SLOVENSKÁ TECHNICKÁ UNIVERZITA
Fakulta informatiky a informačných technológií

BÁZA ZNALOSTÍ A ZRUČNOSTÍ ŠTUDENTOV

(Tímový projekt)

Dokumentácia k projektu

Tím č.10 – ČERNÉ OFCE:
Bc. Martin Macko
Bc. Martin Paulech
Bc. Peter Rada
Bc. Miroslava Romanová
Bc. Tibor Schvartz
Bc. Lukáš Slížik
Študijný odbor: Informačné systémy
Kontakt: team1078@gmail.com

Školský rok: 2007/2008
Dátum: 14. novembra 2007

OBSAH

1 ÚVOD

1.1 Cieľ

Predpokladá sa, že súčasný študent vysokej školy nadobudol pred svojím štúdium na nej určité znalosti a zručnosti, ktoré si počas svojho štúdia môže ďalej rozširovať a zlepšovať sa v nich alebo sa oboznámiť s novými vedomosťami, ktoré získava na škole. Úroveň ovládania znalostí a zručností študenta je dôležitou informáciou a aj poznatkom najmä pre pedagógov, ale aj pre samotných študentov, korí sa týmto spôsobom môžu porovnávať so svojimi spolužiakmi a zistiť na akej úrovni sa ich znalosti nachádzajú.

Cieľom tohto projektu je vytvorenie informačného systému, ktorý bude študentovi slúžiť ako nástroj na sebahodnotenie svojich nadobudnutých znalostí a môže mu pomôcť pri štúdiu napríklad prideľovaním zadaní projektov, ktoré budú preňho ľahšie zvládnuteľné, nakoľko bude ovládať vedomosti potrebné na vyriešenie zadania. Študentovi sa okrem toho môže naskytnúť šanca pracovať v niektorej z firiem, ktoré hľadajú práve takého ako je on sám, so znalosťami potrebnými na získanie danej pracovnej pozície s prislúchajúcim finančným ohodnotením.

Vytvorením takéhoto informačného systému získajú v neposlednom rade aj pedagógovia, ktorým umožní vytvárať si na študenta vďaka jeho znalostiam názor, pohľad, prípadne majú možnosť potvrdiť a zvýšiť preferenciu najlepších študentov, na základe svojich skúseností získaných počas ich vyučovania. Taktiež im umožní vyhľadávať najvhodnejších študentov pre svoje projekty, súťaže, práce, odborné praktiká a pod.

Všeobecnou celkovou podstatou projektu je vytvorenie systému, ktorý uľahčí život na fakulte pedagógom a aj nám študentom.

Tento dokument slúži ako dokumentácia k informačnému systému Bázy znalostí a zručností študentov. V časti Úvod sa nachádza okrem cieľu projektu podkapitola Referencie a Slovník pojmov. Ďalšia kapitola sa venuje špecifikácii požiadaviek systému. Kapitola Analýza je venovaná rozboru systémov s podobnou tematikou, fakultným systémom, ktoré môžu svojimi informáciami prispieť k tomuto informačnému systému a analýze možných technológií. Návrh systému sa venuje konkrétnemu popisu nami navrhovaných jednotlivých častí systému v podobe architektúry systému a logického modelu údajov databázy.

1.2 Referencie

[1] Stránka tímu The Llama team, ktorý riešil rovnaký projekt.
http://www2.dcs.elf.stuba.sk/TeamProject/2005/team06/

[2] Stránka tímu _elf_, ktorý riešili rovnaký projekt.
http://www2.dcs.elf.stuba.sk/TeamProject/2005/team11/

[3] Stránka AIS – Akademický informačný systém.
https://is.stuba.sk/

[4] Stránka Yonban – systém na prideľovanie projektov.
http://www2.fiit.stuba.sk/yonban/index.jsp

[5] Stránka FIIT Moodle - nástroj na elektronické vzdelávanie.
http://moodle.fiit.stuba.sk/moodle/

1.3 Slovník pojmov

DAO (Data Access Object): návrhový vzor, ktorý poskytuje abstraktné rozhranie niektorým databázovým mechanizmom

JDBC API (Java Database Connectivity Application Programming Interface): rozhranie, ktoré definuje prístup klienta k databáze

JDO (Java Data Objects): špecifikácia perzistencie Java objektov

MVC (Model-View-Controller): architektonický vzor; v komplexných aplikáciách sa separujú údaje (model) a používateľské rozhranie (view), takže zmena jedného neovplyvní druhé

Objektovo - relačné mapovanie: programovacia technika, ktorá slúži na konverziu údajov medzi databázou a objektovo - orientovanými programovacími jazykmi; vytvára sa tak dojem virtuálnej objektovej databázy

Oracle Database XE: Oracle Database Express Edition

ORM:	Object - Relational Mapping, objektovo - relačné mapovanie
PHP:	PHP: Hypertext Preprocessor, programovací jazyk, pôvodne určený na tvorbu dynamických web stránok
XML:	Extensible Markup Language

2 ANALÝZA

Táto kapitola sa venuje analýze predchádzajúcich riešených systémov s podobným zadaním, analýze systémov používaných na našej fakulte, analýze technológií a ich výhod a nevýhod v súvislosti s realizáciou projektu.

2.1 Analýza predchádzajúcich systémov

V školskom roku 2005/2006 boli na našej škole realizované dva projekty s veľmi podobnom tematikou. V nasledujúcich dvoch podkapitolách sa venujeme ich bližšej analýze s dôrazom na ich klady a zápory, ktoré pri tvorbe použili.

2.1.1 Analýza systému vytvoreného tímom _elf_

Tím _elf_ vychádzal z dvoch exitujúcich podobných systémov, na ktorých niektorí jeho členovia v minulosti pracovali. Jednalo sa o systémy z oblasti poisťovníctva a z oblasti logistiky. Tento tím vytvoril informačný systém, ktorý získaval zoznam študentov z informačného systému Študent. Na zabezpečenie aktuálnosti a zároveň predchádzaniu zdvojeniu dát, import údajov sa vykonával manuálne na podnet administrátora na konci každého semestra. Študenti si mali vo vlastnom záujme dbať o svoj profil znalostí a zručností. Výstup získaných informácii mal byť použiteľný ďalej v iných informačných systémoch ako napríklad YonBan..

Používateľ systému (vyučujúci alebo externý systém) si mohol určiť kritériá a priority, podľa ktorých si zoradí vybraných študentov. Vyučujúcim teda mali pomáhať získané informácie o doterajších skúsenostiach, schopnostiach a záujmoch študenta. Systém však neposkytoval priame informácie o študijných výsledkoch ani osobné číslo spolu s menom študenta kvôli dodržiavaniu zákona o ochrane osobných údajov (Zákon . 363/2005 Z. z. o ochrane osobných údajov v znení neskorších predpisov).

Medzi požívateľov systému tohto tímu definovali jeho členovia rolu študenta, vyučujúceho, externého systému (poskytoval informáciu o študentovi alebo využíval výstup

z informačného systému Znalosti), administrátora (staral sa o správu systému a jeho správny chod počas prevádzky) a konfigurátora (rozhodoval o tom, aké typy informácií budú prístupné používateľom informačného systému).

Do prípadov použitia tím _elf_ zahrnul napríklad pridanie certifikátu, zručnosti, znalosti, hodnotenia, poskytnutie indexovaného zoznamu študentov, komplexnej alebo čiastočnej informácie o študentovi, známky, import údajov, notifikáciu, nastavenie profilu a správu jednotlivých častí. Vstupy systému boli známky, znalosti, zručnosti, certifikáty, hodnotenia a informácie o predmetoch a výstupy zoznam študentov podľa výsledného indexu, komplexná informácia o študentovi a informácia o predmete.

Komunikáciu s okolím predstavovali webový formulár, web services a import údajov zo Študenta. Systém bol zabezpečený autentifikáciou (login a heslo), autorizáciou a protokolom HTTPS. Zloženie systému vytvoreného týmto tímom pozostávalo z dátovej, aplikačnej a prezentačnej vrstvy. Členovia tímu vytvorili JSP stránky a aplikácie s grafickým rozhraním.

V implementácii systému sa zamerali na využitie služieb aplikačného servera Apache Tomcat, knižnice Apache Axis a Apache Xerces, na zostavenie aplikácii použili Apache Maven a na vývoj prostredia Java Eclipse.

2.1.2 Analýza systému vytvoreného tímom The Llama team

Tento tím vytvoril informačný systém s názvom Pallas, ktorý bol taktiež zameraný na bázu znalostí a zručností študentov a jeho navrhnutie vychádzalo predovšetkým z poznatkov získaných z informačných systémov iných univerzít.

Používateľom systému vytvoreného týmto tímom boli priradené roly študent, administrátor, pedagóg a pracovník pedagogického oddelenia. V systéme členovia tímu využili automatické notifikácie (pravidelné notifikácie študentov, aby si aktualizovali údaje v systéme), možnosť tvorby nových formulárov a ukladanie fotografie, životopisu a ďalších dokumentov v profile študenta. Výstup zo systému sa dal použiť ako vstup pre iné systémy, s ktorými bol prepojený, a poskytoval možnosť automatického vstupu do systému napríklad prostredníctvom SOAP/web services.

Tím poskytol študentovi, aby si svoje nadobudnuté znalosti z technológií v systéme aj otestoval formou testu, ktorý sa následne automaticky vyhodnotil, alebo si študent mohol dať vygenerovať vlastný životopis. Samozrejmosťou pre systém bolo vyhľadávanie a triedenie informácií o študentoch podľa rôznych kritérií a vkladanie poznámok od pedagógov.

Výsledný systém poskytoval nepriamu informáciu o študentovi a jeho študijných výsledkoch, a teda či sa nachádza medzi desiatimi percentami najlepších študentov daného predmetu. Členovia tímu uvažovali aj nad rozšírením systému o informácie týkajúce sa registrovania a prideľovania tém v súvislosti so systémom Yonban.

2.2 Analýza systémov na fakulte

Na našej fakulte existujú a používajú sa v súčasnosti tri informačné systémy, ktoré by mohli obsahovať údaje dôležité pre nami vytváraný systém. V nasledujúcich kapitolách sa nachádza ich bližší popis a informácie, ktoré by sme z nich chceli čerpať.

2.2.1 Akademický informačný systém (AIS)

AIS funguje na našej fakulte približne druhý rok, je to odkúpený elektronický systém zahŕňajúci všetky informácie týkajúce sa predovšetkým študentov a pedagógov navštevujúcich a vyučujúcich na Slovenskej technickej univerzite v Bratislave.

Pre náš systém sme sa rozhodli čerpať údaje najmä z tohto systému, ale dbať pritom na ochranu osobných údajov. Relevantnými informáciami pre projekt by malo byť osobné číslo, meno a priezvisko študenta, email študenta, predmety, ktoré vyštudoval alebo študuje, známky, ktoré sa však v hodnotení nášho systému nezobrazia, ale len použijú.

Nepredpokladáme, že by bolo v našom systéme potrebné uchovávať ešte ďalšie informácie o študentovi, keďže všetky tieto sú v AIS a tým by sme ich iba duplikovali. Z dôvodu uchovávania osobného čísla študenta v našom systéme nebude pre nás problém v prípade potreby získať z AIS o študentovi ďalšie informácie.

Keďže systém je založený na databáze vytvorenej v Oracle, pre ľahšiu prácu s dátami sme si aj my zvolili implementáciu databázy práve v tomto prostredí.

2.2.2 Yonban

Existencia tohto systému sa datuje od roku 2002, kedy bol vytvorený študentmi v rámci predmetu Tímový projekt. Jeho prvoradým cieľom je poskytnúť študentom a pedagógom ucelený systém, zefektívňujúci prácu na projekte vo všetkých jeho životných cykloch.

Poskytuje študentom možnosť registrovať sa na jednotlivé témy dôležitých školských projektov (bakalárska práca, diplomová práca...), ktoré im môžu byť vedúcimi prác následne pridelené. Medzi ďalšie funkcie systému sa radí aj odovzdávanie vyriešených projektov a posudkov na ne.

Keďže sa v systéme nenachádza výsledné hodnotenie študenta za vypracovaný projekt (t. zn. známka, ktorú mu komisia udelila po jeho obhajobe, alebo iné relevantné informácie, ktoré by sme chceli z nášho hľadiska použiť pre náš systém), rozhodli sme sa, že informácie, ktoré sa nachádzajú v systéme Yonban, zatiaľ nepoužijeme pri riešení nášho projektu.

2.2.3 FIIT Moodle

Moodle je systém pre vytváranie kurzov založených na Internete a web stránkach. Je projektom určeným pre podporu sociálneho konštruktivistického rámca vyučovania. Slovo Moodle bolo pôvodne akronymom pre Modular Object Oriented Dynamic Learning Environment (modulárne objektovo orientované dynamické výukové prostredie).

V súčasnosti existuje na FIIT STU server s nainštalovaným Moodle. V rámci tohto systému sa nachádzajú len informácie o hodnotení študenta a ostatné informácie, ktoré sú irelevantné pre vyhodnocovanie znalostí študenta. Systém Moodle poskytuje možnosť vyexportovať hodnotenia študentov pre daný predmet, avšak tieto informácie sú taktiež dostupné prostredníctvom akademického informačného systému AIS STU. Z tohto dôvodu považujeme Moodle ako nepodstatný zdroj informácií pre vyhodnocovanie znalostí študentov.

2.3 Analýza technológií

Táto kapitola sa zameriava na opis technológií, nad ktorými sme uvažovali, a na základe ich výhod príp. nevýhod sme sa rozhodli použiť najvhodnejšie z nich pri realizácii riešenia nášho projektu.

2.3.1 Apache Tomcat

Pri štúdiu technológií vhodných na realizáciu prezentačnej vrstvy projektu Báza znalostí a zručností študentov sme sa najmä kvôli zachovaniu čo najväčšej konzistentnosti a aj iným výhodám rozhodli využiť JSP technológiu. Na zobrazovanie stránok s aktívnym obsahom používateľom nášho systému je potrebné mať na strane serveru nainštalovaný Apache Tomcat.

Apache Tomcat ako technológia predstavuje akýsi „kontajner“ pre servlety, ktorý sa často používa pri implementácií Java Servletov a Java Server Pages technológií. Tomcat dokáže vystupovať aj ako plnohodnotný web server, ale k dispozícií sú aj konektory pre najpoužívanejšie web servery Apache a Microsoft IIS. Tieto konektory umožňujú transparentnú obsluhu požiadaviek na JSP stránky pomocou inštancie Tomcatu a obsluhu ostatného obsahu pôvodným web serverom. Zdrojové kódy sú vo všetkých verziách voľne dostupné na domovskej stránke.

2.3.2 Hibernate

Relačné databázy v súčasnosti nepopierateľne tvoria jadro moderných podnikov. Kým programovacie jazyky ako napríklad Java poskytujú intuitívny, objektovo-orientovaný prístup k jednotlivým entitám, podnikové informácie, ktoré sa skrývajú za týmito entitami sú silne relačné vo svojej podstate. Hlavná výhoda relačných modelov spočíva v tom, že ich dizajn je oddelený od akéhokoľvek programovacieho prístupu.

Jednou zo súčasných technológií, ktorá umožňuje prepojenie relačných dát a objektovo-orientovanej paradigmy prostredníctvom objektovo/relačného mapovania (ORM)

je Hibernate. Dokonca sa pár rokov po jeho objavení stal jednou z vedúcich ORM technológií súčasnosti. Veľkou výhodu Hibernate je fakt, že je to open source.

2.3.3 Java

Java ako programovací jazyk je voľne dostupný pre verejnosť, ktorý má širokú oblasť využitia, od osobných internetových stránok až po využívanie vo vesmírnom programe NASA. Java je od základov vytvorená ako čistý, prehľadný, bezpečný a objektovo orientovaný programovací jazyk. Syntax Javy sa odvodila z programovacieho jazyka C++, ale oproti tomuto jazyku má mnoho výhod, ktoré sú zamerané na dodržiavanie štyroch vyššie uvedených základných princípov.

Aplikácie a aplety napísané v Jave sa spúšťajú pomocou Java Virtual Machine (JVM), ktorý musí byť nainštalovaný na počítači používateľa. JVM vytvára prostredie, v ktorom sa vykonáva program napísaný v tomto jazyku.

2.3.4 J2EE

Java Enterprise Edition je priemyselný štandard pre vyvíjanie prenositeľných, robustných, škálovateľných a bezpečných Java aplikácií na strane servera. Je postavená na solídnych základoch Java SE (Standard Edition) a poskytuje webové služby, model komponentov, manažment a komunikáciu s API.

Hlavné výhody použitia J2EE:

- Jednoduchší vývoj. S pomocou J2EE technológie je vývojárom poskytnutá možnosť lepšieho a rýchlejšieho písania zdrojového kódu.
- EJB – jednoduchšie a lepšie využívanie JavaBeans technológie pomocou používania Plain Old Java Objects (POJO).
- Vylepšené web služby – ideálna implementačná platforma pre SOA (Service – Oriented Architecture)

2.3.5 JSP, PHP a ASP

PHP je „open source“ technológia, a teda jeho veľkou výhodou napríklad oproti ASP je cena. PHP je distribuované zadarmo a nezávislé na platforme. Funguje pod Windows, Linux, Solaris a mnohými inými systémami.

Medzi hlavné nevýhody PHP môžeme zaradiť nefunkčný objektový model. Knižnica základných funkcií PHP je totiž procedurálna a takmer vôbec nevyužíva objekty. Funkcie potom nevyhadzujú ani výnimky, čo v podstate znemožňuje objektové programovanie.

JSP je technológia, ktorá umožňuje vývojárom rýchlo vytvárať webové stránky a aplikácie. JSP je založené na objektovo orientovanom jazyku Java a umožňuje vytváranie robustných webových systémov.

Medzi najvýznamnejšie výhody JSP patrí:

- multi - platformová technológia,
- znovuvyužívanie existujúcich komponentov pomocou JavaBeans a Enterprise Java Beansov (EJB),
- výhoda Javy,
- prenositeľnosť JSP súborov na akúkoľvek inú platformu, web server alebo JSP servlet engine,
- HTML a grafika zobrazovaná v internetovom prehliadači tvorí prezentačnú vrstvu a Java kód (JSP) tvorí implementačnú vrstvu.

JSP a ASP sú svojou funkcionalitou takmer podobné. ASP je automaticky inštalované spolu so serverom IIS, ale nesie so sebou skryté výdavky. Ak potrebujete dodatočné služby ako napríklad možnosť uploadovať súbory, šifrovanie alebo správu e-mailov, je potrebné ich dokúpiť.

ASP aplikácie bežia len na vybraných operačných systémoch. ASP nedosahuje rýchlosť PHP a je založená na COM architektúre. Ak programátor používa VBScript, volá neustále COM objekty, čo zapríčiňuje nepríjemné spomalenie.

Porovnanie hlavných funkcií PHP, JSP a ASP:

Funkcia	ASP	PHP	JSP
Separácia medzi obsahom a logikou	Čiastočné	Čiastočné	Úplné
Cieľový používateľ	Programátor	Programátor	Web vývojár, autor stránky
Jazyk	VBScript	Pearl, C	Java
Rozšíriteľný pomocou third-party komponentov	Nie	Nie	Áno

2.3.6 Spring

Spring Framework je open source aplikácia pre vývoj softvéru. Skladá sa z kolekcie menších modulov, ktoré sú navrhnuté tak, aby pracovali nezávisle a poskytovali lepšiu funkcionalitu. To znamená, že komponenty sa vyvíjajú nezávisle a integráciu zabezpečuje jeden človek.

Medzi jeho ďalšie výhody rozhodne patrí značné zjednodušenie vývoja, nakoľko obsahuje podporu pre webové aplikácie (jednoduché vytvorenie prezentačnej vrstvy, flexibilný Model-View-Controller, podpora JSP, Velocity a ďalších), podporu pre objektovo-relačné mapovanie (štandardné aplikácie Hibernate, TopLink, JDO a ďalšie) alebo aspektovo-orientovanú funkcionalitu.

Modul DAO (Data Access Object) poskytuje abstraktnú vrstvu pre prácu s JDBC API (Java Database Connectivity Application Programming Interface). Jeho hlavné výhody sú, že odstraňuje veľa zbytočného kódu (získanie spojenia, zrušenie spojenia, iterovanie cez výsledky a i.). Správa výnimiek je prevedená z *java.sql.SQLException* do inteligentnej hierarchie Runtime výnimiek, o ktoré sa nemusí starať programátor. Snahou je, aby sa programátor zameral len na prácu s SQL a extrahovanie výsledkov. Za jeho nevýhodu sa dá považovať nemožnosť priamo volať vzdialené objekty.

2.3.7 Oracle

Ide o technológiu ponúkajúcu databázový systém riadenia, umožňujúci transformáciu údajov a dát do informácií. Tento systém umožňuje používateľom vytvárať, aktualizovať a vyberať informácie z databázy. Oracle je v súčasnosti významným nástrojom pri práci s databázovými štruktúrami.

Výhody:

- jednoduchá inštalácia a manažment,
- intuitívne používateľské rozhranie (browser-based),
- kompatibilita s množstvom súčasných technológií (PHP, Java, .NET, XML).

Nevýhody:

- nutnosť zakúpenia komerčnej verzie pre projekty rozsiahlejšieho charakteru (voľne dostupné riešenie ponúka len Oracle Database XE),
- hardvérové obmedzenia v Oracle Database XE (max. 4GB používateľských údajov, max. 1GB RAM).

2.3.8 Zhodnotenie

Z uvedenej analýzy technológií sa nám javí ako najvhodnejšie riešenie použitie Javy, konkrétne J2EE. Vzhľadom na čo najväčšiu konzistentnosť vytváraného systému je výhodné vytvoriť prezentačné vrstvu pomocou JSP technológie, čím sa nám taktiež otvára možnosť efektívneho prepojenia s Oracle databázou a aplikačnou vrstvou vytváraného systému Bázy znalostí a zručností študentov. Už samotnú implementáciu prezentačnej, aplikačnej a dátovej vrstvy budeme riešiť pomocou Spring frameworku, ktorý zjednoduší a urýchli celý proces vytvárania systému. Spring sme zvolili kvôli modulom Hibernate, web services a iným, ktoré tento framework obsahuje.

3 ŠPECIFIKÁCIA POŽIADAVIEK

Táto kapitola sa zaoberá požiadavkami na vytváraný systém. Je rozdelená do štyroch častí. Prvá obsahuje charakteristiku jednotlivých používateľov systému, v druhej je uvedený diagram prípadov použitia, tretia sa venuje špecifikácii funkcií systému a ostatné požiadavky sa nachádzajú v poslednej časti.

3.1 Charakteristiky používateľov

Z hľadiska rolí používateľov systému sú identifikovaní nasledovní používatelia:

Študent Zadáva do systému informácie o vlastných znalostiach a zručnostiach.

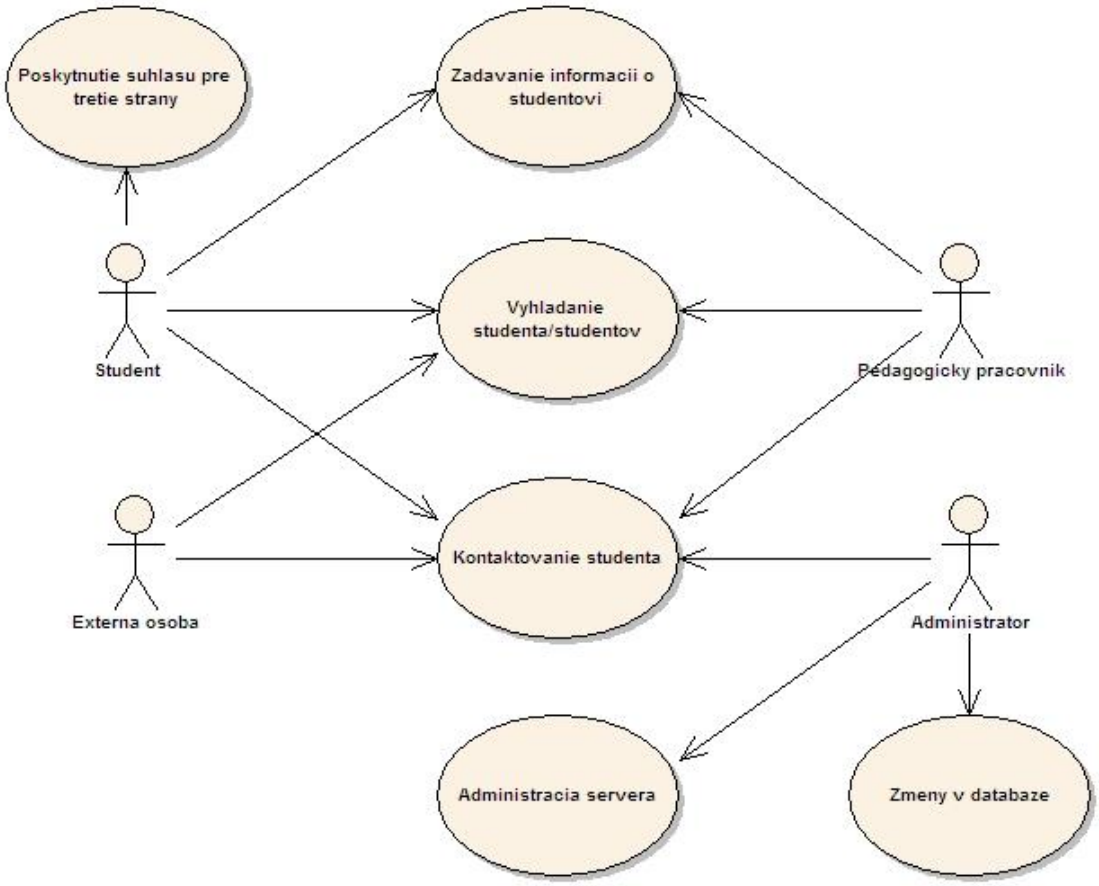
Pedagogický pracovník Zadáva do systému informácie o študentoch. Požaduje od systému znalosti o študentoch.

Administrátor Spravuje systém a má na starosti jeho správny chod počas rutinnej prevádzky.

Externá osoba Využíva znalosti poskytované systémom po súhlase študenta.

3.2 Diagram prípadov použitia

Na obrázku č.1 je znázornený diagram prípadov použitia, ktorý poskytuje prehľadnú informáciu o poskytovanej funkcionalite vo vzťahu k jednotlivým používateľom.



Obr.1: Diagram prípadov použitia

3.3 Požiadavky na funkcie systému

3.3.1 UC - Poskytnutie súhlasu pre tretie strany

Vstup: súhlas/nesúhlas

Výstup: uloženie vstupu do databázy (upravená databáza)

Používatelia: Študent

Opis: evidovanie rozhodnutia študenta zverejniť informácie o sebe aj tretej strane

3.3.2 UC - Zadávanie informácií o študentovi

Vstup: znalosti, zručnosti, študijné výsledky...

Výstup: uloženie vstupov do databázy (upravená databáza)

Používatelia: Študent, Pedagogický pracovník

Opis: vloženie informácií týkajúcich sa znalostí, zručností a študijných výsledkov študenta

3.3.3 UC - Vyhľadanie študenta/študentov

Vstup: požiadavky na študenta

Výstup: študent/zoznam študentov

Používatelia: Študent, Pedagogický pracovník, Externá osoba

Opis: systém nájde študenta/študentov na základe zadaných kritérií hľadania

3.3.4 UC - Kontaktovanie študenta

Vstup: požiadavka na študenta

Výstup: spojenie so študentom

Používatelia: Študent, Pedagogický pracovník, Administrátor, Externá osoba

Opis: pri potrebe spojenia sa so študentom z rôznych príčin s využitím kontaktu na študenta (mail, tel. č., icq, ...)

3.3.5 UC - Administrácia servera

Vstup: údaje týkajúce sa administrácie (napr. prístupové práva používateľov)

Výstup: úprava údajov, nastavenia servera

Používatelia: Administrátor

Opis: zabezpečenie správneho chodu servera

3.3.6 UC - Zmeny v databáze

Vstup: správna alebo chýbajúca informácia v databáze

Výstup: upravená databáza

Používatelia: Administrátor

Opis: v prípade nekonzistentnosti dát v databáze uskutoční administrátor nevyhnutné zmeny

3.4 Ďalšie požiadavky

Bezpečnosť a ochrana informácií

Vzhľadom na to, že systém spracováva a uchováva údaje chránené zákonom o ochrane osobných údajov, musia byť tieto chránené pred náhodným ako aj nezákonným poškodením a zničením, náhodnou stratou, zmenou, nedovoleným prístupom a sprístupnením ako aj pred akýmikoľvek inými neprípustnými formami spracúvania. Systém by mal preto obsahovať štandardné spôsoby ochrany autentifikáciu a autorizáciu.

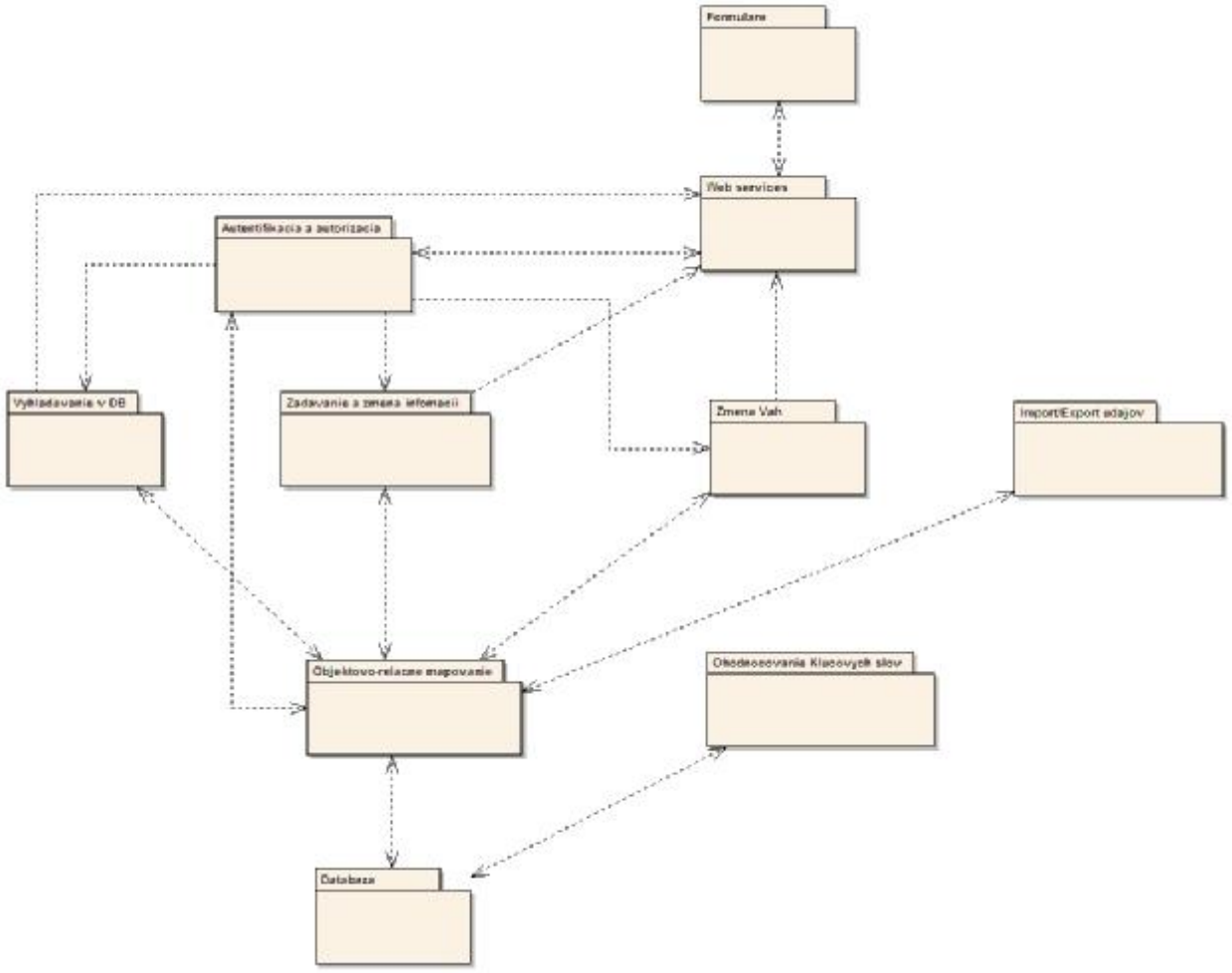
Používateľ, ktorý chce pracovať so systémom, sa musí najskôr prihlásiť pomocou prihlasovacieho formulára. Zadaním a potvrdením mena a hesla sa spustí proces autentifikácie. Zadané údaje sa porovnajú s údajmi, ktoré sú uložené v databáze. Pokiaľ sa nájde zhoda, používateľovi je umožnené prihlásenie sa 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.

4 NÁVRH

4.1 Architektúra systému

4.1.1 Diagram architektúry systému



Obr. 2: Architektúra systému

Používateľ pristupuje do systému cez formuláre (webové stránky). Bude vyzvaný, aby zadal svoje prístupové meno a heslo. Tieto údaje sa cez webové služby pošlú do modulu autentifikácie a autorizácie, kde sa zistí korektnosť zadaných údajov a aké práva daný používateľ má. Následne bude môcť používateľ vykonávať v systéme akcie (podľa prístupových práv).

Vyhľadávanie v DB - autorizovaný používateľ si bude môcť vyhľadať v databáze potrebné informácie. Požiadavka používateľa sa odošle z formuláru do modulu webových služieb, kde sa identifikuje typ požiadavky, zistia sa práva používateľa (modul autentifikácie a autorizácie) a ak používateľ môže v databáze danú akciu vykonať, tak sa táto požiadavka pošle do modulu vyhľadávania v DB, ktorý ju ďalej spracuje a odošle modulu objektovo-relačného mapovania. Tento modul skonvertuje požiadavku do príkazov zrozumiteľných databáze. Databáza požiadavku spracuje a vráti modulu objektovo-relačného mapovania výsledok, ktorý ho skonvertuje do vhodnej formy a odošle modulu vyhľadávania informácií, kde sa spracuje do prezentačnej formy, odošle modulu webových služieb a ten sa postará o správne zobrazenie požadovaného výsledku vo formulári.

Zadávanie a zmena informácií - cez tento modul môže používateľ zadávať a modifikovať informácie v databáze. Používateľ sa najprv autentifikuje a autorizuje v systéme a následne bude môcť zadávať alebo meniť informácie, pre ktoré má práva. Zadá do formuláru požiadavku, ktorá sa odošle modulu webových služieb. Tu sa identifikuje typ požiadavky, oprávnenia a následne sa odošle modulu zadávania a zmeny informácií. V tomto module sa požiadavka spracuje a odošle sa modulu objektovo-relačného mapovania, ktorý požiadavku skonvertuje do jazyka zrozumiteľného databáze. Po odoslaní takto spracovanej požiadavky do databázy sa v nej vykonajú požadované zmeny a výsledok operácie (úspech/neúspech) sa cez modul zadávania a zmeny informácií a modul webových služieb prešíri do formuláru, kde sa zobrazí.

4.1.2 Popis modulov architektúry

Webové služby

Webové služby majú na starosti prekladanie požiadaviek zadaných prostredníctvom formulárov a naopak, prekladanie získaných informácií do webových formulárov.

Medzi webové služby patrí aj samotný proces prihlásenia sa do systému. Webové služby následne poskytujú informácie (login a heslo) o prihlásení pre modul autentifikácie a autorizácie.

Import/export údajov o študentovi

Modul importu/exportu údajov buď získava informácie o študentovi z relevantných zdrojov, ktoré sú spomenuté v časti Analýza systémov na fakulte a posiela ich modulu objektovo-relačného mapovania na spracovanie, alebo už takto spracované informácie o študentovi odosiela do systému Yonban. Toto získavanie informácií sa vykonáva v pravidelných intervaloch (napríklad jedenkrát za semester).

Zmena váh

V rámci procesu vzdelávania občas nastávajú zmeny v systéme výučby a vtedy môže dôjsť aj k zmenám váh, ktoré ovplyvňujú kľúčové slova uchované v databáze. Tieto kľúčové slová sú potrebné pre určenie znalosti študenta. Aby systém mohol reagovať na takéto zmeny a zakaždým určoval čo najpresnejšie znalosti študenta, musí obsahovať modul, pomocou ktorého budú tieto váhy upravené.

Ohodnocovanie kľúčových slov

Určitý používatelia, ktorý úspešne prejdú autentifikáciou majú možnosť vytvárať a modifikovať mieru akou jednotlivé kľúčové slová prispievajú k jednotlivým zručnostiam a znalostiam.

Formuláre

Formuláre tvoria prezentačnú vrstvu systému, zobrazovanú prostredníctvom používateľovho webového prehliadača. Slúžia ako vstupné rozhranie na zadávanie informácií potrebných pre vyhľadávanie študentov s konkrétnymi znalosťami a ako výstupné rozhranie pre zobrazenie informácií o hľadaných študentoch.

Zadávanie a zmena informácií

Poskytuje používateľovi možnosť zmeny určitých údajov v databáze (zmena osobných údajov, pridávanie poznámok pedagóga, zmeny váh predmetov, ...). Na základe úspešnej

autentifikácie systém určí prístupové práva pre daného používateľa, rozsah a typ informácií, ktoré môže modifikovať.

Vyhľadávanie v DB

Databáza poskytuje zmysluplné a relevantné informácie na základe používateľských požiadaviek. Vyhľadávanie je podmienené úspešnou autorizáciou používateľa. Zjednodušene povedané, tento modul vytvára zoznam študentov zoradených podľa zadefinovaných požiadaviek.

Autentifikácia a autorizácia

Tento modul vyhodnocuje informácie, ktoré mu boli poslané z webových služieb. Na základe týchto informácii zistí totožnosť používateľa a podľa totožnosti určí, aké má práva v systéme.

Objektovo – relačné mapovanie

Predstavuje rozhranie medzi ostatnými modulmi a samotnou databázou. Smerom do databázy sú požiadavky transformované do príkazov zrozumiteľných pre konkrétne databázové prostredie (jazyk SQL) a smerom von z databázy sú naopak tieto informácie transformované do objektov (jazyk Java). Tento modul sa v systéme nachádza z dôvodu zvýšenia efektívnosti a jednoduchosti, ktorú prináša objektovo - orientovaná paradigma a práca s objektmi. Viac o objektovo-relačnom mapovaní je popísané v časti Analýza technológií (Hibernate).

Databáza

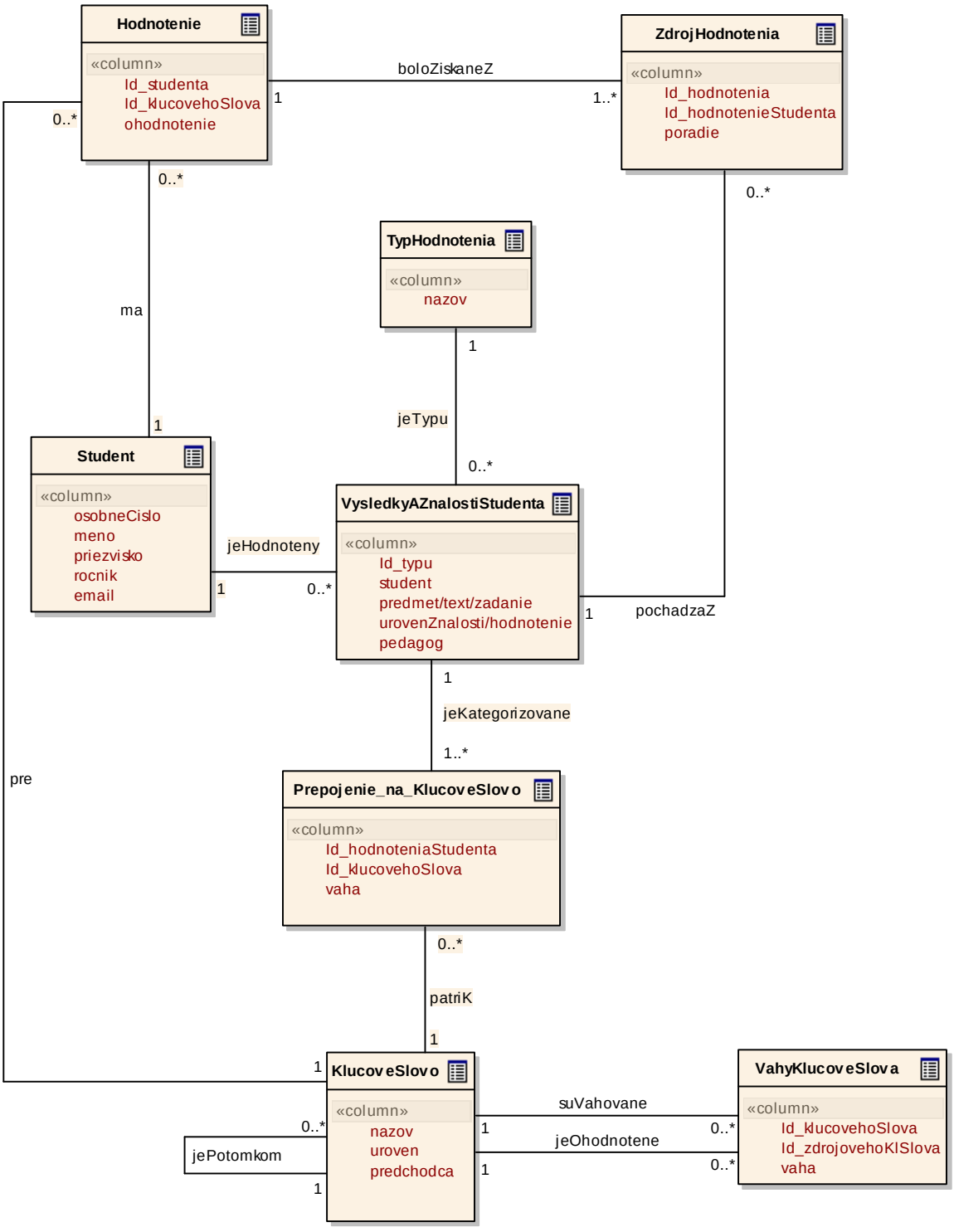
Tvorí jadro celého systému, pretože obsahuje informácie o študentoch (osobné číslo, meno, ročník, e-mail), ich výsledkoch, zručnostiach a znalostiach. Okrem týchto informácií sú v databáze uložené aj prihlasovacie údaje pre registrovaných používateľov systému. Pri autentifikácií sú tieto údaje overované s údajmi v databáze a v prípade ich správnosti je používateľovi umožnená práca so systémov v rozsahu jeho právomocí.

4.2 Databáza

Táto kapitola obsahuje logický pohľad na uchovávané údaje. Vytvorený diagram modelu údajov identifikuje základné entity, ich atribúty a vzťahy medzi nimi. Taktiež Je v tejto kapitole uvedený príklad výpočtu ohodnotenia kľúčových slov.

4.2.1 Diagram modelu údajov databázy (logická úroveň)

Každá entita má svoje atribúty a je zviazaná s inými entitami pomocou vzájomných vzťahov. Na obrázku č.3 je znázornený logický model údajov navrhovanej databázy.



Obr. 3: Logický model údajov

Kľúčové slovo

Celá databáza je postavená na entite KľúčovéSlovo. Táto entita uchováva informácie o hierarchii kľúčových slov (tieto informácie sú uložené v atribútoch *úroveň* a *predchodca*, kde atribút *predchodca* je odkaz na kľúčové slovo, ktoré je priamym predchodcom daného kľúčového slova a kľúčové slovo, ktoré je na najvyššej úrovni bude mať tento atribút vynulovaný). Jedno kľúčové slovo má práve jedného predchodcu (kardinalita 1) a jedno kľúčové slovo môže byť predchodcom žiadneho (nemá nasledovníkov) alebo viacerých kľúčových slov (kardinalita 0..n).

VáhyKľúčovéSlová

Ak je kľúčové slovo predchodcom (má nasledovníkov) iného kľúčového slova, tak existuje väzba medzi entitami KľúčovéSlovo a VáhyKľúčovéSlová. V entite VáhyKľúčovéSlová je uchovaná informácia o tom, s akou váhou prispieva ohodnotenie daného kľúčového slova na ohodnotenie jeho predchodcu.

Väzba *jeOhodnotené:* jedna váha kľúčového slova prislúcha práve k jednému kľúčovému slovu (kardinalita 1) a jedno kľúčové slovo môže byť ohodnotené skrz váhy viacerých kľúčových slov (kardinalita 0..n).

Väzba *súVáhované*: jeden zdroj vo váhe kľúčového slova prislúcha k jednému kľúčovému slovu (kardinalita 1) a jedno kľúčové slovo môže byť zdrojom pre viacero kľúčových slov (kardinalita 0..n).

Študent

V entite Študent sú uchované informácie o jednotlivých študentoch (osobné číslo študenta, jeho meno a priezvisko, v ktorom ročníku práve študuje a jeho kontaktný e-mail).

VýsledkyAZnalostiŠtudenta

V tejto entite budú uložené všetky študijné výsledky a znalosti študenta. V atribúte *Id_typu* je vyjadrené o aký typ ide napríklad:

hodnotenie predmetu - v treťom atribúte entity bude identifikované o aký predmet ide a štvrtý atribút bude prázdny, lebo hodnotenie predmetu pre daného študenta sa bude importovať z externého systému a v databáze sa ukladať nebude a taktiež aj piaty atribút bude prázdny

sebahodnotenie študenta - vtedy tretí atribút bude vyjadrovať čo študent na sebe hodnotí - akú znalosť, vo štvrtom atribúte bude uložená úroveň znalosti, ktorú študent dosiahol a piaty atribút bude prázdny

hodnotenie študenta pedagógom - tretí atribút vyjadruje znalosť, ktorú pedagóg na študentovi hodnotí, štvrtý vyjadruje úroveň znalosti a v piatom atribúte je identifikovaný pedagóg, ktorý hodnotenie do systému vložil

hodnotenie študentského projektu - v treťom atribúte je názov projektu, ktorú študent vypracoval, vo štvrtom je hodnotenie projektu a piaty atribút je prázdny.

Jeden výsledok/znalosť patrí práve k jednému študentovi (kardinalita 1) a jeden študent môže mať viacero výsledkov/znalostí (kardinalita 0..n).

Prepojenie_na_KľúčovéSlovo

V tejto entite sú uchované informácie, ku ktorým kľúčovým slovám daný výsledok/znalosť študenta patrí a s akou váhou ovplyvňuje ohodnotenie kľúčového slova.

Vzťah *VýsledkyAZnalstiŠtudenta-Prepojenie_na_KľúčovéSlovo:* jeden výsledok/znalosť študenta môže byť priradené k viacerým kľúčovým slovám (kardinalita 1..n) a jeden záznam tejto entity patrí k jednému výsledku/znalosti študenta.

Vzťah: *Prepojenie_na_KľúčovéSlovo -KľúčovéSlovo*: záznam tejto entity prislúcha k jednému kľúčovému slovu (kardinalita 1) a jedno kľúčové slovo môže mať viacero záznamov tejto entity (kardinalita 0..n).

Hodnotenie

V tejto entite sú uchované informácie aké má študent ohodnotenie kľúčových slov. Sú tu informácie ako ku ktorému študentovi a ktorému kľúčovému slovu dané hodnotenie patrí a samotné ohodnotenie.

Vzťah *Hodnotenie-Študent*: Jeden jedno hodnotenie patrí jednému študentovi (kardinalita 1) a jeden študent má viacero hodnotení (kardinalita 0..n).

Vzťah *Hodnotenie-KľúčovéSlovo:* jedno hodnotenie prislúcha k jednému kľúčovému slovu (kardinalita 1) a jedno kľúčové slovo môže mať viacero hodnotení (kardinalita 0..n).

ZdrojHodnotenia

Entita na uchovanie toho, ktoré výsledky/znalosti študenta najviac prispeli k ohodnoteniu kľúčového slova pre daného študenta. Atribútom je poradie - hodnotu 1 bude mať ten zdroj, ktorý najviac prispel k ohodnoteniu kľúčového slova.

Vzťah *ZdrojHodnotenia- VýsledkyAZnalstiŠtudenta*: Jeden zdroj prislúcha k jednému výsledku/znalosti študenta (kardinalita 1) a jeden výsledok/znalosť študenta môže byť vo viacerých zdrojoch (kardinalita 0..n).

Vzťah *ZdrojHodnotenia-Hodnotenie*: jeden zdroj patrí k jednému hodnoteniu (kardinalita 1) a jedno hodnotenie má viacero zdrojov (kardinalita 0..n).

Typ

Táto entita uchováva typy výsledku/znalosti študenta. Môže nadobúdať hodnoty ako napr. "Sebahodnotenie", "ŠtudentskéPráce" atď. Jeden výsledok/znalosť študenta je jedného typu (kardinalita 1) a jeden typ môže byť vo viacerých hodnoteniach (kardinalita 0..n).

4.2.2 Príklad výpočtu základného ohodnotenia kľúčových slov

Majme v tabuľke *Študent* (Tab.1) nasledovný záznam:

osobné č.	*meno*	*priezvisko*	*ročník*	*e-mail*
20851	Ján	Malý	3.	janko@it.sk

Tab.1: Študent

a pre tohto študenta budeme chcieť vypočítať ohodnotenia kľúčových slov, ktoré máme v tabuľke *KľúčovéSlová* (Tab.2)*:*

názov	*úroveň*	*predchodca*
programovacie jazyky	1	-
architektúra systémov	1	-
procedurálne prog. jazyky	2	programovacie jazyky
objektovo orientované prog. jazyky	2	programovacie jazyky
funkcionálne prog. jazyky	2	programovacie jazyky

Tab.2: KľúčovéSlová

V nasledujúcej tabuľke (Tab.3) sú váhy, ktorými prispievajú ohodnotenia jednotlivých predmetov k ohodnoteniu kľúčových slov:

kľúčové slovo	*predmet*	*váha*
architektúra systémov	architektúra informačných systémov	0,6
architektúra systémov	manažment v softvérovom inžinierstve	0,1
procedurálne prog. jazyky	algoritmizácia a programovanie	0,5
procedurálne prog. jazyky	dátové štruktúry a algoritmy	0,2
objektovo orientované prog. jazyky	objektovo orientované programovanie	0,4
objektovo orientované prog. jazyky	vývoj programov pre platformu Java2	0,4
funkcionálne prog. jazyky	funkcionálne a logické programovanie	0,8

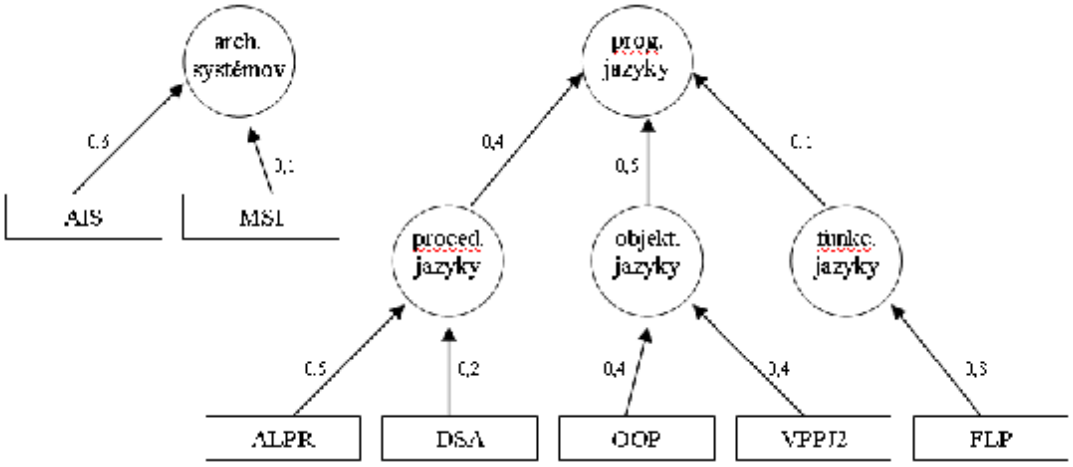
Tab.3: Váhy na ohodnotenie predmetov

a v tabuľke váh, ktorými prispievajú ohodnotenia kľúčových slov k ohodnoteniu ich predchodcov sú tieto záznamy (Tab.4):

kľúčové slovo	zdrojové kľúčové slovo	váha
programovacie jazyky	procedurálne prog. jazyky	0,4
programovacie jazyky	objektovo orientované prog. jazyky	0,5
programovacie jazyky	funkcionálne prog. jazyky	0,1

Tab.4: Váhy

Obrázok č. 4 znázorňuje hierarchiu kľúčových slov spolu s prepojeniami na predmety (názvy predmetov sú uvedené v skratkách).



Obr. 4: Hierarchia kľúčových slov

Teraz si môžeme ukázať ako vypočítať ohodnotenia kľúčových slov pre študenta Jána Malého.

Vezmeme si tabuľku kľúčových slov (Tab. 2). Zoberieme prvé kľúčové slovo ("programovacie jazyky") a vyhľadáme všetky predmety, ktoré prispievajú k ohodnoteniu tohto kľúčového slova (v Tab. 3). Zistíme, že taký predmet neexistuje (k ohodnoteniu tohto kľúčového slova prispievajú len iné kľúčové slová, ale žiaden predmet). Toto slovo si teda zatiaľ odložíme bokom (spracujeme ho neskôr).

Zoberieme druhé slovo ("architektúra systémov") a vyhľadáme predmety (v Tab. 3). V tomto prípade nájdeme dva, ktoré ovplyvňujú hodnotenie tohto slova, a to "architektúra informačných systémov" s váhou 0,6 a "manažment v softvérovom inžinierstve" s váhou 0,1.

Teraz potrebujeme zistiť ohodnotenie daných predmetov (AIS a MSI) pre študenta Jána Malého (túto informáciu dostaneme z externého systému - odtiaľ sme zistili že predmet AIS mal ohodnotený na 78 bodov a predmet MSI na 98 bodov).

Keď máme tieto informácie, tak môžeme kľúčové slovo "architektúra systémov" ohodnotiť. Jeho hodnotenie bude váženým súčtom zistených bodov podľa váh: hodnotenie prvého predmetu * váha, s ktorou prispieva prvý predmet k ohodnoteniu kľúčového slova + hodnotenie druhého predmetu * váha, s ktorou prispieva druhý predmet k ohodnoteniu kľúčového slova + ...

Čiže výsledné hodnotenie kľúčového slova "architektúra systémov" pre študenta Jána Malého bude 78*0,6 + 98*0,1=56,6 lebo predmet AIS mal Ján Malý ohodnotený na 78 bodov a toto hodnotenie prispieva k ohodnoteniu kľúčového slova váhou 0,6 a podobne pre predmet MSI je to 98*0,1. Takto prejdeme všetky ostatné kľúčové slová, ktoré máme v Tab. 3. Dostali

by sme napr. takéto výsledky: "procedurálne prog. jazyky" má ohodnotenie 52,1; " objektovo orientované prog. jazyky" má 68,7 a "funkcionálne prog. jazyky " má 34,9.

Nakoniec nám ešte zostanú slová, ktoré sme si odložili bokom, lebo sme nenašli žiaden predmet, ktorý by prispieval k ich ohodnoteniu (v našom prípade je to len kľúčové slovo "programovacie jazyky"). Vezmeme tieto slová a hľadáme, ktoré kľúčové slová prispievajú k ich ohodnoteniu (Tab. 4). V našom príklade pre kľúčové slovo "programovacie jazyky" to sú "procedurálne programovacie jazyky", "objektovo orientované programovacie jazyky" a "funkcionálne programovacie jazyky".

Ohodnotenia týchto kľúčových slov sme už vypočítali a ohodnotenie kľúčového slova "programovacie jazyky" bude váženým súčtom daných kľúčových slov, teda v našom príklade by to bolo 52,1*0,4 + 68,7*0,5 + 34,9*0,1 = 58,7.

4.2.3 Hierarchia kľúčových slov

Ako hierarchiu kľúčových slov budeme využívať ACM klasifikáciu - http://oldwww.acm.org/class/1998/. Keďže je táto klasifikácia veľmi rozsiahla, použijeme z nej len vhodnú časť. To znamená, že niektoré kategórie použijeme do hĺbky 3 a niektoré iba do hĺbky 2, prípadne len do hĺbky 1, podľa toho, ako veľmi sú dané kategórie relevantné s obsahom výučby na fakulte. V prílohe A sú uvedené prvé dve úrovne ACM klasifikácie.

Prílohy

Príloha A – ACM klasifikácia

- **A. General Literature**
  - A.0 GENERAL
  - A.1 INTRODUCTORY AND SURVEY
  - A.2 REFERENCE (e.g., dictionaries, encyclopedias, glossaries)
  - A.m MISCELLANEOUS
- **B. Hardware**
  - B.0 GENERAL
  - B.1 CONTROL STRUCTURES AND MICROPROGRAMMING (D.3.2)
  - B.2 ARITHMETIC AND LOGIC STRUCTURES
  - B.3 MEMORY STRUCTURES
  - B.4 INPUT/OUTPUT AND DATA COMMUNICATIONS
  - B.5 REGISTER-TRANSFER-LEVEL IMPLEMENTATION
  - B.6 LOGIC DESIGN
  - B.7 INTEGRATED CIRCUITS
  - B.8 PERFORMANCE AND RELIABILITY (C.4)
  - B.m MISCELLANEOUS
- **C. Computer Systems Organization**
  - C.0 GENERAL
  - C.1 PROCESSOR ARCHITECTURES
  - C.2 COMPUTER-COMMUNICATION NETWORKS
  - C.3 SPECIAL-PURPOSE AND APPLICATION-BASED SYSTEMS (J.7)
  - C.4 PERFORMANCE OF SYSTEMS
  - C.5 COMPUTER SYSTEM IMPLEMENTATION
  - C.m MISCELLANEOUS
- **D. Software**
  - D.0 GENERAL
  - D.1 PROGRAMMING TECHNIQUES (E)
  - D.2 SOFTWARE ENGINEERING (K.6.3)
  - D.3 PROGRAMMING LANGUAGES

- D.4 OPERATING SYSTEMS (C)
- D.m MISCELLANEOUS

- **E. Data**
  - E.0 GENERAL
  - E.1 DATA STRUCTURES
  - E.2 DATA STORAGE REPRESENTATIONS
  - E.3 DATA ENCRYPTION
  - E.4 CODING AND INFORMATION THEORY (H.1.1)
  - E.5 FILES (D.4.3, F.2.2, H.2)
  - E.m MISCELLANEOUS
- **F. Theory of Computation**
  - F.0 GENERAL
  - F.1 COMPUTATION BY ABSTRACT DEVICES
  - F.2 ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY (B.6, B.7, F.1.3)
  - F.3 LOGICS AND MEANINGS OF PROGRAMS
  - F.4 MATHEMATICAL LOGIC AND FORMAL LANGUAGES
  - F.m MISCELLANEOUS
- **G. Mathematics of Computing**
  - G.0 GENERAL
  - G.1 NUMERICAL ANALYSIS
  - G.2 DISCRETE MATHEMATICS
  - G.3 PROBABILITY AND STATISTICS
  - G.4 MATHEMATICAL SOFTWARE
  - G.m MISCELLANEOUS
- **H. Information Systems**
  - H.0 GENERAL
  - H.1 MODELS AND PRINCIPLES
  - H.2 DATABASE MANAGEMENT (E.5)
  - H.3 INFORMATION STORAGE AND RETRIEVAL
  - H.4 INFORMATION SYSTEMS APPLICATIONS
  - H.5 INFORMATION INTERFACES AND PRESENTATION (e.g., HCI) (I.7)
  - H.m MISCELLANEOUS
- **I. Computing Methodologies**

- I.0 GENERAL
- I.1 SYMBOLIC AND ALGEBRAIC MANIPULATION
- I.2 ARTIFICIAL INTELLIGENCE
- I.3 COMPUTER GRAPHICS
- I.4 IMAGE PROCESSING AND COMPUTER VISION
- I.5 PATTERN RECOGNITION
- I.6 SIMULATION AND MODELING (G.3)
- I.7 DOCUMENT AND TEXT PROCESSING (H.4, H.5)
- I.m MISCELLANEOUS

- **J. Computer Applications**
  - J.0 GENERAL
  - J.1 ADMINISTRATIVE DATA PROCESSING
  - J.2 PHYSICAL SCIENCES AND ENGINEERING
  - J.3 LIFE AND MEDICAL SCIENCES
  - J.4 SOCIAL AND BEHAVIORAL SCIENCES
  - J.5 ARTS AND HUMANITIES
  - J.6 COMPUTER-AIDED ENGINEERING
  - J.7 COMPUTERS IN OTHER SYSTEMS (C.3)
  - J.m MISCELLANEOUS
- **K. Computing Milieux**
  - K.0 GENERAL
  - K.1 THE COMPUTER INDUSTRY
  - K.2 HISTORY OF COMPUTING
  - K.3 COMPUTERS AND EDUCATION
  - K.4 COMPUTERS AND SOCIETY
  - K.5 LEGAL ASPECTS OF COMPUTING
  - K.6 MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS
  - K.7 THE COMPUTING PROFESSION
  - K.8 PERSONAL COMPUTING
  - K.m MISCELLANEOUS