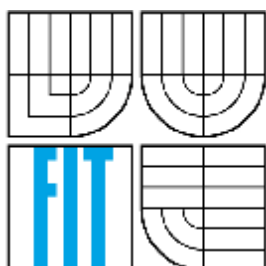




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
CENTRUM VÝPOČETNÍCH A INFORMAČNÍCH
SLUŽEB

FACULTY OF INFORMATION TECHNOLOGY
COMPUTER AND INFORMATION SERVICES CENTRE

STUDIJNÍ STATISTIKY NA PORTÁLU

STUDY STATISTICS FOR PORTAL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAELA GRUZOVÁ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROMÍR MARUŠINEC, Ph.D., MBA

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Centrum výpočetních a informačních služeb

Akademický rok 2006/2007

Zadání diplomové práce

Řešitel: **Gruzová Michaela, Bc.**

Obor: Informační systémy

Téma: **Studijní statistiky na portálu**

Kategorie: Databáze

Pokyny:

1. Prostudujte technologii Oracle, Portal VUT a jazyk PHP.
2. Prostudujte datové schéma ST01 a schéma portálu.
3. Analyzujte potřeby on-line studijních statistik útvaru prorektora pro záležitosti studentů.
4. Navrhněte vybrané potřeby výstupy jako tabulkové výstupy a grafy.
5. Implementujte navržené výstupy jako součást systému portál VUT.
6. Zhodnoťte přínosy práce a navrhněte možné rozšíření.

Literatura:

- Podle pokynů vedoucího.

Při obhajobě semestrální části diplomového projektu je požadováno:

- Splnění prvních 2 úkolů zadání.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

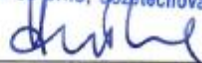
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Marušinec Jaromír, Ing., Ph.D.,** CVIS VUT

Datum zadání: 28. února 2006

Datum odevzdání: 22. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2



prof. Ing. Tomáš Hruška, CSc.
děkan FIT

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Slečna

Jméno a příjmení: **Bc. Michaela Gruzová**
Id studenta: 49560
Bytem: Sadová 5, 664 48 Moravany
Narozena: 20. 06. 1983, Brno
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
diplomová práce

Název VŠKP: Studijní statistiky na portálu
Vedoucí/školitel VŠKP: Marušinec Jaromír, Ing., Ph.D.
Ústav: Centrum výpočetních a informačních služeb VUT
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

Abstrakt

Diplomová práce se zabývá návrhem a implementací studijních statistik jako součást Portálu Vysokého učení technického v Brně. Rozebírá strukturu Portálu VUT a technologie využívané při jeho vytváření. Jedná se o serverový skriptovací jazyk PHP, dotazovací jazyk SQL a kaskádové styly CSS. Popisuje databázovou technologii Oracle a databázové schéma st01, což je rozsáhlé schéma, které obsahuje data využívaná informačními systémy. Analyzuje výchozí stav webových aplikací a stav na jednotlivých fakultách. Rozebírá různé způsoby řešení určitých částí aplikace a rozvádí nejvhodnější z nich. Na závěr popisuje samotnou implementaci studijních statistik a jejich začlenění do centrálního informačního systému.

Klíčová slova

WWW, vysokoškolský portál, informační systém, cluster, databáze, Oracle, Studis, Teacher, histogram četnosti, sloupcový graf, koláčový graf, statistika, HTML, XHTML, PHP, CSS, JpGraph

Abstract

The master's thesis discusses the project and implementation of study statistics of the Portal of Brno University of Technology. It analyses the structure of the BUT portal and the technologies used for its creation. The technologies we are speaking about are the server-side scripting language PHP, query language SQL and cascading style sheets CSS. It describes the Oracle database technology and the st01 database scheme, a large scheme containing data used by information systems. It analyses the initial state of web applications and the situation at single faculties. The thesis analyses different solutions of certain parts of the application and studies the most appropriate ones. At the end it describes the implementation of study statistics and its integration into the central information system.

Keywords

WWW, university portal, information system, cluster, database, Oracle, Studis, Teacher, frequency diagram, bar graph, pie graph, statistics, HTML, XHTML, PHP, CSS

Citace

GRUZOVÁ, M. *Studijní statistiky na portálu*. Brno: Vysoké učení technické v Brně, Fakulta informačních technologií, 2007. 75 s. Vedoucí diplomové práce Ing. Jaromír Marušinec, Ph.D., MBA.

Studijní statistiky na portálu

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně pod vedením Ing. Jaromíra Marušince, Ph.D., MBA. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

.....
Bc. Michaela Gruzová
21.5.2007

Poděkování

Ráda bych poděkovala vedoucímu práce Ing. Jaromíru Marušincovi, Ph.D, MBA., dále Ing. Rudolfu Musilovi, Ing. Tomáši Kreuzwieserovi a Ing. Markovi Strakošovi za poskytnuté rady a odbornou pomoc.

© Bc. Michaela Gruzová, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	5
1 Úvod	7
2 Statistika.....	9
2.1.1 Způsoby zobrazení dat.....	9
3 Specifikace požadavků.....	11
4 Analýza výchozího stavu	12
4.1.1 Portál VUT.....	12
4.1.2 Studis.....	13
4.1.3 Teacher	15
4.1.4 Apollo.....	17
4.1.5 Stav na fakultách.....	17
5 Technologie.....	20
5.1 Implementační prostředky	20
5.1.1 HTML, XHTML	20
5.1.2 PHP	21
5.1.3 CSS.....	24
5.1.4 SQL	25
5.2 Knihovny	26
5.2.1 ADODB	26
5.2.2 JpGraph.....	26
5.3 Software.....	27
5.3.1 Subversion	27
5.4 Hardware.....	28
5.4.1 Databázový server.....	29
5.4.2 Aplikační cluster serverů	29
6 Databáze VUT	33
6.1 Oracle 10g.....	33
6.2 Struktura databáze VUT	34
6.2.1 Konvence	36
6.3 Nový datový model	36
6.3.1 Popis nově vytvořených tabulek	39
7 Návrh řešení	42
7.1 Konvence používané při implementaci portálu.....	42
7.1.1 Adresářová struktura	42

7.1.2	Připojení k databázi	43
7.1.3	Autentizace uživatele	43
7.1.4	Jazykové verze	44
7.2	Model případů použití	44
7.2.1	Student	44
7.2.2	Vyučující	45
7.2.3	Uživatel	45
7.2.4	Přihlášený uživatel	45
7.3	ER diagram	45
8	Popis vlastní implementace	48
8.1	Realizace v aplikaci Studis	48
8.1.1	Statistika hodnocení předmětu	48
8.1.2	Statistika vážených průměrů	53
8.1.3	Statistika termínů zkoušek	54
8.1.4	Parametrizace	54
8.2	Realizace v aplikaci Teacher	55
8.3	Realizace na Portálu VUT	56
8.3.1	Statistika státní závěrečné zkoušky	56
8.3.2	Statistika úspěšnosti ukončení studia	58
8.3.3	Počet studentů po programech	59
8.3.4	Počet studentů po ročnících	60
8.3.5	Rozdělení a počet studentů na fakultách	60
8.3.6	Statistika přijímacího řízení	61
9	Závěr	62
	Literatura	63
	Seznam použitých zkratk	64
	Přílohy	65
	Příloha 1 ER diagram, část schématu st01	65
	Příloha 2 Model případů použití	67
	Příloha 3 SQL dotazy	68

1 Úvod

V současné době patří Internet k velmi rychle se rozvíjejícím médiím. Ještě nedávno byla většina informací na webu statických. Tento přístup je však v dnešní době zastaralý a uživatelé vyžadují především dynamické stránky, které jim poskytují interaktivní přístup k informacím všeho druhu. K dynamickým stránkám patří i webový portál Vysokého učení technického v Brně a studijní statistiky, které jsou předmětem této diplomové práce. Cílem práce je návrh studijních statistik a jejich implementací jako součást Portálu VUT.

Práce je rozdělena do šesti základních kapitol. První kapitola se věnuje úvodu do statistiky a statických metod. Statistické metody se zabývají získáváním, zpracováním a zobrazením hromadných dat. Jejich cílem je popsat vlastnosti hromadných jevů a odhalit souvislosti mezi nimi. Získaná data mohou být zobrazena pomocí různých typů grafů nebo tabulek.

V druhé kapitole je popsána specifikace požadavků na nově vytvořenou aplikaci. Další kapitola se týká analýzy výchozího stavu. Je zde popsán centrální informační systém VUT, do kterého patří webové aplikace Portál, Studis, Teacher a systém Apollo. Portál nabízí různé informace a služby pro jeho uživatele. Jeho první část slouží jako prezentační web VUT pro cizí návštěvníky a druhá jako primární zdroj informací pro jeho klienty po přihlášení. Aplikace Studis je určena studentům VUT, slouží pro podporu výuky a jsou zde převážně informace týkající se studia studenta. Webové rozhraní Teacher je využíváno pedagogickými pracovníky pro zadávání informací, podle kterých se orientují studenti a podle kterých je organizována výuka. U každé aplikace je zhodnocen současný stav statistik v ní implementovaných. V závěru kapitoly je rozebrán stav studijních statistik na jednotlivých fakultách.

Čtvrtá kapitola je věnována popisu technologií. Jsou zde uvedeny implementační prostředky používané při vývoji Portálu VUT, které jsem využila i při implementaci studijních statistik. Jedná se o skriptovací jazyk PHP, jazyk pro specifikaci rozvržení dokumentu HTML, standardizovaný dotazovací jazyk SQL a kaskádové styly CSS. Dále jsou popsány knihovny ADODB, JpGraph a software Subversion určený pro správu verzí zdrojových souborů aplikací. Poslední část kapitoly je věnována popisu databázového serveru a clusteru webových serverů.

Pátá kapitola seznamuje se strukturou centrálního datového skladu VUT a databázovým systémem Oracle 10g. Popisuje nový datový model vytvořený pro statistiky a jeho začlenění do již existujících struktur centrální databáze VUT.

Šestá kapitola se zabývá návrhem řešení. Nejdříve jsou popsány konvence, které se dodržují při implementaci webových aplikací. Jedná se o adresářovou strukturu systému a způsob ukládání souborů, funkce používané pro připojení k databázi, způsob autentizace

uživatele a jazykové verze Portálu. Je zde uveden model případů použití s popisem a ER diagram obsahující tabulky důležité pro statistiky. Důležité tabulky jsou i detailněji popsány.

Poslední kapitola je zaměřena na popis vlastní implementace. Popisuje postupy implementace jednotlivých statistik a jejich začlenění do částí centrálního informačního systému. Každá statistika je odlišná a proto kapitola popisuje různé způsoby získávání dat a také odlišný grafický výstup jednotlivých statistik.

2 Statistika

Statistika je naukou, jak získat informace z numerických dat. V běžné řeči se slovem statistika často míní znázorňování číselných údajů přehlednou formou. Jejím cílem je nahradit spoustu čísel pouze několika, a přitom charakterizovat a rozpoznat informace, které jsou v nich ukryty.

Statistické metody umožňují data znázorňovat, analyzovat vizuálně a numericky, zkoumat jejich struktury, určovat závislosti a dělat závěry o datech. Uplatňují se jak na akademických pracovištích, tak i ve všech profesích, kde pracovník potřebuje vyřešit problém systematickým způsobem nebo musí vyhodnotit závěry výzkumu provedeného někým jiným pro svoji potřebu. Statistika hraje významnou roli při politickém rozhodování, v psychologickém výzkumu, sociologii, pedagogice, ve světě ekonomiky a obchodu. V některých z těchto disciplín vznikly i specializace se zaměřením na statistické úlohy.

Statistiku můžeme rozdělit na tři části:

1. Získávání dat – zahrnuje metody pro sběr dat, které odpoví na předem určenou otázku.
2. Analýza dat – organizace dat a jejich popis použitím tabulek, grafů a dalších matematických prostředků. Této oblasti se někdy říká popisná statistika a je velmi důležitá právě pro tuto práci.
3. Statistické usuzování – cílem je získat závěry o určitých jevech. Také sem patří zhodnocení, jak jsou závěry spolehlivé. Této části se říká inferenční statistika.

2.1.1 Způsoby zobrazení dat

Účelem popisné statistiky je data přehledně zpřístupnit graficky, tabulkově a výpočtem různých statistických charakteristik tak, aby byly dobře patrné jejich statistické vlastnosti. Rozhodnutí, jestli zobrazit údaje v tabulce nebo pomocí obrázku záleží na samotném řešiteli. Tabulkové metody jsou vhodnější, pokud chceme vybrané údaje uvést v přesném tvaru. Přestože lze data jednoduchým způsobem zobrazit v tabulce, pro uživatele není tento způsob příliš přehledný. Grafy umožňují přehlednější orientaci a poskytují větší názornost než tabulky. Nevýhodou je menší přesnost a určité zjednodušení, protože vše nelze vždy zobrazit.

Používá se několik typů grafů:

- **Spojnicový graf** – umožňuje zobrazení souvislých dat v čase a jejich srovnání se společnou stupnicí. Je ideální pro zobrazení trendů v datech ve stejných intervalech. Ve spojnicovém grafu jsou kategorie rovnoměrně rozloženy podél vodorovné osy a všechny hodnoty dat jsou rovnoměrně rozloženy podél svislé osy.

- **Sloupcový graf** – je tvořen ze sloupců, které mají stejnou šířku a výška vyjadřuje velikost příslušné hodnoty. Každý sloupec odpovídá jedné třídě. Sloupcové grafy se používají k porovnání jedné nebo více hodnot. Mohou být horizontálně nebo vertikálně orientované.
- Speciální typ sloupcového grafu je **histogram**. V něm osa X odpovídá hodnotám proměnné a osa Y absolutním nebo relativním četnostem. Histogram může mít symetrický tvar nebo může být zešikmen na pravou, respektive levou stranu. Také může mít jeden, dva nebo více vrcholů. Histogram prokládáme někdy ideální křivkou, která se nazývá hustota. Tvar histogramu se často porovnává s hustotou, která se nazývá Gaussova křivka nebo normální křivka. Gaussova křivka je symetrická křivka zvonovitého tvaru a data s tímto rozložením se nazývají normálně rozložená data.
- **Kruhový (výsečový) graf** – je znázornění pomocí výsečí kruhu, kde každé třídě odpovídá jedna výseč. Velikosti obsahu výsečí odpovídají četnostem třídy.

3 Specifikace požadavků

Cílem této práce je zpracování a přehledné zobrazení hromadných dat. Databáze VUT obsahuje obrovské množství čísel a účelem statistiky je nahradit je pouze několika a přitom charakterizovat a vyzvednout informace, které jsou v nich ukryty.

Studijní statistiky určené pro studenta, by měly převážně zpracovávat výsledky studentova studia a porovnávat jeho výsledky s jinými studenty. Student by měl vidět histogramy hodnocení předmětů, které studuje a také předmětů, které studoval dříve. Kromě celkového hodnocení by měly být k dispozici i výsledky hodnocení jednotlivých termínů zkoušek vypsanych vyučujícími a také histogramy výsledků jednotlivých pokusů o složení zkoušky. Pro větší motivaci studenta by student mohl vidět svůj vážený průměr a také umístění v ročníku podle tohoto průměru. Součástí by měl být i histogram vážených průměrů.

Statistiky určené pro vyučující se budou týkat převážně hodnocení předmětů, které vyučují nebo u kterých zadávají hodnocení. Vyučující by měli vidět souhrnné histogramy hodnocení u těchto předmětů a také histogramy hodnocení pro všechny ostatní vyučující, kteří zadávali hodnocení. Budou tak moci porovnat svoje hodnocení s ostatními vyučujícími. K dispozici by také měly být histogramy výsledků jednotlivých pokusů o složení zkoušky.

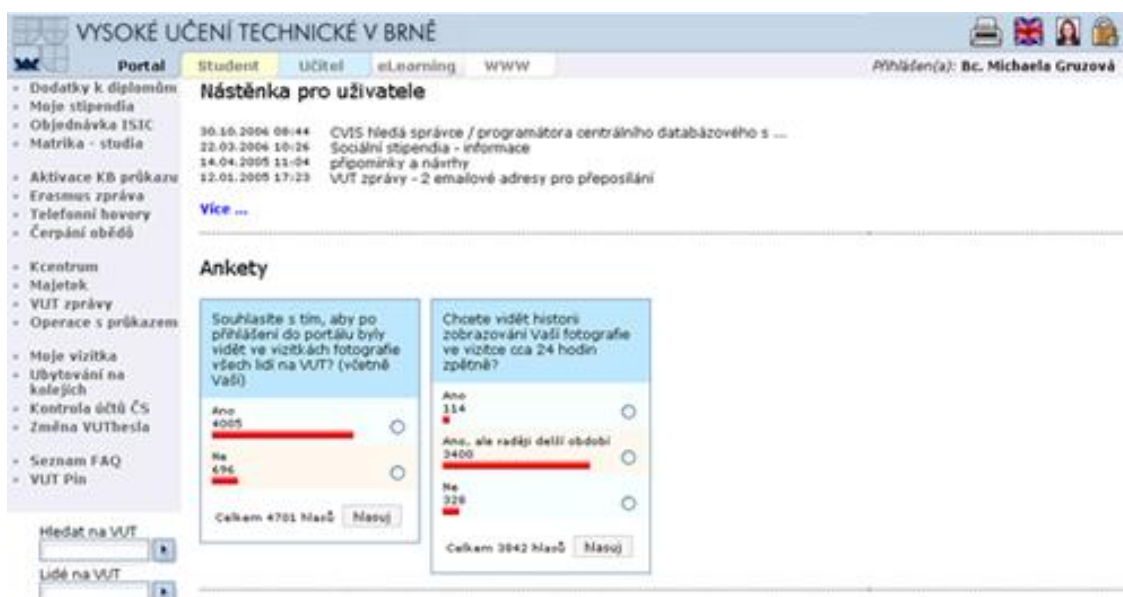
Další část statistik by měla být přístupná jak vyučujícím, tak i studentům. Neměla by být ale veřejná, takže by se měla zobrazovat pouze po přihlášení. Bude se týkat především úspěšnosti dokončení studia na fakultách a počtu a rozdělení studentů na fakultách. Měly by zde být zobrazeny počty studentů v jednotlivých programech na fakultách, počty studentů v jednotlivých programech a ročnících na fakultách a také statistika rozmanitosti studentů na fakultě. Také by měla být dostupná statistika týkající se státní závěrečné zkoušky, tzn. histogram celkového hodnocení jednotlivých pokusů o vykonání státní závěrečné zkoušky, histogramy hodnocení ústní části SZZ a diplomové práce.

U všech statistik je důležité, aby se zobrazovaly pouze oprávněným osobám a nikdo jiný k nim neměl přístup.

4 Analýza výchozího stavu

4.1.1 Portál VUT

Vysokoškolský informační portál je velmi širokým pojmem pro označení informačního systému přístupného obvykle přes webové rozhraní a nabízející různé informace a služby pro jeho uživatele. Portál VUT má dvě části, jedna slouží jako prezentační web VUT pro cizí návštěvníky a druhá jako primární zdroj informací pro jeho klienty po přihlášení – konkrétně zaměstnance VUT, pedagogy, studenty a další. Důležitým požadavkem na portál je schopnost spravovat co největší počet školských agend. Portál spravuje osobní informace všech uživatelů, kteří jsou schopni se přihlásit a vhodně je prezentuje. Také spravuje webové prezentace součástí VUT, které jsou navzájem odlišné, ale mají společný základ, který prostředím portálu poskytuje.



Obrázek 4.1 Ukázka prostředí aplikace Portál

Po přihlášení na Portál VUT se uživatel dostane do části, kde najde důležité informace, které se ho týkají. Je zde například modul moje stipendia, který zahrnuje studijní, sociální a ubytovací stipendia. Uživatel zde může o stipendium požádat a také vidí všechna stipendia, která mu byla přidělena a která pobírá. V další aplikaci si může student objednat ISIC kartu nebo průkaz Komerční banky.

Aplikace čerpání obědů slouží pro informaci, kolik obědů zaměstnanec vyčerpal a na kolik má nárok. Operace s průkazem zase zobrazuje seznam všech karet uživatele a u nich operace, které byly s průkazy prováděny. V modulu majetek je seznam majetků uživatele. Také

je zde aplikace pro FAQ (*Frequently Asked Questions*), kde může uživatel zadat dotaz a zároveň vidí dotazy od jiných uživatelů spolu s odpověďmi na ně.

V modulu Moje vizitka může uživatel editovat vlastní životopis a kontaktní údaje, které se zobrazují po jeho vyhledání na webu VUT. Také zde může nastavit, zda má být jeho fotka ve vizitce zobrazena veřejně i uživatelům bez oprávnění.

Součástí portálu jsou také VUT zprávy. Po přihlášení má uživatel přístup ke své schránce, kde vidí seznam přijatých a odeslaných zpráv a také může poslat zprávu ostatním osobám, které mají vztah k VUT. V nastavení si může nastavit přeposílání zpráv na e-mailovou adresu. Tyto VUT zprávy jsou využívány ve všech aplikacích portálu. Většinou jsou zprávy posílány, pokud si student něco registruje nebo si něco objedná, např. pokud si objedná kartu ISIC, zažádá o stipendium nebo si registruje výuku. Zprávy jsou také zasílány, pokud vyučující zadá do systému zápočet nebo hodnocení studenta. Vyučující mohou posílat i hromadné zprávy studentům.

Portál obsahuje publikační systém, který umožňuje uživatelům publikovat obecné dokumenty. Systém využívá pro přístup do jednotlivých složek protokol LDAP a rozhoduje tak, jaké operace uživateli nad danou složkou povolí vykonat.

Portál VUT obsahuje jen velmi málo statistik. Jednou z nich je statistika přijímacího řízení, která zobrazuje informace o výsledcích jednotlivých kol přijímacího řízení. Jsou zde např. tyto informace: počet podaných přihlášek a přihlášených uchazečů na fakultu, počet uchazečů, kteří se zúčastnili přijímacích zkoušek, včetně přijímacích zkoušek v náhradním termínu, počet uchazečů, kteří splnili podmínky přijetí (včetně uchazečů nepřijatých z kapacitních důvodů), počet uchazečů, kteří nesplnili podmínky přijetí (včetně uchazečů, kteří se nedostavili k přijímací zkoušce), počet uchazečů přijatých ke studiu, bez uchazečů přijatých ke studiu až na základě výsledku přezkoumání původního rozhodnutí a počet uchazečů přijatých celkem.

Pro studenty slouží část portálu nazvaná Studis, pro vyučující existuje rozhraní s názvem Teacher.

4.1.2 Studis

Studis je webová aplikace, která je určena studentům VUT. Slouží pro podporu výuky a jsou zde převážně informace týkající se studia studenta. Tuto aplikaci využívá Fakulta architektury, Fakulta chemická, Fakulta podnikatelská, Fakulta strojního inženýrství a Fakulta výtvarných umění.

Aplikace je přehledně rozčleněna do modulů, přičemž každý modul představuje jednu položku v navigačním menu. Vzhledem k tomu, že aplikaci používá více fakult a každá potřebuje mít v jiný čas aktivní jiné moduly, jsou moduly rozděleny na statické a dynamické.

Část statická je zobrazena stále a druhá část je generována dynamicky podle parametrizace. U dynamických modulů je možné nastavit datum a čas, kdy má být modul zobrazen v menu a tedy kdy má být přístupný studentům. Také je možné nastavit určité parametry, které určují chování aplikace, např. rok, pro který má být pro danou fakultu aplikace spuštěna.

The screenshot shows the 'Studium' section of the Studis application. It features a navigation menu on the left with options like 'Osobní informace', 'Kontaktní adresy', and 'Studium'. The main content area displays a table titled 'Výběr studia:' with a dropdown menu set to '01.09.2003 | FSI | Strojni inženýrství | Inženýrská informatika a automatizace - aktivní'. Below the table is a 'Ukázat' button.

ok. rok	datum	popis změny	stup	roč.	obor	program	fakulta
2006	01.09.2006	11 - řádný zápis	2	1	Inženýrská informatika a automatizace	Strojní inženýrství	FSI
2005	14.02.2006	19 - změna zařazení - programu, formy, oboru, ročníku	1	3	Strojní inženýrství - MS	Strojní inženýrství	FSI
2005	01.09.2005	11 - řádný zápis	1	3	Strojní inženýrství	Strojní inženýrství	FSI
2004	01.09.2004	11 - řádný zápis	1	2	Strojní inženýrství	Strojní inženýrství	FSI
2003	01.09.2003	11 - řádný zápis	1	1	Strojní inženýrství	Strojní inženýrství	FSI

Obrázek 4.2 Ukázka prostředí aplikace Studis

Mezi moduly, které jsou aktivní stále, patří osobní informace, kontaktní adresy, studium a moje závěrečná práce. V prvních dvou najde student informace, týkající se jeho osobních údajů a adres. V modulu studium jsou informace o minulých i aktuálních studiích přihlášeného studenta. Je zde seznam všech studií, přičemž jedno je vždy nastaveno jako „aktivní“ a k němu se vztahují zobrazené záložky. Student zde může mezi studii přepínat, přičemž po vybrání jiného studia se vždy záložky aktualizují – jsou zobrazeny vždy jen ty, které má daná fakulta zapnuté. V části moje závěrečná práce se studentovi zobrazují informace o přidělené závěrečné práci pro aktuální akademický rok. Také zde jsou odkazy pro vygenerování popisného souboru závěrečné práce, vzoru titulního listu, desek závěrečné práce a licenční smlouvy.

Moduly dynamické se vztahují vždy k nějakému určitému časovému horizontu v roce (např. registrace) nebo je používají pouze některé fakulty (např. rozvrhy). Jedním z nejdůležitějších modulů je elektronický index, který je obdobou papírové formy indexu. Student zde vidí zapsané předměty i s jejich výsledky za celé studium přehledně rozdělené podle akademických let a semestrů. Po kliknutí na název předmětu se student dostane na detail předmětu, kde se může registrovat na termíny zkoušek nebo zadání projektů. Do elektronického indexu budou umístěny studijní statistiky, které se budou týkat hodnocení předmětu a zkoušek.

K dalším modulům patří různé registrace – registrace předmětů do individuálního plánu, registrace studijní skupiny, oboru, vyučování a registrace závěrečné práce. Všechny registrace jsou založeny na principu „kdo se dříve zaregistruje, tomu je registrace přidělena“. U všech student vidí aktuální počet a maximální počet registrací, a tak ví, jestli je kapacita už naplněna nebo se může ještě zaregistrovat.

Ve Studisu také student najde několik modulů, které se týkají rozvrhů. Je zde individuální rozvrh studenta, který zobrazuje jen zaregistrovanou výuku studenta, rozvrhy ostatních

studentů, vyučujících a rozvrhy předmětů. Dále zde najde jazykový rozřazovací test, který vyplní a podle výsledku je podle pokročilosti umístěn do příslušné jazykové skupiny. V modulu hodnocení předmětů může student vyjádřit svou spokojenost či nespokojenost s výukou jednotlivých předmětů.

Poslední moduly se týkají ukončení studentova studia. Je to přihláška k SZZ, kde si student vybere termín konání SZZ a zkontroluje a potvrdí výsledky svého studia. A kontrola dodatku k diplomu, kde student potvrdí nebo nepotvrdí správnost údajů, které se tisknou na diplom.

Jediná statistika implementovaná v systému Studis se týká výsledků hodnocení předmětů studenty. Statistika je rozdělena na dvě části podle časového hlediska – aktuální období a minulé období. První zobrazuje aktuální statistická data k danému předmětu. Data jsou zobrazována v textovém a grafickém formátu (sloupcové grafy). Jsou zde vypsané jednotlivé otázky, které se týkají hodnocení předmětu a informace o odpovědích studentů na ně.

V druhé části je zobrazena tabulka obsahující předešlé akademické roky včetně příslušných semestrů od spuštění systému po současnost. Výběrem příslušné položky je zobrazen seznam předmětů, které byly vyučovány právě ve vybraném období. V tabulce jsou informace o předmětu a informace týkající se statistiky – počet studentů, kteří předmět navštívili, počet odeslaných anket k danému předmětu a procentuálně vyjádřený počet odeslaných anket vůči zapsaným studentům. Po výběru předmětu se student dostane na statistiku anket pro aktuální období.

4.1.3 Teacher

Teacher je aplikace určená pedagogickým pracovníkům. Toto webové rozhraní slouží vyučujícím pro zadávání informací, podle kterých se orientují studenti a podle kterých je organizována výuka. Rozhraní využívá Fakulta architektury, Fakulta podnikatelská, Fakulta strojního inženýrství a Fakulta výtvarných umění. Vývoj této aplikace začal ve stejném období jako aplikace Studis a proto je jejich základ velmi podobný. Systém je opět postaven na modulární struktuře, každý modul tvoří záložku v menu.



Obrázek 4.3 Ukázka prostředí aplikace Teacher

První část aplikace se týká předmětů. Přihlášený pracovník má k dispozici seznam předmětů, u kterých je veden jako garant nebo správce, a proto může vyplňovat nebo editovat jejich karty. Zadávají se zde údaje jako obsahová anotace, cíl, charakteristika získaných vědomostí a znalostí, podmínky udělování zápočtů, forma zkoušek a způsob a pravidla výsledné klasifikace, kontrolovaná výuka, prerekvizity, osnova a literatura předmětu. Pro tisk karet předmětů je zde také vytvořena tisková sestava. K jednotlivým předmětům může garant přidávat další správce, kteří budou také moci editovat kartu předmětu.

Druhou částí systému jsou zkoušky, které jsou dále rozděleny na zápis hodnocení dle ECTS, zápis bodového hodnocení, zápis zápočtů, vypisování termínů a vypisování společných termínů. V jednotlivých záložkách pracovník opět vidí seznam předmětů, kterých je garantem nebo správcem, přičemž v záložce vyučující může přidávat právo na zápis hodnocení nebo zápočtů dalším osobám. Rozdíl mezi zápisem hodnocení dle ECTS a zápisem bodového hodnocení je ten, že v první aplikaci vyučující zapisuje hodnocení studenta pouze pomocí „A“, „B“, „C“, „D“, „E“, „F“. Ve druhé zapisuje získané body studenta – minimum je 0 a maximum 100 bodů. V modulu zápis zápočtů se udělují zápočty. Ve všech třech aplikacích může vyučující poslat informační e-mail různým skupinám studentů – všem studentům na seznamu, vybraným studentům ze seznamu, všem hodnoceným nebo všem nehodnoceným studentům. V modulu vypisování termínů vyučující vypisuje termíny zkoušek, na které se potom studenti ve Studisu registrují. Vypisování společných termínů je velmi podobné, ale je zde možné vypsát jeden termín zkoušky, který bude společný pro více předmětů.

Třetí částí jsou závěrečné práce s moduly moje zadání, vypisování předběžných zadání, vypisování zadání a seznam předběžných zadání. V části moje zadání vidí přihlášený uživatel předběžná a konečná zadání, u kterých je správce, garant nebo vedoucí. Zadání zde může filtrovat podle různých kritérií, může je editovat, kopírovat nebo smazat. V části vypisování zadání a vypisování předběžných zadání vytváří nová zadání a předběžná zadání. Také je zde možné registrovat studenty na závěrečnou práci, přidělit studentovi práci nebo studentovu registraci zamítnout. Další modul obsahuje seznam předběžných zadání pro celou fakultu. Opět je možné seznam mnoha způsoby filtrovat, aby se usnadnilo vyhledání určitých zadání. Rovněž je možné zadání schválit nebo zamítnout. To však mohou jenom oprávněné osoby. Studenti se mohou ve Studisu registrovat jen na schválená zadání, ta neschválená jim nejsou k registraci nabídnuta. V tomto modulu jsou také vytvořeny tiskové sestavy pro jednotlivá zadání nebo pro skupinu zadání.

Rozvrhy jsou další skupinou v systému. Jsou podobné jako v aplikaci Studis, vyučující si může vyhledat a zobrazit rozvrh ostatních pedagogů, studentů a také jednotlivých předmětů.

Poslední částí jsou skupiny, kam patří seznam vyučovacích skupin s možným tiskem a také odesílání e-mailů různým skupinám studentů. U obou aplikací je velké množství filtrů, které umožňují lepší výběr skupiny studentů podle aktuálních potřeb uživatele.

V této aplikaci jsou k dispozici opět jen statistiky týkající se hodnocení předmětů, které jsou popsány v kapitole 4.1.2 Studis.

4.1.4 Apollo

I když Apollo nepatří k webovým aplikacím, zařadila jsem ho do tohoto popisu, protože se o něm budu zmiňovat v návrhu řešení a je tedy důležité vědět, co název Apollo znamená. Apollo je informační systém primárně určen pro pedagogické a provozní pracovníky VUT.

Je členěn na provozně logické celky reprezentující jednotlivé moduly. Ty jsou reprezentovány DLL knihovnamí, které mají přesně definované rozhraní, a proto lze provádět aktualizace samotných modulů nezávisle na sobě. Také uživateli není spuštěna celá aplikace, ale pouze určitý modul, čímž je systém méně náročný na provoz.

Apollo je tenký aplikační klient v třívrstvé architektuře. To znamená, že se klient nepřipojuje přímo k databázi, ale na aplikační servery s názvem Akira. Klient při připojení provede oznámení o tom, že hodlá navázat spojení s jedním z serverů Akira. Ty odpoví a informují klienta o počtu již připojených uživatelů. Klient informaci vyhodnotí a provede připojení k serveru, který má nejmenší počet uživatelů, takže se předpokládá jeho nejmenší zatížení.

Data z databáze jsou opět získávána pomocí právě připojeného aplikačního serveru, který provede uložený SQL dotaz a výsledek předá klientovi. Touto operací vzniká další vrstva, která je vložena mezi klienta a databázový server. Jedná se o vrstvu reprezentující aplikační server, která tímto vytváří třívrstvou architekturu. Akira je server této střední vrstvy postavený na technologii Asta.

Samotná komunikace mezi klientem a serverem probíhá přes zabezpečené připojení za pomoci protokolu SSL (*Secure Sockets Layer*). Pokud chce klient provádět jakékoliv operace s daty, neodesílá ani nevytváří SQL dotaz, ale řekne serveru, který z uložených SQL dotazů má vykonat. Během této operace je provedena kontrola oprávněnosti této operace, která zabraňuje neoprávněnému uživateli v získání nebo modifikaci dat. Pro zrychlení předávání dat mezi serverem a klientem jsou data komprimovaná a v případě potřeby jsou posílána pouze po částech na požádání.

4.1.5 Stav na fakultách

Většina fakult využívá centrální informační systém se statistikami, které obsahuje. Ostatní fakulty mají studijní statistiky ve svém fakultním systému nebo nemají žádné. Výsledky průzkumu pro zjištění stavu studijních statistik na fakultách:

- **Fakulta chemická** – na fakultním webu nemají žádné statistiky, studenti a vyučující využívají centrální informační systém.

- **Fakulta výtvarných umění** – žádné studijní statistiky na fakultním webu, vyučující a studenti využívají centrální informační systém.
- **Fakulta strojího inženýrství** – momentálně přístupné statistiky vyhodnocení vědy a výzkumu. Vyučující a studenti využívají centrální informační systém.
- **Fakulta podnikatelská** – na fakultním webu nejsou žádné statistiky, vyučující a studenti využívají centrální informační systém.
- **Fakulta informačních technologií** – používá vlastní fakultní informační systém a v něm následující statistiky. Histogram výsledků každého hodnocení (projekt, test, zkouška apod.). Viditelné jen pro účastníka daného hodnocení (studenta). Histogram hodnocení pro autora hodnocení, případně garanta nebo vyučujícího. Pro studijní oddělení a učitele přehled zapsaných výsledků (kolik studentu v kterém předmětu uspělo, kolik neuspělo, kolik nedostalo ani bod). V průběhu přijímacího řízení je pro zaměstnance k dispozici on-line statistika úspěšnosti zkoušek.
- **Fakulta stavební** – má vlastní informační systém s následujícími statistikami.
 1. Pro vyučující (výběr omezen na předměty příslušného ústavu, pokud má vyučující vyšší oprávnění, vidí i data ostatních vyučujících svého ústavu, jinak vidí jen svá vlastní data):
 - § Hodnocení zkušebních termínů – úspěšnost získání zápočtu nebo složení zkoušky podle jednotlivých předmětů, zkoušejících a termínů. Včetně zobrazení histogramů.
 - § Klasifikace studentů – zobrazení klasifikace studentů podle jednotlivých předmětů, zkoušejících, termínů a typu hodnocení (úspěšný, neúspěšný, nedostavil se, omluven, neomluven atd.).
 - § Hodnocení kvality výuky – hodnocení předmětu jako celku, hodnocení kvality výuky přednášejících a cvičících, hodnocení znalostí studentů.
 - § Dále systém obsahuje různé seznamy studentů a přehledy, jako například přehled přihlášek pro zaměření pro 5. ročník a obory pro 3. ročník, přehled objednávek povinně volitelných a volitelných předmětů pro následující akademický rok, výpis z listu studia konkrétního studenta pro předměty příslušného ústavu (včetně hodnocení, počtu zápisů předmětu apod.), přehled závěrečných prací příslušného ústavu, seznamy studentů s možností zobrazení počtu studentů, kteří mají předmět zapsán a přitom mají zápočet/zkoušku splněn v daném akademickém roce, zápočet/zkoušku uznán v daném akademickém roce, zápočet/zkoušku splněn již z předchozích let.
 2. Agendy studijního oddělení (vše výše uvedené, referentky však mají možnost pracovat s daty všech ústavů a všech studentů):

- § Seznamy studentů pro SZZ – různé možnosti filtrace.
- § Počty studentů – podle studijních programů, ročníků, oborů, zaměření, stavu studia a studijních skupin.
- § Problematičtí studenti – všichni studenti nezapsaní pro aktuální akademický rok, kteří nemají ukončeno studium, studenti se souběžnými studii, studenti s nízkým počtem kreditů apod.
- **Fakulta elektrotechniky a komunikačních technologií** – má vlastní informační systém obsahující různé seznamy, které zároveň umožňují získávat statistické údaje. Čistě specializované statisticky jsou pouze výsledky v předmětech. Slouží vždy pouze osobám, které mají příslušný vztah nebo oprávnění (garant, proděkan atd.). Veřejné nejsou žádné. Pro studenty jsou pouze zveřejňovány statistiky výsledků v předmětech, které hodnotilo více jak 20% studentů.

Jak je vidět z výše uvedeného průzkumu, některé fakultní informační systémy obsahují více studijních statistik než samotný centrální informační systém. Statistiky jsou důležitou součástí webových aplikací, a proto by neměly chybět ani v centrálním informačním systému. Cílem této práce je tedy rozšířit možnosti tohoto systému a poskytnout uživatelům z fakult, které používají tento systém, důležité statistické informace.

5 Technologie

Tato kapitola obsahuje popis technologií, jež se používají při vývoji portálu Vysokého učení technického v Brně, a které jsem využila při implementaci studijních statistik.

5.1 Implementační prostředky

5.1.1 HTML, XHTML

HTML (*Hypertext Markup Language*) je jazyk pro specifikaci rozvržení dokumentu a hypertextových odkazů. Používá přesně definovanou syntaxi speciálních příkazů, které prohlížeč nezobrazuje, ale které určují, jak se zobrazí obsah dokumentu. Současně jazyk umožňuje vytváření interaktivních dokumentů pomocí speciálních hypertextových odkazů, které propojují dokument s jinými dokumenty.

Základní syntaxe a sémantika jazyka je definována ve standardu HTML. Ten se vyvíjí pod dohledem konsorcia W3C (*World Wide Web Consortium*), jeho specifikaci je možné najít na adrese <http://www.w3c.org>. Poslední verze schválená W3C je HTML 4.01. Je to poslední verze jazyka HTML, který se již dále nevyvíjí, protože má být nahrazen novějším XHTML (*Extensible Hypertext Markup Language*). Vývoj tedy pokračuje jiným jazykem, ale specifikace HTML platí dál – je tedy možné vyvíjet nové aplikace a vytvářet dokumenty podle HTML 4.01 všude tam, kde tento jazyk stačí. Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

Jazyk HTML je založen na obecnějším jazyku SGML (*Standard Generalized Markup Language*), který tvoří jakýsi základ pro všechny značkovací jazyky. Ten definuje všechny elementy pro označování dokumentů, pomocí kterých se dá nadefinovat vše, od hieroglyfů až po HTML, takže teoreticky ostatní značkovací jazyky nejsou zapotřebí. Bohužel tato velká šíře záběru jazyka je i jeho problémem. Pracovat s ním vyžaduje nejen složité a nákladné nástroje, ale i dokonalé pochopení podstaty a vlastností SGML, což je pro velkou část uživatelů příliš složité. Pro účely generování webových stránek tak vznikla zjednodušená instance SGML ve formě HTML jazyka, který zjednodušeně popisuje nejpoužívanější konstrukce a vyhýbá se těm extrémně složitým.

Jelikož vznikla potřeba definovat další značkovací jazyky, například pro zpracování síťových dokumentů, byl ze SGML vyvinut samostatný formátovací značkovací jazyk XML (*Extensible Markup Language*). Jsou v něm vynechány různé prvky SGML, které se v jazycích typu HTML nedají uplatnit a jiné elementy jsou v něm zjednodušeny, takže je celý jazyk lépe srozumitelný a lépe se s ním pracuje. Jazyk HTML však standardu XML nevyhovuje, a proto

vznikla alternativa XHTML (*Extensible Hypertext Markup Language*). XHTML je v podstatě jazykem HTML přeformulovaným do prostředí XML, snaží se podporovat co největší podmnožinu funkcí a prvků jazyka HTML, ale dodržuje přitom přísnější pravidla XML. XHTML 1.0 je první specifikace jazyka, jejímž cílem bylo převedení staršího jazyka HTML tak, aby vyhovoval podmínkám tvorby XML dokumentů a přitom byla zachována zpětná kompatibilita. Existuje ve třech verzích: *Strict*, *Transitional* a *Frameset*. Jediným podstatným rozdílem proti HTML je, že veškeré prvky jsou párové a musí být uzavřeny. Hodnoty všech atributů musí být zapsány v uvozovkách, všechny tagy a jejich atributy musí být napsány malými písmeny. Další verze je XHTML 2.0, která už není zpětně kompatibilní se svými předchůdci.

Jazyky HTML a XHTML nabízí řadu různých způsobů pro definici vzhledu dokumentu, obsahují prvky a jejich vlastnosti, které ho dovedou ovlivňovat. Důležitější je však obsah dokumentu, a proto značkovací jazyky obsahují mnohem více možností strukturování obsahu dokumentu, které neovlivňují přímo jeho výsledný vzhled. Jsou to záhlaví oddílů, strukturované seznamy, odstavce, oddělovače, titulky a vložené obrázky. Tyto prvky by měly umět definovat všechny značkovací jazyky, aniž by to ovlivnilo jejich výsledný vzhled v prohlížeči.

Webová prezentace portálu VUT se snaží držet normy HTML 4.01 Transitional, která zajišťuje zpětnou kompatibilitu se staršími kódy a zároveň umožňuje použít moderní zobrazování dokumentů s definicí kaskádových stylů CSS.

Základní struktura HTML dokumentu:

```
<HTML>
  <HEAD>
    <TITLE>Jméno dokumentu</TITLE>
  </HEAD>
  <BODY>
    Zobrazovaný obsah dokumentu.
  </BODY>
</HTML>
```

5.1.2 PHP

PHP je skriptovací jazyk běžící na straně serveru, jehož skripty se začleňují přímo do HTML stránek. Aby server poznal, že se nejedná o obyčejné HTML soubory, ukládají se s příponou .php. Vždy když server obdrží požadavek na soubor končící s touto koncovkou, předá jej systému PHP. Ten provede všechny příkazy uložené v souboru a výsledek předá zpět serveru, který výslednou stránku zobrazí klientovi. Výsledkem běhu PHP skriptu je pak obyčejný HTML dokument.

Jazyk PHP byl vytvořen v roce 1994, kdy Rasmus Lerdorf naprogramoval množinu skriptů sloužících k evidenci přístupu k jeho stránkám. Postupně se na vývoji podíleli další lidé a systém se od své první verze s primitivní funkcionalitou vyvinul v mocný programovací nástroj s velkou oblibou a podporou vývojářů. Název PHP původně znamenal *Personal Home Page*, ale později byl změněn na *Hypertext Preprocessor*. Poslední verzí je PHP 5.0.

PHP může být zkompileováno jako samostatný interpret CGI nebo jako modul Apache. V PHP nastaveném jako interpret CGI je skript PHP pokaždé interpretován, webový server pro něj vytvoří novou instanci interpretu, která tento skript zpracuje. To u větších systémů vede ke snížení výkonu webového serveru. Pokud je PHP zkompileováno jako modul Apache, běží ve stejném adresovém prostoru jako sám proces webového serveru a proto poskytuje větší výkon. Některé funkce, jako např. trvalé spojení s databází, jsou dostupné pouze, pokud je PHP nainstalováno jako modul.

Mezi hlavní výhody skriptovacích jazyků patří fakt, že je skript vykonán na serveru a tak na straně klienta odpadají problémy s případnou nepodporou tohoto vývojářského nástroje. Pro korektní činnost programu stačí každý prohlížeč zobrazující HTML. To s sebou nese i další výhodu, že se klientovi nikdy nedostane zdrojový text programu. Za nevýhodu lze považovat to, že takovéto programy nemohou bezprostředně reagovat na události způsobené uživatelem.

Obliba PHP je natolik vysoká, že je považován svým způsobem za standard a málokterý webhosting jej nepodporuje. Používá se pro tvorbu složitých redakčních systémů, elektronických obchodů, webových e-mailových služeb, systémů pro správu e-mailové konference, webových fór atd. Hlavními konkurenty PHP jsou Perl, ASP (*Microsoft Active Server Pages*), JSP (*Java Server Pages*).

Některé vlastnosti PHP:

- **Výkonnost**

PHP je velmi efektivní. Jeden jednoduchý a nenáročný server je schopen obsloužit mnoho požadavků denně. Jádro jazyka PHP má velmi dobře optimalizovanou dobu odezvy potřebnou pro webové aplikace.

- **Zabudované knihovny**

PHP obsahuje mnoho zabudovaných funkcí, které mimo jiné zahrnují: matematické funkce všeho druhu, funkce pro práci s textovými řetězci, pro podporu regulárních výrazů, funkce pro práci s datem a časem, pro práci se soubory a adresáři, síťové funkce, funkce pro odesílání elektronické pošty a práci s elektronickou poštou pomocí protokolu IMAP, funkce pro čtení informací ze síťových zařízení pomocí protokolu SNMP, funkce pro zpracování dat z webových formulářů, podporu cookies, funkce pro spouštění externích programů a zpracování jejich výstupu, podporu komprese a dekomprese dat pomocí metody používané programem gzip a mnoho dalších.

- **Výstup PHP**

Výstupem PHP skriptů nemusí být vždy nutně HTML kód, ale při použití specializovaných knihoven i obrázky, PDF soubory či XLS exporty.

- **Integrace databází**

PHP má schopnost připojovat se bez jakýchkoli prostředníků k mnoha databázovým systémům – Oracle, Sybase, FireBird, Informix, Solid, Adabas D, Hyperwave, mSQL, MySQL a PostgreSQL. Při využití obecného rozhraní ODBC (*Open Database Connectivity Standard*) je možné se připojit ke kterékoliv databázi, která poskytuje ovladač ODBC, včetně produktů společnosti Microsoft a mnoha jiných.

- **Nízké náklady, zdrojový kód**

PHP je *Open source*. *Open source* se netýká pouze volně šiřitelného softwaru, ale také otevřenosti zdrojového kódu. Každý má přístup ke zdrojovému kódu a na rozdíl od komerčních produktů si v jazyce může cokoliv upravit nebo přidat. Volný přístup ke zdrojovým kódům způsobuje, že autoři jsou nuceni k dodržování standardů. Pokud nejsou dodržovány, deviace jsou označeny jako chyby a pokud je neopraví autor, opraví je někdo jiný. Nízké náklady PHP favorizují před konkurenčním programovacím jazykem ASP, který je komerčním produktem firmy Microsoft. Nejnovější verze PHP je přístupná na domovské stránce <http://www.php.net>.

- **Objektová orientace**

PHP je částečné i plně objektově orientovaný jazyk, záleží pouze na použité verzi. Starší verze PHP 4 má méně knihoven a rozšíření a není plně objektová. Novější verze PHP 5 zavádí již plně objektově orientovaný jazyk.

- **Výuka PHP**

Syntaxe jazyka je založena na jiných programovacích jazycích, především na C a Perlu. Pokud programátor jeden z nich ovládá (nebo jazyk na C postavený – C++, Java), bude prakticky schopen ihned programovat v PHP.

- **Přenositelnost**

Důležitou vlastností PHP je jeho nezávislost na platformě. Pracuje pod operačními systémy UNIX (Linux, FreeBSD, Solaris atd.) i různými verzemi Microsoft Windows.

V případě Portálu VUT je použit jazyk PHP 4, který běží na serveru Apache s operačním systémem RedHat Linux. Apache HTTP Server je softwarový webový server s otevřeným kódem pro Linux, BSD, Microsoft Windows a další platformy. Je vyvíjen skupinou vývojářů, cílem projektu je poskytnout bezpečný, výkonný a rozšiřitelný server, který poskytuje HTTP služby v souladu se současným standardem. Název vznikl z anglického slovního spojení „*A patchy server*“ (záplatovaný server). Jako indiánský symbol je ve znaku ptačí pero.

Vývoj Apache začal v roce 1993 v NCSA (*National Center for Supercomputing Applications*) na Illinoiské univerzitě. Původní jméno projektu bylo NCSA HTTPd. V roce 1994 vývojářský tým opustil hlavní programátor Rob McCool. Tím došlo ke zpomalení vývoje a v roce 1998 k úplnému zastavení. NCSA HTTPd mezitím už používali správci webových serverů a dodávali k němu vlastní úpravy – patche. Významnou úlohu sehráli Brian Behlendorf a Cliff Skolnick, kteří založili e-mailovou konferenci a začali sběr úprav koordinovat. První veřejná verze byla vydána v dubnu 1995. Poté došlo ke kompletnímu přepsání kódu a založení *Apache Group*, která je dnes základem vývojářského týmu.

Apache je nejpopulárnějším webovým serverem na Internetu od dubna 1996. Podle výsledků měření společnost Netcraft v listopadu 2005 více než 70% webových stránek používalo právě Apache. Důvody pro používání serveru Apache:

- výkonnost, spolehlivost
- ohromné konfigurační možnosti
- modul PHP umožňuje dynamické změny obsahu v závislosti na návštěvníkovi
- je zdarma, vyvíjí ho mezinárodní komunita vývojářů

5.1.3 CSS

CSS (*Cascading Style Sheets*) neboli kaskádové styly jsou kolekcí metod pro grafickou úpravu webových stránek. První návrh normy byl zveřejněn v roce 1994, v roce 1996 byla vydána specifikace CSS 1 a v roce 1998 verze CSS 2. Historie CSS je vlastně historií jejich podpory ve webových prohlížečích, každý prohlížeč totiž styly interpretuje odlišně, některé prohlížeče neznají některé vlastnosti atd.

CSS se využívá k formátování obsahu HTML a ve srovnání s formátováním pomocí atributů v HTML formátovací schopnosti rozšiřuje. Styly umožňují přesně určit, jak bude který element vypadat. Na rozdíl od atributů se stylem může definovat jednotný vzhled elementu pro celý dokument a to jediným zápisem pro příslušný element. Stejně tak se pomocí stylu může určit odlišné formátování pro třeba jen jediný výskyt určitého elementu. Tím se zkrátí délka vytvořeného kódu a tento kód se stane mnohem přehlednější. Při vhodně zvoleném formátování můžeme definovat jednotný vzhled jednoho elementu a tím provádět změnu grafického vzhledu celé aplikace jedním zápisem, což je značně efektivní.

Do stránek HTML lze kaskádové styly připojit více způsoby. Pokud chceme předepsat styl pouze jednomu konkrétnímu prvku, můžeme napsat definici stylu přímo do jeho značky, což je tzv. přímý styl. Pro celý dokument můžeme použít tabulku stylů zapsanou v hlavičce stránky. Nebo můžeme styly nadefinovat v externím souboru s koncovkou .css a načítat je do stránky přes příkaz link v hlavičce HTML dokumentu. Poslední uvedená možnost je nejobvyklejší a je použita i na Portálu VUT.

5.1.4 SQL

SQL je standardizovaný dotazovací jazyk, který se používá pro práci s daty v relačních databázích. SQL je zkratka anglických slov *Structured Query Language*, což znamená strukturovaný dotazovací jazyk. První prototypová implementace tohoto jazyka vznikla roku 1974 v laboratořích společnosti IBM a byla použita ve firemním systému s názvem *R*. Probíhal zde výzkum relačních databází a bylo nutné vytvořit sadu příkazů pro jejich ovládání. Jazyk se nejdříve jmenoval SEQUEL (*Structured English Query Language*). K vývoji jazyka se přidaly další firmy a koncem 70. let se jazyk přejmenoval na SQL.

Jelikož začal vzrůstat význam relačních databází, bylo nutné jazyk standardizovat. První standard byl přijat v roce 1986 a byl pojmenován jako SQL-86. V dalších letech se ukázalo, že SQL-86 obsahuje některé nedostatky a naopak v něm nejsou obsaženy některé důležité prvky týkající se hlavně integrity databáze. V roce 1992 byl proto přijat nový standard SQL-92. Zatím nejnovějším standardem je SQL3 (SQL-99), který reaguje na potřeby nejmodernějších databází s objektovými prvky.

Cílem SQL je poskytnout vývojářům standardní metodu přístupu k datům uloženým v databázovém systému, která by byla nezávislá na dalších použitých vývojových nástrojích. Na jednu stranu se jedná o jazyk poměrně mocný a současně jednoduchý (stačí si do angličtiny přeložit, co vlastně chceme, a samotný příkaz bude vypadat velmi podobně). SQL patří do kategorie tzv. deklarativních programovacích jazyků, což znamená, že kód jazyka nepíšeme v samostatném programu, ale vkládáme jej do jiného programovacího jazyka, který je již procedurální.

Jazyk SQL také není možné chápat striktně jako dotazovací, skládá se z několika částí. Některé části jsou určeny pro administrátory a návrháře databázových systémů, jiné pro koncové uživatele a programátory. První část je označována jako *DDL (Data Definition Language)* a jedná se o jazyk pro vytvoření vlastní databázové struktury. Druhou částí je *DML (Data Manipulation Language)*, pomocí které s daty manipulujeme – můžeme data vkládat, upravovat, mazat a provádět nad nimi dotazy.

5.2 Knihovny

5.2.1 ADODB

Vzhledem k tomu, že přístup z jazyka PHP do databází není standardizován a je implementován pro každou databázi rozdílně, je pro zjednodušení přístupu ke zdrojům dat na Portálu VUT použita knihovna ADOdb (*Database Abstraction Library for PHP and Python*). ADOdb je databázová abstraktní knihovna, která patří mezi jedny z nejrychlejších knihoven přístupných pro jazyky PHP a Python. Tvoří abstraktní vrstvu mezi databázovými stroji a uživatelskými skripty, které pracují pomocí jazyka SQL se zdroji v databázi.

Velkou výhodou této knihovny je absolutní přenositelnost, zjednodušující vlastnost při napojování do databáze, přehlednost a hlavně poskytnutí možnosti změny typu databáze, např. z MySQL na Oracle, přepsáním jediného parametru při připojování k databázi. ADOdb tedy umožňuje připojení k několika druhům databází bez nutnosti změny kódu ve skriptech. Jedinou podmínkou je nutnost stejné datové struktury. ADOdb umožňuje vývoj aplikace na jiné databázové platformě mimo prostor centrální databáze. Na knihovnu se vztahuje licence GPL.

5.2.2 JpGraph

JpGraph je objektově orientovaná knihovna pro vytváření a zobrazování grafů. Je to přídavná knihovna do PHP, která je zároveň v jazyce PHP napsaná. Pomocí JpGraph je možné vytvářet velké množství různorodých grafů, od základních sloupcových, bodových, křivkových, koláčových až po polární, tří-dimenzionální, mapové nebo obrázkové.

Pro svou činnost vyžaduje grafickou knihovnu GD2, která je součástí kompilace webového serveru, což je Apache. Samotná knihovna se skládá z několika základních vykreslovacích skriptů. Do zdrojového kódu PHP je potom nutné vložit část knihovny, nadefinovat datové zdroje pro přísun dat a popisků ke grafu a o zbytek se postará knihovna. Výsledné grafy jsou obrázky formátu PNG, GIF nebo JPG. Knihovna je přístupná ve dvou verzích, Standard a Pro. První verze je zdarma, ale vztahuje se na ni licence QPL. Verze Pro je určena ke komerčnímu využití.

Kromě JpGraphu existuje mnoho dalších nástrojů pro generování grafů. Uvedu některé z nich:

- **FusionCharts** – komponenta pro generování animovaných interaktivních grafů ve formátu flash. Je použitelná v libovolném skriptovacím jazyku (.NET, ASP, JSP, PHP apod.), statickém HTML i prezentacích PowerPoint, podporuje XML datové rozhraní. Existuje v plné verzi, která nabízí až padesát různých 2D and 3D grafů. Free verze nabízí jen dvacet možností grafů a vizuálních efektů.

- **Graiddle** – produkt, který může být použit ke generování několika typů grafů. Obsahuje několik specializovaných tříd pro generování histogramů, koláčových a křivkových diagramů. Lze nakonfigurovat několik grafických aspektů, jako je barva, popisek, souřadnice. Diagramy jsou generovány pomocí funkcí GD knihovny jako barevné obrázky, které mohou být zobrazeny v aktuálním skriptu nebo uloženy do souboru ve formátu PNG. Na produkt se vztahuje licence GPL.
- **AnyChart Flash Chart Component** – řešení, založené na technologii Macromedia Flash, pro tvorbu animovaných, kompaktních, interaktivních a poutavých grafů pro web a desktopové aplikace. Umožňuje nastavení řady možností (průhlednost, barvu, URL odkazy, titulky apod.), prostřednictvím XML souboru, bez nutnosti modifikace Flash zdroje. Je kompatibilní se všemi skriptovacími jazyky (ASP, PHP, Perl, JSP atd.) a databázemi. Cena personální licence je 38,5 EUR.
- **PHP Advanced Graph and Chart Collection** – grafická komponenta poskytující řešení pro začlenění dynamických a interaktivních grafů a diagramů do PHP aplikací a webových stránek. Umožňuje generování mnoha různých grafů. Existuje v několika verzích, které se liší cenou a počtem poskytovaných typů grafů. Trial verze je zdarma.
- **Libchart** – je jednoduchá knihovna pro tvorbu grafů pomocí PHP. Grafy jsou uloženy jako obrázek ve formátu PNG. Knihovna umožňuje tvorbu sloupcových (vertikálních nebo horizontálních) a kruhových grafů. Pro její využívání je nutné mít na serveru PHP s podporou GD a FreeType. Knihovna Libchart je relativně jednoduchá, ale umožňuje tvorbu pouze tří druhů grafů. Na knihovnu se vztahuje licence LGL (*GNU Lesser General Public License*).

Knihovna JpGraph je zaimplementovaná jako součást portálu VUT. Je již použita při generování některých grafů, a proto jsem ji využila i já v rámci této práce.

5.3 Software

5.3.1 Subversion

Subversion je software, který slouží pro správu verzí zdrojových souborů aplikací. Je to systém, který udržuje kontrolu nad skupinou projektů, uchovává veškeré provedené změny a v případě potřeby umožňuje návrat k dřívější verzi. Je to systém, umožňující práci více vývojářů na jedné aplikaci nebo zdrojovém kódu.

Subversion patří do kategorie *version control* nástrojů, je přímým konkurentem známého systému CVS (*Concurrent Versions System*). Je šířen pod licencí *Apache Software Foundation*,

kteřá umožňuje jeho bezplatné komerční použití, k dispozici jsou i zdrojové kódy. Systém je postaven na dvouvrstvé architektuře klient – server, lze ho provozovat na platformách Windows, Linux, OS2 a dalších. Velkou výhodou Subversion je, že se dá ovládat pomocí příkazové řádky či shellu nebo pomocí grafického rozhraní SVN tortoise. Toto rozhraní umožňuje přehledně a efektivně pracovat s verzemi souborů, práce s ním je velmi pohodlná a intuitivní. Na oddělení vývoje v Centru výpočetních a informačních služeb je použita tato kombinace: server na platformě Linux a klient SVN tortoise na platformě Windows.

Základní částí je centrální úložiště repository obsahující soubory daného projektu. Uchovávají se zde jednotlivé verze souborů, které jsou uživatelům dostupné. Další částí je pracovní kopie (*working copy*), což je kopie nějakého stavu repository určená k lokálním úpravám. Tím je zajištěno, že uživatel nikdy nepracuje přímo se zdrojovými kódy, ale pouze s kopiemi.

Základní operace:

- **Checkout** – provádí se pouze jednou a slouží ke stažení pracovní kopie na lokální disk z repository.
- **Commit** – po modifikaci zdrojového kódu je tímto příkazem možné uložit změnu do repository. Tato změna je označena jako revize, při každém nahrání modifikovaných souborů je číslo revize inkrementováno. Je vhodné u commitu uvést čeho se změna týká, co bylo upraveno. Informace o tomto kroku mohou být odeslány vývojářům systému emailem, což pomáhá udržovat přehled o provedených změnách.
- **Update** – příkaz provádí synchronizaci změn v databázi s naší lokální kopií. Samozřejmě se neprovede přepsání našich změn těmi, které jsou v databázi, ale klient na tento rozdílový stav upozorní a je pouze věcí uživatele, zda změny potvrdí nebo si nechá svou verzi.

Pro zjištění rozdílu pracovní kopie a repository slouží nástroj diff, který přehledně zobrazuje veškeré změny a je také použit pro přenos informací o nových verzích. Hlavní smysl uchovávání historie jednotlivých změn je možnost vrátit se o jeden nebo více kroků zpět k verzi, která fungovala. Snadné porovnání dvou rozdílných verzí nás zavede na ono místo, kde se nachází možný zdroj problémů.

5.4 Hardware

Celý systém portálu se skládá z několika serverů, které se starají o základní funkce a navzájem spolupracují. Servery jsou mezi sebou propojeny vnitřním síťovým okruhem s rychlostí 1Gbit.

5.4.1 Databázový server

Centrální databáze Oracle je provozována na hlavním databázovém serveru OGRE. Je to čtyřprocesorový stroj na platformě Intel XEON s 24GB operační paměti. Operačním systémem je distribuce linuxu RedHat.

5.4.2 Aplikační cluster serverů

Dostupnost webových stránek a informačního systému vysoké školy je velmi důležitá, proto systém musí mít vysokou odolnost proti výpadku. Systém by měl být odolný jak proti běžným výpadkům hardwarového charakteru, tak i proti softwarovému selhání jednotlivých komponent a funkčních bloků systému. Také by měl být navržen efektivně, aby dokázal zvládnout výkonové špičky, které se objevují zejména při konkurenční registraci předmětů, zkoušek, termínů apod. Nedostupnost informačních systémů pro studenty a vyučující může způsobit velké komplikace. Řešení náročných výpočetních úloh jedním strojem by trvalo velmi dlouho a konečného výsledku bychom se také nemuseli vůbec dočkat. Proto je na VUT použito řešení pomocí clusteru serverů.

Cluster je seskupení volně vázaných počítačů, které spolu úzce spolupracují a navenek mohou pracovat jako jeden počítač. Clustery lze nasazovat v několika typech úloh, podle toho se také liší požadavky na hardwarovou a softwarovou konfiguraci celého clusteru. Jsou obvykle nasazovány pro zvýšení výpočetní rychlosti nebo spolehlivosti s větší efektivitou než by mohl poskytnout jediný počítač. Jejich cena je většinou nižší než cena jediného počítače o srovnatelné rychlosti nebo spolehlivosti.

Podle typu úloh dělíme clustery na tyto kategorie:

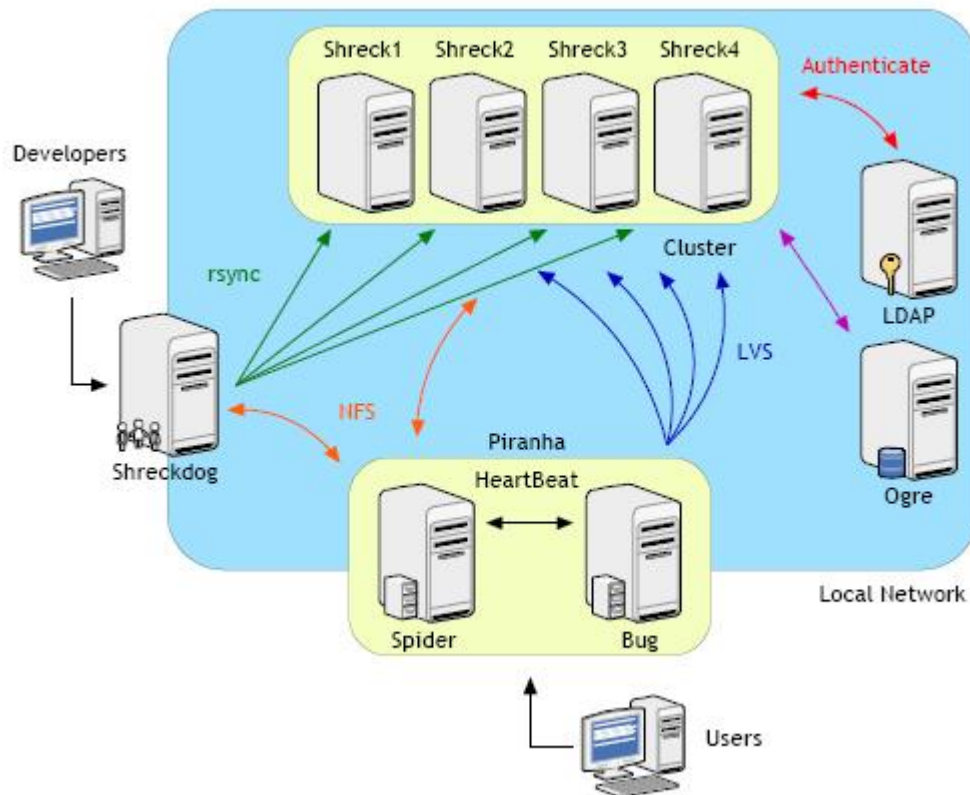
- **Výpočetní clustery** (*high-performance computing*) – mohou sloužit k paralelním výpočtům složitých početních úloh, např. faktorizace na prvočísla, analýza velkého množství statistických dat, simulace vývoje počasí apod. Používají se většinou specializované víceprocesorové stroje propojené vysokorychlostní linkou nebo obyčejné počítače třídy PC. Úlohy určené pro urychlení za pomoci výpočetního clusteru, musí být speciálně navrženy. Pro linuxové výpočetní clustery se používá souhrnné označení *Beowulf* podle stejnojmenného projektu NASA, který realizoval první výpočetní cluster a některé podpůrné části v jádře systému. Nejde o označení konkrétního softwaru.
- **Clustery pro vysokou dostupnost aplikací** (*high availability*) – zabezpečují vysokou dostupnost aplikací a služeb. Jsou založeny na principu synchronizace dat jednotlivých uzlů s uzlem, který je v daném okamžiku primární. Při výpadku je provoz celé sítě převeden na druhý server a to včetně identity předešlého uzlu, takže klient nic nepozná a systém se tváří jako stále dostupný.

- **Clustery rozdělující zátěž** (*load balancing*) – spojení více počítačů do jednoho celku za účelem rozdělení zátěže. Prosazují se tam, kde je třeba zajistit co nejrychlejší odezvy na velké množství požadavků přicházejících na server. Cluster se tváří jako síťový server, na jednotlivé uzly clusteru jsou přesměrovány požadavky od různých klientů. Tento cluster je vhodný pro webové servery, ale také může být použit pro vyrovnávání zátěže sítě. Existuje několik metod, jak rozdělovat požadavky na jednotlivé servery:
 - § **Round Robin** – požadavky jsou rovnoměrně rozdělovány na reálné servery.
 - § **Weighted Round Robin** – jednotlivým serverům je navíc přiřazena váha dle výkonnosti nebo očekávané zátěže. Pokud má server vyšší váhu, dostane nový požadavek prioritněji a je tím více využíván než ostatní servery s nižší váhou. Servery se stejnými váhami dostávají požadavky rovnoměrně rozloženy.
 - § **Least Connection** – požadavky jsou přiřazeny serverům, které zpracovávají nejméně aktivních požadavků.
 - § **Weighted Least Connection** – rozděluje požadavky stejně jako *Least Connection* algoritmus podle stavu aktivních požadavků na serverech, ale v úvahu bere navíc přednastavené váhy pro jednotlivé servery. Tento algoritmus je používán implicitně.
 - § **Locality-Based Least-Connection** – přiděluje požadavky ze stejné IP adresy na jeden server, pokud je dostupný a nevytížený. Pokud je nedostupný, přidělí požadavek méně vytíženému serveru a uchová informaci pro budoucí přesměrování na tento server.
 - § **Locality-Based Least-Connection with Replication** – pracuje na stejném principu jako předchozí algoritmus a zároveň odstraňuje jeho podstatnou nevýhodu. Ta se může projevit v případě delšího provozu, kdy IP adresy zahrnují stále jeden server a ostatní jsou nevyužity. Jednou za čas se tedy provede replikace, kdy se nejvytíženějšímu serveru smaže seznam IP adres. Ty se znovu přerozdělí mezi ostatní servery v momentě, když z nich přijde nový požadavek.
 - § **Destination Hashing** – přiděluje požadavky serverům podle staticky definované tabulky cílových IP adres.
 - § **Source Hashing** – přiděluje požadavky serverům podle staticky definované tabulky zdrojových IP adres klientů.
 - § **Shortest Expected Delay** – vybírá server, u kterého je předpokládaná nejnižší doba zpracování. Doba zpracování je vypočtena podle vzorce $(C_i + 1) / U_i$, kde C_i je počet požadavků, které server zpracovává nebo má zpracovat a U_i je statisticky definovaná váha serveru.

§ **Never Queue** – předá požadavek prvnímu serveru, který nezpracovává žádný požadavek. Pokud jsou všechny obsazeny, postupuje podle algoritmu *Shortest Expected Delay*.

Na VUT je využito clusteru serverů pro rozdělení zátěže. Tento cluster funguje tak, že existuje více webových serverů, které jsou před uživatelem schovány a navenek se jeví jako jeden server. Uvnitř clusteru se efektivně rozdělují požadavky na zpracování mezi jednotlivé aplikační servery. Pro běh informačního systému je použito řešení *RedHat Cluster System*, kde podpora pro provoz clusteru je přímo zakomponována v jádře distribuce RedHat Linux. Tato služba se jmenuje *IPVSADM* a zajišťuje přerozdělování a přesměrování všech TCP spojení a UDP datagramů na fyzické servery podle desíti integrovaných algoritmů, které jsou distribuovány s *Linux Virtual Server (LVS)*. *Linux virtual server* představuje systém pro vyrovnávání zátěže a zajištění vysoké dostupnosti.

IPVSADM ovládá program *pulse* běžící v režimu daemon, který je součástí *RedHat Clusteru*. Tento program kontroluje všechny servery zapojené do clusteru a zjišťuje, jestli jsou dostupné a fungují v rámci standardních parametrů. Pokud nalezne nefunkční server, je z clusteru vyloučen. K připojení serveru dojde, až když je opět schopen provozu.



Obrázek 5.1 Schéma clusteru serverů VUT

Aplikační cluster na VUT používá pro přidělování požadavků algoritmus *wlc (Weighted Lest-Connection)*. Ten rozděluje požadavky na servery, které mají nejméně aktivních požadavků ke zpracování. Také bere v úvahu přednastavenou váhu pro jednotlivé servery. Jeho výhodou je uchování existující session pro momentálně připojeného uživatele, tím odpadá i riziko vytvoření nové session do databáze při každé změně stránky.

Konfigurace IPVSADM umožňuje nastavit viditelnost serverů přímo nebo skrytí za překladačem adres NAT (*Network Address Translation*). To je použito i na VUT, kde celý cluster běží na vnitřní síti a navenek se projevuje pouze jednou IP adresou s doménou pro adresaci webových aplikací. Server viditelný zvenku se nazývá Piranha. Piranha se tváří jako jeden server, ve skutečnosti se ale jedná o dva servery Spider a Bug propojené v režimu Master – Slave. Každý z těchto strojů má svoji privátní IP adresu a zároveň disponují další společnou IP adresou. Společnou adresu používá server, který má v té chvíli status Master. Součástí každého systému je datový sklad obsahující dokumenty a jiný datový materiál, který může mít větší objem. Aby se data zbytečně nereplikovala na každý server clusteru, jsou uložena fyzicky právě na serverech Spider a Bug. Mezi těmito servery se datový sklad fyzicky synchronizuje každou hodinu. Aby byl datový sklad online přístupný na aplikačních serverech, aniž by tam byl fyzicky nakopírován, je vytvořena vazba mezi Piranhou a aplikačními servery pomocí technologie NFS (*Network File System*).

Dalším důležitým serverem clusteru je Shreckdog. Ten je použit jako vývojový a testovací server. Běží na něm lokálně testovací verze tří systémů Portál, Studis a Teacher. Na tomto serveru lokálně pracují a vyvíjejí vývojáři systémů a po odladění nové verze aplikace se pomocí Subversion a rsync přenesou aktualizované zdrojové kódy na cluster a tím nasadí do ostrého provozu.

6 Databáze VUT

6.1 Oracle 10g

Centrální databáze VUT obsahuje celou agendu vysoké školy, což je obrovské množství dat. Rozsáhlou skupinu tvoří například informace týkající se studia, osob se vztahem k VUT, vědy a výzkumu a další. Celkový objem dat má přitom vzrůstající tendenci. Jako primární zdroj dat používá VUT několik instancí databáze Oracle.

Databáze Oracle je dlouhodobě známa díky svému špičkovému výkonu, spolehlivosti a zároveň i bohaté funkcionalitě. Umožňuje ukládat data a získávat přístup k nim způsobem odpovídajícím relačnímu modelu. V relační databázi jsou data ukládána do tabulek a veškeré operace pracují s těmito tabulkami. Tabulka je tedy základním stavebním kamenem pro budování celé databáze. Každá tabulka v databázi má jedinečné jméno identifikující její obsah. Tvoří ji jeden nebo více sloupců, které jsou uspořádány ve specifickém pořadí směrem zleva doprava. Sloupce odpovídají jednotlivým atributům entity, v jedné tabulce musí mít každý sloupec jiný název kvůli identifikaci. V rámci celé tabulky mají všechny hodnoty dat v jednom sloupci stejný datový typ. Data jsou ukládána jako řádky tabulky.

Ve sloupcích tabulky je možné vytvářet integritní omezení. Po jeho vytvoření musí všechny řádky tabulky splňovat podmínky uvedené v definici omezení. V relační databázi má každá tabulka sloupec (nebo kombinaci sloupců), který se označuje jako primární a jehož hodnota jednoznačně identifikuje řádek v tabulce. Žádné dva řádky tabulky nemohou mít stejný primární klíč, primární klíč obsahuje jedinečnou hodnotu pro každý řádek. Omezení cizí klíč určuje typ vztahu mezi tabulkami. Cizí klíč je sloupec v tabulce, jehož hodnota odpovídá hodnotě primárního klíče v jiné tabulce. Pokud je tabulka spojena s více tabulkami, může obsahovat více cizích klíčů. Cizí klíče tvoří základní část relačního modelu, jelikož vytváří relaci mezi tabulkami. Omezení v databázi umožňují zajistit referenční integritu dat. Referenční integrita zaručuje, že budou všechny odkazy v rámci databáze platné a všechna omezení budou splněna.

Kromě uložení dat v relačním formátu podporuje Oracle i objektově orientované struktury jako jsou abstraktní datové typy a metody. Objekty mohou být provázány s jinými objekty nebo mohou další objekty obsahovat. Při použití relačních i objektově orientovaných struktur jsou data interními strukturami databáze logicky mapována na soubory. Různé typy dat jsou ukládány samostatně, tato logická rozdělení se nazývají tabulkové prostory. Tabulkový prostor představuje logické rozdělení databáze, všechny databáze obsahují alespoň jeden (*SYSTEM*). Seskupením uživatelů nebo aplikací do dalších tabulkových prostorů je možné

usnadnit údržbu a zvýšit výkon databáze. Jednotlivé tabulkové prostory se skládají z jednoho nebo více souborů na disku, tzv. datových souborů, které poskytují fyzické úložiště pro data databáze. Každý datový soubor může patřit pouze k jednomu tabulkovému prostoru.

V polovině roku 2006 byla na VUT databáze Oracle 9i nahrazena vyšší verzí Oracle 10g. Oracle 10g je moderní multiplatformní databázový systém s velice pokročilými možnostmi zpracování dat, vysokým výkonem, dostupností, bezpečností a snadnou škálovatelností. Jsou zde začleněny sofistikované mechanismy automatizované správy databáze, díky kterým může databáze automaticky monitorovat svůj stav a adaptovat se. Podle provedených studií se správa databáze zjednodušila až o 50% oproti verzi 9i. Snadná správa databáze a s tím související nízké náklady na provoz se tak stávají další silnou stránkou databáze Oracle.

Tento databázový systém byl navržen pro robustní databázové sklady a je schopen zvládat a obsluhovat požadavky jednotlivých systémů, které jsou na VUT v provozu. Uživatelé se přes tyto systémy připojují do centrální databáze a pracují s informacemi uloženými v datovém skladu.

6.2 Struktura databáze VUT

Databáze na VUT se skládá ze čtyř instancí. První je označena CDBX, obsahuje ostrá data a využívají ji informační systémy běžící v ostrém provozu. Druhá je CISD a slouží k vyvíjení a testování nových aplikací a komponent v rámci vývojového oddělení. Instance je vždy jednou za týden aktualizována daty z ostré databáze. Jsou v ní tedy většinou aktuální údaje. Třetí instancí je CISC, slouží k dlouhodobému vyvíjení systému a narozdíl od CISD v ní nejsou úplně aktuální data. Data v určitém schématu se vždy na požádání aktualizují z ostré databáze. Čtvrtou instancí je CISB, která také slouží k vývoji. Data zde ale nejsou shodná s CDBX, především osobní údaje jsou promíchány a změněny. Důvodem je to, že tuto instanci využívají například studenti pro vývoj svých projektů pro VUT a mohlo by dojít ke zneužití těchto dat.

Data jsou podle logických celků členěna do databázových schémat. Schéma je sada objektů vlastněná uživatelským účtem, lze je použít k oddělení určité sady databázových objektů od ostatních. Uživatelský účet má důležité vazby s objekty databáze, protože uživatelé jsou vlastníky databázových objektů. Příklad existujících schémat:

- **ST01** – rozsáhlé schéma uchovávající informace týkající se studia. Jedná se o přehled studií, změny studia, studijní programy a obory, individuální plán studenta, elektronický index s výsledky studia jednotlivých studentů, také data týkající se rozvrhů, závěrečných prací atd. Toto schéma je z hlediska studijních statistik nejdůležitější, jelikož jsou v něm všechny potřebné informace pro statistiky.
- **BRUTISADM** – další rozsáhlé schéma, ve kterém původně byla data z informačního systému Brutis. Postupně se ale rozšiřovalo a dnes obsahuje například osobní informace,

kontakty, informace o organizačních jednotkách VUT, data z oblasti vědy a výzkumu, data týkající se karet VUT.

- **DOKUMENTACE** – schéma, ve kterém jsou informace týkající se dokumentace centrální databáze.
- **LOGGER** – v tomto schématu se uchovávají informace o změnách nebo mazání záznamů v důležitých tabulkách. Kromě změněných sloupců, je zde i čas změny a kdo změnu provedl, takže je pak možné snadno dohledat, kdo změnil data v tabulce a kdy.

Kromě těchto schémat, je v databázi ještě mnoho dalších, které vznikly pro potřeby externích systémů a fakult, které jejich prostřednictvím nahrávají data z jejich vnitřních informačních systémů pro potřeby centrální evidence a zpracování.

Pro udržení konzistence dat v takto velkém datovém skladu je třeba zavádět určité typy omezení a kontrol a to jak na straně aplikací a jednotlivých systémů, tak i na straně samotné databáze. Ty se aplikují za pomoci integritních omezení a databázových triggerů. Triggery jsou procedury, které se spouštějí v případě, že dojde ke specifikované události databáze. Mohou být použity ke zvětšení referenční integrity, vynucení dodatečného zabezpečení nebo vylepšení dostupných možností auditování. Existují dva typy triggerů. První jsou triggery příkazů, které se spustí jednou při každém odpovídajícím příkazu. Druhé jsou triggery řádků spouštějící se jednou pro každý řádek tabulky ovlivněný odpovídajícím příkazem.

Kromě klasického schématu tabulek, klasických a materializovaných pohledů se používají i procedury a funkce, které jsou vytvořeny jazykem PL/SQL, a které jsou doménou Oracle. Procedura je blok příkazů uložených v systémovém katalogu. Pomocí ní lze uchovávat často používanou aplikační logiku v rámci databáze. Procedury nevracejí volajícímu programu žádné hodnoty. Funkce jsou podobné jako procedury, ale na rozdíl od nich mohou volajícímu programu vrátit hodnoty. Je možné spouštět funkce, které poskytuje přímo platforma Oracle nebo vytvářet vlastní a volat je pomocí příkazů SQL. Procedury a funkce jsou vynikajícím prostředkem k získávání informací z databáze, jelikož hlavní dotazy jsou centralizované v databázi a klientské aplikace volají pouze rozhraní procedury. Procedury a funkce lze uspořádat do logických seskupení pomocí programových balíků. Ty jsou velice užitečné při administrativních úlohách, které jsou nutné k řízení procedur a funkcí. Různé prvky balíku lze definovat jako veřejné nebo soukromé. Veřejné prvky jsou přístupné uživateli balíku, soukromé jsou mu skryty. Soukromé prvky mohou zahrnovat procedury volané jinými procedurami v rámci balíku.

Pro přístup k datům slouží řada nástrojů, které jsou dodávány v rámci databázového řešení. Ty jsou primárně určeny pro administrátory a databázové operátory. Pro uživatele je třeba vytvořit rozhraní, přes které mohou data zobrazovat, přidávat, odebírat nebo měnit.

6.2.1 Konvence

Všechny tabulky v centrální databázi VUT mají některé společné znaky, při jejich vytváření se dodržují určité konvence. Například primární klíč tabulky má stejný název jako tabulka a je k němu přidána přípona `_id`. Cizí klíče mají stejné názvy jako ve své mateřské tabulce. Všechny názvy číselníků začínají předponou `c_`, názvy pohledů prefixem `v_`, materializované pohledy prefixem `mv_` apod.

Každá tabulka, kromě číselníků, obsahuje tři povinné sloupce:

1. **Upd_id** – slouží k identifikaci uživatele, který provedl záznam nebo změnu daného záznamu v tabulce.
2. **Upd_ts** – datum a čas vložení nebo modifikace příslušného záznamu v tabulce. Provádí databázová funkce `SYSDATE`.
3. **Status** – slouží k určení stavu, ve kterém se daný záznam nachází. Může nabývat těchto hodnot:
 - 0 – platná data, synchronizace z jiných systémů.
 - 1 – data logicky smazána, nelze provést fyzické vymazání, protože existují návaznosti na tato data.
 - -1 – data určená k fyzickému smazání po určité době.
 - 9 – platná data, nově vytvořený záznam.

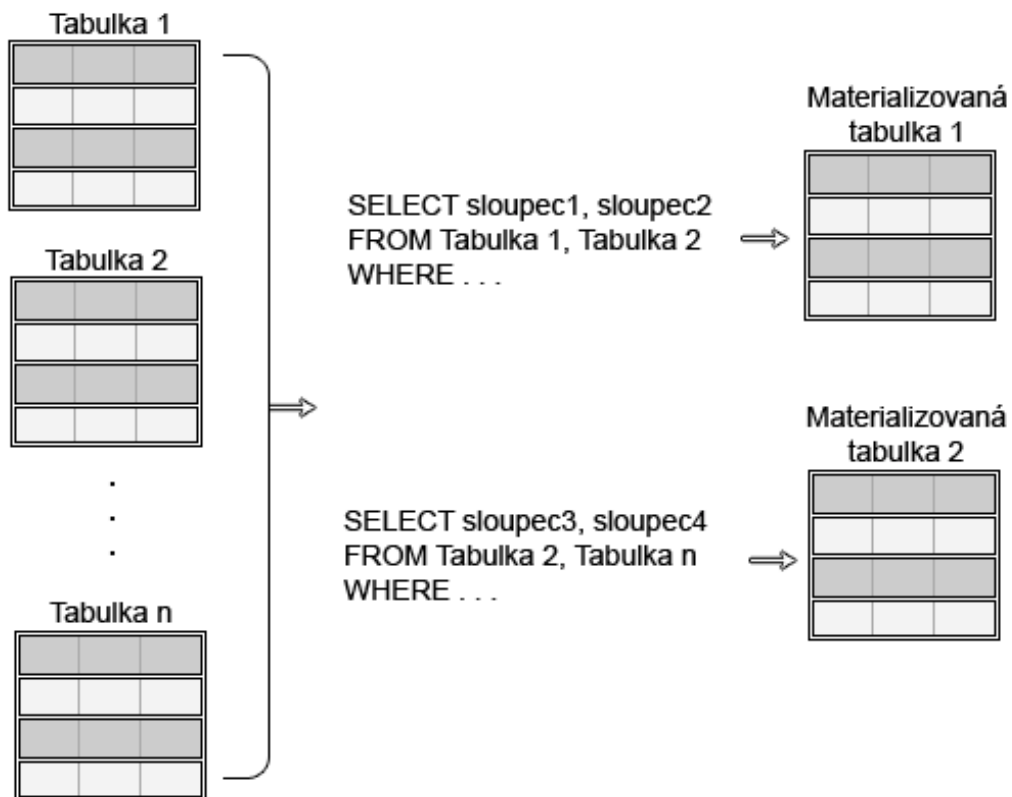
6.3 Nový datový model

Některé dotazy pro získávání statistických údajů se jevily jako neefektivní a zároveň neúměrně zatěžovaly databázový server. Proto bylo nutné vytvořit několik nových materializovaných tabulek, do kterých jsou předem uložena výsledná data pro zobrazení. Materializované tabulky jsou objekty obsahující odvozená data. Slouží k souhrnům a přípravným výpočtům dat, umožňují zvýšit rychlost dotazů zahrnujících agregace nebo spojení tabulek. Použitím nových tabulek se dotazy pro získání dat značně zjednodušily a zároveň se doba jejich provádění velmi urychlila.

Tabulky se vytváří pomocí SQL příkazů, které slouží k jejich naplnění daty. Aktualizaci dat zajišťuje skript napsaný v jazyce Perl, který běží na serveru Shreckdog. Ten pomocí SQL výrazu pro vytvoření tabulky získá aktuální data z databáze, provede konkatenci jednotlivých sloupců v rámci každého řádku a uloží je do asociativního pole jako indexy. Jako oddělovač při konkatenci je většinou použit znak nebo skupina znaků, které se ve zdrojových tabulkách nevyskytují.

Data v asociativním poli jsou poté porovnávána s daty získanými z materializované tabulky a data v tabulce jsou upravována podle aktuálních dat. Některé záznamy mohou být přidány, jiné mohou být smazány, modifikovány nebo mohou zůstat beze změny. Při modifikaci dochází k zamykání na úrovni řádků, zbytek tabulky je tedy přístupný pro získávání dat. Ve skriptu zajišťujícího aktualizaci, se nastavuje první čas provedení aktualizace a poté perioda, po které se bude aktualizace opakovat.

Asociativní pole (hash) je pole, jehož prvky nejsou indexovány pomocí posloupnosti celých čísel, ale pomocí klíčů. Klíčem může být číslo nebo textový řetězec. Každý klíč v poli je unikátní. Všechny páry klíč – hodnota nejsou oproti normálnímu poli nijak uspořádány, takže hashe nemají žádný první ani poslední prvek. Počet prvků v poli se může dynamicky měnit. Asociativní pole představují výkonný programátorský nástroj a jsou velmi užitečným rysem jazyka Perl. Pole jsou v Perlu implementována pomocí rozptýlených tabulek, prefixem pro označení proměnné typu hash je znak procento (%). Mechanismem hashování se dosahuje výborné rychlosti v přístupu k uloženým položkám, data tedy mohou být v hash tabulce díky výkonným algoritmům velice rychle nalezena.



Obrázek 6.1 Vytvoření materializovaných tabulek

Místo materializovaných tabulek by bylo možné použít i materializované pohledy. Aktualizace dat v pohledu je ale pomalejší a méně spolehlivá než výše uvedený použitý způsob řešení. U materializovaného pohledu existují čtyři typy obnovení:

- **Rychlé** – pomocí protokolů pohledu se sledují změny, ke kterým došlo od posledního obnovení. Na základě nastavených kritérií obnovení jsou do materializovaného pohledu pravidelně vkládány pouze změněné údaje. Protokol materializovaného pohledu je tabulka, která obsahuje záznamy úprav hlavní tabulky v materializovaném pohledu. Jsou v něm tedy uložena pouze dočasná data. Změněné záznamy jsou vloženy do protokolu, použity během obnovení a poté vymazány. Protokol je uložen ve stejné databázi a schématu jako hlavní tabulka a může být použit pouze u jednoduchých materializovaných pohledů. V případě, že několik pohledů používá stejnou tabulku, sdílejí také stejný protokol.
- **Úplné** – u tohoto obnovení jsou při každém spuštění nahrazena všechna data materializovaného pohledu. Čas potřebný k provedení úplného obnovení může být velmi dlouhý a během něj jsou data v pohledu uzamčena.
- **Vynucené** – při tomto obnovení se nejprve zjistí, jestli lze spustit rychlé obnovení. Pokud to není možné, je provedeno úplné obnovení.
- **Žádné** – při zvolení této možnosti data nebudou obnovována.

Pomocí materializovaných pohledů nelze zajistit referenční identitu mezi dvěma relačními tabulkami, na kterých jsou tyto pohledy založeny. Pokud jsou tabulky obnoveny v rozdílnou dobu nebo pokud u hlavních tabulek během obnovení proběhnou transakce, je pravděpodobné, že materializované pohledy nebudou odrážet referenční integritu těchto tabulek. Pro tento problém existuje několik řešení. Některé z nich jsou velmi neefektivní, jako například provedení obnovení ručně po uzamčení hlavních tabulek nebo po ukončení procesů v databázi a načasování obnovení na dobu, kdy se hlavní tabulky nepoužívají. Další možností je použití spojení souvisejících pohledů do skupin pro obnovení. Skupina pro obnovení slouží ke koordinaci plánů obnovení jejich členů a zajišťuje i referenční integritu.

Standardní mechanismus obnovy materializovaných pohledů byl z dlouhodobého hlediska nespolehlivý, pomalý a po dobu materializace byl pohled nedostupný. Z tohoto důvodu byly vyvinuty skripty, které všechny výše uvedené problémy řeší.

6.3.1 Popis nově vytvořených tabulek

Tabulka st01.mv_statistika_hod_szz

Obsahuje data získaná z tabulek `studium`, `ukonceni_studia_neuspesne`, `ukonceni_studia` a číselníku `c_ukonceni_predmetu`. Slouží pro statistiku celkového hodnocení státní závěrečné zkoušky. Dotaz pro vytvoření tabulky je uveden v příloze 3.7.

Seznam atributů:

- **fakulta_id** – fakulta, pro kterou se záznam vztahuje
- **rok** – rok státní závěrečné zkoušky
- **znamka_znak** – známka dle hodnocení ECTS („A“, „B“, „C“, „D“, „E“, „F“)
- **pocet** – počet studentů, kteří při celkovém hodnocení SZZ získali uvedenou známku
- **termin** – značí termín SZZ (buď první nebo druhý termín)

Tabulka st01.mv_statistika_hod_ust_szz

Materializovaná tabulka je určena pro statistiku hodnocení ústní části závěrečné zkoušky. Je naplněna daty z tabulek `studium`, `ukonceni_studia_neuspesne`, `ukonceni_studia` a číselníku `c_ukonceni_predmetu`. Seznam atributů:

- **fakulta_id** – fakulta, pro kterou se záznam vztahuje
- **rok** – rok státní závěrečné zkoušky
- **znamka_znak** – známka podle hodnocení ECTS („A“, „B“, „C“, „D“, „E“, „F“)
- **pocet** – počet studentů, kteří při hodnocení ústní části SZZ získali uvedenou známku
- **termin** – značí termín SZZ (buď první nebo druhý termín)

Tabulka st01.mv_statistika_hod_dp

Z této tabulky se získávají data pro histogram hodnocení diplomové práce. Je vytvořena nad tabulkami `studium`, `ukonceni_studia`, `ukonceni_studia_neuspesne` a číselníku `c_ukonceni_predmetu`. Seznam atributů:

- **fakulta_id** – fakulta, pro kterou se záznam vztahuje
- **rok** – rok státní závěrečné zkoušky
- **znamka_znak** – známka dle hodnocení ECTS („A“, „B“, „C“, „D“, „E“, „F“)
- **pocet** – počet studentů, kteří při hodnocení diplomové práce získali uvedenou známku
- **termin** – značí termín SZZ (buď první nebo druhý termín)

Tabulka st01.mv_statistika_osp_stud

Tabulka slouží pro získání dat pro statistiku úspěšnosti studia na jednotlivých tabulkách. Její data jsou získána z tabulek `zmena_studia`, `studium`, `rocnik`, `obor`, `program`

a číselníku `c_typ_studia`. Dotaz pro vytvoření tabulky je uveden v příloze 3.8. Seznam atributů:

- **fakulta_id** – fakulta, které se záznam týká
- **skolni_rok** – rok, pro který se záznam vztahuje
- **typ_studia_id** – typ studia studentů
- **zapsano** – počet studentů, kteří byli v uvedeném roce zapsaní na dané fakultě do uvedeného typu studia
- **ukonceno** – počet studentů, kteří z celkového počtu zapsaných, úspěšně ukončili studium (je jedno ve kterém roce)
- **zapsano_muži** – počet mužů, kteří byli v uvedeném roce zapsaní na dané fakultě do uvedeného typu studia
- **ukonceno_muži** – počet mužů, kteří z celkového počtu zapsaných mužů, úspěšně ukončili studium (v kterémkoliv roce)
- **zapsano_zeny** – počet žen, které byly v uvedeném roce zapsány na dané fakultě do uvedeného typu studia
- **ukonceno_zeny** – počet žen, které z celkového počtu zapsaných žen, úspěšně ukončily studium (v kterémkoliv roce)

Tabulka st01.mv_statistika_pocet_stud

Je postavena nad tabulkami `program`, `studium`, `zmena_studia`, `stupen`, `rocnik` a `obor` ze schématu `st01` a tabulkou `person` ze schématu `brutisadm`. Je využívána jako podklad pro statistiku o počtu a rozdělení studentů na fakultách. Dotaz pro vytvoření materializované tabulky je uveden v příloze 3.9. Seznam atributů:

- **fakulta_id** – značí fakultu pro uvedený záznam
- **zkratka** – zkratka fakulty
- **rok_platnosti** – rok platnosti záznamu
- **aktivni** – počet všech aktivních studentů na celé fakultě v daném roce
- **muži** – počet všech mužů studujících v daném roce na fakultě
- **zeny** – počet všech žen studujících na fakultě v daném roce
- **cesi** – počet Čechů, kteří měli aktivní studium na fakultě v uvedeném roce
- **slovaci** – počet všech Slováků, kteří studovali v uvedeném roce na fakultě
- **cizinci** – počet cizinců studujících na fakultě v uvedeném roce (jedná se o studenty s národností jinou než je česká nebo slovenská)

Tabulka st01. mv_statistika_hodn_predmetu

Materializovaná tabulka je postavena nad tabulkou `el_index` ze schématu `st01` a data z ní jsou využívána pro celkovou statistiku bodového hodnocení předmětu. Dotaz pro vytvoření tabulky je uveden v příloze 3.10. Seznam atributů:

- **aktualni_predmet_id** – id předmětu, kterého se záznam týká
- **popis** – popis hodnocení, může nabývat hodnot „0-9“, „10-19“, „20-29“, „30-39“, „40-49“, „50-59“, „60-69“, „70-79“, „80-89“, „90-100“
- **pocet** – počet studentů, kteří v daném předmětu získali toto hodnocení

7 Návrh řešení

7.1 Konvence používané při implementaci portálu

Při vytváření aplikací centrálního informačního systému jsou používány určité konvence, které budu dodržovat i v implementaci studijních statistik. V této podkapitole se zmíním o některých z nich.

7.1.1 Adresářová struktura

Centrální informační systém má tři části a zdrojové kódy každé části jsou uloženy ve vlastním adresáři – Portal, Teacher a Studis. Vznikem tří nezávislých instancí systémů se však hodně věcí duplikovalo a vznikaly nekonzistence ve verzích. Proto byl nad aplikacemi zaveden adresář `_base`, který obsahuje knihovny univerzálních funkcí používaných ve všech systémech, konfigurační soubory pro jednotlivé verze systémů a sklad zdrojových souborů externích knihoven. Vzhledem k tomu, že některé moduly bylo potřeba využívat ve všech třech systémech (např. vyhledávání osob), vznikly tzv. globální moduly, které jsou také uloženy v `_base`. Adresář zároveň obsahuje citlivé informace jako loginy a hesla k databázím, a proto byl k jeho obsahu zakázán přístup v konfiguraci webservru Apache. Tím se posílila bezpečnost a zmenšila možnost napadení přes webové rozhraní.

V jednotlivých systémech existují tyto adresáře:

- **app** – obsahuje soubory jednotlivých aplikací systémů, které obsahují načítání dat z databáze, editaci, mazání dat apod. Samotné zobrazení je provedeno v samostatném souboru uloženém v adresáři `templates`.
- **css** – soubory s definovanými kaskádovými styly
- **images** – obrázky použité v aplikacích
- **js** – knihovny JavaScriptu
- **libs** – lokální knihovny pro aplikaci
- **temp** – adresář slouží jako dočasné úložiště souborů z webu. Soubory starší než dvacet čtyři hodin jsou průběžně promazávány.
- **templates** – šablony s definicí vzhledu dat získaných pomocí souborů v adresáři `app`. Výstupem šablon je většinou HTML kód, ale někdy je potřeba také generovat obsah do formátu PDF. Výhodou rozdělení kódu do souboru aplikace umístěném v `app`

a šablony umístěné v templates je ten, že máme pouze jeden soubor samotné aplikace a z něj můžeme pro zobrazení dat volat různé soubory šablon.

7.1.2 Připojení k databázi

Pro připojení k databázi se na portálu využívá knihovna ADOdb. V ní však chybí některé detaily, jako například použití vázaných proměnných nebo logování dotazů. Proto byla vytvořena třída `oracle.class`, která zajišťuje rozhraní pro volání dotazů. Nejčastěji je ve třídě používaná funkce `run_sql`, která umožňuje používat při volání sql dotazu vázané proměnné. Vzhledem k tomu, že tato funkce je na portálu využívána téměř ve všech aplikacích, budu ji používat i při implementaci studijních statistik. Ukázka použití funkce:

```
$rs = run_sql($sql_dotaz, $binded_array);  
if ($rs === false) { return 0; }
```

Kde `$sql_dotaz` reprezentuje dotaz, který má být proveden a `$binded_array` je asociativní pole vázaných proměnných s jejich hodnotami, které budou doplněny do dotazu před jeho vykonáním. Podmínka testuje výsledek dotazu.

7.1.3 Autentizace uživatele

Autentizací se rozumí prokázání, že je někdo tím, za koho se vydává. Do technik ověřování totožnosti patří hesla, elektronické podpisy, biometrická měření a detekce hardwaru. Na Portálu VUT se používá autentizace pomocí uživatelského jména a hesla. Hesla jsou jednoduše implementovatelná, snadno použitelná a nepotřebují žádná vstupní zařízení. Základní princip je jednoduchý. Heslo zná jen uživatel a systém. Dokud nikdo jiný nebude znát toto heslo a ani ho nikdo nebude schopen uhodnout, je tato metoda bezpečná. Pro ukládání uživatelských jmen a hesel je použita centrální databáze, kde jsou uloženy zakódované údaje pomocí jednosměrného hešovacího algoritmu. Po prokázání uživateli totožnosti vzhledem k databázi je provedena autentizace pomocí protokolu LDAP a to vůči autentizačnímu serveru. Zde se získají práva uživatele, která se využívají především v publikačním systému portálu. Zabezpečení přenosu dat během přihlašovací procedury je zabezpečeno protokolem SSL.

Po prokázání uživateli totožnosti, je nutné si nějak zapamatovat, že už je identifikován, a že nemusí svoji totožnost znovu prokazovat. Protokol HTTP je bezstavový a není v něm nikde zabudované, jak uchovávat stav mezi dvěma transakcemi. Když uživatel požaduje nějakou stránku a pak jinou, neposkytuje HTTP žádnou možnost, jak by se dalo zjistit, že oba požadavky přišly od jedné osoby. Řešením tohoto problému je tedy ke sledování informací o uživateli použít proměnných `$_SESSION`, poté co prokázal totožnost přihlašovacím mechanismem. `$_SESSION` jsou předávány v průběhu navázané relace a jejich obsah mění vzhled a funkčnost jednotlivých skriptů ve webových aplikacích.

7.1.4 Jazykové verze

Většina prezentací a aplikací, které jsou součástí Portálu, vyžaduje vícejazyčný výstup informací. Znamená to, že všechny vypisované textové řetězce, kterými systém v těchto aplikacích komunikuje s uživatelem, musí být jazykově nastavitelné. Nejsou tedy pevně zakotveny v programovém kódu, ale odděleně ve skladu textů v různých jazykových mutacích. Skladem textů je soubor `func_tr_data.php` společný pro všechny části Portálu. Ten obsahuje dvourozměrné asociativní pole jazykových výrazů, kde prvním rozměrem je identifikace výrazu nebo řetězce a druhým jednotlivé jazykové mutace. Pole je kdykoliv rozšiřitelné pro další jazyky, stačí pouze přidat další mutaci výrazu. Příklad části pole:

```
$tr_mem['but_title'] = array (
    'Vysoké Učení Technické v Brně',
    'Brno University of Technology');
```

Pro vložení správné jazykové verze textu se používá funkce `get_lang`, která podle nastavené proměnné rozliší typ jazyka a vloží do stránky správnou jazykovou verzi textu. Rozeznávání aktuálně zapnutého jazyka je realizováno dvěma způsoby. V systémech, do kterých mají přístup pouze přihlášení uživatelé, se hodnota s indexem aktuálního jazyka uchovává v `$_SESSION` uživatele. Tím je navenek neviditelná a pomocí přepínacích prvků snadno ovladatelná. Toto řešení ale nelze použít v prezentační části portálu, protože zde uživatel nemusí být autentizován a není pro něj vytvořena `$_SESSION`. Index jazyka se zde tedy uchovává v globálních polích GET nebo POST a každý odkaz obsahuje parametr `lang`, který určuje zvolený jazyk. Předávání a doplňování parametrů do všech odkazů a formulářů je realizováno funkcí `output_add_rewrite_var('lang', $GLOBALS['lang_id'])`.

7.2 Model případů použití

Model případů použití znázorňuje možnosti, které budou nabízeny v systému jednotlivým uživatelům. Model případů použití je znázorněn v příloze 2. Hlavními aktéry v něm jsou student, vyučující, přihlášený a nepřihlášený uživatel.

7.2.1 Student

Tento uživatel bude mít po přihlášení do systému Studis přístup k histogramu vážených průměrů v ročníku a programu, který studuje. Uvidí svůj vážený průměr a umístění v ročníku. U předmětů zapsaných v elektronickém indexu bude mít možnost dostat se na modul se statistikou k předmětu, kde bude statistika celkového hodnocení předmětu, histogram hodnocení pro vyučující, kteří zadávali u předmětu hodnocení a hodnocení pro jednotlivé termíny pokusů

udělení zkoušky, tedy napokolikáté studenti zkoušku úspěšně složili. Také uvidí histogram hodnocení u jednotlivých termínů zkoušky vypsanych vyučujícími.

7.2.2 Vyučující

Po přihlášení do aplikace Teacher uvidí vyučující u předmětů, u kterých mohou zadávat hodnocení, celkovou statistiku hodnocení a také se dostanou na modul se statistikou předmětu, který byl popsán v minulé podkapitole (7.2.1). U předmětů, které jsou ukončeny zápočtem, se zobrazí informace kolik studentů zápočet dostalo a kolik ne.

7.2.3 Uživatel

Na webu VUT se nepřihlášený uživatel dostane na statistiku přijímacího řízení, kde se dozví, kolik studentů se v daný rok hlásilo na fakultu, kolik jich bylo přijato po přijímacím řízení, kolik bez přijímacího řízení, kolik studentů bylo zapsáno atd.

7.2.4 Přihlášený uživatel

Přihlášený uživatel uvidí v menu Portálu odkaz na modul se statistikami, které se budou týkat především fakult. Např. statistika úspěšnosti ukončení studia na jednotlivých fakultách, počet a rozdělení studentů na fakultách atd.

7.3 ER diagram

V příloze 1 je ER diagram, který zobrazuje část schématu st01. Jsou v něm především tabulky, které jsou použity při implementaci studijních statistik.

- **EL_INDEX** – tabulka obsahuje informace o elektronickém indexu. Cizí klíče vytváří vazbu na studium studenta, aktuální předmět a tabulku evidence. Jsou zde uloženy například informace o zápočtu (zda je nebo není udělen, datum udělení nebo neudělení zápočtu, kdo zápočet udělil), o výsledku zkoušky (bodové hodnocení, známka, datum zápisu výsledku), zda student předmět absolvoval nebo ne atd.
- **PREDMET** – data o předmětech vyučovaných na fakultách. Jsou zde informace o fakultě, která předmět vyučuje, název a zkratka předmětu, od kterého a do kterého roku je předmět vyučován, jazyk výuky apod.
- **AKTUALNI_PREDMET** – zatímco v tabulce PREDMET jsou obecné informace o předmětu, AKTUALNI_PREDMET obsahuje aktuální informace o předmětu pro daný rok. Například ústav, kde je předmět vyučován, semestr a rok, minimální, maximální a aktuální počet registrovaných studentů, kredity, povinnost, typ ukončení předmětu atd.

- **STUDIUM** – zde jsou záznamy o všech studiích studentů. Je zde fakulta, ke které studium patří, per_id studenta, zda je studium aktivní nebo není, pořadí, začátek a rok studia, kontrola studia atd.
- **ZMENA_STUDIA** – v této tabulce jsou všechny informace o změnách studentova studia. Je zde vazba na studium, ročník a stav studia. Dále je zde školní rok, pro který změna platí, datum změny, studijní skupina studenta apod.
- **ROCNIK** – obsahuje informace o ročnících – číslo ročníku, maximální a minimální počet kreditů v ročníku, vazbu na obor, ke kterému patří.
- **OBOR** – tabulka s jednotlivými obory obsahující program, ke kterému obor patří, název oboru, název oboru anglicky, kód oboru, cíl oboru, znalosti, praxe, maximální a minimální počet kreditů, garant oboru apod.
- **PROGRAM** – tabulka studijních programů, obsahuje například název programu, název anglicky, zkratka, kód, délka studia, přidělovaný titul, cíl programu, znalosti, průběh studia, praxe.
- **UKONCENI_STUDIA** – obsahuje informace o ukončení studia – především informace o diplomové práci a státní závěrečné zkoušce – téma, datum, hodnocení, členové komise, zvláštní ocenění apod.
- **UKONCENI_STUDIA_NEUSPESNE** – tabulka má stejnou strukturu jako UKONCENI_STUDIA, ale obsahuje informace o neúspěšných ukončeních studia. Pokud tedy student neuspěje při SZZ, vytvoří se záznam v této tabulce.
- **TERMIN** – uchovává data o vypsaných termínech zkoušek – název termínu, minimální, maximální a aktuální počet registrovaných studentů, začátek a konec zkoušky, místo konání, datum odkdy a dokdy je možné se na termín registrovat apod.
- **ZADANI** – obsahuje údaje o vypsaných zadáních – název zadání, vedoucí zadání, maximální počet řešitelů, literatura atd.
- **ZKOUSKA_PROJEKT** – vazební tabulka mezi termínem, zadáním a aktuálním předmětem.
- **EVIDENCE** – slouží pro uložení hodnocení termínů zkoušek nebo zadání. Jsou zde položky jako maximální počet bodů, počet získaných bodů, známka, datum klasifikace, kdo výsledek uložil, pořadové číslo pokusu atd.
- **REGISTRACE_TERMINU, REGISTRACE_ZADANI** – zde se ukládají registrace studentů na termín (zadání).
- **C_UKONCENI_PREDMETU** – číselník obsahující pro každou fakultu a rok záznamy o možnostech a typech ukončení studia. Obsahuje například položky hodnocení známkou, známka znak, slovní hodnocení, slovní hodnocení anglicky, minimální počet získaných bodů, maximální počet získaných bodů atd.

- **C_TYP_STUDIA** – číselník, který obsahuje záznamy pro všechny typy studia (bakalářský, doktorský, magisterský, magisterský navazující a celoživotní vzdělávání).
- **C_FORMA_STUDIA** – číselník obsahující záznamy o všech formách studia, jako například prezenční studium, kombinované studium, distanční studium atd.

8 Popis vlastní implementace

Samotná aplikace je rozdělena do tří částí a každá z nich se nachází v jednom ze tří informačních systémů (Studis, Teacher, Portál). Každý z těchto systémů má vlastní základní grafické rozhraní, které je předdefinováno za pomoci kaskádových stylů a které by se nemělo měnit, např. hlavní menu aplikace, hlavní titulek modulu atd. Většinou jsou grafické prvky v systémech podobné, jen se liší barvami. Grafickou podobu vnitřního vzhledu jednotlivých modulů je možné upravit za pomoci definice vlastních kaskádových stylů. Ty ale nesmí být v rozporu s již vytvořenou šablonou. Také je důležité zvolení barevné palety, které by mělo brát ohled na vzhled celého systému. Snažila jsem se tedy veškeré statistiky barevně sladit s aplikací, ve které se nacházejí. Svoje vlastní styly jsem přidala do již vytvořených souborů s nadefinovanými styly.

Statistiky zpracovávají velké množství dat a proto je zde riziko, že budou příliš pomalé. U implementace každé statistiky jsem tedy provedla analýzu několika možností a vybrala to nejlepší řešení. U některých bylo nutné provést optimalizaci dotazu nebo vytvořit novou tabulku s přepočítanými daty, u jiných bylo nutné ukládat vygenerované grafy jako obrázky a poté je jen zobrazovat, aby se nemusela znovu načítat data a vykreslovat graf.

Pro samotné generování grafů jsem použila objektově orientovanou knihovnu JpGraph, která umožňuje vytvářet grafy sloupcové, křivkové, bodové, koláčové, ale také mapové, obrázkové nebo tří-dimenzionální. Knihovna už je zaimplementovaná jako součást portálu VUT, a proto vždy stačí jen načtení příslušných souborů.

8.1 Realizace v aplikaci Studis

8.1.1 Statistika hodnocení předmětu

Statistika hodnocení předmětu by se měla zobrazovat jak v informačním systému pro vyučující tak i pro studenty. Aby nedocházelo k nežádoucím duplikacím zdrojového kódu, vytvořila jsem statistiku jako globální modul, na který je možné se odkazovat z obou aplikací. Vzhled modulu je závislý na aplikaci, ve které je spuštěn, jsou vždy použity styly příslušné aplikace. Modul se jmenuje `gm_statistika_predmet.php` a je stejně jako ostatní globální moduly umístěn v adresáři `_base`. Soubor se šablonou je `gm_statistika_predmet_templ.php`. Samotné vygenerování grafu je implementováno v souboru `graph_sloupcovy.php`, ve kterém jsou podle předaných parametrů získána odpovídající data k zobrazení a samotné vykreslení grafu pomocí knihovny už probíhá vždy stejně. Pokud je tedy potřeba vygenerovat

nějaký sloupcový graf, použije se tento soubor, do kterého se pouze přidá část kódu pro získání dat a samotný kód pro vygenerování již není nutné znovu přepisovat.

Modul statistika hodnocení předmětu obsahuje tři části a to celkovou statistiku hodnocení předmětu, statistiku hodnocení předmětu pro jednotlivé pokusy o úspěšné složení zkoušky a pro jednotlivé vyučující. V každé z nich je nejdříve zobrazen formulář pro výběr roku, pomocí kterého je možné dostat se na statistiku předmětu i v jiných letech. Nabízeny jsou jen roky, kdy byl předmět vyučován.

Student se na modul dostane přes položku v menu Elektronický index, kde kliknutím na název předmětu přejde na detail předmětu. Zde je umístěn odkaz Celková statistika hodnocení předmětu, který vede na aplikaci se statistikami týkajícími se předmětu.

8.1.1.1 Celková statistika hodnocení předmětu

Výsledkem této statistiky jsou dva histogramy celkového hodnocení předmětu. První z nich je založen na hodnocení dle ECTS a druhý na bodovém hodnocení. Dle ECTS vyučující hodnotí studenta pomocí „A“, „B“, „C“, „D“, „E“, „F“. Při bodovém hodnocení se udělují studentovi body, minimum je 0 a maximum 100 bodů. Hodnocení jednotlivých předmětů je uloženo v tabulce `st01.el_index`, která obsahuje záznamy pro všechny předměty zapsané jednotlivými studenty.

Nejprve se budu věnovat statistice hodnocení dle ECTS. Pro získání dat jsem nejdříve vyzkoušela rychlost dotazu (viz. příloha 3.1), který pro všechny typy hodnocení předmětu počítá pomocí `count()`, kolik jich je v tabulce `st01.el_index` u příslušného předmětu. Typy hodnocení se získávají z číselníku `st01.c_ukonceni_predmetu`. Dotaz tedy vrací všechny typy hodnocení a k nim počty studentů. Pokud příslušnou známkou nebyl hodnocen žádný student, vrací typ hodnocení a u něj nulu. Dotaz trvá v průměru 0,61 s.

Další dotaz (příloha 3.2) využívá k počítání prvků metodu `group by`. Nevýhodou tohoto dotazu je, že vrací počty jen pro ty známky, které se v elektronickém indexu objevují. Pokud tedy například žádný student není v předmětu ohodnocen známkou „A“, ve výsledku tato známka není zahrnuta. Dalším dotazem se tedy musí získat šablona pro hodnocení a podle této šablony vypisovat výsledná data. Výhodou dotazu je jeho rychlost – 0,031 s. Pokud bude dotaz spouštěn často, bude uložen v Cache paměti a doba trvání se ještě zkrátí.

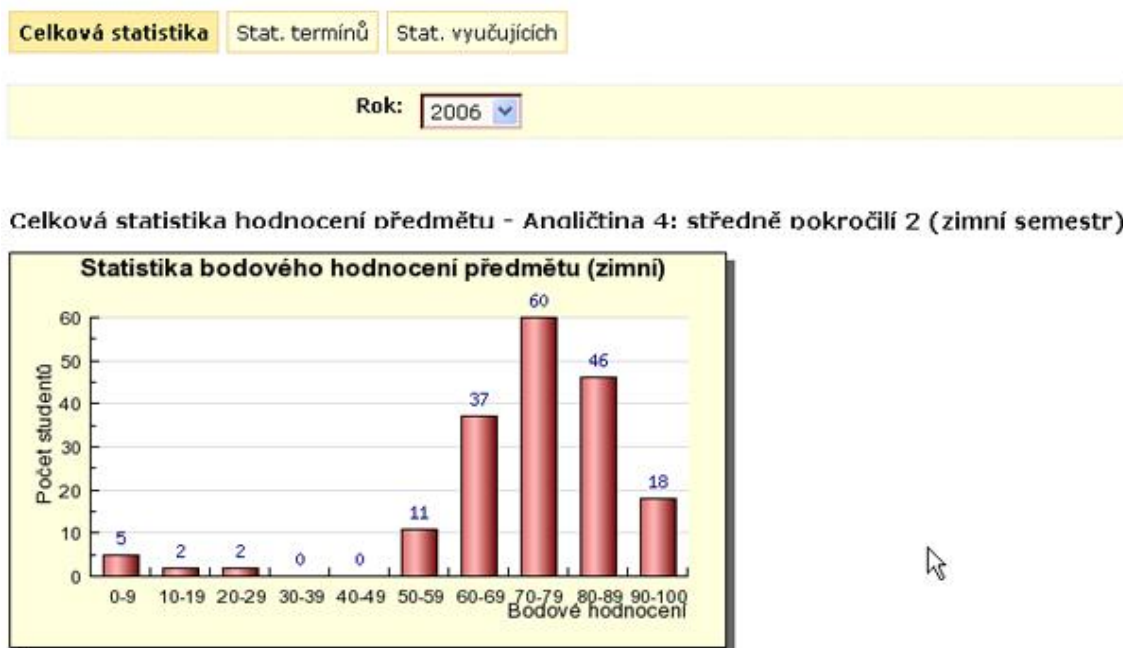
Další možností je předpřipravit si data do nové tabulky. Vzhledem k tomu, že rychlost získávání dat z této tabulky by nebyla o moc větší než při použití druhého dotazu, zvolila jsem použití dotazu uvedeného v příloze 3.2.

Dotaz pro získání dat pro bodové hodnocení byl náročnější jak na čas, tak i na vytížení databáze. Zvolila jsem tedy možnost vytvoření nové materializované tabulky s názvem `st01.mv_statistika_hodn_predmetu`. Dotaz vytvoření je uveden v příloze 3.10 a samotná tabulka byla popsána v kapitole 6.3.1.

Dalším problémem je zobrazení samotného histogramu. Nabízí se zde několik možností, a to zobrazení přímo vygenerovaného grafu nebo dočasné uložení vygenerovaného grafu jako obrázku a poté zobrazování jen uloženého obrázku. Po provedené analýze jsem zjistila, že samotné vygenerování grafu trvá v průměru 0,098 s. Zobrazování uloženého obrázku by zde bylo neefektivní vzhledem k časté změně dat a také vzhledem ke kontrole oprávnění pro přístup k němu. Proto jsem tedy u obou histogramů zvolila zobrazení přímo vygenerovaného grafu.

Grafický výstup statistiky je zobrazen na obrázku 8.1. V horní části se zobrazují tlačítka pro přepínání mezi jednotlivými částmi modulu. Poté je nabízen formulář pro výběr roku zobrazované statistiky a následuje histogram celkového hodnocení předmětu dle ECTS doplněný o informaci, kolik studentů předmět absolvovalo a kolik ne. Pokud jsou k dispozici data pro bodové hodnocení, zobrazí se histogram bodového hodnocení.

Statistika hodnocení předmětu



Obrázek 8.1 Celková statistika hodnocení předmětu

8.1.1.2 Statistika hodnocení předmětu pro jednotlivé termíny

Cílem této statistiky je zobrazit histogramy hodnocení podle ECTS pro jednotlivé pokusy o úspěšné složení zkoušky. Poskytne nám informace typu, kolik studentů bylo úspěšných při prvním termínu, kolik při druhém, kolik studentů z těch, kteří nebyli úspěšných při prvním termínu, se dostavilo na druhý atd. Nemyslí se tím tedy jednotlivé termíny zkoušek vypsané vyučujícími, kterých může být mnoho, ale spíše pokusy studentů o složení zkoušky, které by

měly být podle směrnice obvykle tři. Hodnocení termínů je uloženo v tabulce st01.evidence, která je pomocí cizího klíče propojena s tabulkou st01.el_index.

K získání dat jsem nejprve využila dotaz uvedený v příloze 3.3, který vezme všechny typy hodnocení a pro ně počítá počet studentů s tímto hodnocením pomocí count(). Výsledkem dotazu jsou všechny typy hodnocení a k nim počty studentů. Doba provádění je průměrně 0,58 s, což je velmi pomalé.

Druhý dotaz (příloha 3.4) počítá prvky pomocí count() a group by a vrací počty jen u těch známek, které se v hodnocení objevují. Druhým dotazem se tedy získá šablona hodnocení pro předmět a podle ní se vypisují data. Dotaz trvá v průměru 0,029 s, a proto není nutné vytvářet nad daty materializovaný pohled, který by získání dat výrazně neurychlil.

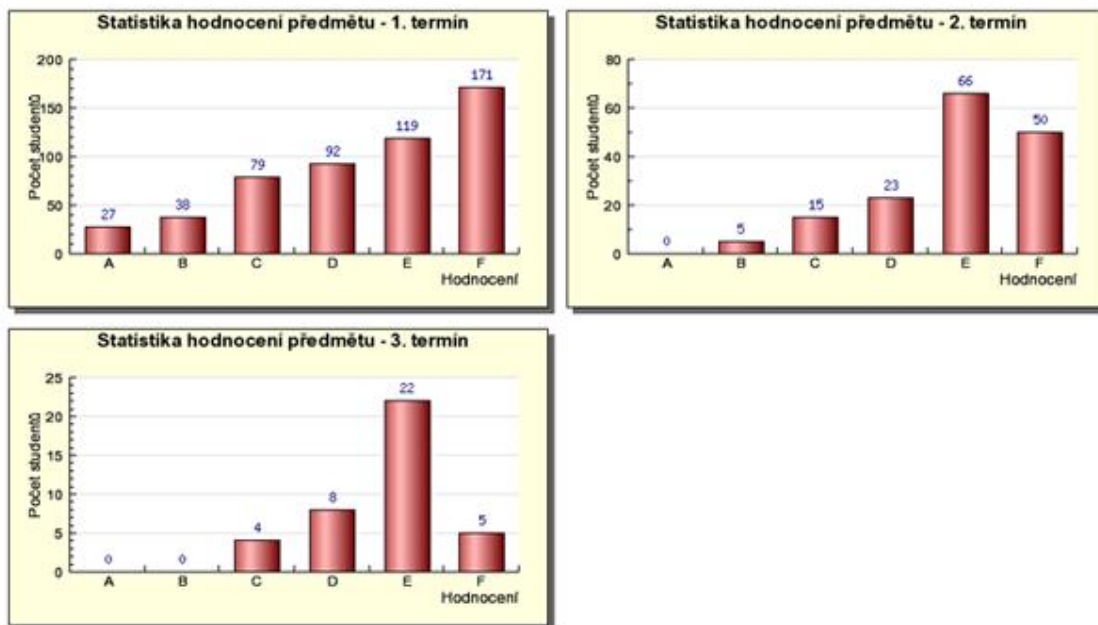
Vygenerování grafu ze získaných dat je opět provedeno pomocí knihovny JpGraph. Vygenerování trvá v průměru 0,088 s, a proto stejně jako v minulém případě zobrazují přímo vygenerované histogramy.

Statistika hodnocení předmětu

Celková statistika **Stat. termínů** Stat. vyučujících

Rok: 2006

Statistika hodnocení předmětu pro jednotlivé termíny - Matematika III (zimní semestr)



Obrázek 8.2 Statistika hodnocení předmětu pro jednotlivé termíny

Grafický vzhled této části je zobrazen na obrázku 8.2, kde je opět vidět v horní části menu pro přepínání mezi statistikami. Dále formulář pro výběr roku a nakonec samotné

histogramy hodnocení. U tohoto předmětu jsou grafy pro všechny tři pokusy o ukončení předmětu, pokud by všichni studenti udělali zkoušku napoprvé, byl by zde pouze jeden graf.

8.1.1.3 Statistika hodnocení předmětu pro jednotlivé vyučující

Hodnocení pro jednotlivé vyučující je poslední částí modulu týkajícího se statistik předmětu. Výsledkem jsou histogramy hodnocení dle ECTS pro všechny vyučující, kteří zapsali nějakou známku studentovi pro daný předmět. Údaje o tom, kdo předmět klasifikoval, jsou v tabulce `st01.el_index`.

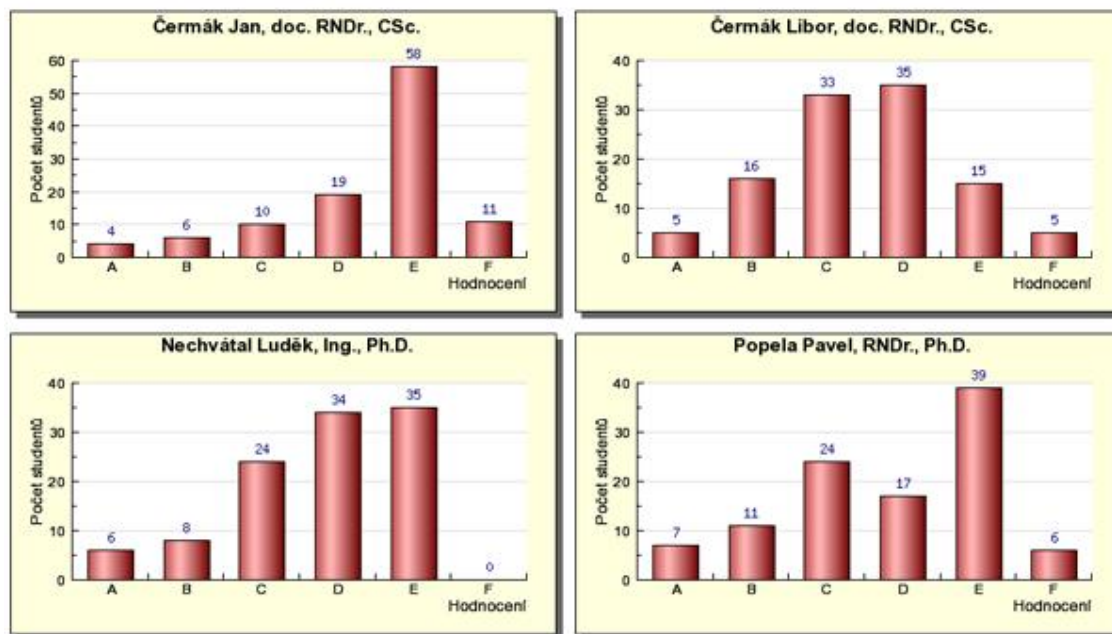
Pro načtení dat se nejprve použije dotaz vracející všechny vyučující, kteří předmět hodnotili. Pro každého z nich se poté získávají data sloužící jako podklad pro histogram. K jejich získání jsem opět porovnávala dva typy dotazů. První je uveden v příloze 3.5 a druhý v příloze 3.6, první trval průměrně 0,689 s a druhý 0,032. Druhý byl tedy výrazně rychlejší, a proto jsem ho použila při implementaci.

Statistika hodnocení předmětu

Celková statistika
Stat. terminů
Stat. vyučujících

Rok:

Statistika hodnocení předmětu pro jednotlivé vyučující - Matematika III (zimní semestr)



Obrázek 8.3 Statistika hodnocení předmětu pro jednotlivé vyučující

Samotné zobrazení statistik je řešeno stejně jako v předchozích kapitolách a to tak, že se zobrazuje přímo vygenerovaný graf. Generování se provádí v souboru `graph_sloupcovy.php` a trvá průměrně 0,107 s.

Na obrázku 8.3 je vidět výstup této aplikace. Nejprve je zobrazeno menu pro statistiky předmětů, poté výběrové pole pro rok a histogramy hodnocení pro všechny vyučující, kteří předmět hodnotili.

8.1.2 Statistika vážených průměrů


Statistika je začleněna do modulu Studium, na který se student dostane pomocí položky v menu se stejným názvem. Jsou zde informace o studii přihlášeného uživatele, student může přepínat mezi jednotlivými studii a podle toho se aktualizují všechny ostatní záložky. Je zde informace o studentově zapsané studijní skupině s odkazem na její rozvrh a seznam studentů a odkaz na potvrzení o studiu ve formátu pdf. Na konec modulu jsem přidala histogram vážených průměrů v ročníku a programu, informace o studentově váženém průměru a umístění podle průměru. Výsledný vzhled stránky je zobrazen na obrázku 8.4.

Studium

Výběr studia:

01.09.2003 | FSI | Strojní inženýrství | Inženýrská informatika a automatizace - aktivní

Zapsaná studijní skupina pro letní semestr akademického roku 2006/2007:

skupina	rozvrh	seznam
40/48		

Formuláře:

- [Potvrzení o studiu \(PDF\)](#)



Váš vážený průměr: **2.3.**

Umístění v ročníku a programu: **240. - 301. z celkového počtu 412.**

Obrázek 8.4 Statistika vážených průměrů

Získání a zpracování dat je přidáno do souboru `vyber_studia.php` a jejich výpis do šablony `tpl_vyber_studia.php`. Nejprve je zjištěno, kolik studentů má stejný a kolik lepší vážený průměr než student. Potom jsou získána data pro samotný histogram.

Po vygenerování grafu se obrázek uloží do adresáře `studis/temp`. Uložení je pouze dočasné, soubory v adresáři se každých dvacet čtyři hodin mažou. Při zobrazení statistiky se nejdříve kontroluje, jestli pro danou fakultu, ročník a program neexistuje v adresáři obrázek se stejným datem. Pokud ano, obrázek se jen zobrazí, pokud ne, musí se znovu načíst data a vygenerovat nový graf. Vzhledem k počtu studentů v ročnících, by tento postup měl statistiku výrazně urychlit. Každý den se pouze prvním studentovi ve stejném ročníku a programu budou statistiky generovat, ostatním už se bude jen zobrazovat předpřipravený obrázek.

8.1.3 Statistika termínů zkoušek

Statistika je začleněna do detailů jednotlivých termínů zkoušek. Přístup k ní je přes položku v menu Elektronický index, odkud se student dostane přes názvy předmětů na detaily jednotlivých předmětů. Tam je seznam vypsaných termínů zkoušek a kliknutím na číslo nebo popis termínu se dostane na detail termínu. Histogram hodnocení termínu je přidán na konec stránky s detailem termínu pod informace o předmětu a vybraném termínu. Zdrojové kódy jsou přidány do souborů `termin_detail.php` a `tpl_termin_detail.php`.

Pro získání dat jsem vyzkoušela několik dotazů. První získával data z tabulky `st01.evidence` a vrátil jen typy hodnocení, které se v tabulce vyskytovaly. Druhý spojoval tabulku `st01.evidence` a `st01.c_ukonceni_predmetu` a vrátil všechny typy hodnocení a k nim počty studentů. První dotaz byl výrazně rychlejší a proto jsem ho použila ve spojení s dotazem, který získal šablonu hodnocení, podle níž se data vypisovala.

Počet studentů registrovaných na termín není tak velký jako například počet studentů v celém ročníku. Proto se dá předpokládat, že se jednotlivé histogramy termínů nebudou zobrazovat tak často jako histogram váženého průměru. Výsledný graf se tedy nebude ukládat na server jako obrázek, jak je tomu v případě statistik váženého průměru, ale bude se zobrazovat přímo vygenerovaný graf. Ukládání obrázků pro každý termín by bylo také neefektivní vzhledem k velkému počtu různých termínů.

8.1.4 Parametrizace

Z průzkumu na fakultách vyplynulo, že ne všechny fakulty používající centrální informační systém budou chtít studentům statistiky zobrazovat. Proto je nutné na spuštění statistik ve Studisu použít parametrizaci umožňující spuštění aplikace nebo její části zvlášť pro každou fakultu a pro určité období. V databázi existují dvě tabulky, které se týkají parametrizace. První z nich je číselník `C_PARAMETR_IS`, do kterého jsem vložila nový parametr s názvem Studis – zobrazení statistik hodnocení předmětů, termínů atd. Druhou tabulkou je `PARAMETR_IS`, do které jsou ukládány parametry, které si nastaví samy fakulty.

Pro spuštění parametrů je v informačním systému Apollo vytvořen modul s názvem Parametrizace, harmonogram sloužící jako rozhraní právě k těmto tabulkám. Oprávněným osobám je zde zobrazen seznam parametrů z číselníku C_PARAMETR_IS a ony mohou u parametru nastavit například odkdy dokdy má být spuštěn, pro jakou fakultu platí atd. Uložením se vytvoří záznam v tabulce PARAMETR_IS. Před samotným zobrazením statistiky v aplikaci Studis se tedy vždy testují nastavené hodnoty v této tabulce patřící k nově vytvořenému parametru a k zobrazení dojde, jen pokud je pro danou fakultu spuštěn.

8.2 Realizace v aplikaci Teacher

Web pro učitele » Zkoušky » Zápis zápočtů

Výběr předmětu pro zápis zápočtů:
 2005/2006 [zimní] - Matematika I (1M)

Typ studia:
 jen aktivní studium

Zkratka: Zobrazit studenty podle termínu:
 --- všechny ---

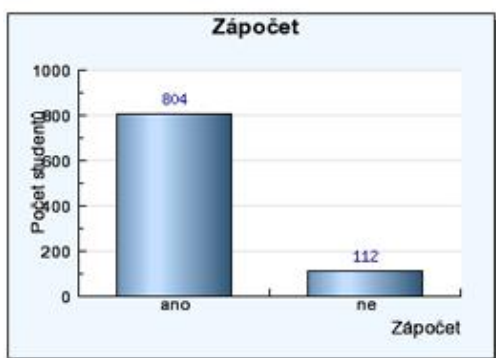
Celkem vyhledáno studentů: 598
 - z toho má zápočet: 597
 Datum udělení zápočtu: 29.04.2007

Výsledek hledání výrazu ""
 [Hledání trvalo 1.7123 s]

E-Mail	Student	Roč.	Skup.	Os. č.	Prog.	Uk.	Zap.
<input type="checkbox"/>	Abraham Martin	1./I.	2C/22	76384	B2341-3	zk,zá	Z 20.12.0
<input type="checkbox"/>	Adamec Martin	1./I.	2B/12	76630	B2341-3	zk,zá	Z 19.12.0
<input type="checkbox"/>	Adamec Petr	1./I.	1F/41	76548	B2341-3	zk,zá	Z 20.12.0
<input type="checkbox"/>	Adámek Pavel	1./I.	2E/95	77507	B3901-3	zk,zá	Z 02.01.0
<input type="checkbox"/>	Alfieri Tomáš	1./I.	2C/23	85978	B2341-3	zk,zá	Z 20.12.0
<input type="checkbox"/>	Altamirano Karo...	1./I.	2B/13	76418	B2341-3	zk,zá	Z 20.12.0
<input type="checkbox"/>	Ančík Zdeněk	1./I.	2E/95	77506	B3901-3	zk,zá	Z 02.01.0
<input type="checkbox"/>	Axman Lukáš	1./I.	2D/31	76625	B2341-3	zk,zá	Z 19.12.0
<input type="checkbox"/>	Bádr Miroslav	1./I.	2K/2	77416	B2341-3	zk,zá	Z 12.12.0
<input type="checkbox"/>	Balcar Petr	1./I.	2K/2	77415	B2341-3	zk,zá	Z 12.12.0

Navigace: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 »

Statistiky



Obrázek 8.5 Statistika udělení zápočtů

Vyučující potřebují stejně jako studenti vidět statistiky týkající se hodnocení předmětů, které vyučují. Do modulu Zkoušky -> Zápis hodnocení dle ECTS a modulu Zkoušky -> Zápis bodového hodnocení je tedy ke každému předmětu přidán odkaz na modul Statistika hodnocení předmětu popsany v kapitole 8.1.1. Vyučující se tak dostanou na všechny části modulu, jako je celková statistika hodnocení předmětu (popsaná v 8.1.1.1), statistika hodnocení předmětu pro jednotlivé termíny (popsaná v 8.1.1.2) a statistika hodnocení předmětu pro jednotlivé vyučující (popsaná v kapitole 8.1.1.3).

Do modulu Zkoušky -> Zápis zápočtů je ke každému předmětu přidán histogram udělení zápočtu. Data se získávají z tabulky `st01.el_index`, kde je položka `zapocteno`, která určuje, jestli student zápočet dostal nebo ne. Získání dat trvá průměrně 0,042 s a zobrazuje se přímo vygenerovaný sloupcový graf. Výsledek je vidět na obrázku 8.5.

8.3 Realizace na Portálu VUT

Do Portálu VUT jsem začlenila modul s názvem Statistika. Odkaz na něj se zobrazuje v menu Portálu všem přihlášeným uživatelům. Pro lepší přenositelnost je modul vytvořen jako globální, může se tedy jednoduše začlenit i do ostatních částí Portálu bez zásahů do zdrojového kódu. Hlavní soubor modulu se nazývá `gm_statistiky.php`, soubor se zobrazením výstupů do HTML je `gm_statistiky_templ.php` a posledním souborem je `gm_statistiky_func.php`, který slouží jako knihovna funkcí. Všechny tři jsou umístěny v adresáři `_base` a jeho podadresářích.

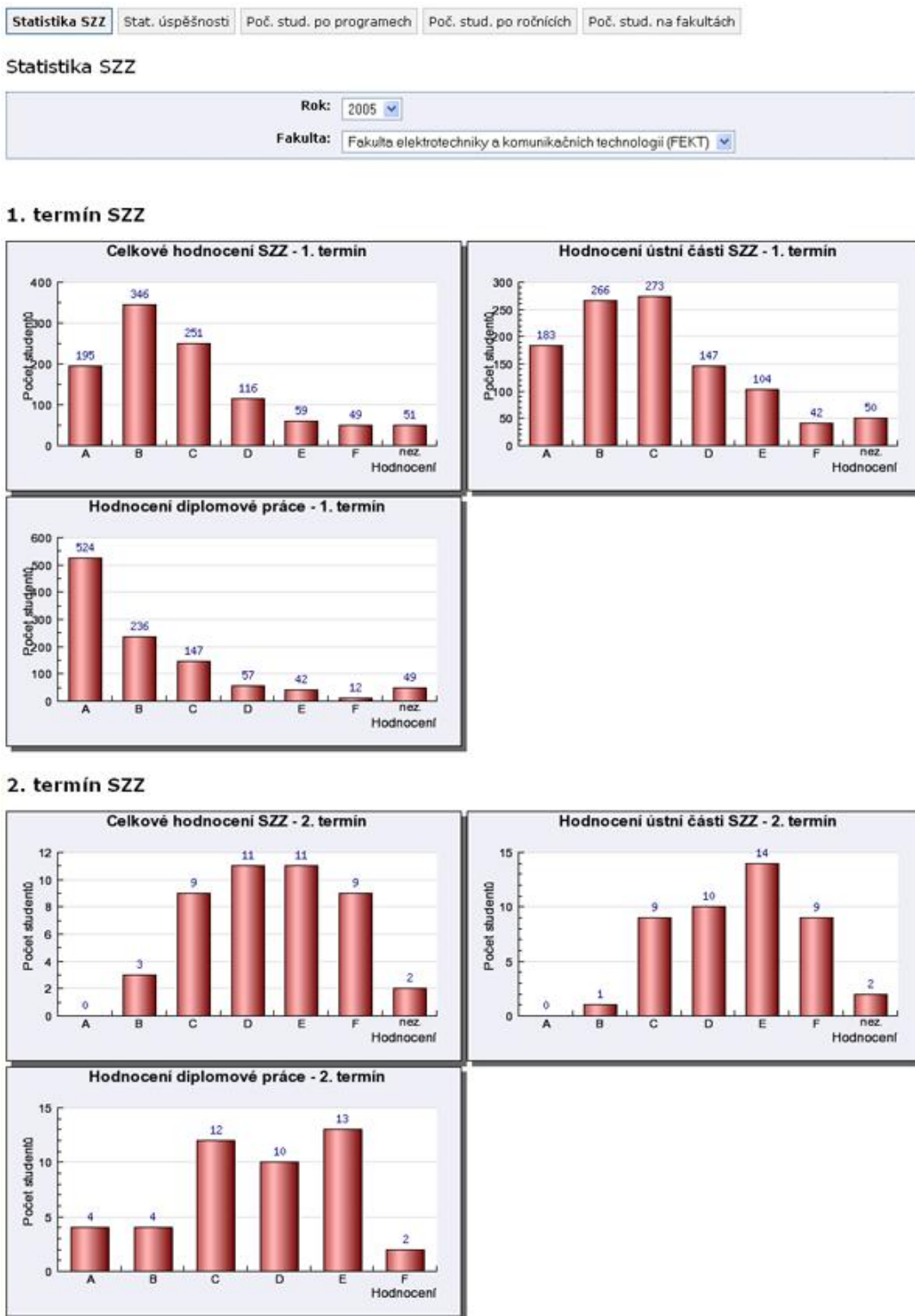
Modul je rozdělen na několik částí, obsahuje statistiky týkající se státní závěrečné zkoušky, úspěšnosti ukončení studia, počtu studentů po programech a po ročnících a rozdělení a počtu studentů na fakultách. U všech se nejdříve zobrazuje menu pro přepínání mezi nimi, formulář sloužící pro výběr roku a fakulty a nakonec samotná statistika.

8.3.1 Statistika státní závěrečné zkoušky

Výsledkem statistiky jsou histogramy pro celkové hodnocení státní závěrečné zkoušky, hodnocení ústní části SZZ a hodnocení diplomové práce. Všechny jsou založeny na hodnocení dle ECTS a zobrazují výsledky jak pro první tak i pro druhý termín SZZ. Celkem je tedy zobrazeno šest grafů, jejichž data se vztahují k vybranému roku a fakultě. Barvy modulu jsou voleny tak, aby ladily s barevnou paletou použitou na Portálu VUT.

Dotaz pro získání všech dat byl velmi pomalý (trval průměrně 1,263 s) a ani při různých optimalizacích se mi dobu trvání nepodařilo snížit. Proto jsem se rozhodla pro vytvoření nových materializovaných tabulek. Výsledkem jsou tabulky `st01.mv_statistika_hod_szz`,

st01.mv_statistika_hod_ust_szz a st01.mv_statistika_hod_dp popsané v kapitole 6.3.1. Příklad dotazu pro vytvoření jedné z tabulek je uveden v příloze 3.7.



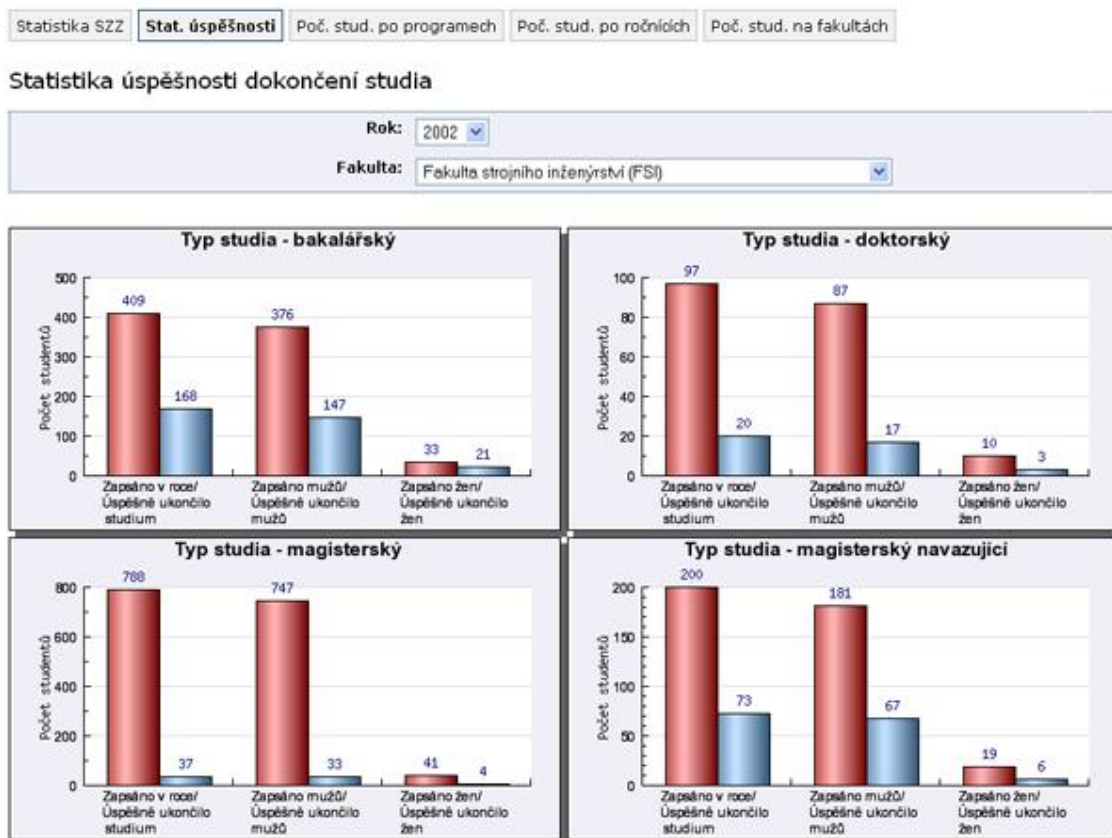
Obrázek 8.6 Statistika státní závěrečné zkoušky

Pro zobrazení grafů se jako nejlepší a také nejrychlejší ukázala možnost dočasného ukládání obrázků s grafy. Při zobrazení histogramů se tedy nejdříve zjistí, jestli pro vybranou fakultu, rok a termín neexistuje obrázek se stejným datem. Pokud ano, dojde k zobrazení obrázku. Pokud ne, dojde k načtení dat a vygenerování nového grafu, který se jako obrázek uloží do adresáře temp. V tomto adresáři dochází každých dvacet čtyři hodin k mazání souborů, měly by zde tedy být pouze aktuální soubory. K získávání a vygenerování grafu bude tedy docházet pouze jednou denně, poté už se budou jen zobrazovat uložené obrázky s grafy.

Grafický výstup je zobrazen na obrázku 8.6. Je zde vidět menu pro výběr určité části modulu, formulář pro výběr roku a fakulty a samotné histogramy poskytující informace o výsledcích státní závěrečné zkoušky na vybrané fakultě ve zvoleném roce.

8.3.2 Statistika úspěšnosti ukončení studia

Cílem statistiky je informovat o úspěšnosti ukončení studia na jednotlivých fakultách. Výsledné histogramy zobrazují pro každý typ studia informaci o tom, kolik bylo ve vybraném roce na fakultě zapsáno do tohoto typu studia studentů a kolik z nich studium úspěšně ukončilo. Jednotlivé třídy tvoří muži, ženy a všichni studenti. Výstup je na obrázku 8.7.



Obrázek 8.7 Statistika úspěšnosti ukončení studia

Po najetí myši na jednotlivé sloupce histogramu se o nich zobrazí doplňující informace. U sloupců reprezentujících počet studentů, kteří ukončili studium, se navíc zobrazí výsledná hodnota v procentech. Ta značí, kolik procent studentů z celkového počtu přijatých dokončilo úspěšně studium. Pro zajištění této funkčnosti je využito vygenerování grafu jako klikací mapy, které umožňuje knihovna JpGraph. Klikací mapa určuje aktivní a neaktivní místa obrázku, který bude fungovat jako odkaz. Mapa je reprezentována párovým tagem <map>, aktivní oblasti jsou reprezentovány tagy <area>. Mapa je spojena s obrázkem pomocí jména – atributu name. V tomto případě není přímo využito toho, aby mapa někam odkazovala, ale toho, že po najetí na odkaz (sloupec grafu) se zobrazí vysvětlující text.

Získání dat pomocí dotazu, který prováděl statistické výpočty, bylo velmi pomalé a neefektivní. Jako řešení se tedy nabízela možnost vytvoření nové materializované tabulky `st01_mv_statistika_osp_stud`. Dotaz pro její vytvoření je uveden v příloze 3.8, popis v kapitole 6.3.1. Rychlost získání dat z nově vytvořené tabulky je několikanásobně vyšší než bez ní. Použitý dotaz pro získání dat z tabulky trvá průměrně 0,031 s.

8.3.3 Počet studentů po programech

Grafický výstup této části modulu je znázorněn na obrázku 8.8. Tabulka obsahuje počty studentů pro jednotlivé programy, typy a formy studia pro vybraný rok a fakultu. Jako výstup je zvolena právě tabulka vzhledem k množství a detailnosti popisu dat.

Statistika SZZ Stat. úspěšnosti **Poč. stud. po programech** Poč. stud. po ročnících Poč. stud. na fakultách

Počty aktivních studentů po programech

Rok: 2006/2007
Fakulta: Fakulta stavební (FAST)

Kód studijního programu	Program	Typ studia	Forma studia	Délka studia	Titul	Počet studentů
B3503	Architektura pozemních staveb	bakalářský	prezenční studium	4	Bc.	93
B3607	Stavební inženýrství	bakalářský	prezenční studium	4	Bc.	2451
B3607	Stavební inženýrství	bakalářský	kombinované studium	4	Bc.	298
B3607	Stavební inženýrství	bakalářský	prezenční studium	4	Bc.	26
B3607	Stavební inženýrství	bakalářský	příjezd na krátkodobý studijní pobyt	4		11
B3609	Stavatelství	bakalářský	prezenční studium	3	Bc.	206
B3646	Geodézie a kartografie	bakalářský	prezenční studium	3	Bc.	190
M3607	Stavební inženýrství	magisterský	prezenční studium	5	Ing.	1195
M3607	Stavební inženýrství	magisterský	kombinované studium	5	Ing.	132
M3607	Stavební inženýrství	magisterský	prezenční studium	5	Ing.	32
M3646	Geodézie a kartografie	magisterský	prezenční studium	5	Ing.	110
N3607	Stavební inženýrství	magisterský navazující	prezenční studium	1,5	Ing.	0
N3607	Stavební inženýrství	magisterský navazující	příjezd na krátkodobý studijní pobyt	1		0
N3646	Geodézie a kartografie	magisterský navazující	prezenční studium	2	Ing.	0
P3607	Stavební inženýrství	doktorský	prezenční studium	3	Ph.D.	220
P3607	Stavební inženýrství	doktorský	kombinované studium	3	Ph.D.	199
P3646	Geodézie a kartografie	doktorský	kombinované studium	3	Ph.D.	3
P3646	Geodézie a kartografie	doktorský	prezenční studium	3	Ph.D.	9
P3917	Soudní inženýrství	doktorský	kombinované studium	3	Ph.D.	34
P3917	Soudní inženýrství	doktorský	prezenční studium	3	Ph.D.	16
SOCR	Socrates	magisterský	hostující student	1		1

Obrázek 8.8 Počet studentů po programech

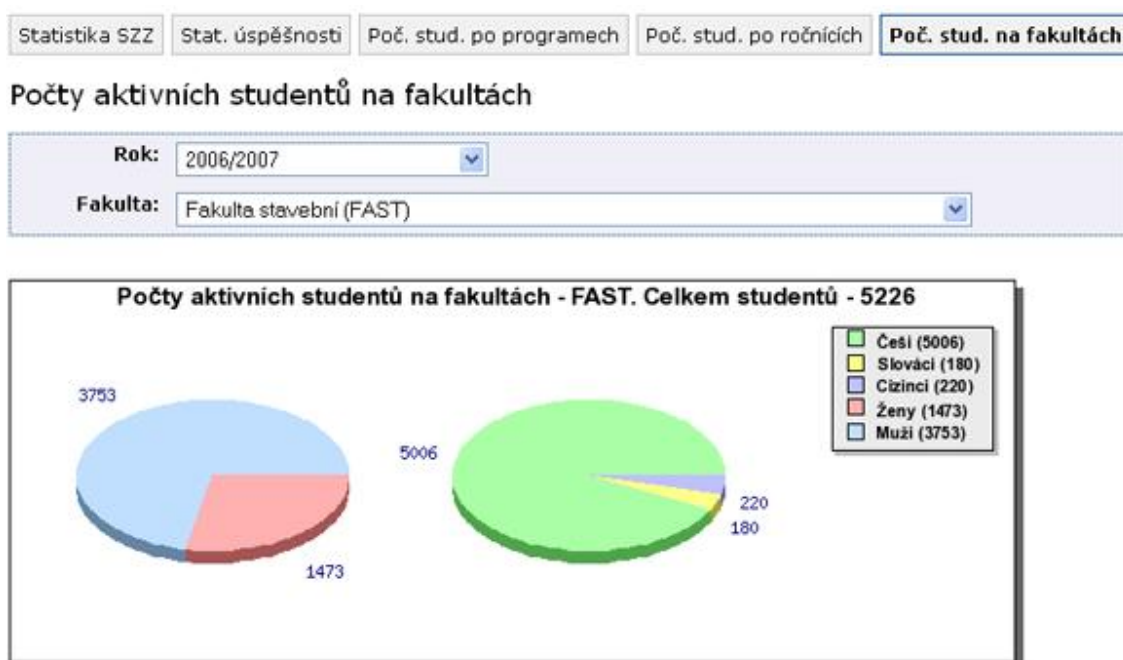
8.3.4 Počet studentů po ročnících

Grafický výstup této statistiky je velmi podobný počtu studentů po programech. Pro vybranou fakultu a akademický rok se zde zobrazují všechny programy a formy studia a k nim jednotlivé počty studentů pro každý ročník. Výstup je vypsán do tabulky, která umožňuje uvést všechny údaje v přesném tvaru.

8.3.5 Rozdělení a počet studentů na fakultách

Na obrázku 8.9 je uvedeno grafické zobrazení modulu týkajícího se počtu studentů na fakultách. V horní části se nachází formulář pro výběr akademického roku a fakulty, ve spodní části graf statistiky tvořený dvěma koláčovými grafy. První zobrazuje rozdělení počtu žen a mužů na fakultě a druhý národnostní rozdělení studentů. V titulku grafu se zobrazuje informace o celkovém počtu studentů na fakultě. Graf je vygenerován jako klikací mapa. Po najetí myši na jednotlivé díly koláče se zobrazí doplňující informace, jako je například počet studentů tvořící výše uvedený v procentech.

Pro získání dat jsem nejdříve vyzkoušela sql dotaz. Vzhledem k tomu, že se v dotazu počítalo mnoho statistických údajů, byl velmi pomalý. Vytvořila jsem tedy novou tabulku s názvem `st01.mv_statistika_pocet_stud`, která je popsána v kapitole 6.3.1. V ní jsou statistické údaje již předpočítány a dotaz pro získání dat z ní už je dostatečně rychlý.



Obrázek 8.9 Rozdělení a počet studentů na fakultách

8.3.6 Statistika přijímacího řízení

Statistika přijímacího řízení je volně přístupná všem návštěvníkům oficiálního webu VUT. Nachází se na webových stránkách VUT v sekci Studium a ECTS -> Statistiky přijímacího řízení. Jako podklad pro data slouží tabulka `st01.mv_statistika_uchazeci_sum`, která již byla vytvořena dříve a používá se i pro statistiku v systému Apollo. Získaná data se zobrazují přehledně v tabulce, kde jsou informace o statistice pro všechny fakulty a typy studia a nakonec souhrnné údaje pro celé VUT. Výsledná tabulka obsahuje počet podaných přihlášek, počet studentů, kteří se dostavili k přijímacímu řízení, počet těch, co splnili požadavky pro přijetí, počet přijatých studentů po přijímacím řízení, bez přijímacího řízení a mimo přijímací řízení, celkový počet přijatých a nepřijatých studentů a počet studentů, kteří byli zapsáni na fakultě.

9 Závěr

Cílem diplomové práce bylo navrhnout a implementovat studijní statistiky do Portálu Vysokého učení technického v Brně. Nejprve byla provedena analýza současného stavu na jednotlivých fakultách a v centrálním informačním systému. Dále byl proveden návrh řešení, byly vytvořeny nové struktury do databáze VUT a na závěr byla vytvořena webová aplikace. Výsledkem je několik rozšiřujících modulů přidaných do již existujících aplikací Studis, Teacher a Portál. V době dokončení této práce probíhá testování modulů a v blízké době by mělo dojít k jejich nasazení do ostrého provozu.

Během práce jsem se detailně seznámila s Portálem Vysokého učení technického v Brně i s aplikacemi Studis a Teacher, které jsou jeho součástí. Měla jsem tak možnost prostudovat některé zajímavé techniky řešení různých problémů. Seznámila jsem se s databázovou technologií Oracle a databázovým schématem st01, které obsahují data využívaná webovými aplikacemi. Bylo nutné prostudovat možnosti a prostředky jazyka PHP, HTML, XHTML a SQL, knihovny JpGraph a možnosti použití těchto nástrojů pro implementaci studijních statistik.

Další vývoj projektu by mohl směřovat například k přepracování statistik přijímacího řízení. Tyto statistiky byly vypracovány spolu s elektronickou přihláškou, jsou zastaralé a při jejich vývoji se nepoužívaly později zavedené konvence pro vývoj Portálu VUT.

Literatura

- [1] Welling, L., Thomsonová, L.: PHP a MySQL – rozvoj webových aplikací. Praha, SoftPress 2004.
- [2] Castagnetto, J., Rawat, H., Schumann, S. a kolektiv: Programujeme PHP profesionálně, Praha, Computer Press 2002.
- [3] Loney, K., Theriault, M.: Mistrovství v Oracle, Praha, Computer Press 2002
- [4] Musciano, Ch., Kennedy, B.: HTML a XHTML, Praha, Computer Press 2000.
- [5] Spainhour, S., Eckstein, R.: Webmaster v kostce, Praha, Computer Press 1999.
- [6] Staníček, P.: CSS kaskádové styly, Brno, Computer Press 2003.
- [7] Hendl, J.: Přehled statistických metod zpracování dat, Praha, Portál 2004
- [8] Hanousek, J., Charamza, P.: Moderní metody zpracování dat, Praha, Grada 1992
- [9] Úvod do CSS. Dokument dostupný na URL <http://www.webtvorba.cz/css/uvod-do-css.html> (květen 2007).
- [10] Skřivan, J.: Databáze a jazyk SQL, 4. 8. 2000. Dokument dostupný na URL <http://interval.cz/clanky/databaze-a-jazyk-sql/> (květen 2007).
- [11] Lim, J.: ADOdb Library for PHP Manual. Dokument dostupný na URL <http://phplens.com/adodb/> (květen 2007).
- [12] Vašíček, Z.: Stručný úvod do SVN. Dokument dostupný na URL <http://merlin.fit.vutbr.cz/FITkit/?pg=navody&cl=20060209svn> (květen 2007).
- [13] JpGraph Manual. Dokument dostupný na URL <http://calliope.ifh.de/ape/html/include.php3/jpgraph-1.12.2/docs/html/index.html> (květen 2007).
- [14] PHP – Graphs and Charts. Dokument dostupný na URL http://www.hotscripts.com/PHP/Scripts_and_Programs/Graphs_and_Charts (květen 2007).
- [15] Wikipedia. Otevřená encyklopedie dostupná na URL <http://cs.wikipedia.org> (květen 2007).

Seznam použitých zkratek

ADODB – Database Abstraction Library for PHP and Python

ASP – Active Server Pages

BUT – Brno University of Technology

CGI – Common Gateway Interface

CSS – Cascading Style Sheets

CVS – Concurrent Versions System

DLL – Dynamic Linking Library

ECTS – European Credit Transfer System

FAQ – Frequently Asked Questions

GPL – General Public License

HTML – Hypertext Markup Language

IMAP – Internet Message Access Protocol

ISIC – International Student Identity Card

JSP – Java Server Pages

LDAP – Lightweight Directory Access Protocol

LVS – Linux Virtual Server

NAT – Network Address Translation

NFS – Network File System

ODBC – Open Database Connectivity Standard

PDF – Portable Document Format

PHP – Hypertext Preprocessor

SGML – Standard Generalized Markup Language

SNMP – Simple Network Management Protocol

SQL – Structured Query Language

SSL – Secure Sockets Layer

SŘBD – Systém řízení báze dat

SZZ – Státní závěrečná zkouška

VUT – Vysoké učení technické

W3C – World Wide Web Consortium

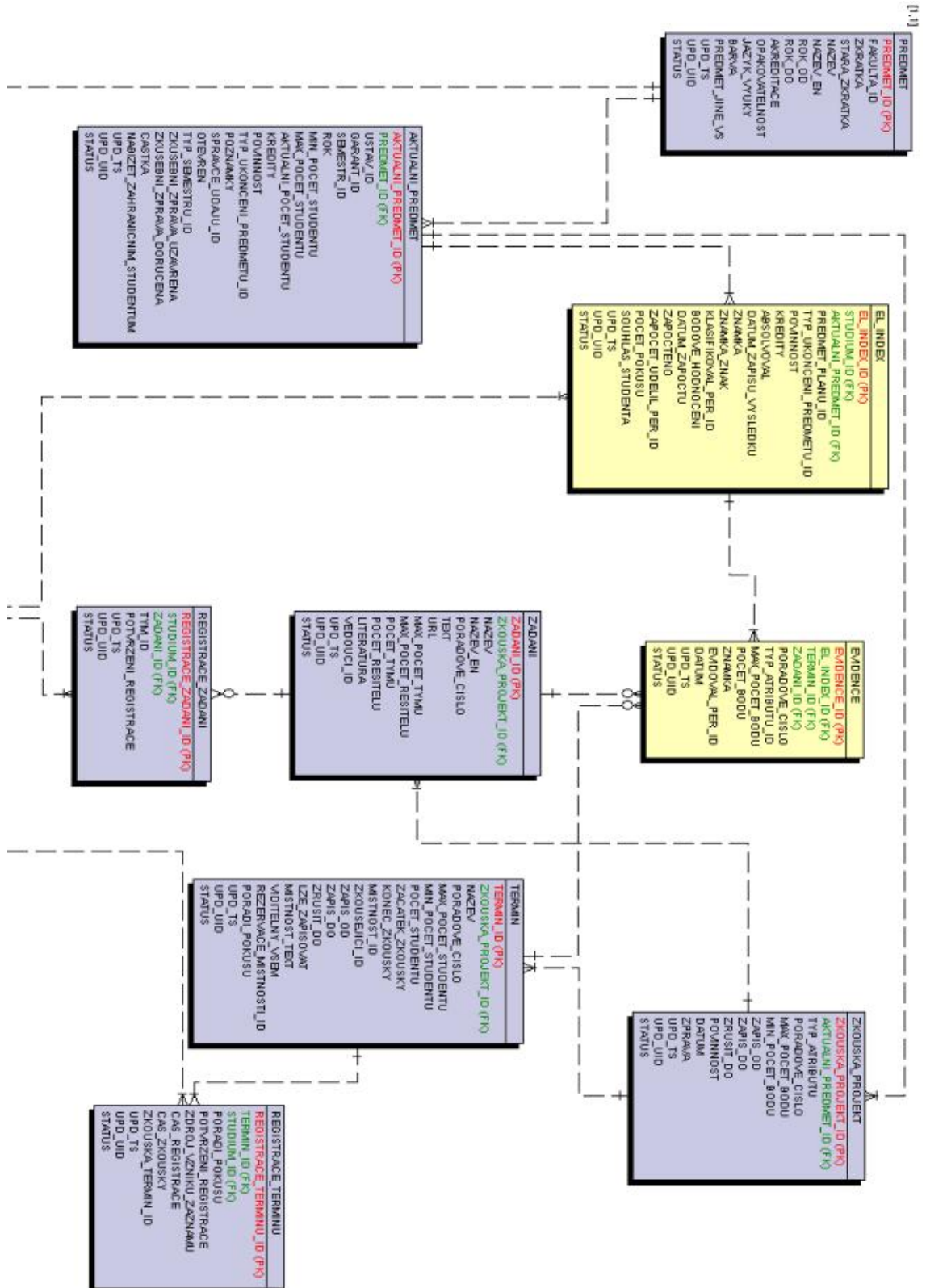
WWW – World Wide Web

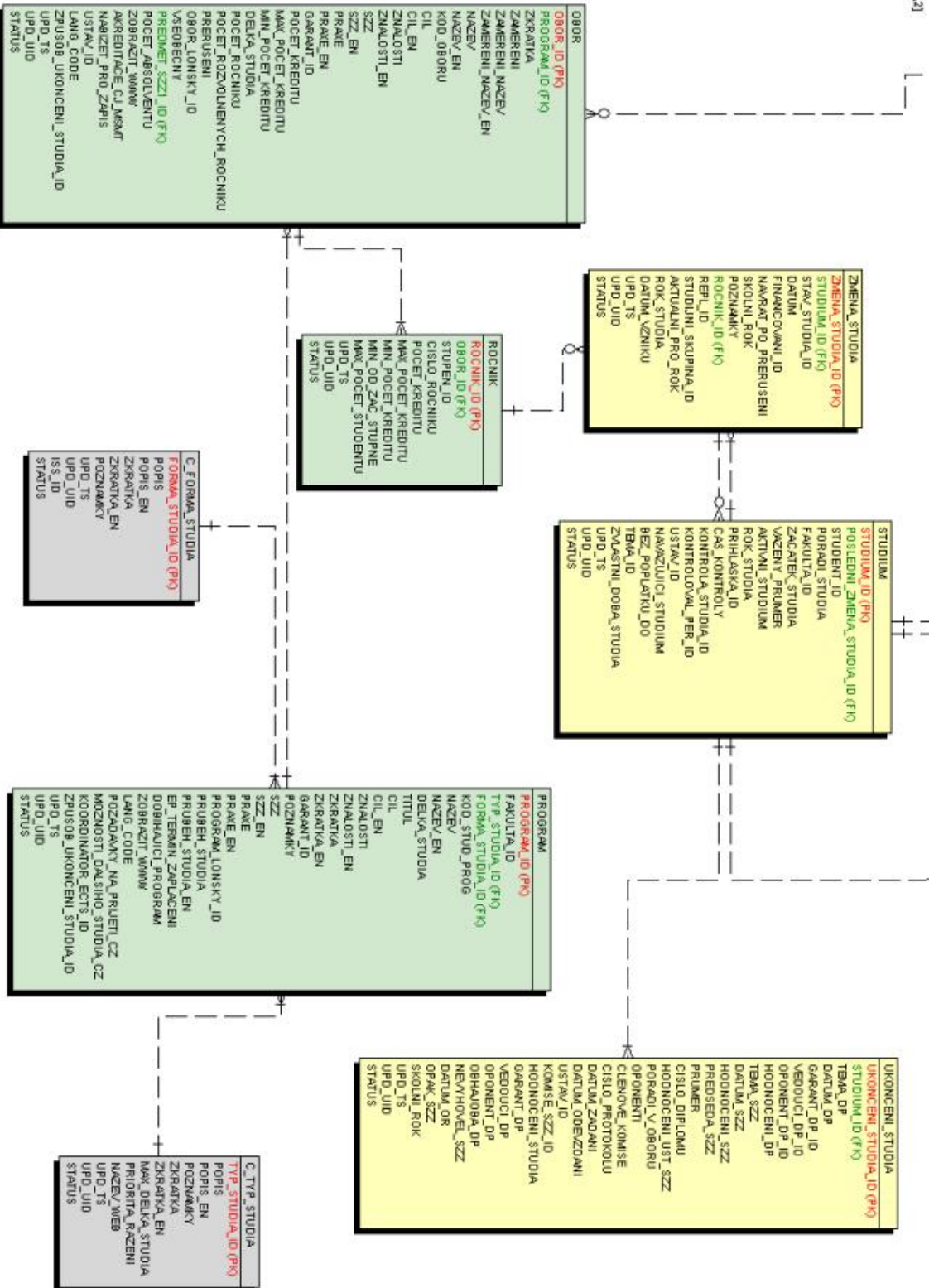
XHTML – Extensible Hypertext Markup Language

XML – Extensible Markup Language

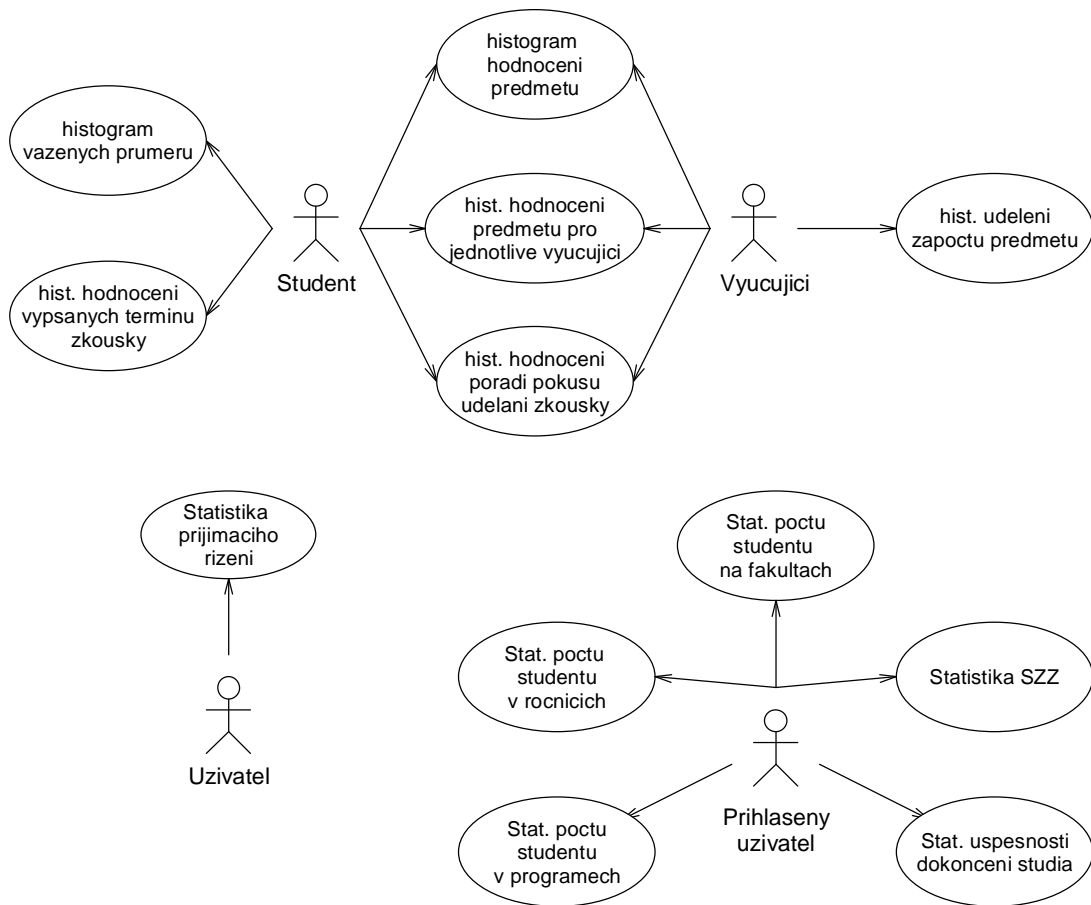
Přílohy

Příloha 1 ER diagram, část schématu st01





Příloha 2 Model případů použití



Příloha 3 SQL dotazy

Příloha 3.1 Dotaz pro celkovou statistiku předmětu 1

```
select cup.znamka_znak as znamka_znak,
       (select count(1)
        from st01.el_index ei2
        where ei2.aktualni_predmet_id = :aktualni_predmet_id
              and ei2.znamka_znak = cup.znamka_znak
              and ei2.status in (0, 9)) as pocet
  from st01.el_index ei, st01.c_ukonceni_predmetu cup
 where ei.aktualni_predmet_id = :aktualni_predmet_id
       and ei.typ_ukonceni_predmetu_id = cup.typ_ukonceni_predmetu_id
       and cup.fakulta_id = :fakulta_id
       and cup.rok = :rok
       and ei.status in (0, 9)
       and cup.status in (0, 9)
 group by cup.znamka_znak
 union
 select decode(ei.znamka_znak, null, ' nez.', ei.znamka_znak) as
        znamka_znak, count(1) as pocet
  from st01.el_index ei
 where ei.aktualni_predmet_id = :aktualni_predmet_id
       and ei.znamka_znak IS NULL
       and ei.status in (0, 9)
 group by ei.znamka_znak order by znamka_znak
```

Příloha 3.2 Dotaz pro celkovou statistiku předmětu 2

```
select decode(ei.znamka_znak, null, ' nez.', ei.znamka_znak) as
        znamka_znak, count(1) as pocet
  from st01.el_index ei
 where ei.aktualni_predmet_id = :aktualni_predmet_id
       and ei.status in (0, 9)
 group by ei.znamka_znak order by znamka_znak
```

Příloha 3.3 Dotaz pro statistiku hodnocení předmětu pro jednotlivé termíny 1

```
select cup.znamka_znak as znamka_znak,
       (select count(1)
        from st01.evidence ev2, st01.el_index ei1
        where ev2.el_index_id = ei1.el_index_id
              and ei1.aktualni_predmet_id = :aktualni_predmet_id
              and ev2.znamka = cup.znamka_znak
              and ev2.poradove_cislo = :poradove_cislo
              and ev2.typ_atributu_id = 7
              and ev2.status in (0, 9)
              and ei1.status in (0, 9)) as pocet
  from st01.el_index ei, st01.c_ukonceni_predmetu cup
 where ei.aktualni_predmet_id = :aktualni_predmet_id
       and ei.typ_ukonceni_predmetu_id = cup.typ_ukonceni_predmetu_id
       and cup.fakulta_id = :fakulta_id
       and cup.rok = :rok
       and ei.status in (0, 9)
       and cup.status in (0, 9)
 group by cup.znamka_znak
 union
 select decode(ev.znamka, null, ' nez.', ev.znamka) as znamka_znak,
        count(1) as pocet
  from st01.evidence ev, st01.el_index ei2
```

```

where ei2.aktualni_predmet_id = :aktualni_predmet_id
and ei2.el_index_id = ev.el_index_id
and ev.znamka IS NULL
and ev.status in (0, 9)
and ei2.status in (0, 9)
group by ev.znamka order by znamka_znak

```

Příloha 3.4 Dotaz pro statistiku hodnocení předmětu pro jednotlivé termíny 2

```

select ev.znamka as znamka_znak, count(1) as pocet
from st01.el_index ei, st01.evidence ev
where ei.aktualni_predmet_id = :aktualni_predmet_id
and ev.el_index_id = ei.el_index_id
and ev.poradove_cislo = :poradove_cislo
and ev.typ_atributu_id = 7
and ei.status in (0, 9)
and ev.status in (0, 9)
group by ev.znamka order by znamka_znak

```

Příloha 3.5 Dotaz pro statistiku hodnocení předmětu pro jednotlivé vyučující 1

```

select cup.znamka_znak as znamka_znak,
(select count(1)
from st01.el_index ei2
where ei2.aktualni_predmet_id = :aktualni_predmet_id
and ei2.znamka_znak = cup.znamka_znak
and ei2.klasifikoval_per_id = :klasifikoval_id
and ei2.status in (0, 9)) as pocet
from st01.el_index ei, st01.c_ukonceni_predmetu cup
where ei.typ_ukonceni_predmetu_id = cup.typ_ukonceni_predmetu_id
and cup.fakulta_id = :fakulta_id
and ei.klasifikoval_per_id = :klasifikoval_id
and cup.rok = :rok
and ei.status in (0, 9)
and cup.status in (0, 9)
group by cup.znamka_znak
union
select decode(ei.znamka_znak, null, 'nez.', ei.znamka_znak) as
znamka_znak, count(1) as pocet
from st01.el_index ei
where ei.aktualni_predmet_id = :aktualni_predmet_id
and ei.klasifikoval_per_id = :klasifikoval_id
and ei.znamka_znak IS NULL
and ei.status in (0, 9)
group by ei.znamka_znak order by znamka_znak

```

Příloha 3.6 Dotaz pro statistiku hodnocení předmětu pro jednotlivé vyučující 2

```

select decode(ei.znamka_znak, null, ' nez.', ei.znamka_znak) as
znamka_znak, count(1) as pocet
from st01.el_index ei
where ei.aktualni_predmet_id = :aktualni_predmet_id
and ei.status in (0, 9)
and ei.klasifikoval_per_id = :klasifikoval_id
group by ei.znamka_znak order by znamka_znak

```

Příloha 3.7 Dotaz pro vytvoření tabulky st01.mv_statistika_hod_szz

```

create table st01.mv_statistika_hod_szz as (
select cup.fakulta_id, cup.rok, cup.znamka_znak as znamka_znak,
(select count(*)

```



```

        from (select us.hodnoceni_szz,
                    us.ukonceni_studia_id,
                    us.skolni_rok,
                    s.fakulta_id
               from st01.ukonceni_studia us, st01.studium s
               where s.studium_id = us.studium_id
                    and s.status in (0, 9)
                    and us.status in (0, 9)
                    and us.ukonceni_studia_id not in
                        (select usn.ukonceni_studia_id
                         from st01.ukonceni_studia_neuspesne usn
                         where usn.status in (0, 9))
               union
               select usn.hodnoceni_szz,
                      usn.ukonceni_studia_id,
                      usn.skolni_rok,
                      s.fakulta_id
               from st01.ukonceni_studia_neuspesne usn, st01.studium s
               where s.studium_id = usn.studium_id
                    and s.status in (0, 9)
                    and usn.status in (0, 9))
        where hodnoceni_szz = cup.znamka_znak
              and skolni_rok = cup.rok
              and fakulta_id = cup.fakulta_id) as pocet,
1 as termin
from st01.c_ukonceni_predmetu cup
where cup.typ_ukonceni_predmetu_id = 9
      and cup.status in (0, 9)
union
select s.fakulta_id, us.skolni_rok as rok, 'nez.' as znamka_znak,
       count(1) as pocet, 1 as termin
   from st01.ukonceni_studia us, st01.studium s
  where s.studium_id = us.studium_id
        and us.hodnoceni_szz IS NULL
        and us.skolni_rok IS NOT NULL
        and us.status in (0, 9)
        and s.status in (0, 9)
        and us.ukonceni_studia_id not in
            (select usn.ukonceni_studia_id
             from st01.ukonceni_studia_neuspesne usn
             where usn.status in (0, 9))
group by s.fakulta_id, us.skolni_rok
union
select cup.fakulta_id, cup.rok, cup.znamka_znak as znamka_znak,
       (select count(*)
        from st01.ukonceni_studia          us,
             st01.studium                  s
        where us.skolni_rok = cup.rok
              and us.hodnoceni_szz = cup.znamka_znak
              and s.studium_id = us.studium_id
              and s.fakulta_id = cup.fakulta_id
              and s.status in (0, 9)
              and us.status in (0, 9)
              and us.ukonceni_studia_id in
                  (select usn.ukonceni_studia_id
                   from st01.ukonceni_studia_neuspesne usn
                   where usn.status in (0, 9))) as pocet,
2 as termin
from st01.c_ukonceni_predmetu cup
where cup.typ_ukonceni_predmetu_id = 9
      and cup.status in (0, 9)

```

```

union
select s.fakulta_id, us.skolni_rok as rok, 'nez.' as znamka_znak,
       count(1) as pocet, 2 as termin
  from st01.ukonceni_studia us, st01.studium s
 where s.studium_id = us.studium_id
       and us.hodnoceni_szz IS NULL
       and us.skolni_rok IS NOT NULL
       and us.status in (0, 9)
       and s.status in (0, 9)
       and us.ukonceni_studia_id in
         (select usn.ukonceni_studia_id
          from st01.ukonceni_studia_neuspesne usn
          where usn.status in (0, 9))
 group by s.fakulta_id, us.skolni_rok)

```

Příloha 3.8 Dotaz pro vytvoření tabulky st01.mv_statistika_osp_stud

```

create table st01.mv_statistika_osp_stud as (
select zs.skolni_rok, s.fakulta_id, p.typ_studia_id,
       count(*) as zapsano,
       count(zs2.zmena_studia_id) as ukonceno,
       count(zap_muži.per_id) as zapsano_muži,
       count(uk_muži.per_id) as ukonceno_muži,
       count(zap_ženy.per_id) as zapsano_ženy,
       count(uk_ženy.per_id) as ukonceno_ženy
  from st01.zmena_studia zs, st01.studium s, st01.rocnik r,
       st01.obor o, st01.program p,
       (select *
        from st01.zmena_studia zs2
        where zs2.status in (0, 9)
          and zs2.stav_studia_id = 61) zs2,
       (select per.per_id
        from brutisadm.person per
        where per.per_sex = 'M'
          and per.status in (0, 9)) zap_muži,
       (select per.per_id
        from brutisadm.person per
        where per.per_sex = 'M'
          and per.status in (0, 9)) uk_muži,
       (select per.per_id
        from brutisadm.person per
        where per.per_sex = 'F'
          and per.status in (0, 9)) zap_ženy,
       (select per.per_id
        from brutisadm.person per
        where per.per_sex = 'F'
          and per.status in (0, 9)) uk_ženy
 where s.studium_id = zs.studium_id
       and zs.skolni_rok =
         (select min(zs2.skolni_rok)
          from st01.zmena_studia zs2
          where zs2.status in (0, 9)
            and zs2.studium_id = zs.studium_id)
       and not exists (select 1
                       from st01.zmena_studia zs2
                       where zs2.status in (0, 9)
                         and zs2.studium_id = zs.studium_id
                         and zs2.datum < zs.datum)
       and zs.rocnik_id = r.rocnik_id
       and r.obor_id = o.obor_id
       and o.program_id = p.program_id)

```

```

    and r.status in (0, 9)
    and o.status in (0, 9)
    and p.status in (0, 9)
    and zs.status in (0, 9)
    and s.studium_id = zs2.studium_id(+)
    and st01.pkg_teacher.f_studium_id2per_id(zs2.studium_id) =
uk_muzy.per_id(+)
    and st01.pkg_teacher.f_studium_id2per_id(zs2.studium_id) =
uk_zeny.per_id(+)
    and s.student_id = zap_muzy.per_id(+)
    and s.student_id = zap_zeny.per_id(+)
group by zs.skolni_rok, s.fakulta_id, p.typ_studia_id, ctyp.popis,
ctyp.popis_en)

```

Příloha 3.9 Dotaz pro vytvoření tabulky st01.mv_statistika_poc_stud

```

create table st01.mv_statistika_pocet_stud as (
select m.fakulta_id, m.zkratka, pr.rok_platnosti,
(select count(1) from st01.program p, st01.studium stud,
st01.zmena_studia z, st01.stupen s, st01.rocnik r, st01.obor o
where stud.status in (0,9)
and s.status in (0,9)
and r.status in (0,9)
and o.status in (0,9)
and p.status in (0,9)
and p.rok_platnosti = pr.rok_platnosti
and p.fakulta_id = m.fakulta_id
and o.obor_id = s.obor_id
and p.program_id = o.program_id
and s.stupen_id = r.stupen_id
and z.rocnik_id = r.rocnik_id
and z.status in (0,9)
and z.skolni_rok = p.rok_platnosti
and stud.studium_id = z.studium_id
and z.aktualni_pro_rok = 1
and z.stav_studia_id > 10
and z.stav_studia_id < 20
) as aktivni,
(select count(1) from st01.program p, st01.studium stud,
st01.zmena_studia z, st01.stupen s, st01.rocnik r, st01.obor o,
brutisadm.person per
where stud.status in (0,9)
and s.status in (0,9)
and r.status in (0,9)
and o.status in (0,9)
and p.status in (0,9)
and p.rok_platnosti = pr.rok_platnosti
and p.fakulta_id = m.fakulta_id
and o.obor_id = s.obor_id
and p.program_id = o.program_id
and s.stupen_id = r.stupen_id
and z.rocnik_id = r.rocnik_id
and z.status in (0,9)
and z.skolni_rok = p.rok_platnosti
and stud.studium_id = z.studium_id
and z.aktualni_pro_rok = 1
and z.stav_studia_id > 10
and z.stav_studia_id < 20
and per.per_id = stud.student_id
and per.per_sex = 'M') as muzi,
(select count(1) from st01.program p, st01.studium stud,

```

```

st01.zmena_studia z, st01.stupen s, st01.rocnik r, st01.obor o,
brutisadm.person per
  where stud.status in (0,9)
    and s.status in (0,9)
    and r.status in (0,9)
    and o.status in (0,9)
    and p.status in (0,9)
    and p.rok_platnosti = pr.rok_platnosti
    and p.fakulta_id = m.fakulta_id
    and o.obor_id = s.obor_id
    and p.program_id = o.program_id
    and s.stupen_id = r.stupen_id
    and z.rocnik_id = r.rocnik_id
    and z.status in (0,9)
    and z.skolni_rok = p.rok_platnosti
    and stud.studium_id = z.studium_id
    and z.aktualni_pro_rok = 1
    and z.stav_studia_id > 10
    and z.stav_studia_id < 20
    and per.per_id = stud.student_id
    and per.per_sex = 'F') as zeny,
(select count(1) from st01.program p, st01.studium stud,
st01.zmena_studia z, st01.stupen s, st01.rocnik r, st01.obor o,
brutisadm.person per
  where stud.status in (0,9)
    and s.status in (0,9)
    and r.status in (0,9)
    and o.status in (0,9)
    and p.status in (0,9)
    and p.rok_platnosti = pr.rok_platnosti
    and p.fakulta_id = m.fakulta_id
    and o.obor_id = s.obor_id
    and p.program_id = o.program_id
    and s.stupen_id = r.stupen_id
    and z.rocnik_id = r.rocnik_id
    and z.status in (0,9)
    and z.skolni_rok = p.rok_platnosti
    and stud.studium_id = z.studium_id
    and z.aktualni_pro_rok = 1
    and z.stav_studia_id > 10
    and z.stav_studia_id < 20
    and per.per_id = stud.student_id
    and per.per_nationalities = 'CZ') as cesi,
(select count(1) from st01.program p, st01.studium stud,
st01.zmena_studia z, st01.stupen s, st01.rocnik r, st01.obor o,
brutisadm.person per
  where stud.status in (0,9)
    and s.status in (0,9)
    and r.status in (0,9)
    and o.status in (0,9)
    and p.status in (0,9)
    and p.rok_platnosti = pr.rok_platnosti
    and p.fakulta_id = m.fakulta_id
    and o.obor_id = s.obor_id
    and p.program_id = o.program_id
    and s.stupen_id = r.stupen_id
    and z.rocnik_id = r.rocnik_id
    and z.status in (0,9)
    and z.skolni_rok = p.rok_platnosti
    and stud.studium_id = z.studium_id
    and z.aktualni_pro_rok = 1

```

```

        and z.stav_studia_id > 10
        and z.stav_studia_id < 20
        and per.per_id = stud.student_id
        and per.per_nationalities = 'SK') as slovaci,
(select count(1) from st01.program p, st01.studium stud,
st01.zmena_studia z, st01.stupen s, st01.rocnik r, st01.obor o,
brutisadm.person per
where stud.status in (0,9)
and s.status in (0,9)
and r.status in (0,9)
and o.status in (0,9)
and p.status in (0,9)
and p.rok_platnosti = pr.rok_platnosti
and p.fakulta_id = m.fakulta_id
and o.obor_id = s.obor_id
and p.program_id = o.program_id
and s.stupen_id = r.stupen_id
and z.rocnik_id = r.rocnik_id
and z.status in (0,9)
and z.skolni_rok = p.rok_platnosti
and stud.studium_id = z.studium_id
and z.aktualni_pro_rok = 1
and z.stav_studia_id > 10
and z.stav_studia_id < 20
and per.per_id = stud.student_id
and (per.per_nationalities <> 'CZ' and per.per_nationalities <>
'SK')) as cizinci
from st01.mv_soucasti_vut m,
(select distinct pr.fakulta_id, pr.rok_platnosti
from st01.program pr
where pr.status in (0,9)
) pr
where m.fakulta_id in (3,4,5,6,11,12,13,20042)
and m.fakulta_id = pr.fakulta_id)

```

Příloha 3.10 Dotaz pro vytvoření tabulky st01.mv_statistika_hodn_predmetu

```

create table st01.mv_statistika_hodn_predmetu as (
select ei.aktualni_predmet_id, '0-9' as popis, count(1) as pocet
from st01.el_index ei
where ei.bodove_hodnoceni >= 0
and ei.bodove_hodnoceni <= 9
and ei.status in (0, 9)
group by ei.aktualni_predmet_id
union
select ei.aktualni_predmet_id, '10-19' as popis, count(1) as pocet
from st01.el_index ei
where ei.bodove_hodnoceni >= 10
and ei.bodove_hodnoceni <= 19
and ei.status in (0, 9)
group by ei.aktualni_predmet_id
union
select ei.aktualni_predmet_id, '20-29' as popis, count(1) as pocet
from st01.el_index ei
where ei.bodove_hodnoceni >= 20
and ei.bodove_hodnoceni <= 29
and ei.status in (0, 9)
group by ei.aktualni_predmet_id
union
select ei.aktualni_predmet_id, '30-39' as popis, count(1) as pocet
from st01.el_index ei

```

```

where ei.bodove_hodnoceni >= 30
      and ei.bodove_hodnoceni <= 39
      and ei.status in (0, 9)
group by ei.aktualni_predmet_id
union
select ei.aktualni_predmet_id, '40-49' as popis, count(1) as pocet
      from st01.el_index ei
      where ei.bodove_hodnoceni >= 40
            and ei.bodove_hodnoceni <= 49
            and ei.status in (0, 9)
group by ei.aktualni_predmet_id
union
select ei.aktualni_predmet_id, '50-59' as popis, count(1) as pocet
      from st01.el_index ei
      where ei.bodove_hodnoceni >= 50
            and ei.bodove_hodnoceni <= 59
            and ei.status in (0, 9)
group by ei.aktualni_predmet_id
union
select ei.aktualni_predmet_id, '60-69' as popis, count(1) as pocet
      from st01.el_index ei
      where ei.bodove_hodnoceni >= 60
            and ei.bodove_hodnoceni <= 69
            and ei.status in (0, 9)
group by ei.aktualni_predmet_id
union
select ei.aktualni_predmet_id, '70-79' as popis, count(1) as pocet
      from st01.el_index ei
      where ei.bodove_hodnoceni >= 70
            and ei.bodove_hodnoceni <= 79
            and ei.status in (0, 9)
group by ei.aktualni_predmet_id
union
select ei.aktualni_predmet_id, '80-89' as popis, count(1) as pocet
      from st01.el_index ei
      where ei.bodove_hodnoceni >= 80
            and ei.bodove_hodnoceni <= 89
            and ei.status in (0, 9)
group by ei.aktualni_predmet_id
union
select ei.aktualni_predmet_id, '90-100' as popis, count(1) as pocet
      from st01.el_index ei
      where ei.bodove_hodnoceni >= 90
            and ei.bodove_hodnoceni <= 100
            and ei.status in (0, 9)
group by ei.aktualni_predmet_id)

```