnotenie (KBH - koeficient bodového hodnotenia):

1	výborne	(1.0)	(nekr. 4.0)	1-	výborne m.	(1.5)	(nekr. 0.0)
2	veľmi dobre	(2.0)	(nekr. 3.0)	2-	veľmi dobre m.	(2.5)	(nekr. 0.0)
3	dobre	(3.0)	(nekr. 2.0)	3-	dobre m.	(3.5)	(nekr. 0.0)
4	dostatočne	(4.0)	(nekr. 0.0)	Z	započítané	(2.0)	(nekr. 0.0)

študijné výsledky troch študentov, výsledok je jedna z týchto hodnôt:

- výborne, výborne m.,
- veľmi dobre, veľmi dobre m.,
- dobre, dobre m.,
- dostatočné,
- započítané,
- uspokojivo

Poznámky:

- **uspokojivo** neznamená, že študent v skúške nevyhovel, znamená to, že nemal predmet zapísaný
- je použitý starý character encoding pre msdos
- export pre 1500 študentov bude mať cca 12 MB
- export obsahuje známky z bakalárskeho aj diplomového projektu



PRÍLOHA B

FIIT STU V BRATISLAVE
11.11.2005 Str. 1

S U M Á R N E H O D N O T E N I E

(za štúdium vrátane roku 2004/05)

Štud.zaradenie: 4 9400 Informatika

V: ručroc = "4" and (osc\$'xxxxx,xxxxx,xxxxx')

AA

Z N Á M K Y

BaP

PREDMETY KREDITY

1 1- 2 2- 3 3- 4 Z

ukon. z!sk.

AA

4 9400	xxxxx	Priezvisko	Meno	xx	xx	xx	xx	xx	xx	x	xx	x.xx
xx		xxx.x										
4 9400	- -	- -		xx	xx	xx	xx	xx	xx	x	xx	x.xx
xx		xxx.x										
4 9400	- -	- -		xx	xx	xx	xx	xx	xx	x	xx	x.xx
xx		xxx.x										

AA

Poznámka:

Do sumárnych výsledkov (BaP, ...) sú zahrnuté, len predmety ukončené, známou alebo "Z"

V "nekreditných" ročníkoch sa všetky predmety započítavajú s váhou "1"

Hodnotenie (KBH - koeficient bodového hodnotenia):

1	výborne	(1.0)	(nekr. 4.0)	1-	výborne m.	(1.5)	(nekr. 0.0)
2	veľmi dobre	(2.0)	(nekr. 3.0)	2-	veľmi dobre m.	(2.5)	(nekr. 0.0)
3	dobre	(3.0)	(nekr. 2.0)	3-	dobre m.	(3.5)	(nekr. 0.0)
4	dostatočne	(4.0)	(nekr. 0.0)	Z	započítané	(2.0)	(nekr. 0.0)



PRÍLOHA C



Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH
TECHNOLÓGIÍ

Ilkovičova 3, 812 19 Bratislava

PONUKA

Báza znalostí a zručností študentov (ZNALOSTI)

Predmet: Tvorba informačného systému v tíme I.
Študijný odbor: Informačné systémy
Október 2005
Tím č.: 11
Vedúci tímu: Ing. Vladimír Grlický

Bc. Slavomír Červeň
Bc. Andrej Fenik
Bc. Martin Kováčik
Bc. Juraj Malečka
Bc. Marián Miština
Bc. Martina Práznovská
Bc. Michal Sabo

Obsah

1 ZADANIE	1
2 ÚVOD	2
3 MOTIVÁCIA	3
3.1 NAŠA MOTIVÁCIA PRE VÝBER TÉMY	3
3.2 MOTIVÁCIA PRE ZÁKAZNÍKA	3
4 HRUBÝ NÁVRH RIEŠENIA	5
4.1 ZABEZPEČENIE POŽIADAVIEK ZADÁVATEĽA	5
4.1.1 Získavanie informácií o študentoch.....	5
Typy informácií o študentoch	5
Typy informačných zdrojov.....	7
4.1.2 Zabezpečenie pravidelného získavania informácií.....	7
4.1.3 Poskytnutie uchovávaných znalostí pre iné systémy.....	7
4.1.4 Konfigurovateľnosť, modulárnosť a univerzálnosť systému.....	8
4.1.5 Bezpečnosť systému.....	8
4.1.6 Ďalšie prínosy nášho riešenia.....	8
4.2 FUNKČNÁ ARCHITEKTÚRA	9
4.2.1 Popis prípadov použitia.....	10
4.3 TECHNICKÁ ARCHITEKTÚRA	12
4.3.1 Architektúra systému.....	12
Prepojenie systému Znalosti so systémom Študent	13
Prepojenie systému Znalosti s ostatnými systémami	13
4.3.2 Návrh použitých technológií.....	14
Spoločné črty riešení	14
Riešenie na platforme ASP.NET.....	15
Riešenie na báze Linux + PHP.....	16
Riešenie na báze Linux/Windows + Java.....	16
Zvolené riešenie.....	16
5 PRIBLIŽNÝ PLÁN PROJEKTU	18
6 TÍM	19
PRÍLOHA A: ZORADENIE TÉM PODEĽA PRIORITY	23
PRÍLOHA B: ROZVRH ČLENOV TÍMU	24

1 Zadanie

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 zužitkovanie 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 začlenení sa do určitého riešiteľského kolektívu. V súčasnosti však na fakulte neexistuje žiadny systém, ktorý by toto "preukazovanie 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, sťažie, 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 Úvod

Náš 7-členný tím sa od počiatku vytváral zo študentov, ktorí mali jednoznačný záujem o tému „Báza znalostí a zručností študentov“. Ako jediný tím z odboru Informačné systémy sme odhodlaní uchádzať sa o túto tému (ďalej len ZNALOSTI). Sme si vedomí konkurencie zo strany ostatných tímov a preto ponúkame svoje nápady a návrh riešenia na posúdenie. V tomto dokumente najprv predstavíme náš hrubý návrh riešenia a nakoniec predstavíme tím.

Projekt ZNALOSTI zahŕňa komplexný systém, ktorý poskytne objektívne informácie o študentoch FIIT. Výstupy z aplikácie pomôžu pedagógom, t. z. zabezpečia prehľad o znalostiach a zručnostiach študentov. Prínos pocítia i študenti pri výbere záverečného, diplomového príp. iného projektu. My navyše ponúkame riešenie, ktoré by svojou konfigurovateľnosťou mohlo byť využité aj v komerčnej sfére.

3 Motivácia

Kapitola uvádza našu motiváciu a dôvody pre výber témy „Báza znalostí a zručností študentov“. Rovnako obsahuje aj motiváciu pre zákazníka, teda čo ho môže viesť k rozhodnutiu prideliť riešenie témy práve nášmu tímu.

3.1 Naša motivácia pre výber témy

Pre výber témy sme sa rozhodli najmä vďaka jej dostatočnému potenciálu rozvinúť naše doterajšie znalosti v odbore štúdiá. Chceme nazbierať ďalšie zručnosti a rozšíriť svoj odborný profil o skúsenosti z tvorby reálne použiteľného informačného systému. Ide totiž o systém, ktorý by mal byť nasadený v reálnych podmienkach. Navyše máme vlastnú konkrétnu víziu riešenia problému.

Je pre nás výzvou vytvoriť produkt, ktorý bude slúžiť potrebám fakulty ale zároveň bude použiteľný aj v komerčnej sfére. Znalosti získané pomocou produktu môžu využiť spoločnosti na získanie prehľadu o uchádzačoch o zamestnanie. Našou snahou bude, aby parametre systému boli porovnateľné so systémami využívanými v komerčnej sfére.

Myslíme si, že výstup projektu by pomohol minimálne pri priradovaní študentov na tímové, záverečné a diplomové projekty. To by v konečnom dôsledku viedlo k efektívnejšiemu priradeniu študenta na úlohu, na ktorú je vhodný, čo sa odrazí na zvýšenej kvalite projektov ako aj na spokojnosti študentov a vyučujúcich.

V neposlednom rade nepochybujeme o dobrej komunikácii s vedúcim tohto projektu.

3.2 Motivácia pre zákazníka

Pri motivácii zákazníka k prideleniu projektu práve nášmu tímu by mohli pomôcť nasledovné faktory:

Znalosti – Členovia tímu disponujú potrebnými znalosťami na riešenie projektu. Ovládajú viaceré technológie, ktoré môžu byť použité na jeho implementáciu. Rovnako sú oboznámení so základnými metodológiami procesu tvorby softvérových a informačných systémov.

Skúsenosti – Takmer všetci členovia sa venovali, alebo venujú vývoju informačných systémov v komerčnej sfére. Ide jednak o menšie firmy ako aj o popredné slovenské a svetové spoločnosti. Disponujú teda skúsenosťami z praxe v oblasti analýzy, návrhu, implementácie, testovania, údržby a komunikácie so zákazníkmi.

Motivácia – Tím je vysoko motivovaný k riešeniu zadania témy. Táto bola takmer jednohlasne zaradená na prvé miesto zo všetkých. Je badateľný entuziazmus, s ktorým tím ako celok pristupuje k dosahovaniu čiastkových cieľov ako aj konečného cieľa, ktorým je úspešný výstup projektu.



Učelnivosť – Vďaka vysokej motivácii jeho členov je tím ochotný osvojiť si nové prístupy a princípy v tvorbe informačných systémov. Rovnako je schopný rozšíriť svoje znalosti potrebných technológií. Výhodou je aj účasť na niektorých relevantných kurzoch prebiehajúcich tento semester v rámci inžinierskeho štúdia.

Úspešnosť – Traja z členov tímu sú držiteľmi pochvalného uznania dekana Fakulty informatiky a informačných technológií. Členovia však dosahujú výborné výsledky nielen v štúdiu, ale aj v komerčnom sektore. Pracujú na vývoji technológií, ktoré sú v praxi reálne zavádzané a používané.

Argumentmi v prospech tímu sú v neposlednom rade aj jeho kvalitné osobnostné zloženie, veľmi dobrá úroveň komunikácie spolu s otvorenou a tvorivou atmosférou v jeho vnútri.

4 Hrubý návrh riešenia

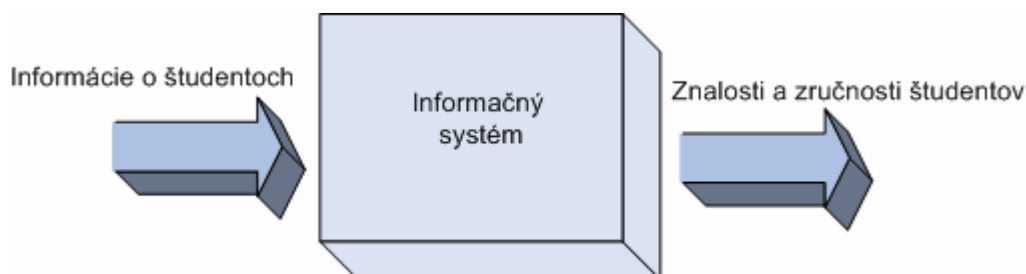
Návrh riešenia vychádza z detailného preštudovania problémovej oblasti a pochopenia požiadaviek na požadované softvérové riešenie. K spresneniu navrhovaného riešenia dôjde vo fáze „Analýza a návrh riešenia“.

4.1 Zabezpečenie požiadaviek zadávateľa

Vzhľadom na to, že hlavným cieľom systému **Báza znalostí a zručností študentov** je umožniť efektívne poskytovanie informačných služieb pre cieľové skupiny používateľov, považujeme za dôležité zabezpečiť pri návrhu a realizácii riešenia nasledovné požiadavky zadávateľa:

4.1.1 Získavanie informácií o študentoch

Základným vstupným údajom, v systéme sú informácie o študentoch. Systém tieto informácie vhodným spôsobom pretransformuje a výstupom zo systému budú znalosti o študentoch a ich zručnostiach ako je to znázornené na obrázku (Obrázok 1).



Obrázok 1: Transformácia údajov v systéme.

Informácie o študentoch môžu byť rôzneho charakteru a môžu do systému vstupovať z rôznych zdrojov.

Definovali sme nasledovné typy informácií a informačných zdrojov:

Typy informácií o študentoch

Tabuľka 1 predstavuje prvé priblíženie, ktoré informácie by sa mohli objaviť vo výslednom informačnom systéme.



Meno študenta
Bc./Ing. štúdium ukončil s vyznamenaním
Téma záverečného projektu
Abstrakt
Téma diplomového projektu
Abstrakt
Ocenenia: pochvalné uznanie dekana za výborné študijné výsledky, záverečný projekt a pod.
Študijné výsledky v predmetoch tvoriacich určitú skupinu (skupiny môžu byť konfigurovateľné), napríklad: <ul style="list-style-type: none">▪ 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)
Súťaže: Programátorská súťaž ACM, regionálne kolo (rok, umiestnenie), CSIDC, ...
Školenia: Cisco Networking Academy a pod.
Mimoškolské aktivity: zamestnanie, účasť na projektoch ...
Ovládané technológie spolu so stupňom ich ovládania
Záujmy študenta
Referencie: <ul style="list-style-type: none">▪ od vedúceho ZP▪ od zamestnávateľa▪ od vedúceho pracovníka počas praxe študenta
Životopis
Študentove vlastné slová, napr. ako sa chce profilovať, vyjadrenie k vyššie uvedenému, príp. doplnenie
Index spoľahlivosti

Tabuľka 1: Hrubý návrh ukladaných dát o študentoch

Komplexný obraz o študentovi nám poskytne tzv. **výsledný index hodnotenia**, ktorý bude vypočítaný algoritmom konfigurovateľnosti na základe vybraných možností (napr. študent je v TOP 5% najlepších študentov z matematických predmetov, ovláda programovacie jazyky C# a Java na výbornej úrovni, nevlastní žiaden certifikát, jeho index spoľahlivosti je 100 a výsledný index hodnotenia je 80).

- Stupnica a významnosť indexov ako aj konkrétne algoritmy budú spresnené vo fáze podrobného návrhu systému.

Typy informačných zdrojov

Priamy vstup od študenta, napríklad: formulár
Priamy vstup od pedagogického pracovníka, napríklad: formulár
Priamy vstup od vedúceho pracovníka počas praxe študenta, napríklad: formulár
Vstup zo systému Študent, ktorý obsahuje študijné výsledky študenta
Vstup z iných systémov

Tabuľka 2: Hrubý návrh typov informačných zdrojov

4.1.2 Zabezpečenie pravidelného získavania informácií

System umožní notifikáciu prostredníctvom e-mailu, ktorá zabezpečí pravidelné získavanie informácií od študenta. V tejto súvislosti sme zaviedli tzv. **index spoľahlivosti študenta**, ktorý sa môže meniť v závislosti od frekvencie obnovovania informácií zo strany študenta a môže vypovedať o jeho spoľahlivosti a schopnosti plniť si svoje úlohy. Bude teda v záujme študenta, aby aktualizoval svoje údaje v systéme častejšie.

4.1.3 Poskytnutie uchovávaných znalostí pre iné systémy

Jednou z častí nášho systému je modul pre poskytovanie uchovávaných znalostí pre iné systémy. Ak uvažujeme nasadenie systému v akademickom prostredí na našej fakulte (FIIT STU), takýmto systémom môže byť systém YONBAN, ktorý slúži na podporu riadenia projektov. Takéto riešenie by výrazne uľahčilo výber a pridelenie študentov do jednotlivých projektov.

Ďalším z možných systémov, ktorý by využíval uchovávané znalosti a zručnosti študentov nášho systému môžu byť portály pracovných príležitostí. Takéto riešenie by zefektívnilo a uľahčilo proces uplatnenia študenta na trhu práce.

Riešenie v uvedenej podobe by vyžadovalo vytvorenie modulov v spomínaných systémoch, ktoré by dokázali spolupracovať s našim systémom.

4.1.4 Konfigurovateľnosť, modulárnosť a univerzálnosť systému

Každý modul systému je **autonómny** subsystém, ktorý obsahuje ucelenú množinu funkcií a dátových štruktúr. **Univerzálnosť** systému spočíva v jeho použití nielen na akademickej pôde, ale aj v komerčnej sfére. Tento cieľ plánujeme dosiahnuť s využitím tzv. zásuvných modulov (pluginov), ktoré budú predstavovať funkčnosť systému. Jednoduchou zmenou modulu potom bude možné meniť a rozširovať funkcionálnosť systému. Napríklad pri nasadení systému povedzme vo firemnom prostredí budú zbierané informácie celkom určite iného charakteru (produktivita práce zamestnancov, a pod.).

Dôležitou vlastnosťou nášho systému je jeho **konfigurovateľnosť**. Pre rôzne cieľové skupiny používateľov systém poskytuje možnosť konfigurácie parametrov (na základe typov informácií o študentoch) a tým umožní získať rôzne druhy výstupných znalostí o študentoch. **Algoritmus konfigurovateľnosti** systému teda umožňuje **nezávislú** konfiguráciu jednotlivých možností získavania výstupných znalostí a ich kombinácie.

4.1.5 Bezpečnosť systému

Oprávnenie pracovať so systémom majú iba **autorizovaní** používatelia. Používateľ môže byť zároveň členom jednej alebo viacerých skupín, pričom jednotlivé skupiny majú rôzne práva pri práci so systémom, ktoré zabezpečí proces **autentifikácie**.

V hrubom návrhu sme navrhli nasledovné typy používateľov:

- študent (osoba)
- Študent (fakultný informačný systém)
- pedagogický pracovník
- vedúci pracovník študenta počas praxe
- administrátor systému

Pri návrhu sa zameriame na dodržiavanie ochrany osobných informácií.

4.1.6 Ďalšie prínosy nášho riešenia

- naše riešenie zabezpečí jednoduchý a rýchly prístup k informáciám nezávisle od toho, či sa používateľ nachádza doma alebo v zahraničí
- precíznym projektovaním a použitou technológiou sa zabezpečí robustnosť, stabilná prevádzka a odolnosť systému voči chybám
- používateľské rozhranie bude maximálne prehľadné s jednoduchou orientáciou

- dostupnosť systému nie je časovo obmedzená. Hlavné služby, ktoré sú potrebné na jeho prevádzku sú spustené neustále
- systém bude viacjazyčný a umožní jeho prípadné využitie aj v zahraničí
- zanalyzujú a navrhnu sa procesy naplňania databázy

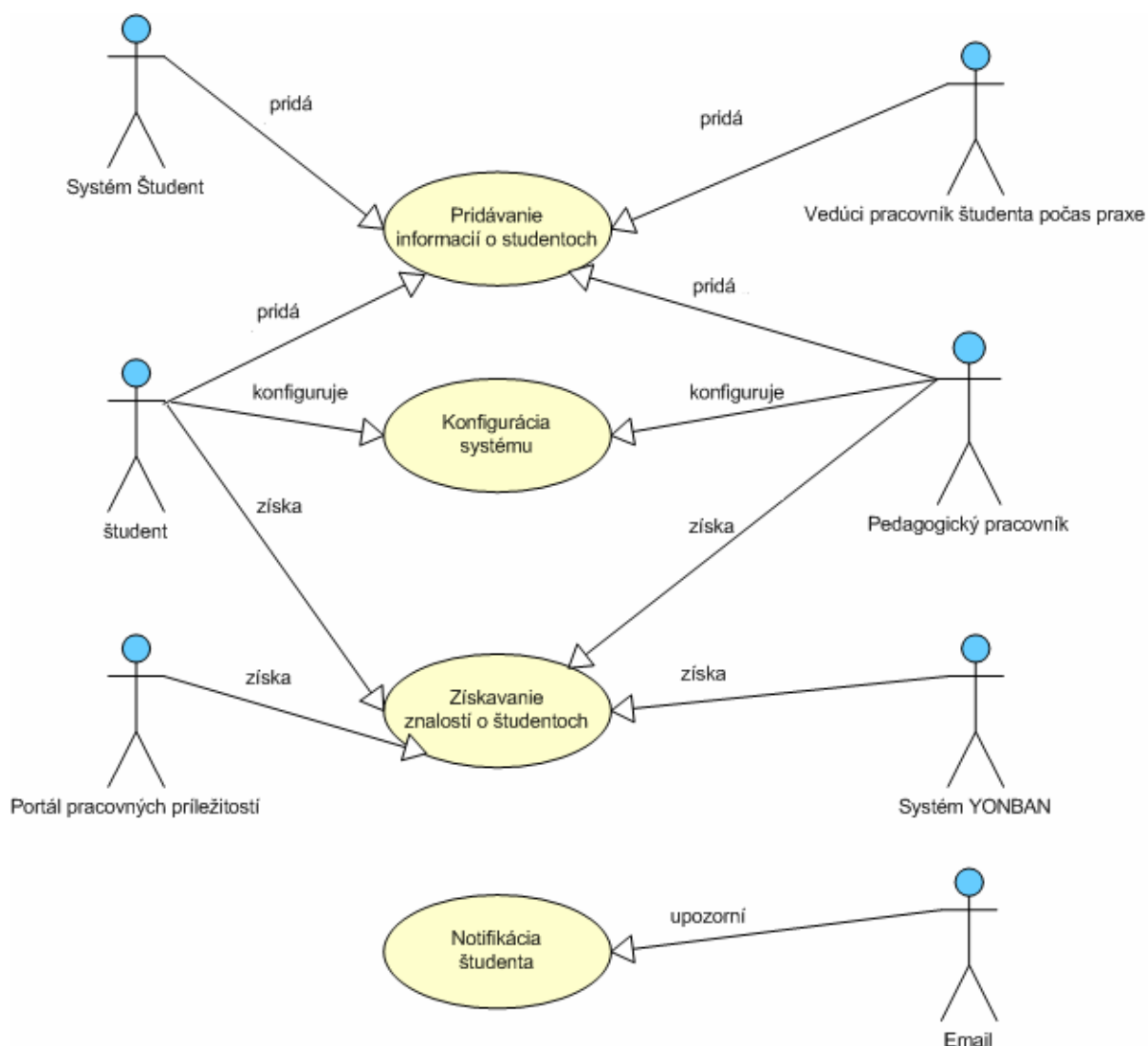
4.2 Funkčná architektúra

Na základe požiadaviek zadávateľa sme definovali základné funkčné oblasti systému:

- pridávanie informácií o študentoch
- konfigurácia systému
- získavanie znalostí o študentoch
- notifikácia študenta

Na nasledujúcom obrázku (Obrázok 2) je v podobe diagramu prípadov použitia znázornený prístup jednotlivých typov používateľov k informáciám uloženým v jednotlivých funkčných oblastiach. Kvôli prehľadnosti nie je uvedený medzi používateľmi administrátor systému (má prístup ku všetkým oblastiam systému).

Tento diagram znázorňuje použitie systému na akademickej pôde. V prípade použitia systému v inej oblasti by sa diagram mierne modifikoval.



Obrázok 2: Diagram prípadov použitia

4.2.1 Popis prípadov použitia

Prípad použitia	Pridávanie informácií o študentoch
Popis	<p>Umožňuje pridávanie informácií o študentoch autorizovaným používateľom. Tieto informácie môžu byť rôzneho druhu.</p> <p>Systém Študent vkladá informácie o študijných výsledkoch študenta. Pedagogický vedúci a vedúci pracovník študenta počas praxe pridávajú hodnotenie študentov a ich referencie. Samotný študent pridáva ostatné typy informácií ako to bolo uvedené vyššie</p>
Funkčnosť	<p>Pridanie informácie</p> <p>Aktualizácia informácie</p>



	Vyhľadanie informácie
--	-----------------------

Prípado použitia	Konfigurácia systému
Popis	<p>Umožňuje konfiguráciu systému na základe rôznych parametrov pre rôzne cieľové skupiny. V diagrame nie je zakreslená možnosť konfigurácie pre systémy (YONBAN, Portály pracovných príležitostí) kvôli prehľadnosti, ale samozrejme aj tieto systémy budú mať možnosť konfigurovať náš systém a získavať požadované znalosti.</p> <p>Každý autentifikovaný používateľ bude mať rôzne možnosti konfigurácie na základe autorizačných oprávnení.</p>
Funkčnosť	<p>Konfigurovanie parametrov pre poskytovanie znalostí</p> <p>Konfigurovanie systému</p>

Prípado použitia	Získavanie znalostí o študentoch
Popis	<p>Umožňuje získavanie znalostí o študentoch a ich zručnostiach pre rôzne cieľové skupiny na základe ich oprávnenia.</p> <p>Študent môže získavať znalosti pre potreby výberu kolegov v rámci tímového projektu.</p> <p>Vedúci pracovník môže získavať znalosti pre potreby výberu študenta v rámci DP alebo ZP a pod.</p>
Funkčnosť	<p>Poskytnutie znalostí na základe rôznych kritérií</p> <p>Poskytnutie zručností na základe rôznych kritérií</p>

Prípado použitia	Notifikácia študenta
Popis	<p>Umožňuje pravidelné zasielanie správ študentom pre potreby obnovenia informácií o znalostiach a zručnostiach. Táto činnosť sa vykonáva automatizovane na základ konfigurácie systému.</p>
Funkčnosť	Zaslanie správy študentovi

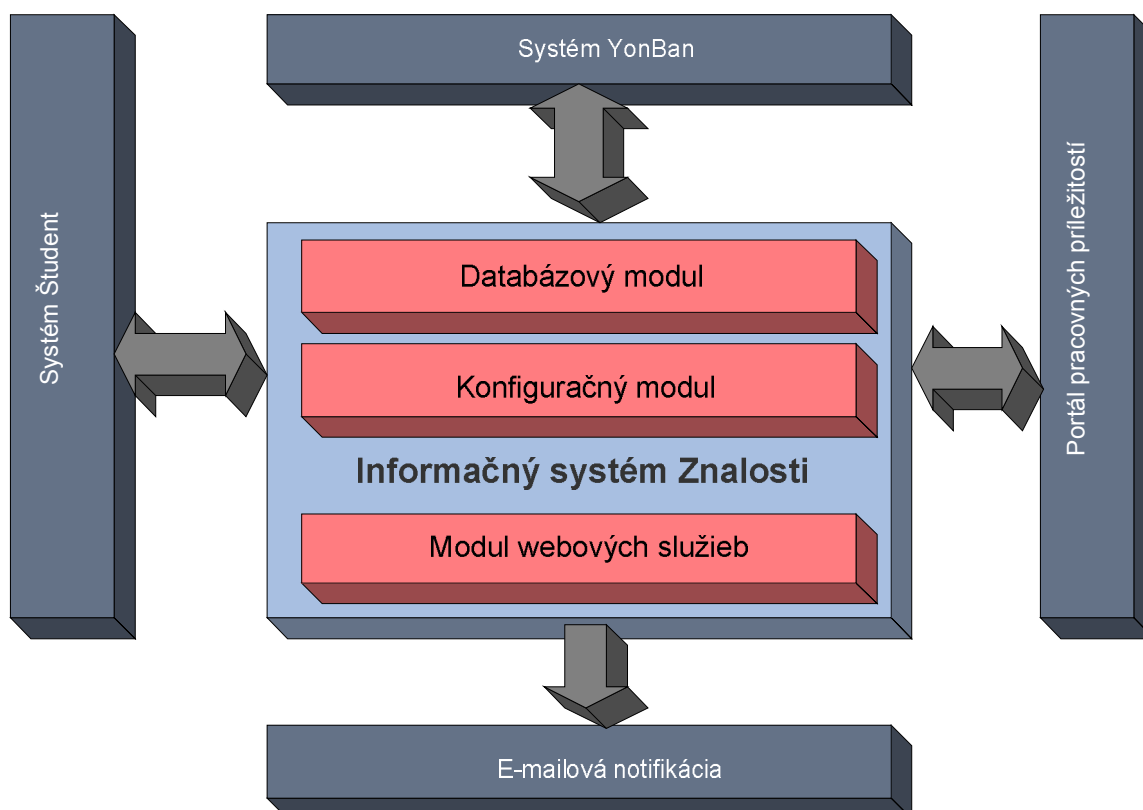
4.3 Technická architektúra

V tejto podkapitole predkladáme možné návrhy technickej realizácie nášho riešenia, definujeme požiadavky na dané technológie, ich výhody a nevýhody a zároveň odporúčame najvhodnejšiu technológiu.

4.3.1 Architektúra systému

Na obrázku (Obrázok 3) môžeme vidieť architektúru systému **Znalosti** ako sme ju navrhli vo fáze hrubého návrhu. Hlavnými časťami systému sú tri moduly: databázový modul, konfiguračný modul a modul webových služieb.

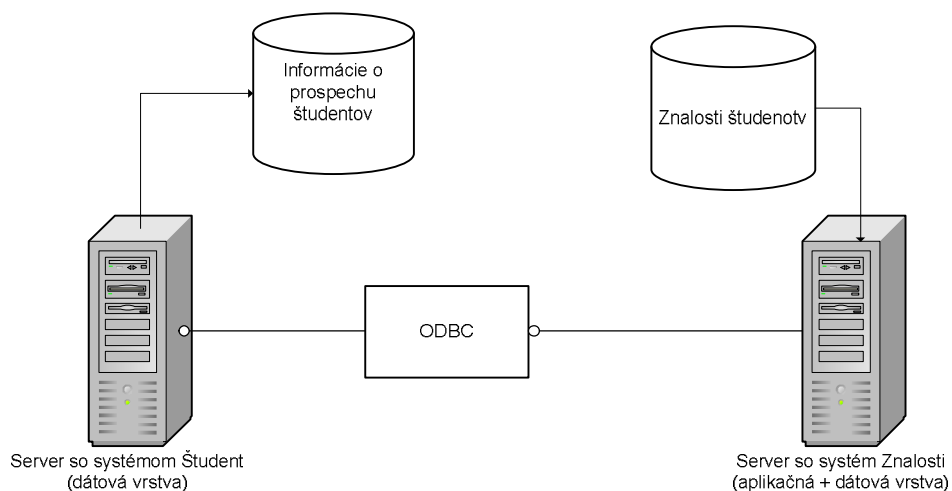
Databázový modul ukladá informácie o študentoch, používateľoch systému a iné. Konfiguračný modul zabezpečuje rôzne možnosti konfigurácie systému, ktorý pozostáva z ďalších podmodulov v závislosti od použitia systému. V prípade, že sa zmení oblasť použitia, pridajú sa nové podmoduly, ktoré budú tvoriť nový konfiguračný modul. Modul webových služieb poskytuje žiadané služby iným systémom, ako to bolo uvádzané vyššie v tejto kapitole.



Obrázok 3: Architektúra systému

Prepojenie systému Znalosti so systémom Študent

Jednou z požiadaviek na navrhovaný IS je použitie údajov z existujúceho IS Študent. Systém Študent podľa nám dostupných informácií používa databázový systém FoxPro, na ktorý je možné napojiť sa pomocou programovacieho rozhrania ODBC, ako to zobrazuje obrázok (Obrázok 4).

**Obrázok 4:** Prepojenie systému znalosti so systémom študent

Prepojenie systému Znalosti s ostatnými systémami

Na prepojenie so systémom Yonban je potrebné doplniť nový modul do tohto systému, ktorý bude vedieť komunikovať so systémom Znalosti. Podľa dostupných informácií je systém Yonban postavený na technológii JSP (Java Server Pages).

Rovnako aj pre systémy pracovných príležitostí by bolo potrebné dorobiť moduly, ktoré by dokázali spracovať výstupné informácie z nášho systému.

V prípade e-mailovej notifikácie je situácia jednoduchá. V systéme bude modul, ktorý bude pravidelne posilať e-maily študentom a upozorňovať ich na potrebu aktualizácie údajov.

Databázový modul

Voľba databázovej technológie závisí od technologického riešenia. O týchto riešeniach sa zmienime nižšie v tejto kapitole.

Konfiguračný modul

Úlohou tohto modulu je umožniť konfiguráciu systému. Ako najvýhodnejší spôsob sa nám javí konfigurácia pomocou webového rozhrania.

Modul webových služieb

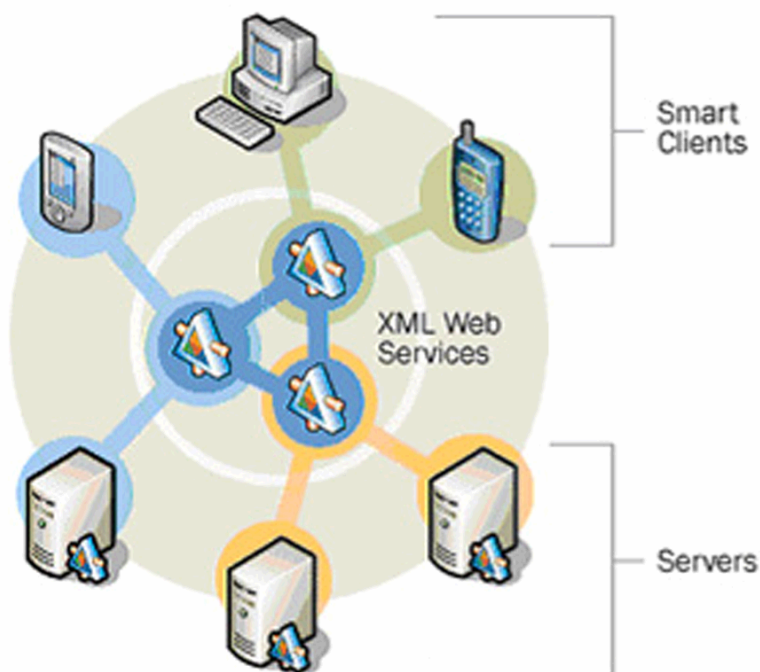
Tento modul zabezpečí komunikáciu s ostatnými systémami. Služby, budú dostupné vo formáte XML.

4.3.2 Návrh použitých technológií

Strategickým rozhodnutím pri návrhu je výber technológií, ktorý musí rešpektovať možnosti fakulty a zároveň poskytovať kvalitný základ pre výsledný systém. Z predošlého je jasné, že sa ponúka niekoľko riešení. V nasledujúcom texte sú najvýznamnejšie riešenia stručne charakterizované.

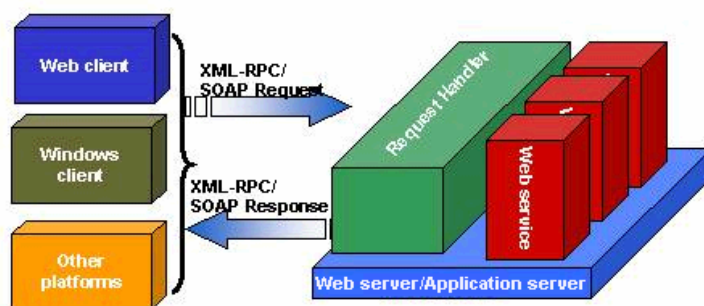
Spoločné črty riešení

Naše navrhované riešenie je založené na architektúre klient-server a využíva XML webové služby. Klienti budú požiadavky posielat' vo forme XML otázok, na ktoré bude server odpovedať vo forme XML odpovedí. Táto situácia je znázornená na nasledujúcom obrázku (Obrázok 5).



Obrázok 5: Klient-server architektúra

Detailnejší pohľad na XML webovú službu ponúka nasledovný obrázok (Obrázok 6).



Obrázok 6: Detailnejší pohľad na XML webovú službu

Klient-server architektúra je pri riešení podobných projektov používaná z viacerých dôvodov:

- pri vhodnom návrhu rozhrania sa nemusíme zaoberať tým, aký typ klienta sa u používateľa nachádza. Používateľ môže na interakciu so systémom použiť vhodne navrhnutého webového klienta (tak, aby dokázal komunikovať s webovou službou), ale napríklad aj klienta v mobilnom telefóne, alebo aplikáciu v operačnom systéme Windows. Klientom taktiež môže byť ďalší informačný systém,
- systém je ľahko prenositeľný na príbuzné architektúry,
- vysoký stupeň modularity umožňuje zapracovať požiadavky na zmenu aj vo vyšších štádiách vývoja,
- moduly sa dajú navrhnuť a implementovať tak, aby boli čo najmenej viazané, a teda použiteľné aj v iných projektoch.

Riešenie na platforme ASP.NET

Spoločnosť Microsoft ponúka riešenia s využitím technológie ASP (active server pages). Nevýhodou tejto technológie je, že je úzko zviazaná s operačným systémom Windows a teda systémy využívajúce túto technológiu sú neprenositeľné na iné operačné systémy. Technológiu .NET je vhodné používať s web serverom Microsoft IIS, ktorý je dodávaný so serverovými verziami OS Windows. Pri využití tejto technológie je vhodné zvoliť niektorý z nasledujúcich databázových serverov:

- Microsoft Jet Engine (súčasť kancelárskeho balíka MS Office)
- Microsoft SQL server
- Microsoft Data Engine (MSDE)

Veľkou nevýhodou takéhoto riešenia je licenčná politika firmy Microsoft. Väčšinu modulov, ktoré by tvorili softvérovú podporu nášho riešenia, by bolo potrebné zakúpiť, čo je v našej situácii veľká nevýhoda.



Riešenie na báze Linux + PHP

Riešenie na báze jazyka PHP je pomerne zaujímavé, pretože poskytuje výbornú funkčnosť a je k dispozícii zadarmo pod licenciou GNU GPL. K PHP existuje veľké množstvo knižníc na rôzne účely (práca s XML, šifrovanie, ...) a väčšina je k dispozícii tiež pod licenciou GNU GPL. Výhodou jazyka PHP je tiež podpora veľkého množstva operačných systémov, web serverov a databázových systémov. PHP sa najčastejšie používa spolu s web serverom Apache a databázovým systémom MySQL (samozrejme je možnosť použiť veľa iných DB, napríklad PostgreSQL, ktorá je šírená pod licenciou GNU GPL) v OS Linux. Takáto architektúra je známa pod pojmom LAMP. Výhodou PHP je taktiež veľmi dobrá dokumentácia s množstvom príkladov. Nevýhodou PHP je, že nie je veľmi vhodný pre väčšie projekty.

Riešenie na báze Linux/Windows + Java

Ďalšou možnosťou je riešenie na platforme Linux s využitím Java Servletov. Toto riešenie, podobne ako predchádzajúce, je prenositeľné aj na operačné systémy MS Windows, MacOS a iné. Využíva štandardný programovací jazyk JAVA, čo umožňuje použiť množstvo existujúcich modulov JAVA (packages) a tým zrýchliť a zjednodušiť vývoj aplikácie. Umožňuje prístup k väčšine existujúcich databázových systémov. Vzhľadom na obrovský potenciál jazyka JAVA toto riešenie tiež umožňuje neskoršie rozšírenie systému na viacero navzájom komunikujúcich systémov. Pri tejto forme riešenia navrhujeme využívať aplikačný server Tomcat a databázový systém PostgreSQL. Výhodou takéhoto riešenia je podobne ako v predchádzajúcom prípade jeho cena – je zadarmo. Ďalším pozitívom je skutočnosť, že toto riešenie je vhodné aj pre veľké projekty.

Zvolené riešenie

Rozhodli sme sa pre riešenie na platforme Linux+Java+PostgreSQL. K tomuto riešeniu sme sa priklonili aj preto, že vidíme obrovský potenciál tejto technológie a chceme jej lepšie porozumieť. V prospech platformy Linux+Java+PostgreSQL hovorí aj skutočnosť, že táto kombinácia už bola mnohokrát odskúšaná a väčšinou boli splnené všetky požiadavky na systém ako aj na jednotlivé komponenty systému. Toto riešenie nemá žiadne špeciálne nároky na hardvér ani softvér. Všetok potrebný hardvér je dostupný na katedre a softvérové komponenty je možné voľne získať z Internetu.

Server (XML webová služba) bude pozostávať z nasledovných súčastí:

- “jazyk”, v ktorom bude webová služba napísaná
- webový/aplikačný server, na ktorom bude webová služba bežať
- databázový server ako úložisko dát pre informačný systém



Na autorizáciu a autentifikáciu je možné použiť niektoré už existujúce riešenie napr. Java Authentication and Authorization Service alebo bezpečnostné možnosti použitej databázovej technológie. Pri návrhu použijeme UML a RUP. Pri testovaní použijeme štandardné postupy a dokumenty (testovací plán, matica požiadaviek a testov, detailné testcase) pre integračné a akceptačné testovanie.

5 Približný plán projektu

Na vypracovanie projektu bude potrebných približne 1680 MH (človekohodín), čo predstavuje prácu siedmich ľudí 10 hodín týždenne počas dvoch semestrov. V tabuľke č. 3 sa nachádza približný plán činností v zimnom semestri. Popri týchto činnostiach bude prebiehať činnosť riadenia tímu spolu s dokumentovaním tejto činnosti (úlohy v tíme, vedenie stretnutí k projektu, zápisy zo stretnutí, projektový denník) a vytvorenie a priebežná aktualizácia web prezentácie stavu projektu.

	Tý	Prednokladaná činnosť
i m n ý	3 –	▪ odovzdanie a prezentácia ponuky, pridelenie témy zadania, vytvorenie plánu projektu
	4 – 8	▪ analýza problému, špecifikácia požiadaviek a hrubý návrh riešenia ▪ vytvorenie dokumentácie analýzy problému, špecifikácie požiadaviek riešenia spolu s hrubým návrhom ▪ priebežná tvorba dokumentácie analýzy problému, špecifikácie požiadaviek riešenia spolu s hrubým návrhom
s e m e s	8 – 9	▪ odovzdanie dokumentácie analýzy problému, špecifikácie požiadaviek riešenia spolu s hrubým návrhom ▪ vytvorenie posudku analýzy, špecifikácie a hrubého návrhu iného tímu ▪ odovzdanie posudku analýzy, špecifikácie a hrubého návrhu iného tímu
	9 – 12	▪ dopracovanie zistených nedostatkov ▪ návrh a implementácia prototypu vybraných častí systému ▪ vytvorenie dokumentácie a používateľskej prezentácie prototypu
r	12	▪ odovzdanie prototypu vybraných častí systému spolu s dokumentáciou a používateľská prezentácia prototypu

Tabuľka 3: Približný plán práce v zimnom semestri

6 Tím

Na záver predstavujeme členov nášho tímu. Všeobecne stačí pripomenúť, že všetci páni absolvovali bakalárske štúdium na Fakulte Informatiky a Informačných Technológií STU a všetci pokračujú v inžinierskom štúdiu v odbore Informačné systémy. Títo študenti pracujú v softvérových firmách, majú bohaté skúsenosti s vývojom informačných systémov a ich tvorbou sa chcú zaoberať aj v budúcnosti.

Bc. Slavomír Červeň

Je absolventom bakalárskeho štúdia na FIIT STU v odbore Informatika, špecializácia Počítačové systémy a siete. V záverečnom projekte sa venoval návrhu a implementácii P2P systému na zdieľanie súborov v jazyku C#. Počas štúdia pracoval vo viacerých firmách, kde nazbieral skúsenosti a znalosti z oblasti administrácie operačných systémov Linux a Windows. Ako lektor vo firme poskytujúcej kurzy výpočtovej techniky sa naučil mnohým veciam, ale hlavne asertívnemu prístupu k ľuďom. Má skúsenosti z programovacími jazykmi C/C++ na platforme Linux aj Windows. Ďalej ovláda skriptovacie jazyky Bash, Cshell, Perl, PHP, databázové technológie MSSQL, MySQL, PostgreSQL, MS Access. Z webových technológií ovláda HTML, DHTML, XML, CSS, JavaScript, VBScript. Ďalej ovláda jazyky VHDL, UML, Pascal, Visual Basic. V poslednom čase sa venuje a pracuje s technológiou .NET a jazykom C#. V súčasnosti okrem práce na projekte pre logistické centrum firmy Siemens (webová aplikácia v ASP.NET) pracuje v softvérovej spoločnosti Softec (vývoj webovej aplikácie v ASP.NET pre poisťovací systém).

Bc. Andrej Fenik

Zbiera praktické skúsenosti s databázovými systémami ORACLE a v programovaní PL/SQL, Perl a XML pri spolupráci na rozsiahlom projekte v komerčnej sfére počas prázdnin a popri škole. Má skúsenosť v prácach na medzinárodnom projekte s rozsiahlymi a modernými procesmi tvorby softvéru. Zručnosti v programovaní v linuxovom prostredí a v Perl-e nadobudol vo voľnom čase, počas štúdia a tiež pri riešení bakalárskeho projektu z oblasti bezpečnosti sietí. Skúsenosti s tvorbou oknových aplikácií v C++/MFC nadobudol už v 2. ročníku štúdia. Ďalšie jeho klady sú: znalosť UML, tvorba analytického dokumentu, technológie Java, databázy Postgre, HTML/CSS/JavaScript.

Bc. Martin Kováčik

Ukončil bakalárske štúdium v odbore informatika s výbornými výsledkami. Má vynikajúcu znalosť jazykov C/C++, SQL, PHP, VHDL a JavaScript a VBscript. Výsledkom jeho bakalárskej práce je webový informačný systém pre potreby regionálneho zastúpenia CISCO akadémie pri FIIT STU. Pri riešení tohto projektu sa oboznámil s vývojom internetových aplikácií založených na architektúre LAMP (Linux, Apache, MySQL, PHP), zdokonalil sa v analýze a návrhu informačných systémov a osvojil si štandardy XML, XHTML a CSS2. Ovláda programovanie sieťových klient-server aplikácií s využitím knižnice WinSock a má skúsenosti s tvorbou aplikácií využívajúcich knižnicu MFC. Obľubuje vývoj aplikácií ako v OS Windows tak aj v OS Linux. Momentálne sa venuje štúdiu jazyka Java. V rámci inžinierskeho štúdia má zapísané predmety Princípy informačných systémov, Bezpečnosť a manažment informačných systémov, Pokročilé databázové technológie, Kvalita programových a informačných systémov a Základy kryptológie, ktoré nepochybne budú znamenať ďalší prínos pri riešení tohto projektu.

Bc. Juraj Malečka

V súčasnosti popri štúdiu pracuje ako výskumný pracovník na projekte „Nástroje pre získavanie, organizovanie a udržiavanie znalostí v prostredí heterogénnych informačných zdrojov“ na pôde FIIT. Praktické skúsenosti získané počas štúdia spoluprácou na rôznych projektoch zahŕňajú tvorbu internetových a intranetových aplikácií skĺbením technológií PHP/MySQL a C++ s použitím MFC a Windows Sockets. Má skúsenosti s návrhom a implementáciou menších informačných systémov v prostredí PHP/MySQL a Microsoft Access a programovaním v jazykoch C/C++ na platforme Linux aj Windows. Ďalšie relevantné znalosti: štandard XML, modelovací jazyk UML. Získal pochvalné uznanie dekana za veľmi dobré študijné výsledky a výborne vypracovaný záverečný projekt s názvom „Prostriedky pre podporu výučby počítačových sietí“.

Bc. Marián Miština

Má skúsenosti s návrhom a tvorbou menších informačných systémov ako sú statické a PHP/MySQL web stránky, ako aj MS Access databázy. Počas práce na projektoch nadobudol zručnosti v návrhu a modelovaní pokročilými metódami ako je UML. Svoje základné znalosti z oblasti bezpečnosti a manažmentu informačných systémov rozvíja aj v rámci rovnomeného aktívneho kurzu. Ovláda programovacie jazyky C a C++ a technológie MySQL, PHP, CSS, HTML, Flash a základy DHTML a JavaScriptu. V súčasnosti spolupracuje v rámci štvorčlenného tímu na tvorbe elektronického obchodu.

Bc. Martina Práznovská



Ukončila bakalárske štúdium informatiky na Fakulte prírodných vied Univerzity sv. Cyrila a Metoda v Trnave. Výsledok bakalárskej práce je portál s testami nemeckého jazyka s automatickým vyhodnocovaním, ktorý je postavený na PHP v spolupráci s databázou MySQL. Okrem jazykov C, Java, HTML, PHP ovláda aj štatistické metódy vyhodnocovania. Vedomosti zo štatistiky a teórie pravdepodobnosti nadobudla v štúdiu odboru Kvantitatívne metódy v ekonómii a podnikaní. Tieto znalosti sa dajú taktiež efektívne využiť pri analyzovaní a riešení tohto projektu. Má skúsenosti s prácou v tíme, ktoré získala počas testovania bankového informačného systému. V tomto projekte môže byť prínosom aj vzhľadom na to, že na pôdu fakulty prichádza z vonkajšieho prostredia.

Bc. Michal Sabo

Praktické skúsenosti s prácou v tíme nadobudol riešením rozsiahlych komerčných produktov v pozícii vývojár. Má skúsenosti s tvorbou menších (aj klient-server) aplikácií v prostredí Windows a Unix/Linux s využitím jazyka C/C++ a knižníc MFC, Qt a GTK. Pri tvorbe web aplikácií sa stretol s jazykom PHP, databázovým systémom MySQL a využil štandardy XHTML, XML a CSS2. V rámci inžinierskeho štúdia má zapísané predmety Pokročilé databázové technológie a Základy kryptológie, ktoré by mohli mať prínos pri riešení projektu.



PRÍLOHY



PRÍLOHA A: Zoradenie tém podľa priority

- | | |
|---|-------------|
| 1. Báza znalostí a zručností študentov | (ZNALOSTI) |
| 2. Systém na evidenciu a prezentáciu absolventov | (ALUMNI) |
| 3. Podpora zverejňovania informácií o fakulte | (FIIT-INFO) |
| 4. Obaľovač na získavanie pracovných ponúk | (WRAPPER) |
| 5. Kandidát na najlepší multimedialny produkt roku 2006 | (EuroPrix) |
| 6. Portál pracovných príležitostí | (JOBS) |
| 7. Tvorba rozvrhov | (ROZVRH) |



2	Feník Andrej
3	Kováčik Martin
4	Malečka Juraj
5	Mišťina Marián
6	Práznovská Martina
7	Sabo Michal



PRÍLOHA D

- Elektronický nosič – CD-ROM k časti PROTOTYP