



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Platforma průmyslové spolupráce

CZ.1.07/2.4.00/17.0041

Název

System pro zajištění zpětné vazby u studentských stáží

Popis a využití

- aplikace a popis využití
- projekt vychází ze současného stavu stáží na vysokých školách zaměřených na IT v České republice
- produkt zahrnuje model pro hodnocení stáží, který se snaží maximalizovat přínosy zpětné vazby a systém pro hodnocení stáží podporující navržený model"

Jazyk textu

- český

Autor (autoři)

- Eva Kučírková

Oficiální stránka projektu:

- <http://lasaris.fi.muni.cz/pps>

Dostupnost výukových materiálů a nástrojů online:

- <http://lasaris.fi.muni.cz/pps/study-materials-and-tools>

Obsah

| | | |
|-------|---|----|
| 1 | Úvod | 1 |
| 2 | Studentské stáže v IT | 3 |
| 2.1 | Vymezení pojmu stáž | 3 |
| 2.1.1 | Historický význam | 3 |
| 2.1.2 | Stáže v dnešní době | 3 |
| 2.2 | Současná situace na vysokých školách v ČR | 4 |
| 2.2.1 | Předmět průzkumu | 4 |
| 2.2.2 | Průzkum veřejně dostupných informací | 5 |
| 2.2.3 | Průzkum na základě rozesílaného dotazníku | 6 |
| 2.2.4 | Výsledky průzkumu | 7 |
| | Organizace stáží a souvislost se studiem | 7 |
| | Využití a zpracování zpětné vazby | 8 |
| 3 | Srovnání stážových modelů | 9 |
| 3.1 | Identifikované modely stáží | 9 |
| 3.1.1 | Povinná stáž v rámci studia | 9 |
| 3.1.2 | Stáž jako volitelný předmět | 10 |
| 3.1.3 | Dobrovolná stáž vázaná na studium | 10 |
| 3.1.4 | Individuální stáž | 10 |
| 3.2 | Kritické zhodnocení | 10 |
| 3.2.1 | Kritéria pro hodnocení | 10 |
| 3.2.2 | Porovnání modelů | 11 |
| 4 | Návrh modelu hodnocení stáží | 13 |
| 4.1 | Model dvoufázových stáží | 13 |
| 4.2 | Možnosti hodnocení stáží | 14 |
| 4.2.1 | Potřeba zpětné vazby | 14 |
| 4.2.2 | Možnosti pro sběr dat | 14 |
| 4.2.3 | Vhodný prostředek pro hodnocení stáží | 15 |
| 4.3 | Návrh modelu hodnocení | 15 |
| 4.4 | Dostupná řešení pro sběr hodnocení | 16 |
| 4.4.1 | Základní řešení | 16 |
| 4.4.2 | Rozsáhlejší aplikace | 16 |
| 4.4.3 | Shrnutí a porovnání | 17 |

| | | |
|-------|--|----|
| 5 | Analýza a návrh systému pro hodnocení stáží | 18 |
| 5.1 | <i>Požadavky na systém</i> | 18 |
| 5.2 | <i>Případy užití</i> | 19 |
| 5.3 | <i>Datový model jádra systému</i> | 20 |
| 5.3.1 | Aspekty | 20 |
| 5.3.2 | Účastníci, společnosti a období | 20 |
| 5.3.3 | Zaznamenávání hodnocení | 21 |
| 5.3.4 | Sporná hodnocení | 21 |
| 5.4 | <i>Rozpracování případů užití</i> | 22 |
| 5.4.1 | Přidání aspektu | 23 |
| 5.4.2 | Generování dvojice dotazníků | 23 |
| 5.4.3 | Ukládání hodnocení | 25 |
| 5.4.4 | Výpočet statistických údajů | 28 |
| | Statistiky jednoduchých aspektů | 28 |
| | Statistiky složených aspektů | 30 |
| 6 | Implementace systému | 32 |
| 6.1 | <i>Struktura implementace</i> | 32 |
| 6.2 | <i>Použité technologie</i> | 32 |
| 6.2.1 | Spring | 33 |
| 6.2.2 | Java Persistence API | 33 |
| 6.2.3 | Spring Web MVC | 34 |
| 6.2.4 | iText | 35 |
| 6.2.5 | JFreeChart | 35 |
| 6.2.6 | Apache POI | 35 |
| 6.3 | <i>Uživatelské rozhraní</i> | 36 |
| 6.3.1 | Vzhled aplikace | 36 |
| 6.3.2 | Dotazníky | 36 |
| 6.3.3 | Zobrazení statistik | 37 |
| 6.4 | <i>Bezpečnost aplikace</i> | 37 |
| 6.5 | <i>Zajištění kvality aplikace</i> | 38 |
| 6.5.1 | Jednotkové testování | 38 |
| 6.5.2 | Funkční testování | 39 |
| 6.5.3 | Testování zobrazení aplikace | 40 |
| 7 | Nasazení systému a přínos práce | 41 |
| 7.1 | <i>Potřebné komponenty pro nasazení</i> | 41 |
| 7.1.1 | Servletový kontejner | 41 |
| 7.1.2 | Databáze | 42 |
| 7.2 | <i>Nasazení aplikace</i> | 42 |
| 7.2.1 | <i>Interim Project</i> | 42 |
| 7.2.2 | Stáže projektu PPS | 42 |

| | | |
|-------|---|-----------|
| 7.2.3 | Možná rozšíření systému | 43 |
| 8 | Závěr | 44 |
| A | Ukázka grafického vzhledu aplikace | 47 |
| B | Ukázka vygenerovaných statistik ve formátu PDF | 54 |
| C | Elektronické přílohy | 58 |

Kapitola 1

Úvod

Studium na vysoké škole přináší studentům nezbytný teoretický základ k tomu, aby se mohli uplatnit v požadovaném oboru. Praxe je však stejně důležitá jako teorie, a proto je na některých vysokých školách součástí studia povinná praxe či stáž. Pokud si student může během studia sám vyzkoušet práci v daném oboru, získá tím velice cenné zkušenosti.

V oborech, jako je například medicína či učitelství, je praxe víceméně běžnou součástí studia u všech vysokých škol v České republice. V oboru informačních technologií je situace jiná. Některé školy nabízejí obory, jejichž součástí je povinná stáž, u jiných to tak není. Většina vysokých škol však nějakým způsobem spolupracuje s průmyslovou sférou. Školy si uvědomují, že uchazeči o práci, kteří mají alespoň nějaké zkušenosti s reálným světem, mají lepší možnost získat požadovanou pozici.

V práci je zkoumána situace na vysokých školách se zaměřením na IT v České republice. Bylo vybráno osm škol, u nichž byl proveden průzkum toho, jaké možnosti stáží studentům nabízí. Zjistil jsem, jak stáže probíhají, jak jsou začleněny do studia a jak či zda vůbec je získávána zpětná vazba. Na základě průzkumu jsem identifikovala několik stážových modelů, které se používají.

Na FI se v rámci projektu *Platforma průmyslové spolupráce* používá model *dvoufázových stáží*, který je v práci představen. Jedním ze znaků tohoto modelu je využití zpětné vazby. Navrhla jsem model hodnocení, který je možno u stáží využít. Jako praktický přínos práce byl navržen a vytvořen systém pro zaznamenávání hodnocení stáží, s jehož pomocí lze jednoduchým způsobem získávat zpětnou vazbu a odhalit problematická místa ve stážovém modelu. Systém byl navržen jako webová aplikace a při jeho tvorbě byl kladen důraz na jednoduchost použití a uživatelskou přívětivost. Aplikace byla vytvořena v jazyku JavaEE [1] s použitím aplikačního rámce Spring [2].

Systém byl nasazen pro stážový program *Interim Project*, který je součástí magisterského studijního oboru *Služby, výzkum, řízení a inovace* na Fakultě informatiky Masarykovy univerzity v Brně. V práci je zhodnoceno

nasazení systému a je nastíněno, jak by bylo možno systém dále rozvíjet.

V druhé kapitole je vysvětleno, co se rozumí pod pojmem stáž. Je provedena analýza současné situace v České republice. Další kapitola přináší přehled různých stážových modelů, které se na vysokých školách používají. Dále je pak představen model *dvoufázových stáží* a systém hodnocení, který jsem pro tento model navrhla. V páté kapitole je analyzován a navržen systém pro hodnocení stáží. Popis implementace systému je obsahem šesté kapitoly. Následující kapitola uvádí popis nasazení systému pro stážisty programu *Interim Project*, shrnuje výsledky a nabízí možnosti dalšího rozšíření aplikace.

Kapitola 2

Studentské stáže v IT

V této kapitole je představeno, co všechno může označovat termín stáž. Jsou rozebrány jednotlivé možnosti stáží a je upřesněno, jakým typem stáží se bude práce dále zabývat. Je provedena analýza současné situace v České republice, jejíž výsledky jsou v kapitole shrnuty.

2.1 Vymezení pojmu stáž

2.1.1 Historický význam

Podle Slovníku spisovné češtiny [3] má slovo stáž dva výklady. Označuje jak delší studijní pobyt na zkušenou, tak i praxi mediků na klinikách. Z historického pohledu byla medicína jediný obor, ve kterém bylo naprosto nutné, aby si studenti procvičili získané vědomosti v praxi. Lékař, který by pouze léta studoval z knih a s realitou se nikdy nesetkal, by pak po nástupu do práce zajisté nadělal více škody než užitku.

2.1.2 Stáže v dnešní době

Dnes se slovem stáž rozumí spíše pobyt na škole v cizí zemi. Student má možnost poznat nové prostředí, procvičit si cizí jazyk a naučit se věci, které se na jeho mateřské univerzitě neučí.

Kromě takovýchto studijních stáží se často setkáváme s pracovními stážemi v zahraničí, které nejsou většinou organizovány fakultou, ale zajišťuje je samostatná organizace. Největší a nejrozšířenější je studentská organizace *AIESEC*.¹ Má pobočky v devíti městech v České republice a spolupracuje s více než třemi tisíci partnery na lokální, národní i mezinárodní úrovni. V rámci programu *Technical Traineeship* mají studenti z oboru informačních technologií možnost vycestovat na půlroční až roční stáž do zahraničí. Uchazeči se mohou dostat na stáž víceméně do celého světa. Tato aktivita je pro studenty čistě dobrovolná.

1. <http://aiesec.cz>

Další věcí, kterou v dnešní době pojem stáž označuje, je studentská stáž v nějaké firmě při studiu. Na některých školách se pro tuto aktivitu používá pojem praxe, ve firmách světové úrovně však převažuje označení stáž. Jemný rozdíl mezi těmito termíny je podle mého názoru v tom, že praktikant se ve firmě má naučit dané činnosti a nepředpokládá se, že by dopředu již něco uměl. Na rozdíl od toho, stážista je někdo, kdo již má nějaké znalosti a dovednosti a ve firmě si je chce procvičit a upevnit. Z tohoto pohledu je pro společnosti výhodnější brát studenty na stáž, neboť se očekává, že stážisté by mohli být pro firmu přínosem.

Informační technologie jsou poměrně mladý a rychle se rozvíjející obor. Pro dobré pracovní uplatnění potřebuje student nejen vynikající teoretické základy, ale i jisté praktické dovednosti. Školy zaměřené na IT učí studenty i praktické věci, nicméně projekty řešené v rámci předmětů na fakultě jsou vždy zčásti umělé a nedokáží zcela simulovat situaci v komerční firmě. Absolventi informatických fakult pak mají často zkreslené představy o tom, jak vlastně praxe vypadá. Je velice potřebné, aby si studenti už v průběhu studia vyzkoušeli práci ve skutečné firmě. Skloubit plnohodnotné zaměstnání s prezenčním studiem je však hodně náročné. Studentská stáž je dobrou příležitostí, jak si otestovat a prohloubit znalosti získané ve škole. Na druhou stranu, zkušenosti ze stáže mohou být studentovi přínosem při studiu.

V této práci se zaměřuji na stáže, které student absolvuje při studiu (v průběhu semestru nebo v době prázdnin) a v České republice. Ve většině případů stáže probíhají přímo v místě studia. Tyto stáže bývají zpravidla neplacené. V ojedinělých případech může být student za svůj přínos firmě oceněn i finančně formou stipendia.

2.2 Současná situace na vysokých školách v ČR

Tato část se zabývá analýzou možností na veřejných vysokých školách v České republice. Analýza byla provedena v březnu 2012.

2.2.1 Předmět průzkumu

Na základě internetového průzkumu jsem stanovila osm informatických fakult v České republice, které budou předmětem dalšího zkoumání. Byly vybrány ty fakulty veřejných vysokých škol, na kterých se vyučuje informatický studijní program se zaměřením na praktické využití a softwarové inženýrství.

Do průzkumu byly vybrány tyto školy:

- České vysoké učení technické v Praze, Fakulta elektrotechnická;
- České vysoké učení technické v Praze, Fakulta informačních technologií;
- Univerzita Karlova v Praze, Matematicko-fyzikální fakulta;
- Masarykova univerzita v Brně, Fakulta informatiky;
- Vysoké učení technické v Brně, Fakulta informačních technologií;
- Vysoká škola báňská - Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky;
- Univerzita Hradec Králové, Fakulta informatiky a managementu;
- Západočeská univerzita v Plzni, Fakulta aplikovaných věd.

2.2.2 Průzkum veřejně dostupných informací

Pro každou z vybraných škol jsem provedla průzkum veřejně dostupných informací. Hledala jsem jakékoliv zmínky o stážích, praxích nebo jiné spolupráci s komerční sférou. Prohledávala jsem zejména internetové stránky škol, fakult i kateder. Zkoumala jsem také archívy aktualit, elektronické časopisy a sborníky.

Z internetového průzkumu vyplývá, že firmy projevují více iniciativy v organizaci stáží než fakulty. Nejrozšířenější možností zisku praktických zkušeností je zúčastnit se letní stáže u IBM v rámci *IBM Academic Internship Program*.²

Výsledky průzkumu jsou shrnuty v tabulce 2.1. Tabulka zachycuje, zda se daná fakulta zapojuje do programu *IBM Academic Internship Program*. Také obsahuje další možnosti stáží, které fakulta veřejně prezentuje. Bylo zjištěno, že stáže jsou studentům nabízeny ve společnostech Honeywell³, EATON⁴, Oracle⁵, RedHat⁶, Barclays Capital⁷ a GSA⁸. Stáže také nabízí severomoravská firma Kvados⁹.

2. <http://www.facebook.com/IBMSmartUniversityCZ>

3. <http://www.honeywell.com>

4. <http://www.eaton.com>

5. <http://www.oracle.com>

6. <http://www.redhat.com>

7. <http://www.barcap.com>

8. <http://www.gsa.europa.eu>

9. <http://www.kvados.cz>

| Škola | IBM | Další nabízené stáže | Dostupnost informací |
|------------|-----|--------------------------|----------------------|
| FEL ČVUT | ano | Honeywell, EATON | aktuality |
| FIT ČVUT | ano | Honeywell | web |
| MF UK | ano | Oracle | web |
| FI MU | ano | Red Hat | aktuality |
| FIT VUT | ano | | mailing list |
| FEI VŠB-TU | ano | Barclays Capital, Kvados | aktuality |
| FIM UHK | ne | | |
| FAV ZUP | ano | GSA | aktuality |

Tabulka 2.1: Veřejně nabízené možnosti stáží

V posledním sloupci tabulky je uvedeno, jak fakulta prezentuje možnosti spolupráce s praxí. Některé fakulty tyto informace uvádí na svých webových stránkách, u jiných je nutno dohledávat nabídky stáží či různých programů ve webových archívech aktualit.

2.2.3 Průzkum na základě rozesílaného dotazníku

Analýzou veřejně dostupných informací bylo zjištěno, že pouze jedna fakulta (FI MU v Brně) nabízí studijní obor, jehož součástí je povinná stáž. U většiny fakult jsou na webu prezentovány pouze nabídky firem. Nikde jsem nenašla, že by se některá fakulta přímo podílela na organizaci stáží. Domnívám se, že důvodem je nedostatečná prezentace informací na internetu. Abych zjistila přesnější informace, rozhodla jsem se do škol rozeslat dotazník.

Dotazník jsem rozeslala zástupcům jednotlivých fakult, kteří mají na dané fakultě na starosti spolupráci s průmyslem. Cílem dotazníku bylo zjistit tyto informace:

- zda fakulta nabízí stáže;
- jak jsou stáže začleněny do výuky (zda jsou povinné, povinně volitelné nebo čistě dobrovolné);
- jaké jsou vstupní podmínky pro absolvování stáže;
- jaká je délka trvání stáže a kdy většinou stáže probíhají;
- jaká je souvislost mezi obsahem stáží a obsahem studia;

- zda se ze stáží získává zpětná vazba a jak se využívá.

2.2.4 Výsledky průzkumu

Bylo zjištěno, že většina oslovených vysokých škol stáže studentům nabízí. Stáže zatím nenabízí pouze FIT ČVUT v Praze, a to proto, že se jedná o mladou fakultu, která zatím nemá dostatek průmyslových partnerů.

Organizace stáží a souvislost se studiem

Žádná z dotázaných fakult nemá stáže zahrnuté jako povinnou součást všech svých studijních oborů. Pouze na FI MU v Brně existuje studijní obor, jehož povinnou součástí je absolvování stáže. Tato informace souhlasí s veřejně prezentovanými daty, uvedenými v sekci 2.2.3.

V polovině dotázaných případů jsou stáže povinně volitelné. To znamená, že studenti nejsou povinni stáž absolvovat, ale pokud ji absolvují, tak za ni získají kredity. U ostatních škol jsou stáže pro studenty čistě dobrovolné. Nejsou součástí studia, ale pokud studenti mají zájem, tak jim škola stáž zprostředkuje.

Rozložení obvyklé délky stáží je uvedeno v tabulce 2.2.

| Délka (v měsících) | méně než 1 | 1 až 3 | více než 3 | je to různé |
|--------------------|------------|--------|------------|-------------|
| Zastoupení | 13 % | 25 % | 38 % | 25 % |

Tabulka 2.2: Rozložení délky stáží

Tabulka 2.3 pak uvádí, kdy většinou stáže probíhají.

| Období | v průběhu semestru | o letních prázdninách | je to různé |
|------------|--------------------|-----------------------|-------------|
| Zastoupení | 50 % | 13 % | 38 % |

Tabulka 2.3: Rozložení doby konání stáží

V polovině případů neexistují žádné předpoklady pro to, aby student mohl nastoupit na stáž. Každý, kdo má zájem, se může stáže zúčastnit. U druhé poloviny škol musí student splnit určité podmínky. Mezi nejběžnější podmínky patří to, že student musí být v určitém semestru či ročníku nebo již mít získaný daný počet kreditů. Ve 30 % případů musí student prokázat svou připravenost při ústním pohovoru. Pouze jedna škola vyžaduje, aby student před stáží absolvoval určitý předmět.

Také jsem se snažila zjistit, jak jsou stáže propojeny se studiem. V 71 % případů si studenti na stážích vyzkouší teorii, kterou se naučili ve škole. Také mohou v rámci stáže vypracovat bakalářskou nebo diplomovou práci. Na 43 % stáží se student musí naučit pracovat s technologiemi, se kterými nemá možnost se setkat ve škole. Na MFF UK v Praze nemají přímo stanovenou vazbu stáží a studia, je to čistě individuální a závisí to jak na studentovi, tak na firmě, ve které je stáž absolvována. FAV ZU v Plzni požaduje, aby nabízené stáže nebyly pouze rutinní činností. Musí se jednat o kreativní práci, kde student uplatní dovednosti získané ve škole i svou schopnost naučit se novým věcem.

Využití a zpracování zpětné vazby

Bylo zjištěno, že polovina škol systematicky získává a zpracovává zpětnou vazbu. Čtvrtina škol zpětnou vazbu sice získává, ale pouze nahodile (jen u některých stáží). Ostatní školy zpětnou vazbu nezískávají.

Všechny školy, které získávají zpětnou vazbu, ji získávají od studentů. Polovina z těchto škol navíc získává zpětnou vazbu i od zástupců firmy, kde je student na stáži.

Zpětná vazba od studentů se ve většině případů získává formou psané zprávy. Pouze v jednom případě vyplňuje student dotazník, ve kterém odpovídá na kvantitativní i kvalitativní otázky ohledně své stáže. Zástupci firmy také ve většině případů píšou zprávu, pouze jedna škola požaduje vyplnění dotazníku. Nejedná se o však o tu školu, která používá dotazníky pro získání zpětné vazby od studentů.

Zpětnou vazbu jednotlivé školy využívají k různým účelům. U 67 % dotázaných se archivuje. U poloviny škol se využívá pro úpravu stážového modelu. Může se jednat o modifikaci vstupních požadavků na studenty, změny v celkové organizaci stáže nebo dokonce o změnu firem, u kterých se stáž vykonává. Polovina škol, které nabízejí stáž jako volitelný předmět, využívá zpětné vazby k hodnocení studentů. Pouze jediná škola vyhodnocuje zpětnou vazbu pro získání statistického přehledu, který je pak možno prezentovat dalším zájemcům o stáže i veřejnosti.

Školy, které používají zpětnou vazbu pro úpravu stážového modelu, získávají informace většinou formou psané zprávy od studenta. Vyhodnocení těchto zpráv však musí být podle mého názoru poměrně náročné. Také škola, která vytváří statistické přehledy, musí ručně zpracovávat zprávy od studentů. Naopak školy, které používají pro sběr dat dotazníky, získané informace pouze archivují.

Kapitola 3

Srovnání stážových modelů

Na základě provedeného průzkumu jsem identifikovala různé stážové modely, které se na školách používají. V této kapitole je provedeno jejich porovnání a zhodnocení. Pro každý model jsem se snažila najít a popsat jeho přínosy i nedostatky.

3.1 Identifikované modely stáží

Z průzkumu vyplývá, že organizace stáží na školách je poměrně různorodá. Nedá se jednoduše identifikovat nějaký vzor, který by se opakoval. Každá škola si stáže organizuje individuálně. Jako nejdůležitější znak stážového modelu se mi jeví to, jak jsou stáže začleněny do studia. Stáž může tvořit povinnou součást studia, být povinně volitelná nebo úplně dobrovolná. Podle tohoto rozdělení jsem identifikovala následující modely.

3.1.1 Povinná stáž v rámci studia

Tento typ stáže se na jiných oborech označuje také jako povinná praxe. Jedná se o nedílnou součást studia. Před nástupem na stáž musí student splnit určité podmínky. Student musí být v odpovídajícím ročníku či semestru studia, mít daný počet kreditů nebo mít absolvované stanovené předměty. Ze stáže je studentovi pak uděleno hodnocení. Stáže většinou probíhají v průběhu semestru a trvají zpravidla tři měsíce až půl roku. Náplň stáže by měla navazovat na teoretické znalosti získané ve škole. Student by si měl v praxi vyzkoušet to, co se ve škole naučil. V rámci stáže může student vypracovat bakalářskou nebo diplomovou práci. Z těchto stáží se získává zpětná vazba, většinou formou psané zprávy od studenta. Zpětná vazba slouží především k hodnocení studenta. Někdy se také vypracovávají statistické přehledy, aby mohla škola na veřejnosti prezentovat úspěšnost stážistů.

3.1.2 Stáž jako volitelný předmět

Jedná se o podobný model jako povinná stáž. Hlavní rozdíl je v tom, že stáž není pro studenty oboru povinná, ale povinně volitelná. Pokud má student zájem, může se na stáž přihlásit. Toto je většinou organizováno v rámci nějakého předmětu, ze kterého pak student po absolvování stáže získá zápočet a kredity. Obsah stáží by měl rozvíjet znalosti získané ve škole, nicméně náplní práce mohou být i dovednosti, které se student musí nejdříve naučit. Zpětná vazba se z těchto stáží získává a slouží nejen k hodnocení studentů, ale také k úpravě stážového modelu. Může se jednat například o modifikaci vstupních předpokladů, které musí student splňovat, aby si stáž mohl zapsat.

3.1.3 Dobrovolná stáž vázaná na studium

Tyto stáže nejsou pro studenty nijak začleněné do studia. Jejich absolvování je čistě dobrovolné a hlavním přínosem pro studenta jsou zkušenosti, které získá. Stáže jsou organizovány školou a tak mohou existovat určité podmínky, které musí student splnit, aby mohl stáž absolvovat. Většinou se jedná o vstupní pohovor či psaný test. Student se na stáži musí na základě svých znalostí naučit vykovávat jednotlivé úkoly. Znalosti a dovednosti, které se student na stáži naučil, pak může použít k vypracování své bakalářské nebo diplomové práce. Stáže probíhají většinou v době letních prázdnin a trvají několik měsíců. Zisk zpětné vazby u těchto stáží nebývá systematický.

3.1.4 Individuální stáž

Jedná se o úplně volný stážový model, který nemá určená přesná pravidla. Stáže nejsou začleněny do studia, škola pouze studentům zprostředkuje nabídky firem. Vstupní podmínky závisí na konkrétní firmě. Obsah stáže a souvislost se studiem nejsou pevně stanoveny, závisí to na dané firmě a studentovi. Délka trvání a to, kdy stáže probíhají, je čistě individuální věc. Zpětnou vazbu škola z těchto stáží nezískává.

3.2 Kritické zhodnocení

3.2.1 Kritéria pro hodnocení

V této části jsou shrnuty výsledky analýzy a porovnání identifikovaných modelů. Snažila jsem se navolit taková kritéria, aby poukázala na přínosy

i nedostatky, které jsou dány odlišností jednotlivých modelů. Pro ohodnocení modelů byla použita tato kritéria.

- **Velikost cílové skupiny** udává, jaké množství studentů se stáží účastní.
- **Připravenost stážistů** popisuje, jak jsou studenti na stáž připraveni. Zde se míní teoretická příprava a znalosti získané ve škole.
- **Kvalita stážistů** znamená podíl šikovných stážistů. Při posuzování tohoto kritéria jsem brala v úvahu to, že u studentů, kteří mají stáž povinnou, může být menší motivace a snaha být ve firmě přínosem. Naopak ten, kdo se na stáž přihlásí dobrovolně, bude pravděpodobně vykonávat kvalitnější práci, protože chce být užitečný a chce se něčemu naučit.
- **Pracovní náplň** vyjadřuje, jaká je náplň práce stážistů. Práce by měla být kreativní a rozvíjet schopnosti studenta. Pokud student na stáži vykonává například administrativní práci, která nesouvisí s jeho zaměřením, tak pro něj není stáž přínosná. Pokud budou stáže pojaty příliš masově, tak se může stát, že ve firmách není dostatek odpovídajících pozic a stážisté nebudou dostatečně využiti.
- **Motivace studentů** ukazuje, jak jsou studenti motivovaní k práci, kterou na stáži vykonávají. Toto kritérium odráží to, zda jsou studenti za stáž nějak ohodnoceni, například získají zápočet nebo nějaké kredity.
- **Modifikovatelnost modelu** se projevuje jako schopnost dynamicky reagovat na změny a upravovat stážový model. Typicky to souvisí s tím, jak se v daném modelu využívá zpětná vazba.

3.2.2 Porovnání modelů

Podle uvedených kritérií jsem provedla srovnání stážových modelů. Hlavní přínosy a nedostatky jsou shrnuty v tabulce 3.1.

3. SROVNÁNÍ STÁŽOVÝCH MODELŮ

| Stážový model | Přínosy | Nedostatky |
|-------------------|--|---|
| Povinná stáž | velká cílová skupina připravenost stážistů motivace studentů | kvalita stážistů pracovní náplň modifikovatelnost modelu |
| Volitelná stáž | připravenost stážistů kvalita stážistů motivace studentů modifikovatelnost modelu | menší cílová skupina |
| Dobrovolná stáž | kvalita stážistů pracovní náplň modifikovatelnost modelu | menší cílová skupina motivace studentů |
| Individuální stáž | kvalita stážistů | malá cílová skupina připravenost stážistů motivace studentů modifikovatelnost modelu |

Tabulka 3.1: Kritické zhodnocení stážových modelů

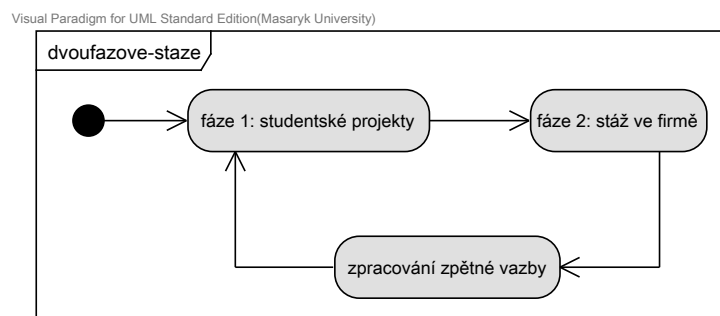
Kapitola 4

Návrh modelu hodnocení stáží

V této části je představen model *dvoufázových stáží*, který se používá na FI MU v Brně v rámci jedné z klíčových aktivit OPVK projektu *Platforma průmyslové spolupráce*. Jedním z hlavních znaků tohoto modelu je využití zpětné vazby. V kapitole jsou analyzovány různé možnosti pro sběr dat. V závěru kapitoly je vypracován model pro hodnocení stáží, jehož cílem je poskytnout zadavatelům rychlý a celkový pohled na úspěšnost stážového modelu.

4.1 Model dvoufázových stáží

Na FI MU se používá model dvoufázových stáží. Základní myšlenka modelu je zachycena na obrázku 4.1. V první fázi student pracuje ve škole na



Obrázek 4.1: Model dvoufázových stáží

projektu, který je zadán externí firmou. Při práci na tomto projektu se student seznámí s používanými technologiemi. Druhá fáze je pak stáž v dané firmě. Tím, že student je již seznámen s technologiemi a nástroji, které se ve firmě používají, je pro firmu mnohem větším přínosem než stážista, kterého je potřeba nejdříve zaškolit. Po skončení stáže poskytne student i firma

zpětnou vazbu na úspěšnost stáže. Na jejím základě je pak možno modifikovat vstupní podmínky na studenty i průběh první fáze stáží.

4.2 Možnosti hodnocení stáží

4.2.1 Potřeba zpětné vazby

Jednou ze základních vlastností představeného modelu je možnost využití zpětné vazby pro modifikaci modelu samotného. Tím, že je model rozdělen na dvě fáze, je možno přizpůsobovat první fázi tak, aby vedla k co nejefektivnějšímu naplnění fáze druhé. Předpokládá se, že stáže budou probíhat opakovaně, proto je potřeba po proběhnutí jedné etapy stáží získat informace o její úspěšnosti. Na základě těchto dat pak bude upraven průběh první fáze tak, aby v příští etapě stáží bylo dosaženo lepších výsledků. Cílem je hodnocení stážového modelu jako takového, ne hodnocení jednotlivých studentů. Otázkou je, jak získat informace o proběhnutých stážích a jak tyto informace vyhodnotit pro získání relevantních výsledků.

4.2.2 Možnosti pro sběr dat

Při sběru dat je důležité si uvědomit tyto hlavní faktory [4]:

- jaký je smysl hodnocení;
- pro koho budou výsledky hodnocení přínosem;
- na jaké otázky se snažíme získat odpovědi;
- jaká forma informací je potřebná pro zodpovězení těchto otázek;
- jaká bude cena (prostředky, čas) za získání odpovědí.

Na základě těchto faktorů je možno stanovit formu pro získání dat. Metody pro získání dat můžeme rozdělit na aktivní a pasivní. Pasivní metody jsou metody, u kterých není přímá interakce s dotazovaným subjektem. Příkladem takovýchto metod je zkoumání již existujících dat, pozorování jednotlivých subjektů či analýza obrazových zdrojů. Hlavní nevýhodou těchto metod je potřeba velkého úsilí ze strany tazajících.

Aktivními metodami rozumíme metody, u kterých je potřeba aby dotazovaný subjekt aktivně poskytl informace tazajícím. Mezi základní metody aktivního sběru dat patří:

- dotazník;

- pohovor;
- případová studie.

Pohovor a případová studie dokáží poskytnout poměrně přesná a konkrétní data ohledně jednotlivých tázaných subjektů. Vedle toho dotazníky slouží spíše pro získání celkového přehledu mezi všemi subjekty. Dotazníky poskytují strukturovaná data, takže vyhodnocování a získání statistických výsledků je jednodušší. Vyhodnotit výsledky pohovoru nebo případové studie je velice časově náročné a díky nejednoznačnosti přirozeného jazyka mohou být získané údaje poněkud nepřesné.

4.2.3 Vhodný prostředek pro hodnocení stáží

Cílem hodnocení stáží je získat celkový pohled na úspěšnost stáží v jednotlivých firmách. Je snaha získat přehled o tom, ve kterých oblastech jsou stáže problematické a jak by se mohla zlepšit příprava studentů na stáž. Je důležitější získat obecný přehled o stavu stáží v jednotlivých firmách než detailní informace o konkrétních studentech. Zisk hodnocení by pro dotázané neměl být časově náročný a neměl by je příliš obtěžovat. Nasbírané údaje by mělo být možno jednoduše sumarizovat a zjistit statistiky úspěšností stáží v jednotlivých firmách.

Z těchto důvodů se jako nejvhodnější prostředek pro sběr informací jeví dotazník. Dotazníkem by měli dotázaní hodnotit úspěšnost jednotlivých aspektů stáží. Je potřeba sbírat odpovědi jak od studentů, tak od zástupců firem, ve kterých budou studenti na stáži.

4.3 Návrh modelu hodnocení

Běžný dotazník slouží pouze ke sběru odpovědí od jedné osoby. U stáží však dochází k interakci mezi studentem a firmou (která je zastoupena styčnou osobou), což je nutné při hodnocení brát v úvahu. Pro efektivní hodnocení stáží je potřeba sbírat data od zástupců firem i od stážistů. Na základě souvislostí v hodnocení obou stran pak bude možno stanovit výsledky. Cílem je, aby hodnocení bylo objektivní. Je potřeba eliminovat příliš subjektivní názory, které mají malou vypovídající hodnotu, ale mohou silně zkreslovat statistiku.

Pro hodnocení stáží navrhuji tento model:

- stanoví se aspekty stáží, které budou hodnotit zástupce společnosti i student;

- pro každý aspekt se zadají kvantitativní otázky pro oba dotázané;
- v případě, že je rozdíl odpovědí větší než stanovená prahová hodnota, nebude odpověď započítána do statistiky;
- pokud je rozdíl menší než daný práh, určí se průměr z odpovědí a tato hodnota se připočítá k celkové hodnotě aspektu;
- hodnoty aspektů se evidují zvlášť pro různé organizace, aby pak bylo možno srovnávat úspěšnosti stáží;
- výsledkem hodnocení bude srovnání jednotlivých aspektů mezi sebou, podle čehož je možno určit, na který aspekt je potřeba se více zaměřit při přípravě studentů.

4.4 Dostupná řešení pro sběr hodnocení

4.4.1 Základní řešení

Pro vytváření on-line průzkumů a dotazníků existuje spousta nástrojů. Mezi nejpoužívanější patří například Survey Monkey¹ nebo Questionform². Jedná se o nástroje pro vytváření různých typů dotazníků. Tyto nástroje jsou v základní verzi zdarma, v rozšířenějších verzích se jedná o placené služby. Základní verze bývá zpravidla omezená počtem dotazníků a odpovědí a neumožňuje pokročilé vlastnosti, jako například dynamickou navigaci v dotazníku nebo zabezpečení připojení.

Běžně používanou a jednoduchou na ovládání je možnost vytvoření dotazníku v rámci služby Google Docs.³ Uživatel si navolí otázky, které chce v dotazníku mít, rozešle e-mailem odkaz na dotazník a odpovědi pak může sledovat v tabulce či jako grafické shrnutí. Hlavní nevýhodou je nemožnost kontroly toho, kdo vlastně na dotazník odpovídá. Dotazník se zobrazí po kliknutí na odkaz komukoliv a jeden uživatel tak může klidně odpovídat i vícekrát. Toto je při větších průzkumech poměrně závažné omezení, neboť může dojít ke znehodnocení získaných dat.

4.4.2 Rozsáhlejší aplikace

Příkladem rozsáhlejšího a placeného systému pro sběr dat je služba WorldAPP KeySurvey.⁴ Jedná se o robustní aplikaci umožňující vytvářet vy-

1. <http://www.surveymonkey.com>

2. <http://questionform.com>

3. <http://docs.google.com>

4. <http://www.keysurvey.com>

soce konfigurovatelné webové průzkumy a analyzovat získaná data. Systém mimo jiné umožňuje sledování IP adres, náhodné generování otázek, dynamicky generované otázky na základě předešlých odpovědí, variabilní vzhled dotazníku podle typu zařízení, omezení přístupu nebo vytváření průzkumů pro mobilní zařízení. Systém také poskytuje API, tudíž je možné jej integrovat do jiných systémů.

4.4.3 Shrnutí a porovnání

Cílem hodnocení stáží je získat přehled o celkovém stavu stáží, proto je dostačující, aby dotazník umožňoval definovat jednoduché kvantitativní otázky. Není potřeba logika pro navigaci dotazníkem či jiné pokročilé vlastnosti, je však nutné, aby hlasování mělo omezený přístup a bylo možno dohledat, kdy a kdo již dotazník odpověděl.

Všechna existující řešení jsou zaměřena na jednu cílovou osobu. U hodnocení stáží je však potřeba sbírat zpětnou vazbu od stážistů i zástupců firem a na základě podobnosti dat pak provádět vyhodnocení. Tuto logiku žádné z nabízených řešení neimplementuje, a tudíž by použití nějakého existujícího systému vyžadovalo netriviální ruční zpracování výsledků.

Kapitola 5

Analýza a návrh systému pro hodnocení stáží

Praktickým přínosem práce bylo navržení a implementace systému pro hodnocení stáží. Tato kapitola uvádí požadavky na systém a na jejich základě navrhuje případy užití navrženého systému. Je představen datový model aplikace. Hlavní důraz je kladen na jádro systému, čili na zaznamenávání a výpočty statistických údajů. Aplikace je namodelována v jazyku UML 2.1 [5].

5.1 Požadavky na systém

Cílem práce bylo vytvořit aplikaci umožňující sběr a vyhodnocování oboustranné zpětné vazby. Student a zástupce společnosti budou hodnotit zadané aspekty. Je potřeba, aby zodpovězení dotazníků bylo co nejméně časově náročné. Ze získaných dat pak bude možno generovat přehledné statistiky, ze kterých by mělo být možno rychle získat údaje o tom, na co dát důraz při přípravě studentů na stáž.

System bude umožňovat:

- evidenci studentů;
- evidenci firem a jejich zástupců;
- evidenci aspektů, které se budou hodnotit;
- shlukování aspektů do skupin;
- evidenci hodnotících období;
- generování dvojic dotazníků pro studenta a zástupce firmy;
- rozeslání e-mailů s žádostí o vyplnění dotazníků;
- vyplňování dotazníků bez potřeby přihlášení se do aplikace;
- zobrazení statistiky hodnot jednotlivých aspektů vztahujících se k danému období a pro danou společnost;

- zobrazení globální statistiky ze všech období a společností;
- zobrazení statistiky pro skupiny aspektů (agregující data všech aspektů ve skupině);
- zobrazení informace o tom, kolik úsilí by se mělo věnovat zlepšení nejkritičtějších aspektů;
- export statistiky ve formátu PDF (*Portable Document Format*);
- export naměřených dat ve formátu XLS (*Microsoft Excel Spreadsheet*) tak, aby se dala případně dále zpracovávat;
- import seznamů studentů, firem a zástupců z textového souboru;
- hromadné generování dotazníků i hromadné rozeslání e-mailů;
- správu uživatelů přes administrační rozhraní.

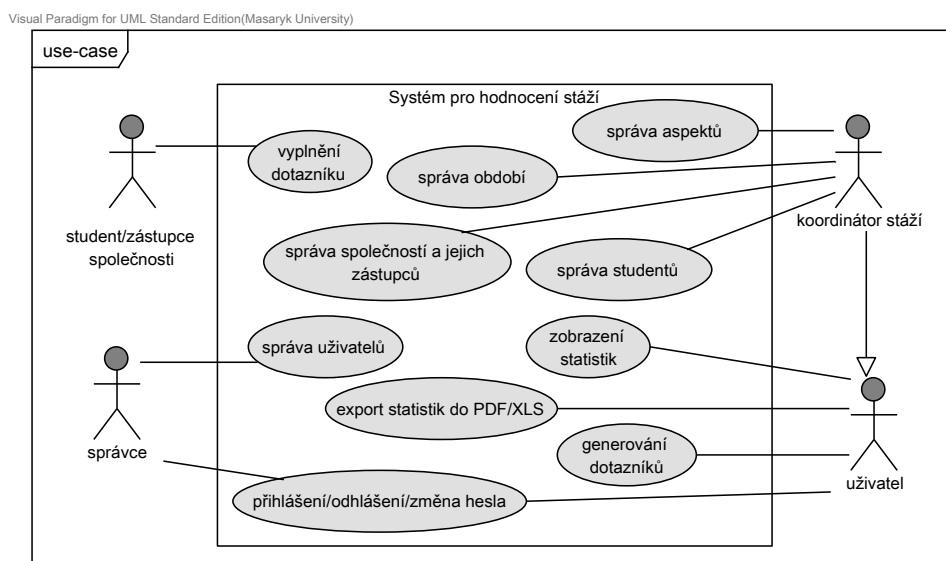
5.2 Případy užití

Analýzou požadavků jsem v systému identifikovala čtyři uživatelské role:

- **Student/zástupce společnosti** - tento uživatel má možnost pouze odpovídat dotazníky. Do systému se nepřihlašuje, zobrazení dotazníku bude pouze na základě speciálního odkazu.
- **Běžný uživatel** - tento uživatel patří k týmu koordinátorů stáží. Uživatel se do systému musí přihlásit. Po přihlášení může generovat dotazníky a rozesílat žádosti pro jejich vyplnění. Může si prohlížet statistiky a případně generovat zprávy ve formátu PDF a XLS.
- **Koordinátor stáží** - tato role má stejná práva jako běžný uživatel, navíc může zadávat a upravovat data, která slouží jako podklady pro dotazníky. Definuje aspekty a hodnotící období, zadává nebo importuje studenty, společnosti a jejich zástupce. Všechny spravované subjekty může také upravovat či mazat.
- **Správce aplikace** - spravuje uživatele a definuje uživatelské role. Na rozdíl od ostatních uživatelů nemá vůbec možnost pracovat s daty týkající se hodnocení stáží.

Tyto uživatelské role a případy užití, které mohou provádět, jsou znázorněny na obrázku 5.1.

5. ANALÝZA A NÁVRH SYSTÉMU PRO HODNOCENÍ STÁŽÍ



Obrázek 5.1: Diagram případů užití

5.3 Datový model jádra systému

5.3.1 Aspekty

Základní jednotkou, pro kterou se zaznamenává hodnocení, je aspekt (Feature). Aspekt představuje určitou malou oblast, která se bude hodnotit. Každý aspekt má jméno, zkrácené jméno, otázku pro studenta, otázku pro zástupce společnosti a *náznak k řešení*. Otázky jsou kvantitativní a ptají se na stejnou věc, pouze jsou jinak formulované. Náznakem k řešení se rozumí konkrétní rada, kterou by měli koordinátoři podniknout, aby došlo ke zlepšení úrovně aspektu.

Aspekty je možno shlukovat do skupin tak, že se pro daný aspekt definuje jeho rodič. Z pohledu datového modelu je možno, aby tato hierarchie byla i vícestupňová.

5.3.2 Účastníci, společnosti a období

Hodnocení mohou ukládat studenti (Student) a zástupci společností (Manager). Jelikož některé vlastnosti mají tyto entity společné, je zavedena nadtřída Participant, aby nedocházelo k duplikaci kódu. Také je pak

možno mít jednu třídu pro manipulaci s těmito entitami. Kromě společných vlastností má student navíc UČO (univerzitní číslo osoby) a zástupce společnosti odkaz na společnost, ke které přísluší. E-mail každého účastníka a UČO studenta musí být unikátní.

Společnost (`Organization`) má název a unikátní IČO (identifikační číslo organizace).

Další třídou, která je pro ukládání informací důležitá, je období (`Period`). Období obsahuje rok, název a poznámku a představuje jednu etapu stáží, pro kterou se hodnocení ukládá. Při vzniku období se generuje časová známka, aby bylo možno období chronologicky řadit podle data přidání.

5.3.3 Zaznamenávání hodnocení

Pro ukládání hodnocení aspektu slouží třída `FeatureRate`. Hodnocení je vždy vázáno na období, ve kterém byla dvojice dotazníků vytvořena a také na společnost, ve které hodnocená stáž proběhla. Ukládá se počet hlasů a součet jednotlivých hodnot hlasování.

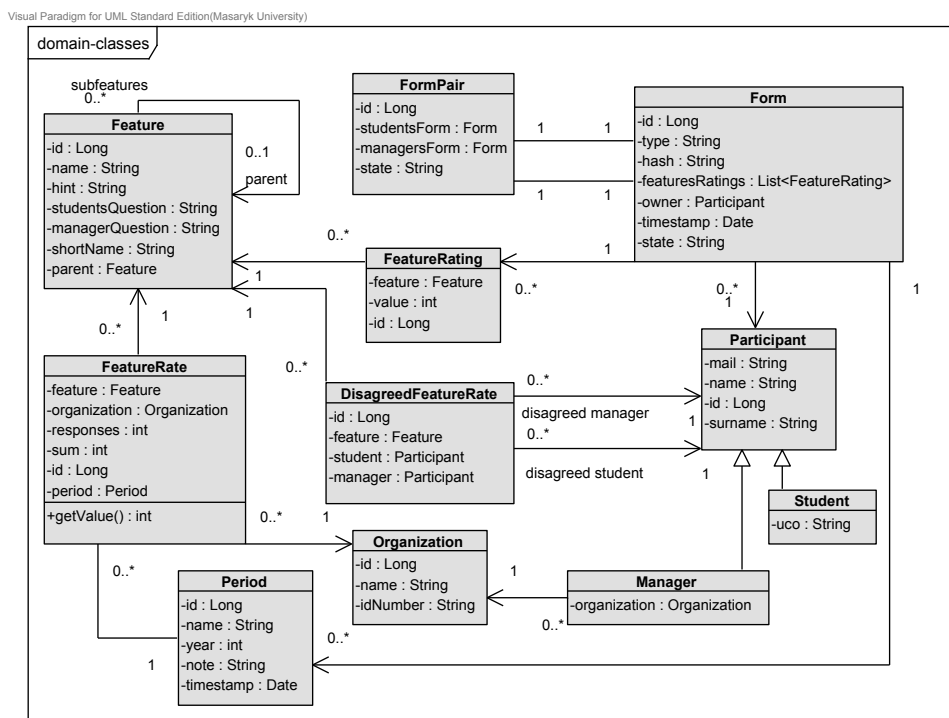
Třída `FeatureRating` slouží pro uložení hlasu jedné osoby pro konkrétní aspekt. V dotazníku (`Form`) je seznam objektů této třídy. Dotazník je vždy vázán na osobu, která jej odpovídá. Dále v sobě obsahuje stav, časovou známku a jednoznačný identifikační řetězec, který se používá k vytvoření unikátní adresy pro zobrazení dotazníku. Stav dotazníku může být nový, otevřený a uložený. Časová známka pak podle daného stavu indikuje čas vytvoření, otevření nebo uložení dotazníku.

Dva dotazníky vždy tvoří dvojici (`FormPair`), kde jeden patří studentovi a druhý zástupci společnosti. Stav dvojice dotazníků ukazuje, zda je dvojice nová (nikdo zatím neodpověděl), odpovězená studentem nebo zástupcem společnosti či vyplněná (odpovězená oběma).

5.3.4 Sporná hodnocení

Jak již bylo uvedeno výše, model hodnocení je postaven na tom, že dva lidé hodnotí daný aspekt, a pokud jsou jejich názory rozdílnější než stanovený práh, tak se do výsledků nezapočítávají. Pro ukládání těchto hodnot slouží třída `DisagreedFeatureRate`. Obsahuje odkazy na oba účastníky a aspekt, v jehož hodnocení se neshodli. Díky této třídě lze zjistit, kolikrát se daný účastník neshodl se svým protějškem. Dá se jednoduše získat i počet sporných hodnocení pro daný aspekt, což může znamenat například to, že otázka je špatně položená.

Diagram doménových tříd aplikace je znázorněn na obrázku 5.2.



Obrázek 5.2: Diagram doménových tříd

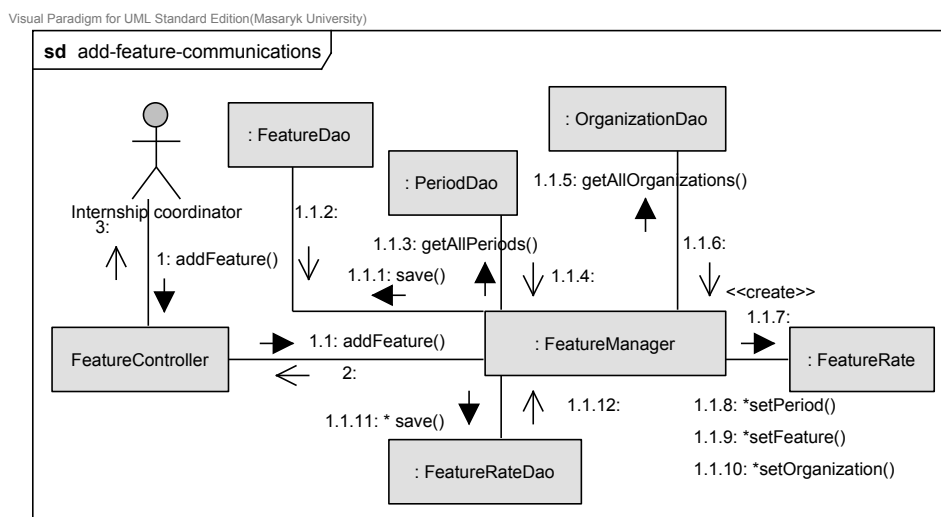
5.4 Rozpracování případů užití

Navržený systém bude webová aplikace, která dodržuje přístup model-pohled-řadič (*Model-View-Controller*). Uživatelské požadavky jsou zpracovávány řadičem, který získá data z modelu (třídy, které poskytují aplikační logiku) a pomocí webové stránky (pohled) je zobrazí. Přístup k datům zabezpečují třídy, které jsou vytvořeny podle návrhového vzoru DAO (*Data Access Object*) [1]. Každý DAO objekt poskytuje metody pro základní manipulaci s daty jedné entity. Typicky se jedná o přidání, smazání, modifikaci a vyhledávání. Nad DAO vrstvou je postavená vrstva služeb, která tvoří rozhraní aplikační logiky.

Následuje rozpracování zajímavých případů užití, ve kterém je vidět, jak jednotlivé vrstvy aplikace komunikují mezi sebou.

5.4.1 Přidání aspektu

Při přidávání aspektu je kromě uložení entity potřeba vytvořit a uložit objekty pro ukládání hodnocení (`FeatureRate`), a to pro všechny existující společnosti a pro každé období. Tato činnost je zachycena na komunikačním diagramu 5.3.



Obrázek 5.3: Komunikační diagram přidání aspektu

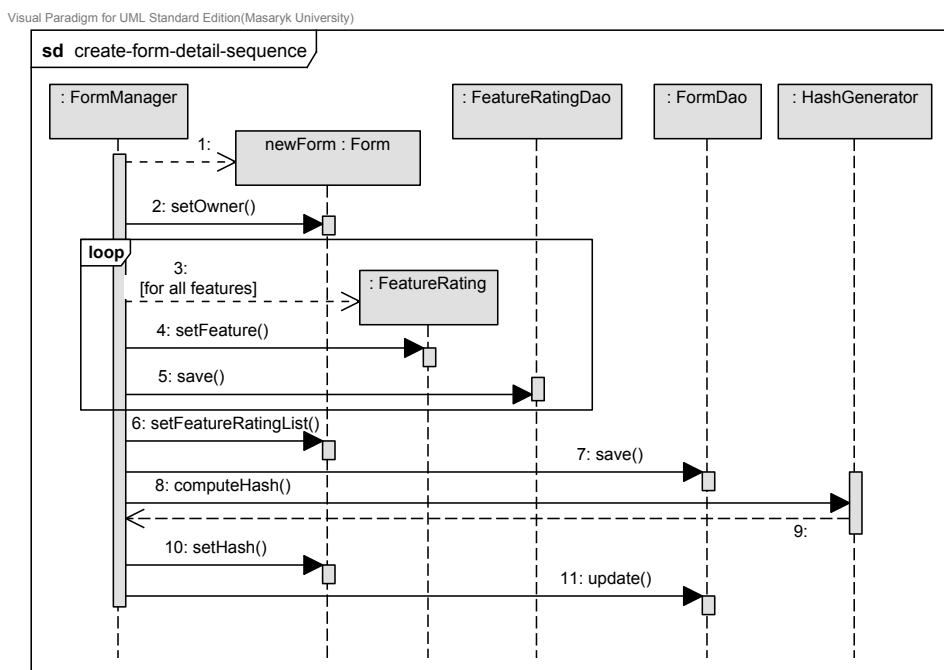
Pokud se do systému přidává společnost nebo období, tak je potřeba provést podobné akce, aby byla zajištěna existence objektů pro ukládání jakéhokoliv hodnocení.

5.4.2 Generování dvojice dotazníků

Při vytváření dvojice dotazníků si uživatel zvolí studenta a zástupce společnosti, pro které se mají dotazníky vytvořit. Také si vybere seznam aspektů, jejichž otázky mají být do dotazníku zahrnuty. Nejdříve proběhne vytvoření jednotlivých dotazníků, které je zachyceno na diagramu 5.4. Pro každý aspekt je potřeba vytvořit a uložit objekt `FeatureRating`, který slouží pro uložení konkrétní hodnoty aspektu. Poté, co se dotazník uloží a vygeneruje se mu primární klíč, je možno určit jeho jednoznačný řetězcový identifikátor, který slouží pro adresování dotazníku.

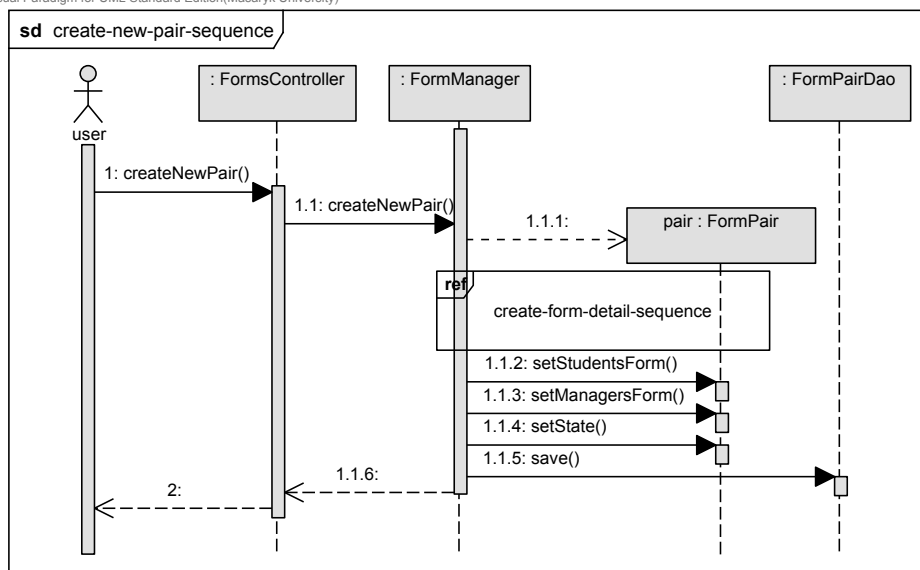
Po vytvoření obou dotazníků se nastaví stav dvojice dotazníků jako nový a dvojice se uloží do databáze, což je znázorněno na obrázku 5.5.

5. ANALÝZA A NÁVRH SYSTÉMU PRO HODNOCENÍ STÁŽÍ



Obrázek 5.4: Sekvenční diagram vytvoření dotazníku

Visual Paradigm for UML Standard Edition(Masaryk University)



Obrázek 5.5: Sekvenční diagram vytvoření nové dvojice dotazníků

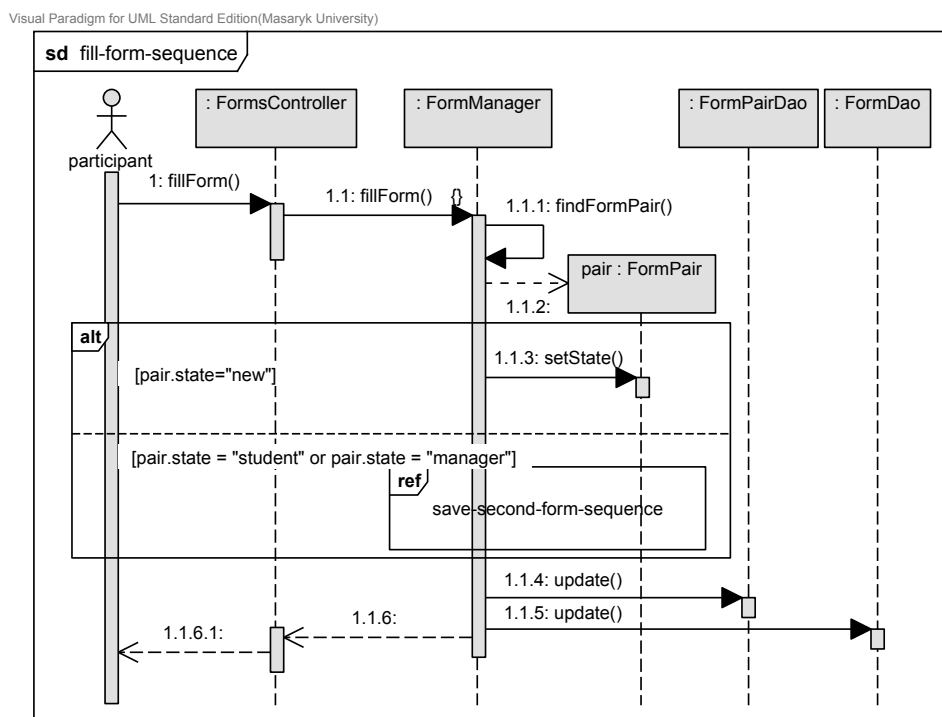
5.4.3 Ukládání hodnocení

Při ukládání dotazníku je nejdříve potřeba najít dvojici, do které dotazník patří. Pokud druhý dotazník z dvojice ještě nebyl vyplněn, první dotazník se pouze uloží do databáze. Toto chování je znázorněno na diagramu 5.6.

V případě, že ukládaný dotazník je již druhý z dvojice, dochází k porovnání hodnocení. Pokud je rozdíl odpovědí menší než zadaný práh, určí se průměrná hodnota, která se přidá k hodnotě daného aspektu (objekt `FeatureRate`). V opačném případě se uloží informace o tom, kdo s kým se v hodnocení daného aspektu neshodl. Podrobnosti jsou zakresleny na obrázku 5.7.

Entita dotazník v sobě obsahuje také položky stav a časová známka. Časová známka se generuje vždy při změně stavu dotazníku. Při vytvoření dotazníku je jeho stav nový a časová známka obsahuje čas vytvoření. V okamžiku, kdy si uživatel dotazník zobrazí, se stav změní na otevřený a nastaví se čas, kdy byl dotazník otevřen. Uživateli je dovoleno si dotazník zobrazovat do té doby, než uplyne určený časový limit. Po jeho uplynutí už nebude možno s dotazníkem pracovat. Pokud uživatel dotazník uloží, nastaví se jeho stav na uložený a také se zaznamená čas uložení. Uložený dotazník není možno znovu otevírat. Pokud by uživateli vypršela doba pro

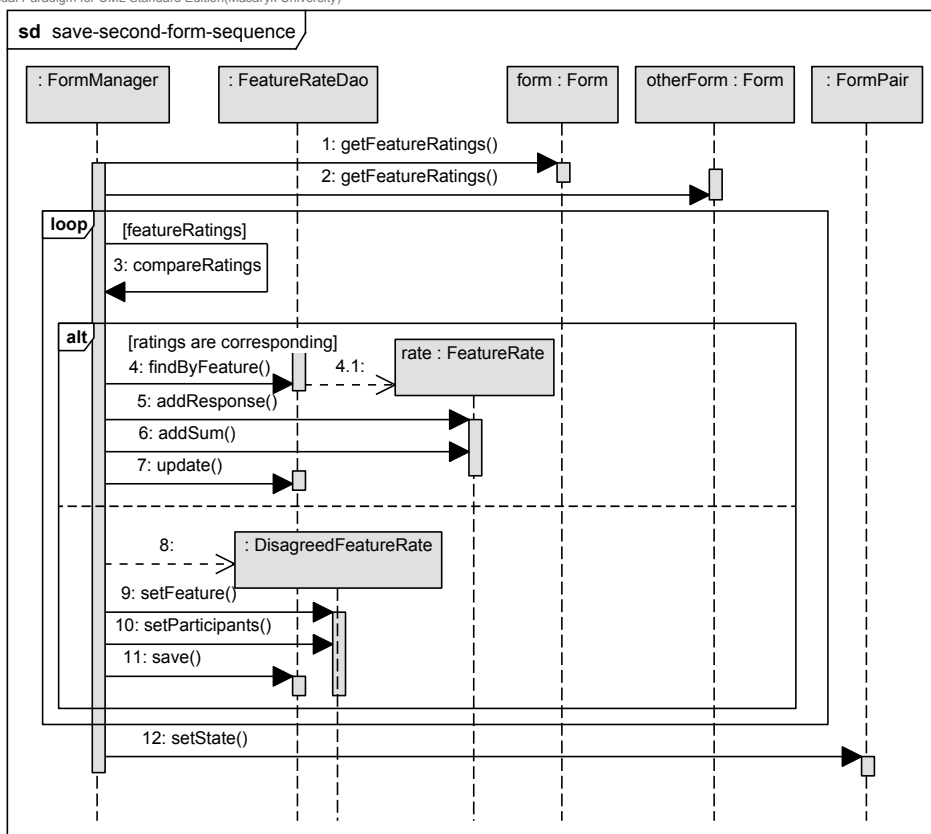
5. ANALÝZA A NÁVRH SYSTÉMU PRO HODNOCENÍ STÁŽÍ



Obrázek 5.6: Sekvenční diagram ukládání dotazníku

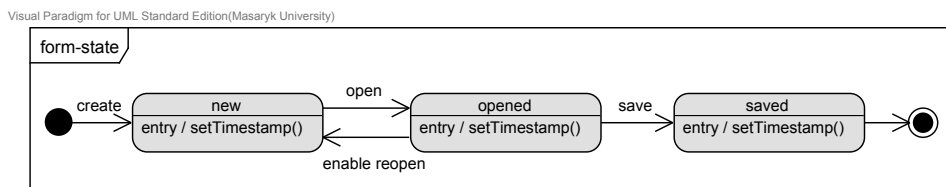
5. ANALÝZA A NÁVRH SYSTÉMU PRO HODNOCENÍ STÁŽÍ

Visual Paradigm for UML Standard Edition(Masaryk University)



Obrázek 5.7: Sekvenční diagram uložení druhého dotazníku z páru

zodpovězení dotazníku, je možno požádat o jeho znovuootevření. Přejchody mezi jednotlivými stavy jsou znázorněny na obrázku 5.8.



Obrázek 5.8: Stavový diagram dotazníku

5.4.4 Výpočet statistických údajů

Statistiky jednoduchých aspektů

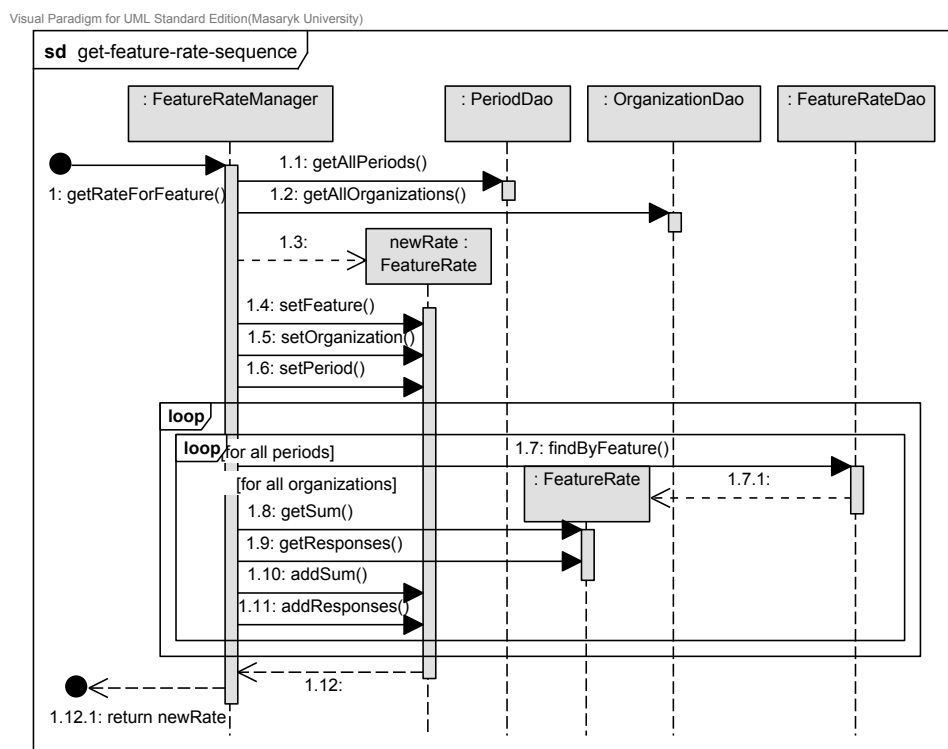
Statistika hodnocení aspektů sestává ze dvou odlišných částí. Zobrazují se přímo naměřené hodnoty aspektů tak, aby je bylo možno porovnat mezi sebou. Kromě toho se uživateli také zobrazí kritické aspekty a u každého z nich je odhad, jakou část úsilí by měl věnovat jeho zlepšení. Za kritické jsou považovány ty aspekty, jejichž hodnota je menší než průměr. Množinu těchto aspektů označme A_k . Proměnná $\overline{hodnota}$ vyjadřuje aritmetický průměr z hodnot aspektů. Potřebné úsilí $u(a)$ pro aspekt a se určí podle vzorce 5.1.

$$u(a) = \frac{\overline{hodnota} - hodnota(a)}{\sum_{b \in A_k} (\overline{hodnota} - hodnota(b))} \quad (5.1)$$

Při výpočtu statistik si uživatel zvolí, pro jaké období a pro kterou společnost se statistika vypisuje. Také je možno si nechat vypsat globální statistiky ze všech společností i přes všechna období. Metody potřebné pro výpočet statistik jsou sdruženy v rozhraní `FeatureRateManager`. Toto rozhraní obsahuje metody pro zjištění hodnoty aspektu, zjištění všech hodnot aspektů, výpočet průměru a výpočet celkového rozpětí zlepšení (*total improvement count*), což je hodnota ve jmenovateli výrazu 5.1.

Hodnoty aspektů (objekty `FeatureRate`) v sobě obsahují období a společnost, pro které jsou zaznamenány. V případě, že se počítá hodnota ze všech období nebo společností, je nejdříve potřeba hodnoty správně nakumulovat. Průběh výpočtu hodnoty aspektu ze všech období a ze všech společností je znázorněn na diagramu 5.9.

5. ANALÝZA A NÁVRH SYSTÉMU PRO HODNOCENÍ STÁŽÍ



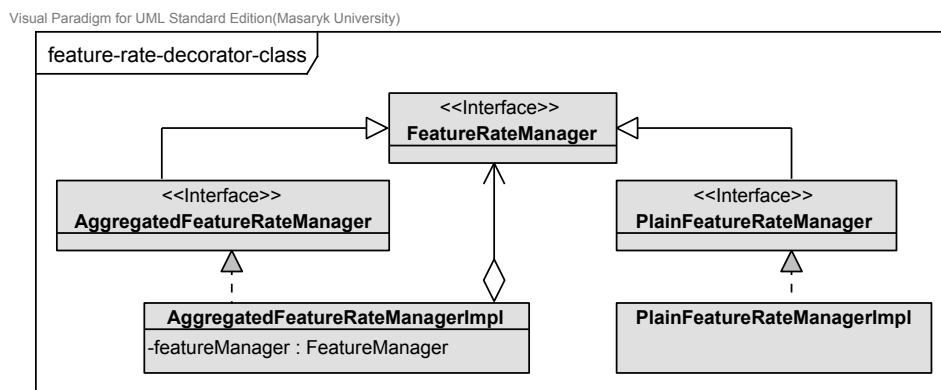
Obrázek 5.9: Sekvenční diagram získání kumulované hodnoty aspektu

Statistiky složených aspektů

Jedním z požadavků na systém bylo, aby se aspekt mohl skládat z více *podaspektů*. Ve výpočtu statistických údajů se s těmito podaspekty může pracovat jako s plnohodnotnými aspekty, nebo se mohou jejich hodnoty agregovat do hodnoty rodičovského aspektu. Z tohoto důvodu existují dvě implementace rozhraní `FeatureRateManager`.

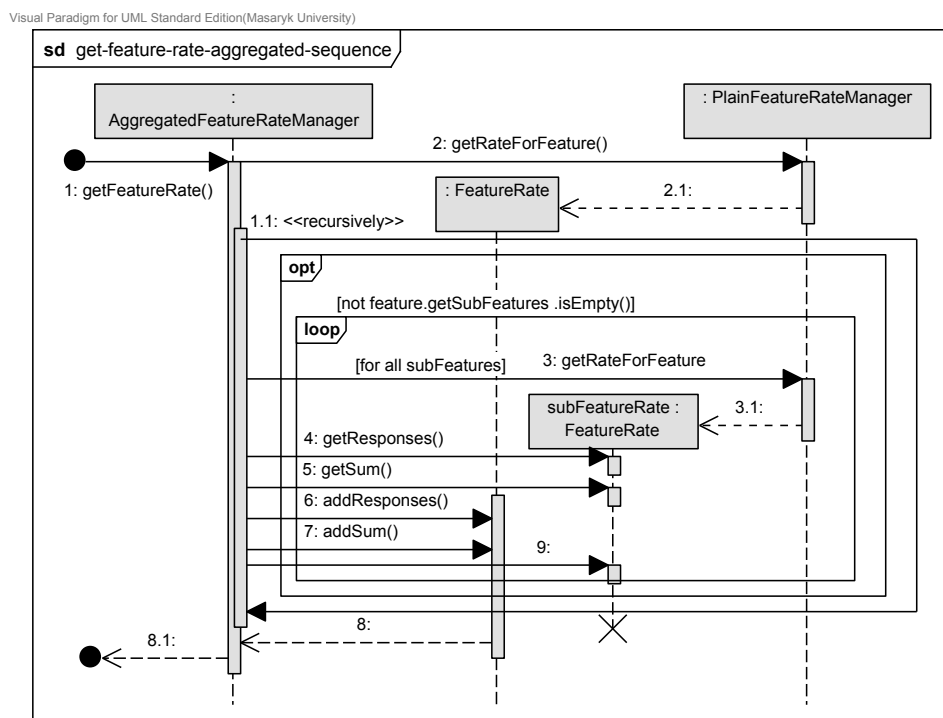
Třída `PlainFeatureRateManagerImpl` implementuje základní logiku získávání hodnot jednotlivých aspektů.

Druhá implementace, třída `AggregatedFeatureRateManagerImpl`, poskytuje hodnoty pro agregované aspekty. Při agregaci je nejdříve potřeba získat hodnoty jednotlivých aspektů, proto je tato třída vytvořena podle návrhového vzoru *dekorátor* [6], což je zakresleno na diagramu 5.10.



Obrázek 5.10: Použití návrhového vzoru *dekorátor* pro agregaci hodnot aspektů

Průběh metody, která počítá hodnotu agregovaného aspektu je znázorněn na obrázku 5.11. K hodnotě aspektu se připočítávají hodnoty všech podaspektů. Jelikož hierarchie může být víceúrovňová, je jí potřeba celou rekurzivně projít a připočítat všechny hodnoty.



Obrázek 5.11: Sekvenční diagram získání agregované hodnoty aspektu a všech jeho potomků

Kapitola 6

Implementace systému

V této kapitole je popsána implementace a testování systému. Jsou zmíněny technologie, které jsem použila pro vývoj aplikace. Je popsán vzhled uživatelského rozhraní. V závěru kapitoly je popsáno, jak jsem se snažila zajistit kvalitu aplikace a vytvořit tak stabilní a robustní systém.

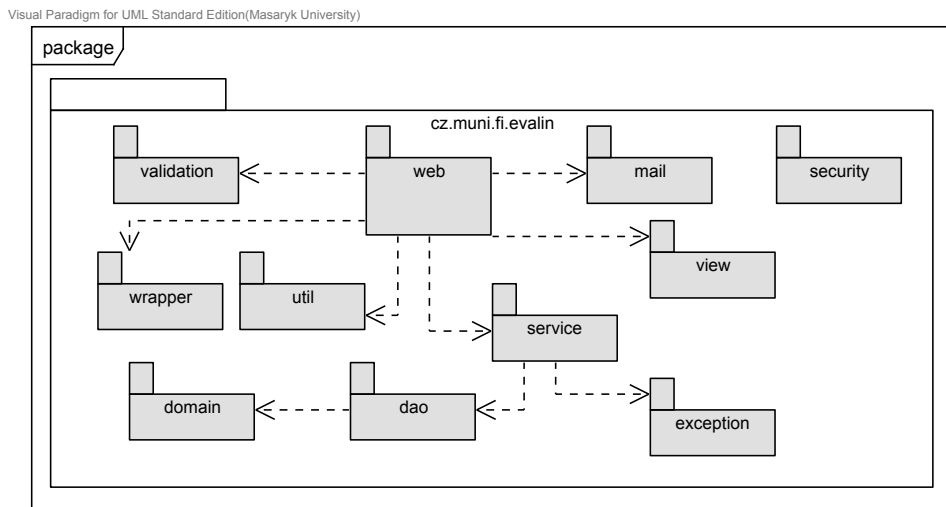
6.1 Struktura implementace

Zdrojové kódy aplikace jsou v balíku `cz.muni.fi.eval.in`, kde jsou dále rozděleny do několika dílčích balíků. Doménové třídy (entity) jsou v balíku `domain`. K nim přistupují DAO objekty, které jsou součástí balíku `dao`. Rozhraní aplikační logiky pak poskytují služby, které jsou v balíku `service` a pomocí DAO objektů manipulují s daty. Vstupním bodem do aplikace jsou třídy v balíku `web`, které fungují jako řadič (*controller*). Tyto třídy přistupují ke službám a k dalším pomocným balíkům. V balíku `view` jsou třídy, které vykreslují výstup aplikace v jiných formátech než HTML. Balík `mail` sdružuje třídy, které slouží pro ustavení spojení a posílání e-mailů. Ve webové aplikaci se pro validaci formulářů používají validátory z balíku `validation`. Pomocné třídy v balíku `util` slouží pro konverzi datových typů. Třídy z balíku `wrapper` představují obálky pro ukládání dočasných informací a předávání dat mezi různými součástmi aplikace. Samostatným balíkem je `security`, kde jsou třídy, které se používají pro zajištění bezpečnosti aplikace. Struktura všech balíků a jejich závislosti jsou znázorněny na obrázku 6.1.

6.2 Použité technologie

V této části jsou popsány klíčové technologie, které jsem použila pro implementaci aplikace. Aplikace se sestavuje pomocí nástroje Maven¹. V konfiguračním souboru `pom.xml` se nadefinují závislosti projektu a používané

1. <http://maven.apache.org>



Obrázek 6.1: Diagram balíků aplikace

knihovny, a Maven pak vše potřebné stáhne a aplikaci sestaví. Také poskytuje podporu pro spouštění jednotkových i integračních testů, spouštění i nasazování aplikace.

6.2.1 Spring

Pro implementaci aplikace jsem použila aplikační rámec Spring [2], který umožňuje obrácené řízení (*inversion of control*), což znamená, že závislosti jednotlivých objektů se určují až za běhu. Díky mechanismu vkládání závislostí (*dependency injection*) je zajištěno, že potřebné objekty vznikají až dynamicky za běhu a programátor se tak nemusí starat o jejich vytvoření. Vytváření závislostí se řídí anotacemi `@Autowired` v kódu. Díky těmto vlastnostem je práce s aplikačním rámcem Spring přímočará a programátorsky přívětivá.

6.2.2 Java Persistence API

Pro přístup k datům jsem použila objektově relační mapování pomocí Java Persistence API. Jedná se o standard, který je definován v rámci specifikace Enterprise Java Beans 3.0 - JSR 220 [7], běžně se však používá i samostatně. Programátor definuje datový model a vazby mezi entitami pomocí anotací.

Jako implementaci JPA jsem použila nástroj Hibernate². Jedná se o jednu z nejpoužívanějších a nejspolehlivějších implementací JPA, mezi další patří například TopLink Essentials³ nebo OpenJPA⁴.

Hibernate zajišťuje vytváření databázového schématu z definovaného datového modelu a programátor se tak vůbec nemusí starat o vytváření tabulek ani zajištění referenční integrity. Pro dotazování se používá *Java Persistence Query Language*. V dotaze se specifikuje, jaké objekty a s jakými vlastnostmi chce programátor získat. Názvy databázových tabulek se v dotazech vůbec nevyskytují. JPA je nezávislé na poskytovateli databáze, používaná databáze se pouze nastaví v konfiguračních souborech.

6.2.3 Spring Web MVC

Pro vytváření webových aplikací se používají různé aplikační rámce. Mezi jejich nejdůležitější vlastnosti patří mechanismus pro řízení toku stránek, svázání formulářů s datovými objekty, validace formulářů, nastavení mapování URL či jednotný způsob zpracování chybových hlášek. Příkladem takovýchto rámců jsou Java Server Faces⁵, Apache Struts⁶ nebo Spring Web MVC, což je součást aplikačního rámce Spring. Ve své aplikaci jsem se rozhodla použít právě Spring Web MVC, protože se velice dobře integruje do aplikace, kde se Spring používá pro vkládání závislostí.

Mezi hlavní přínosy použití tohoto rámce patří:

- Mapování URL. Nastavení toho, na jakém URL bude daná stránka, se provádí přímo v kódu řadiče pomocí anotací. Je také možné, aby součástí adresy byla proměnná, což se dá využít například pro zobrazení detailu o objektu s daným ID.
- Řízení toku stránek a předávání atributů. To, která stránka se zobrazí při dané akci, je určeno návratovou hodnotou metody řadiče. Metoda vrací název JSP stránky, která se má vykreslit, případně může vracet adresu, na kterou se má přesměrovat řízení. V případě, že řadič předává pohledu nějaká data, se dané metodě předává model, do něž se potřebné objekty nastaví. Když potřebuje metoda v řadiči zpracovávat nějaké informace zadané uživatelem, objeví se tato data jako atribut modelu v parametru metody. Toto chování se konfiguruje také pomocí anotací.

2. <http://hibernate.org>

3. <http://oss.oracle.com/toplink-essentials-jpa.html>

4. <http://openjpa.apache.org>

5. <http://www.javaserverfaces.org>

6. <http://struts.apache.org>

- Práce s formuláři. Formulář je vždy svázán s atributem modelu. Při uložení formuláře je tento atribut předán metodě řadiče. Uživatel si definuje validátory, které kontrolují stav jednotlivých položek daného objektu. Pokud některá položka nespĺňuje podmínky, je validátorem odmítnuta a uživateli se ve formuláři u dané položky zobrazí příslušná chybová zpráva. Pokud chceme, aby ve formuláři byla předvyplněná data, předáme potřebný objekt jako atribut modelu. V případě, že jsou položky objektu další objekty, je pro ně potřeba definovat *property editor*, který určuje, jak se mají tyto objekty zobrazovat.

6.2.4 iText

Pro exportování statistik ve formátu PDF jsem použila knihovnu iText⁷. Pomocí této knihovny se vytvoří PDF dokument, do kterého se dynamicky generují aktuální informace o hodnotách aspektů včetně grafického shrnutí. Pro vykreslení dokumentu je implementována třída, která slouží jako pohled a do níž řadič předává data. Výsledkem zobrazení tohoto pohledu je to, že se uživateli v prohlížeči vykreslí PDF dokument s aktuálními statistikami. Příklady exportovaných dokumentů jsou v příloze B.

6.2.5 JFreeChart

K vykreslování grafů, které se pak nastavují do zpráv v PDF je použita knihovna JFreeChart⁸. Vykresluje se sloupcový graf s absolutními hodnotami aspektů a výsečový graf s rozložením úsilí pro zlepšení kritických aspektů. U každého grafu je legenda, u které jsou napsány přesné hodnoty aspektů v procentech.

6.2.6 Apache POI

Kromě zpráv v PDF umožňuje aplikace i export zdrojových dat ve formátu XLS. Pro vytváření těchto dokumentů jsou použity části knihovny Apache POI,⁹ která poskytuje funkce pro manipulaci s dokumenty ve formátech společnosti Microsoft. Třída, která generuje XLS dokument, je implementována jako další pohled aplikace, do nějž jsou řadičem poskytována data. Exportovaný dokument obsahuje tři listy. Na prvním jsou odpovědi studentů, na druhém jsou odpovědi zástupců společností a na třetím jsou

7. <http://itextpdf.com>

8. <http://www.jfree.org/jfreechart>

9. <http://poi.apache.org>

vzorce, které dopočítávají průměrné hodnoty. Při výpočtu průměru se zohledňuje práh pro vyhodnocení shody odpovědí. Výsledky hodnot aspektů v tomto dokumentu odpovídají interně počítaným hodnotám, které se uživateli zobrazují na webu. Pro přehlednost je do XLS dokumentu přidáváno formátování a barevné podbarvení důležitých buněk.

6.3 Uživatelské rozhraní

6.3.1 Vzhled aplikace

Při návrhu uživatelského rozhraní jsem se snažila klást důraz na přehlednost a jednoduchost. Po přihlášení do aplikace se uživateli zobrazí dlaždicové menu, jehož položky závisí na tom, jaké role má uživatel přiděleny. Běžnému uživateli se zobrazují položky Dotazníky a Statistiky. Koordinátorovi se mimoto zobrazují odkazy Studenti, Společnosti a zástupci, Aspekty a Období. Správci aplikace se zobrazují položky Období a Správa uživatelů. V záhlaví aplikace jsou odkazy na hlavní menu aplikace a o úroveň výš. Také se zobrazuje zvolené období, se kterým uživatel zrovna pracuje. Ve spodní části aplikace je odkaz na úpravu uživatelského účtu a odhlášení ze systému.

Agendy pro správu uživatelů, aspektů, dotazníků, studentů, společností a zástupců a období jsou podobné. Vždy se zobrazí tabulka s výpisem všech položek a odkaz pro přidání a případně hromadný import dat. V tabulce je odkaz na zobrazení detailu, editaci a smazání daného záznamu. U všech entit platí pravidlo, že pokud je již položka vázaná na zaznamenané hodnocení, tak ji už není možné smazat. Provádí-li uživatel nějakou nevratnou akci, například smazání položky, je ještě vyzván dialogem k potvrzení.

Ukázka vzhledu hlavních částí aplikace je obsahem přílohy A.

6.3.2 Dotazníky

Při vytváření dotazníků si z nabídky uživatel vybere studenta a zástupce společnosti, pro něž se mají dotazníky vytvořit. U nabízených aspektů si zaškrtně ty, které chce do dotazníku zahrnout. Pokud jsou aspekty členěny na rodiče a podaspekty, je toto vizuálně rozlišeno. Pokud je vybrán nějaký podaspekt, generují se do dotazníku otázky pouze pro vybrané podaspekty. Pokud je vybrán pouze rodičovský aspekt, generují se otázky pro něj. Je také možné hromadně vytvářet dotazníky importem souboru, ve kterém je na každém řádku UČO studenta a IČO společnosti. Pro rozeslání e-mailů

klikne uživatel na ikonku dopisu u dané dvojice dotazníku. Také je možno rozesílat dopisy hromadně. Při akcích, které trvají delší dobu, jako je například rozesílání dopisů nebo import větších souborů, se uživateli zobrazuje okénko s informací, že daná akce probíhá.

E-mail, který přijde studentovi a zástupci společnosti, obsahuje odkaz na dotazník. Na základě tohoto odkazu si jej odpovídající otevře (nikam se nepřihlašuje) a může s ním pracovat po určenou dobu. Po uplynutí časového limitu od prvního otevření skončí platnost dotazníku a již jej nepůjde zobrazit. Koordinátor stáží může povolit znovuotevření dotazníku. V dotazníku se zobrazují otázky, jejichž hodnocení se provádí na stupnici pomocí hvězdiček. Maximum je pět hvězdiček a hvězdičky je možno dělit na poloviny. Hvězdičkové hodnocení je implementováno s pomocí jQuery Star Rating pluginu.¹⁰ Díky tomuto pluginu se u formulářového prvku *radio button* místo standardních tvarů zobrazují hvězdičky. Příklad dotazníku je v příloze na obrázku A.5. Když uživatel dotazník uloží, nemůže si jej už znova zobrazit.

6.3.3 Zobrazení statistik

Na stránce pro zobrazení statistik se vykresluje sloupcový graf s absolutními hodnotami aspektů a výšečový graf s rozložením úsilí pro zlepšení kritických aspektů. V horní části stránky si uživatel může zvolit, pro kterou společnost a pro které období se statistiky vypisují. Také je možno zvolit všechna období i všechny společnosti. Mimo to si uživatel vybere, zda se agregují data ze složených aspektů, nebo zda se každý aspekt považuje za samostatný. Při změně požadovaných vlastností dochází k automatickému překreslení stránky. Pro vykreslení grafů v HTML stránce se používá jQuery knihovna Flot¹¹. Stránka se statistikami také obsahuje odkazy pro export dat do PDF a do XLS. Při exportu dat se aplikují zvolené filtry na společnost, období a agregaci hodnot. Horní část rozhraní pro výpis statistik je zachycena v příloze na obrázku A.6, obrázek A.7 pak znázorňuje statistiku kritických aspektů.

6.4 Bezpečnost aplikace

Pro zajištění bezpečnosti aplikace jsem použila aplikační rámec Spring Security.¹² Jedná se víceméně o standard pro zabezpečení aplikací, které

10. <http://www.fyneworks.com/jquery/star-rating>

11. <http://code.google.com/p/flot>

12. <http://static.springsource.org/spring-security/site>

jsou postaveny na aplikačním rámci Spring. Informace o uživateli, kteří se mohou do aplikace přihlásit, jsou v databázi a získávají se pomocí třídy implementující rozhraní `UserDetailsService`. Každý uživatel může mít přiřazeny uživatelské role. V konfiguračním souboru pro Spring Security je pak určeno, která role má přístup ke kterým částem aplikace. Toto nastavení je také možno provést přímo v kódu pomocí anotací. Je možno zabezpečit i vrstvu služeb, takže pokud uživatel nemá potřebnou roli, tak se volaná metoda nevykoná.

Uživatelé se do aplikace přihlašují zadáním uživatelského jména a hesla. Hesla se do databáze ukládají hašovaná algoritmem SHA-256, což je v dnešní době považováno za bezpečné [8], neboť současnými výpočetními prostředky je nemožné najít původní obraz hašované zprávy nebo dvě zprávy, které by po hašování měly stejný výsledek [9].

Ani správce aplikace se nedostane k heslu v otevřené podobě. Při vytváření účtu správce přidělí heslo, které si pak uživatel může změnit. V případě zapomenutí hesla může správce přidělit uživateli heslo nové.

Při odesílání e-mailů se aplikace připojuje na externí poštovní server. Nastavení spojení je v konfiguračním souboru aplikace. Pro vytvoření připojení se používá protokol SMTP¹³ zabezpečený pomocí protokolu SSL¹⁴.

6.5 Zajištění kvality aplikace

6.5.1 Jednotkové testování

Aplikaci jsem se snažila vyvíjet tak, aby byla kvalitní a robustní. V průběhu vývoje aplikační logiky systému vznikaly jednotkové testy, které pokrývají vrstvu DAO objektů a vrstvu služeb. Jednotkové testy byly vytvářeny pomocí testovacího rámce JUnit 4.5.¹⁵ Tvorba jednotkových testů pro metody, které počítají statistické údaje, byla poměrně složitá. Nejdříve bylo potřeba vytvořit modelová data, na kterých se ručně napočítal předpokládaný výsledek, a ten se následně použil pro porovnání v testech. Testy metod, které počítají statistiku z agregovaných aspektů, jsou poměrně výpočetně náročné, protože dochází k opakovanému volání velkého počtu operací. V jednotkových testech používám pro ukládání dat databázi H2¹⁶, která má schopnost běžet pouze v paměti, ale přesto trvá spuštění všech testů poměrně dlouhou dobu.

13. <http://tools.ietf.org/html/rfc5321>

14. <http://tools.ietf.org/html/rfc6101>

15. <http://www.junit.org>

16. <http://www.h2database.com>

6.5.2 Funkční testování

Pro funkční testování aplikace jsem použila nástroj Selenium¹⁷. Jedná se o doplněk prohlížeče, který umožňuje nahrávat a následně spouštět různé testovací scénáře. Z vytvořených testů jsem exportovala kód pro JUnit a vytvořila jsem tak testy, které se automatizovaně vykonávají přímo z Java kódu. Tyto funkční testy se pouštějí při sestavení aplikace pomocí nástroje Maven. Aplikace se přeloží a nasadí na vestavěný servletový kontejner. Následně dochází ke spuštění nástroje Selenium, který sám nastartuje prohlížeč a vykonává postupně jednotlivé testy. V kódu funkčních testů se vždy po vykonání nějaké akce ověřuje, zda se na stránce požadovaná informace zobrazila. Celý tento proces je plně automatizovaný a vývojář tak pouze testy spustí a pak vidí výsledek. Jedná se o poměrně efektivní způsob, jak provádět funkční testování aplikace.

Funkční testy se spouští vždy nad prázdnou databází a simulují životní cyklus aplikace. Při testování posílání e-mailů se dopisy posílají na konkrétní adresy, u kterých je na straně poštovního serveru nastaveno zahazování. Z pohledu aplikace je důležité, aby dopis odešel a byl zpracován poštovním serverem.

Testují se následující případy užití:

- prvotní nastavení aplikace, manipulace s obdobími;
- správa uživatelských účtu, omezení práv podle rolí;
- správa a import studentů;
- správa a import společností a zástupců;
- správa aspektů a shlukování do skupin;
- správa a import dotazníků;
- posílání e-mailů;
- hlasování uživatelů;
- zobrazení statistik a export PDF.

17. <http://seleniumhq.org>

6.5.3 Testování zobrazení aplikace

Při vývoji aplikace jsem používala prohlížeč Firefox 11. V dnešní době však uživatelé používají různé prohlížeče, ve kterých se některé prvky zobrazují jinak. Případně se může stát, že prohlížeč některý prvek vůbec nepodporuje, a tak se vzhled aplikace poněkud rozpadne. Toto je pro uživatele zajisté velmi nepříjemné, proto jsem testovala zobrazení aplikace v prohlížečích Internet Explorer 9, Safari 5.1, Google Chrome 18 a Opera 11. Odstranila jsem drobné vizuální chyby a aplikaci jsem doladila tak, aby se všude zobrazovala správně. Zobrazení stránky s dotazníkem, což je klíčová součást aplikace, jsem testovala i na mobilních telefonech s operačním systémem Android 2.3.3 a iOS 5. Aplikace není optimalizovaná pro mobilní zařízení, nicméně díky jednoduchému a přehlednému vzhledu je na nich použitelná.

Kapitola 7

Nasazení systému a přínos práce

Kapitola popisuje nasazení aplikace pro stážový program *Interim Project*, což je povinná stáž studentů magisterského oboru *SSME*¹ na FI MU. Také jsou zváženy další možnosti nasazení. Na základě zkušeností získaných nasazením projektu jsou zhodnoceny přínosy práce a navržena možná rozšíření.

7.1 Potřebné komponenty pro nasazení

7.1.1 Servletový kontejner

Systém je vyvíjen jako webová aplikace, která se nasazuje do servletového kontejneru. Pro nasazení své aplikace jsem použila servletový kontejner Tomcat². K tomu, aby adresa aplikace byla uživatelsky přívětivá, bylo potřeba ještě nakonfigurovat několik věcí.

Tomcat se ve výchozím nastavení spouští v systému pod neprivilegovaným uživatelským účtem, takže může poslouchat pouze na portech vyšších než 1023. Ve výchozím nastavení se používá port 8080, který je potřeba specifikovat v adrese. Jelikož pro uživatele by měl být skryt port, na kterém aplikace běží, tak je potřeba aby Tomcat poslouchal na portu 80, což je známý (*well known*) port pro protokol HTTP, takže se do adresy nepíše. Pro řešení tohoto problému jsem použila nastavení překladu cílového portu na firewallu.

K tomu, aby mohla aplikace běžet na vlastní poddoméně, jsem nastavila virtuálního hostitele (*virtual host*) v konfiguraci Tomcatu. Na tomto hostiteli je aplikace nasazená do kořenového kontextu. Uživatelé tak pro přístup k aplikaci používají adresu tvaru `staze.domena.com`.

1. <http://www.fi.muni.cz/admission/master/aplinfo/ssme>
2. <http://tomcat.apache.org>

7.1.2 Databáze

Aplikace používá databázi PostgreSQL³, i když díky JPA by změna databáze znamenala pouze malou změnu v konfiguračních souborech. Název používané databáze a přístupové údaje k ní jsou uloženy v konfiguračním souboru aplikace. Před spuštěním systému musí databáze existovat. Tabulky se vytváří automaticky při startu aplikace. Jelikož nepřihlášený uživatel nemá možnost se systémem pracovat, tak je potřeba, aby došlo k importu prvotního administrátorského účtu do databáze, aby se dal systém vůbec použít. SQL příkazy pro vytvoření tohoto účtu jsou ve skriptu `import.sql`, který je umístěn v adresáři aplikace. Hibernate, což je poskytovatel JPA, kterého v systému používám, po vytvoření databázového schématu zajistí vykonání příkazů z tohoto skriptu.

Při prvotním přihlášení do systému musí uživatel přidat období, aby se dalo se systémem pracovat. Při používání systému je vždy zvoleno nějaké období, ke kterému se všechny informace vážou. Toto období se uživateli zobrazuje v pravém horním rohu aplikace.

7.2 Nasazení aplikace

7.2.1 *Interim Project*

Bylo dohodnuto nasazení aplikace pro stážisty programu *Interim Project*, což je povinná stáž pro studenty oboru *Služby - výzkum, řízení a inovace (SSME)* na FI MU v Brně. Aplikace byla spuštěna do provozu. Již proběhlo zadání aspektů, které se budou hodnotit. Také došlo k nainportování dat studentů a společností a vygenerování dvojic dotazníku. Provedla jsem vše potřebné pro to, aby mohl být zahájen sběr dat. Dalším krokem je spuštění hodnocení, což musí provést koordinátoři programu. Díky tomu, že systém umožňuje rozeslat dotazníky hromadně, je spuštění hodnocení otázkou několika kliknutí.

7.2.2 Stáže projektu PPS

Aplikace byla vytvořena v rámci OPVK projektu *Platforma průmyslové spolupráce*. V tomto projektu se používá model dvoufázových stáží, který byl představen v části 4.1. V tomto modelu se využívá zpětná vazba pro určení podmínek první fáze stáží. Pokud se ukáže, že nasazení systému pro SSME bylo přínosné, je v plánu nasadit systém i pro stáže tohoto projektu.

3. <http://www.postgresql.org>

První kolo stáží se plánuje na léto 2012. Na konci stážového období by měl proběhnout sběr hodnocení tak, aby koordinátoři projektu měli výsledky před začátkem podzimního semestru. Na základě těchto výsledků pak bude možno modifikovat podmínky první fáze stážového modelu pro další etapu stáží.

7.2.3 Možná rozšíření systému

Systém v současné podobě slouží k rychlému získání celkového hodnocení stážového modelu. Nicméně toto hodnocení je poměrně hrubozrné, neboť odpovědi na některé zásadní otázky není možno vyjádřit na hvězdičkové škále. Pro získání detailnějšího popisu by bylo potřeba mít možnost vytvářet i otázky jiného typu, například výběr z možností.

Také by bylo zajímavé mít možnost definovat otázky pouze pro jednu stranu (studenta nebo zástupce firmy), které by se pak nevyhodnocovaly na základě shody. Statistiky by pak mohly poskytnout konkrétnější informace o stavu jednotlivých částí stáže. Takto by se dalo například zjistit, které konkrétní předměty či aktivity fakulty jsou pro stážisty přínosné.

S tím souvisí i možnost integrace vyplňování studentských zpráv. Stáže je většinou potřeba nějak papírově dokladovat, tudíž studenti po skončení sepiší písemnou zprávu, která poskytne detailní informace o průběhu a přínosu stáže. Toto ve většině případů zpracovává každý student individuálně, a proto se forma těchto zpráv může dost lišit. Zprávy zadané do systému by mohly být v jednotné formě exportovány a archivovány.

Při nasazení systému pro stážisty SSME jsem narazila na nutnost vyexportovat data z jejich databáze a tato data následně importovat do systému. Je poměrně neefektivní uchovávat data o stážistech na více místech, takže by bylo užitečné do systému doplnit další funkcionalitu potřebnou pro administraci stáží. Systém pro hodnocení stáží by tak mohl posloužit jako jádro pro vznik komplexního informačního systému pro evidenci stážistů.

Kapitola 8

Závěr

V práci jsem prozkoumala současnou situaci studentských stáží na vysokých školách v České republice. Na základě průzkumu jsem stanovila čtyři základní stážové modely, které jsem následně zhodnotila a porovнала. Jako nejužitečnější se mi zdá model *volitelných stáží*, ve kterém jsou stáže součástí studia (studenti za ně dostanou kredity), ale nejsou v rámci studia povinné.

K tomu, aby mohli koordinátoři stáží dynamicky reagovat na vývoj technologií i situaci na trhu práce, je potřeba zpracovávat zpětnou vazbu stáží. Pro hodnocení stáží jsem navrhla model založený na porovnávání shody odpovědí studenta a zaměstnance společnosti, aby se eliminovaly příliš subjektivní názory.

Cílem práce bylo vytvořit systém, který by umožňoval sběr hodnocení podle tohoto modelu. Systém jsem vytvořila jako webovou aplikaci s pomocí aplikačního rámce Spring. Persistence dat je vyřešena pomocí Java Persistence API. Při návrhu uživatelského rozhraní jsem kladla důraz hlavně na jednoduchost a uživatelskou přívětivost aplikace. Sběr odpovědí probíhá vyplněním dotazníku s hodnocením pomocí hvězdiček na kvantitativně formulované dotazy. Odkaz na dotazník obsahuje unikátní identifikátor a ze systému se rozesílá e-mailem. Vyplnění dotazníku je pro dotázané otázkou několika minut.

Z nasbíraného hodnocení si pak koordinátoři mohou zobrazovat statistiku hodnot aspektů a statistiku rozložení úsilí pro zlepšení kritických aspektů. Tyto statistiky je možno filtrovat podle období, organizace a případně agregovat složené aspekty. Statistiku je možno exportovat do formátu PDF a XLS.

Třídy, které poskytují aplikační logiku systému jsou kompletně pokryty jednotkovými testy. Také jsem vytvořila sadu funkčních testů, které vedly k odhalení několika chyb ve funkcionalitě uživatelského rozhraní.

Aplikace byla nasazena pro stážisty programu *Interim Project* a nyní probíhá sběr hodnocení. Při používání aplikace se ukázalo, že systém by bylo možno rozšířit o další funkcionalitu, což by mohlo vést až k vytvoření plnohodnotného informačního systému pro evidenci studentských stáží.

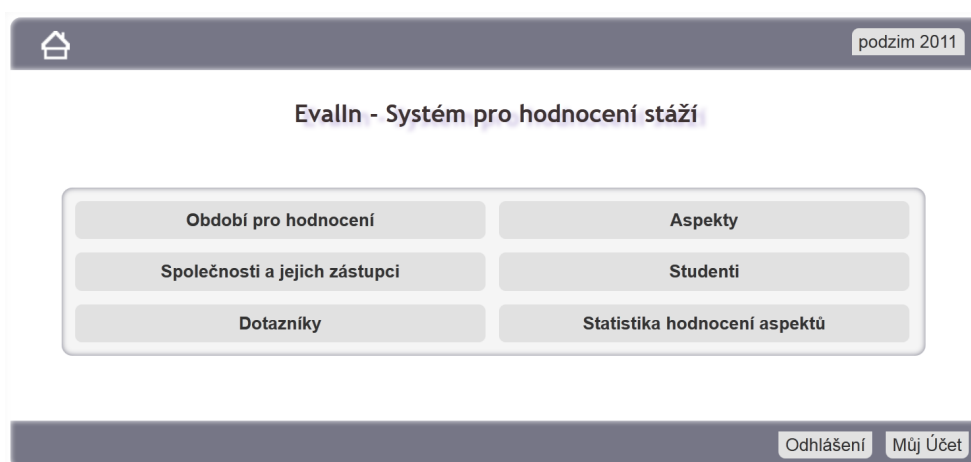
Podle mého názoru je výsledkem této práce kvalitní a robustní aplikace pro hodnocení stáží, která dokáže rychle poskytnout celkový pohled na úspěšnost stážového modelu a odhalit tak jeho kritická místa. Úspěšnost systému dokládá i to, že je plánováno aplikaci nasadit i na hodnocení letních stáží projektu Platforma průmyslové spolupráce. Jedním z cílů tohoto projektu je vytvoření efektivního modelu stáží. Věřím, že aplikace bude k tomu nápomocná a bude tak pro projekt přínosem

Literatura

- [1] D. Alur, J. Crupi, and D. Malks. *Core J2EE patterns: best practices and design strategies*. Prentice Hall PTR, 2003.
- [2] C. Walls and R. Breidenbach. *Spring in action*. Manning Publications Co., third edition, 2011.
- [3] J. Filipec and V. Červená. *Slovník spisovné češtiny pro školu a veřejnost*. Academia, 1994.
- [4] E. Taylor-Powell, S. Steele, and M. Douglass. Planning a program evaluation. *Madison, WI: University of Wisconsin Cooperative Extension*, 1996.
- [5] B. Oestereich. *Analyse und Design mit UML 2.1*. Oldenbourg Wissenschaftsverlag, 2006.
- [6] J.W. Cooper. *Java design patterns: a tutorial*. Addison-Wesley Professional, 2000.
- [7] L. DeMichiel and M. Keith. JSR-220: Enterprise JavaBeans, Version 3.0-Java Persistence API. *Sun Microsystems, EJB*, 3, 2007.
- [8] J.H. Burrows. Secure hash standard. Technical report, DTIC Document, 2012.
- [9] Q. Dang, National Institute of Standards, and Technology (US). *Recommendation for applications using approved hash algorithms*. US Department of Commerce, National Institute of Standards and Technology, 2009.

Příloha A

Ukázka grafického vzhledu aplikace



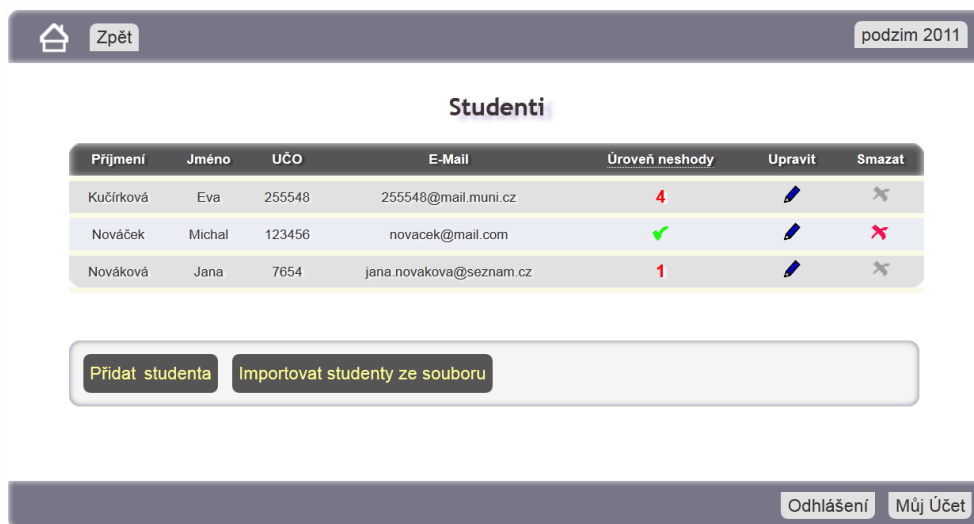
Obrázek A.1: Hlavní stránka aplikace

The screenshot shows a web application interface for editing an aspect. The main content area is titled "Upravit aspekt" and contains the following fields:

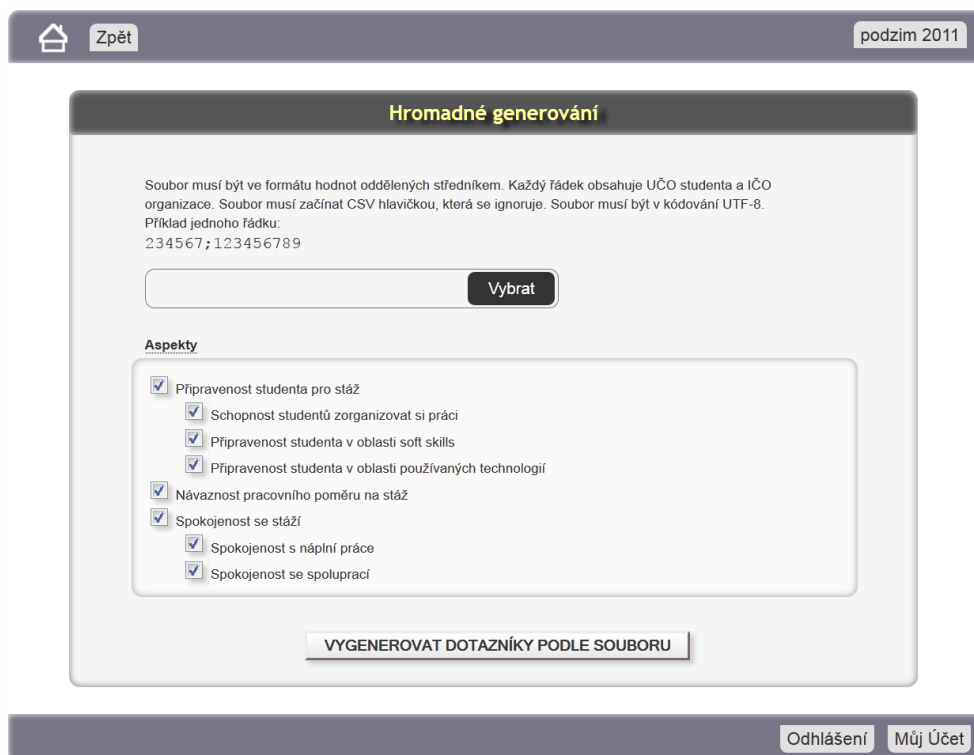
- Jméno:** Text input field containing "Spokojenost se stáží".
- Zkrácené jméno:** Text input field containing "Spokojenost".
- Rodičovský aspekt:** Dropdown menu with the selected option "-- Žádný --".
- Náznak k řešení:** Text input field containing "Zvýšit oboustrannou spokojenost se stáží".
- Otázka pro studenta:** Text input field containing "Jak jste byl na stáží spokojený?".
- Otázka pro zástupce společnosti:** Text input field containing "Jak jste byl spokojený se studentem na stáží?".
- Podaspekty:** Two text input fields containing "Spokojenost s náplní práce" and "Spokojenost se spoluprací".

At the bottom of the form is a button labeled "UPRAVIT ASPEKT". The top navigation bar has a home icon, "Zpět", and "podzím 2011". The bottom navigation bar has "Odhlášení" and "Můj Účet".

Obrázek A.2: Formulář pro přidávání/úpravu aspektu



Obrázek A.3: Agenda pro správu studentů



Obrázek A.4: Dialog pro hromadné generování dotazníků

Hodnocení stáží

Dotazník vyplňuje: Jan Novák (ABC s.r.o.)

Hodnotí se student: Eva Kučírková

Jak moc byste si chtěl ve firmě nechat studenta jako zaměstnance?

Jak moc byl student při nástupu na stáž seznámen s používanými technologiemi?

Jak dobře se vám zdály komunikační schopnosti studenta na stáži?

Do jaké míry byl student schopen si zorganizovat přidělenou práci?

Jak jste byl spokojen s tím, co student na stáži mohl vykonávat?

Jak se vám spolupracovalo se studentem?

[ULOŽIT DOTAZNÍK](#)

Obrázek A.5: Dotazník

🏠 Zpět
podzim 2011

Statistika hodnocení aspektů

Zvolené období: podzim 2011

Zvolená společnost: všechny společnosti

Export do PDF

Export do XLS

Agregovat složené aspekty

Hodnocení jednotlivých aspektů

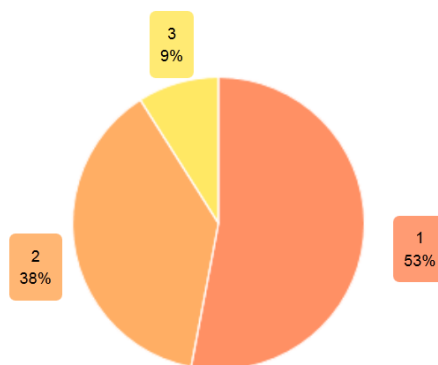
| Č. | Aspekt | Úspěšnost |
|----|---|-----------|
| 1 | Návaznost pracovního poměru na stáž | 83 % |
| 2 | Připravenost studenta pro stáž | 70 % |
| 3 | Schopnost studentů zorganizovat si práci | 70 % |
| 4 | Spokojenost se spoluprací | 65 % |
| 5 | Připravenost studenta v oblasti používaných technologií | 60 % |
| 6 | Spokojenost s náplní práce | 50 % |
| 7 | Připravenost studenta v oblasti soft skills | 45 % |

Obrázek A.6: Horní část rozhraní pro výpis statistik

A. UKÁZKA GRAFICKÉHO VZHLEDU APLIKACE

Co je potřeba zlepšit - rozložení úsilí

| Č. | Co je potřeba řešit | Procento úsilí |
|----|--|----------------|
| 1 | U studentů klást důraz na rozvíjení soft skills | 53 % |
| 2 | Přizpůsobit náplň práce stážisty jeho možnostem | 38 % |
| 3 | Zlepšit přípravu studentů v potřebných technologiích | 9 % |



Obrázek A.7: Statistika kritických aspektů

Příloha B

Ukázka vygenerovaných statistik ve formátu PDF

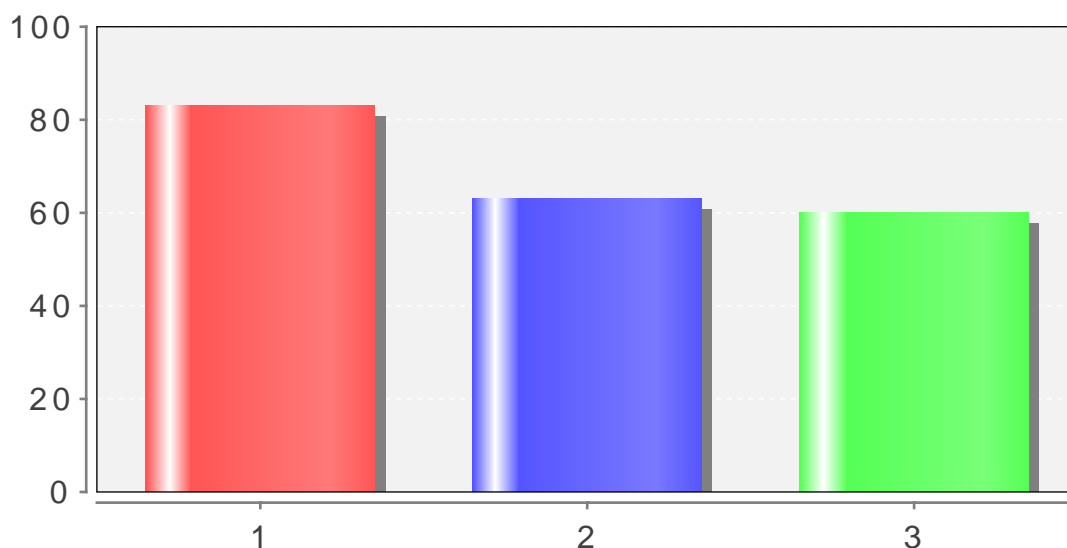
Tato příloha obsahuje ukázky exportovaných statistik ve formátu PDF. Bylo exportováno hodnocení všech organizací pro zvolené období. První zpráva ukazuje statistiku agregovaných hodnot aspektů. Jelikož ve statistice jsou pouze tři aspekty, jsou oba grafy na jedné stránce. Druhá zpráva zachycuje statistiku jednotlivých hodnot aspektů. Aby byly grafy s vyšším počtem aspektů přehledné, generuje se každý obrázek na samostatnou stránku.

Statistika hodnocení aspektů - podzim 2011

Celkové hodnocení (pro všechny organizace)

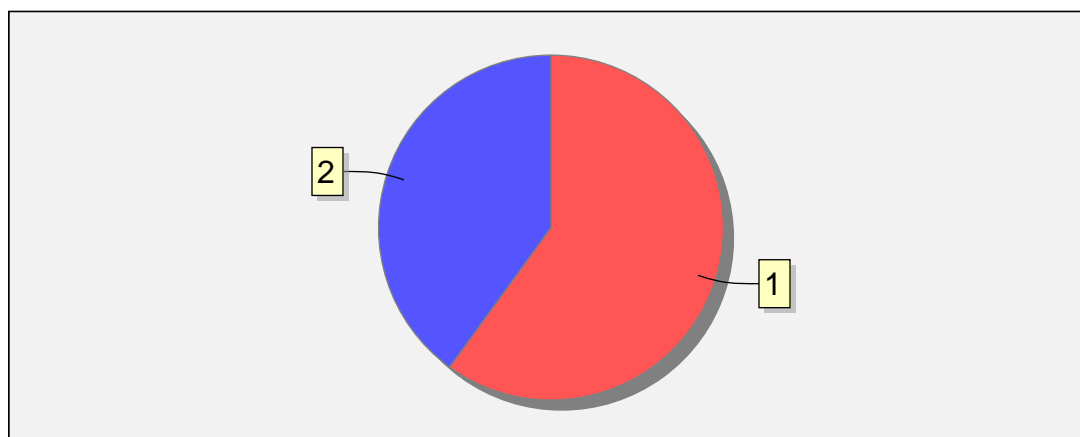
Vytvořeno: 12.5.2012

Hodnocení jednotlivých aspektů



- 1: Návaznost pracovního pom. ru na stáž (83%)
- 2: P ipravenost studenta pro stáž (63%)
- 3: Spokojenost se stáží (60%)

Co je pot eba zlepšit - rozložení úsilí



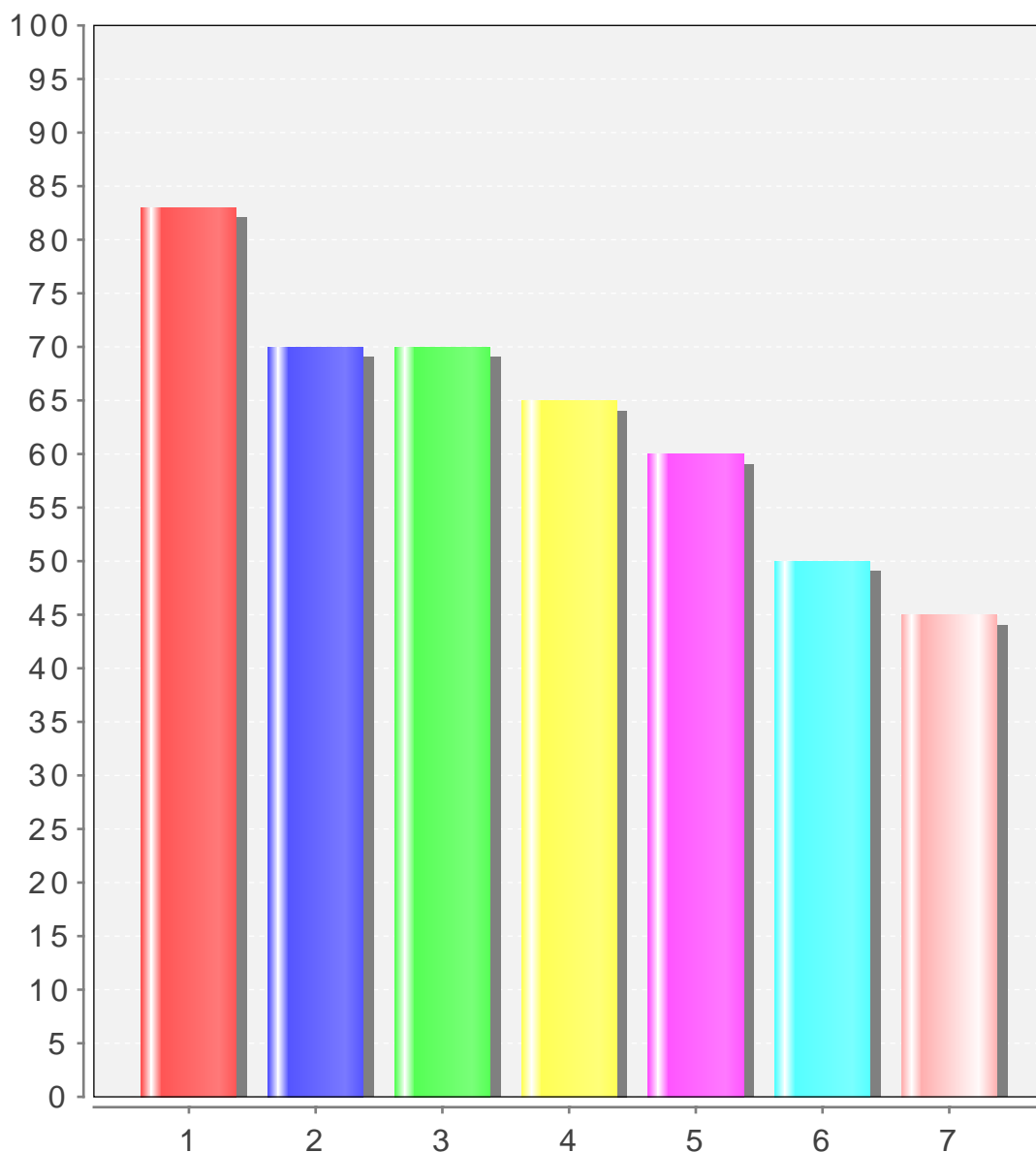
- 1: Zvýšit oboustrannou spokojenost se stáží (60%)
- 2: Lépe p ipravovat studenty na stáž (40%)

Statistika hodnocení aspektů - podzim 2011

Celkové hodnocení (pro všechny organizace)

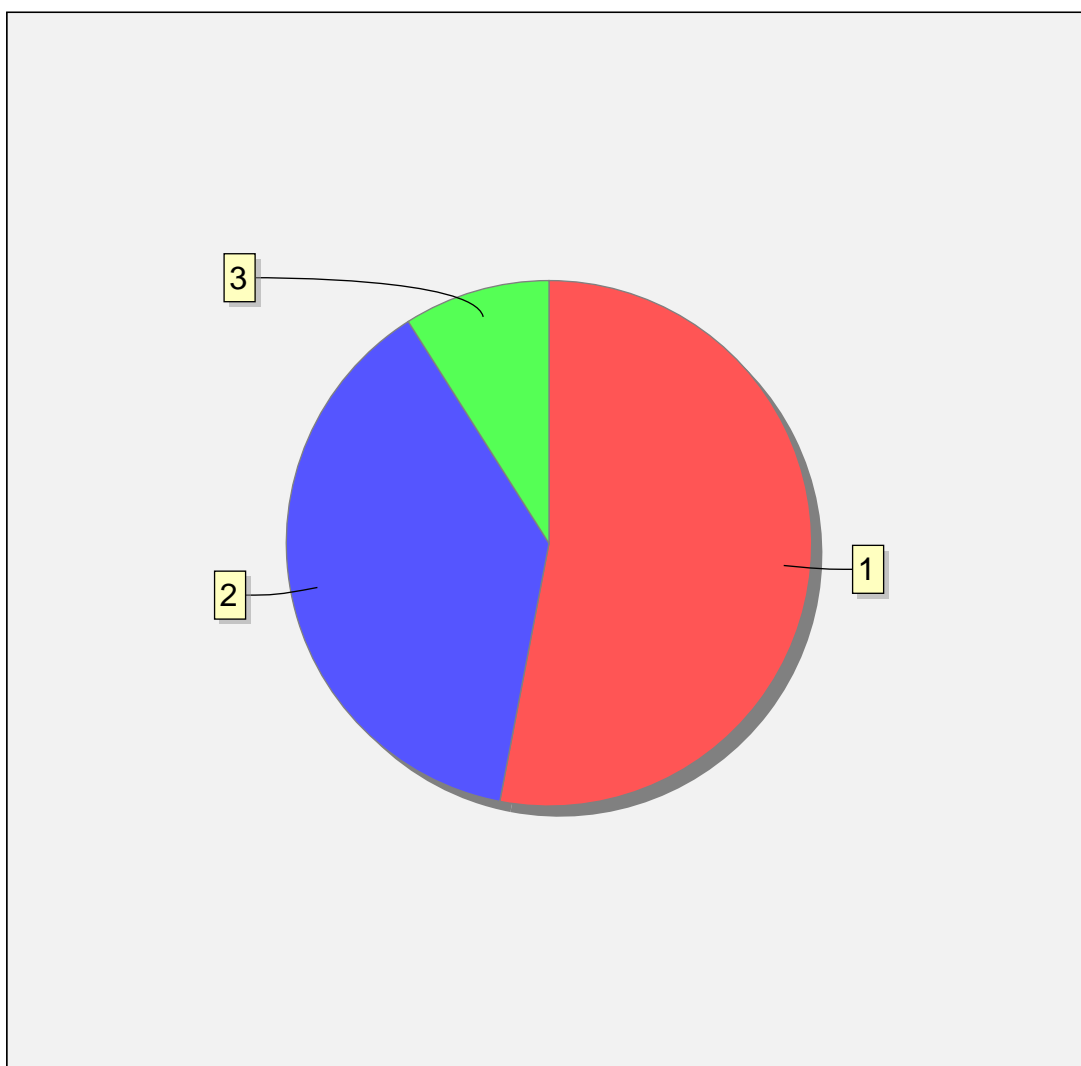
Vytvořeno: 12.5.2012

Hodnocení jednotlivých aspektů



- 1: Návaznost pracovního poměru na stáž (83%)
- 2: Přípravenost studenta pro stáž (70%)
- 3: Schopnost studenta zorganizovat si práci (70%)
- 4: Spokojenost se spoluprací (65%)
- 5: Přípravenost studenta v oblasti používaných technologií (60%)
- 6: Spokojenost s náplní práce (50%)
- 7: Přípravenost studenta v oblasti soft skills (45%)

Co je potřeba zlepšit - rozložení úsilí



■ 1: U student klást draz na rozvíjení soft skills (53%)

■ 2: P izp sobit nápl práce stážisty jeho možnostem (38%)

■ 3: Zlepšit p ípravu student v pot ebných technologiích (9%)

Příloha C

Elektronické přílohy

Elektronickou přílohou práce je archív zdrojových souborů aplikace. Aplikace je sestavována nástrojem Maven, takže struktura projektu dodržuje konvence tohoto nástroje. Jednotlivé části projektu jsou:

- `pom.xml` – konfigurační soubor aplikace, obsahuje nastavení závislostí projektu a používaných knihoven;
- `src` – adresář se zdrojovými kódy;
- `src/test` – zdrojové kódy jednotkových i funkčních testů;
- `src/main/java` - zdrojové kódy tříd aplikace;
- `src/main/webapp` - JSP stránky, konfigurační soubory a další zdroje (obrázky, styly, skripty) používané ve webové aplikaci;
- `src/main/resources` – konfigurační soubory, balíky jazykových verzí a ostatní zdroje používané aplikací;
- `src/main/resources/eval.properties` – v tomto souboru je nastavení aplikace (používaná databáze, poštovní server i mezní hodnota pro porovnávání hodnocení studenta a zástupce společnosti).