Bakalářská práce

České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Informační systém pro vícezdrojové financování

Tomáš Chamra
Otevřená informatika - Informatika a počítačové vědy

Květen 2014
Vedoucí práce: doc. Ing. Zdeněk Kouba, CSc.

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Tomáš C h a m r a

Studijní program: Otevřená informatika (bakalářský)

Obor: Informatika a počítačové vědy

Název tématu: Informační systém pro vícezdrojové financování

Pokyny pro vypracování:

1. Seznamte se s vícezdrojovým financováním na elektrotechnické fakultě a vypracujte funkční požadavky pro informační systém, který umožňuje vedoucímu odborné skupiny efektivně spravovat finanční zdroje a přiřazení pracovníků ke zdrojům.
2. Seznamte se s technologií Java Enterprise Edition 7.
3. Navrhněte informační systém splňující funkční požadavky vypracované v bodě 1.
4. Navrhněte testy pro ověření funkcionality implementovaného informačního systému. V co největší míře testy koncipujte jako jUnit testy.
5. Informační systém implementujte a jeh funkcionalitu ověřte pomocí navržených testů. Protokol o provedení testů bude součástí bakalářské práce.

Seznam odborné literatury:

[1] Goncalces, A.: Beginning Java EE 7, Apress, 2013, ISBN13: 978-1-4302-4626-8
[2] Juneau, J.: Java EE 7 Recipes – A problem-Solution Approach, APress, 2013, ISBN 978-1-4302-4425-7

Vedoucí bakalářské práce: doc. Ing. Zdeněk Kouba, CSc.

Platnost zadání: do konce zimního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 6. 12. 2013

Poděkování / Prohlášení

Rád bych poděkoval vedoucímu této práce, doc. Ing. Zdeňku Koubovi, CSc., za ochotu, trpělivost a poskytnuté konzultace. Dále bych chtěl poděkovat mým rodičům za to, že mne podporovali po celou dobu studia.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 21. 5. 2014

.....

Abstrakt / Abstract

Tato bakalářská práce se zabývá návrhem a implementací informačního systému pro správu vícezdrojového financování. Vychází z potřeb elektrotechnické fakulty ČVUT. Cílem vyvinutého systému je pomoci vedoucím výzkumných skupin a jednotlivých projektů získat přehled o plánovaných výplatách zaměstnanců a stavech jednotlivých finančních zdrojů, ze kterých jsou tito zaměstnanci placeni. Systém je postaven na technologii Java Enterprise Edition a kromě evidování údajů o zaměstnancích a projektech dokáže také naplánovat výplaty. Je tedy možné snadno vidět, kde jaké prostředky chybí a tudíž je nutné je najít v jiných zdrojích.

Klíčová slova: enterprise java, informační systém, vícezdrojové financování

This bachelor thesis focuses on design and implementation of the information system for management of multi-resource funding, based on needs of the Faculty of Electrical Engineering at CTU. The goal of the developed system is to help research group leaders and project leaders to have an overview about planned payouts and balances of financial resources used to pay their employees. System is built using the Java Enterprise Edition technology and in addition to keeping an evidence of information about employees and projects, it can also plan future payouts. This allows leaders to see how much money is missing, so they can pay them from other resources.

Keywords: enterprise java, information system, multi-resouce funding

Title translation: Information System for Multi-Resource Funding

Obsah

1 Úvod ... 1
1.1 Motivace ... 1
1.2 Cíl ... 1
2 Plánování výplat ... 2
2.1 Skládání výplat zaměstnanců ... 2
2.2 Způsob zadávání údajů ... 2
2.3 Výpočet dostupných prostředků ... 3
2.4 Plánování výplat ze zdroje ... 3
2.5 Pomocný zdroj ... 3
2.6 Plánování výplat z projektu ... 3
2.6.1 Přeplánování ... 4
3 Návrh informačního systému ... 5
3.1 Požadavky ... 5
3.2 Uživatelské role ... 5
3.3 Případy užití ... 6
3.4 Datový model ... 7
3.4.1 Uživatelské účty a role ... 8
3.4.2 Osoby ... 9
3.4.3 Výzkumné kupiny ... 9
3.4.4 Projekty a zdroje ... 9
3.4.5 Přiřazení zdrojů k osobám ... 9
3.4.6 Výplaty ... 10
3.4.7 Tarifní tabulka ... 10
4 Implementace informačního systému ... 11
4.1 Použité technologie ... 11
4.1.1 Java 7 ... 11
4.1.2 Maven ... 11
4.1.3 Java EE 7 ... 12
4.1.4 Glassfish 4 ... 12
4.1.5 PrimeFaces ... 12
4.1.6 PrettyFaces ... 12
4.1.7 MySQL ... 13
4.2 Výběr vhodného frameworku ... 13
4.3 Uživatelské rozhraní ... 13
4.4 Rozložení uživatelského rozhraní ... 13
4.4.1 Seznam obrazovek ... 14
4.5 Lokalizace ... 15
4.6 Zabezpečení ... 16
5 Testování ... 17
5.1 Úrovně testování ... 17
5.2 Testování v Mavenu ... 17
5.3 Testování informačního systému ... 17
5.4 Systémové testy ... 18
5.4.1 Přihlášení, odhlášení, změna hesla ... 18
5.4.2 Osoby ... 18
5.4.3 Smlouvy ... 19
5.4.4 Platové výměry ... 19
5.4.5 Skupiny ... 20
5.4.6 Projekty ... 20
5.4.7 Zdroje ... 21
5.4.8 Periody ... 21
5.4.9 Výkazy o skutečném čerpání ... 22
5.4.10 Přiřazování zdrojů k osobám ... 22
5.4.11 Plánování výplat ... 23
5.4.12 Výplaty ... 23
5.4.13 Administrace uživatelů ... 24
5.4.14 Tarifní tabulka ... 24
6 Možná rozšíření do budoucna ... 26
6.1 Dohody s hodinovou sazbou ... 26
6.1.1 Stipendia a příplatky za vedení ... 26
6.2 Import dat ... 27
6.3 Externí ověřování přihlašovacích údajů ... 27
7 Závěr ... 28
Literatura ... 29
A Návod k instalaci ... 30
A.1 Kompilace ... 30
A.2 Nastavení databáze ... 30
A.3 Autentizace ... 31
A.4 Nahrání aplikace na server ... 31
A.5 První přihlášení ... 32
B Protokol o provedení testů ... 33
B.1 Unit testy ... 33
B.2 Systémové testy ... 33
C Obsah přiloženého CD ... 35
D Zkratky ... 36

Tabulky / Obrázky

A.1. Nastavení JDBC poolu 1 30
A.2. Nastavení JDBC poolu 2 31
A.3. Nastavení JDBC zdroje 31
A.4. Nastavení autentizace 31
B.5. Systémové testy 1 33
B.6. Systémové testy 2 34
C.7. Obsah přiloženého CD 35

2.1. Vztah zaměstnanců a zdrojů.....2
3.1. Datový model 8
4.1. Rozložení stránky 14
A.1. Úvodní stránka 1..... 32
A.2. Úvodní stránka 2..... 32

Kapitola 1
Úvod

Naplánovat výplaty pro celý tým lidí, kde každý pracuje na různých projektech a je financován z různých finančních zdrojů, může být náročný úkol.

Každý zaměstnanec patří do určitého oddělení či výzkumné skupiny, a zároveň pracuje na jednom či více projektech. Každý projekt má své finanční zdroje, z nichž jsou hrazeny mzdy jednotlivých zaměstnanců. V případě, že daný zaměstnanec pracuje na více projektech, mzdy se v odpovídajícím poměru skládají z jednotlivých zdrojů.

Cílem informačního systému pro správu vícezdrojového financování je pomoci vedoucím skupin a projektů s plánováním jednotlivých výplat a zároveň jim umožnit, aby měli neustále přehled o stavu jednotlivých zdrojů a měli tedy jistotu, že všechny mzdy budou pokryty bez překročení rozpočtu daných projektů.

1.1 Motivace

Tato bakalářská práce vznikla ze dvou různých důvodů. Prvním důvodem byla má snaha o seznámení se s technologií Java Enterprise Edition. Mnohokrát jsem se s ní setkával v souvislosti s většími webovými aplikacemi a různými informačními či bankovními systémy, ale chyběla mi motivace se s ní naučit pracovat.

Shodou okolností se objevila potřeba vytvoření systému pro plánování financování projektů na některých součástech elektrotechnické fakulty, a to se tak stalo tématem této bakalářské práce. Vzhledem k velice specifickému účelu požadovaného informačního systému se mi nepodařilo nalézt žádný jiný systém, který by podobnou funkcionalitu nabízel, a tudíž bylo nutné vytvořit zcela nové řešení.

1.2 Cíl

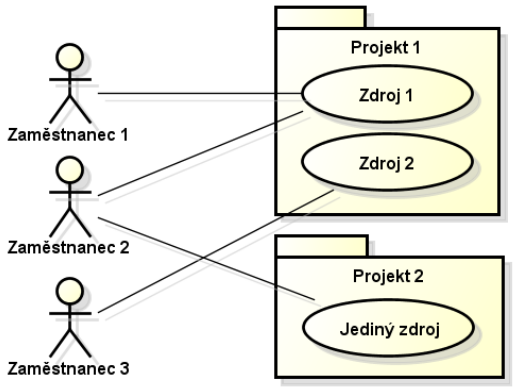
Cílem této práce je navrhnout, implementovat a otestovat informační systém pro správu vícezdrojového financování na elektrotechnické fakultě ČVUT. Informační systém má být postaven na technologii Java EE a uživatelské rozhraní tohoto systému musí být kompletně v anglickém jazyce.

Kapitola 2
Plánování výplat

Před začátkem samotného návrhu informačního systému bych rád objasnil, jakým způsobem dochází ke skládání výplat jednotlivých zaměstnanců a jak bude informační systém tyto výplaty plánovat.

2.1 Skládání výplat zaměstnanců

Mzdy pro jednotlivé zaměstnance jsou placeny obvykle z několika zdrojů. Zaměstnanci zpravidla pracují na jednom či více projektech, přičemž tyto projekty disponují rozpočtem, který může být dále členěn do různých položek. Každou takovou položku budeme nadále označovat pojmem „zdroj“. Zaměstnanci jsou pak placeni vždy z konkrétních zdrojů, ke kterým jsou přiřazeni, jak je vidět na následujícím schématu:



Obrázek 2.1. Příklad možných vztahů mezi zaměstnanci, projekty a zdroji

Situaci však komplikuje fakt, že jednotlivé zdroje poskytují finanční prostředky postupně, v určitých obdobích (periodách), pro které je vždy pevně dána částka, kterou lze čerpat. Zaměstnanci však mají platy pevně dané svým platovým výměrem, a tak je nutné jim odměny poskládat a případné chybějící prostředky pokrýt odjinud.

2.2 Způsob zadávání údajů

Pro začátek je vhodné si rozvrhnout, jak budou údaje o zdrojích financí ukládané.

Ke každému projektu bude náležet jeden či více zdrojů, a každý z těchto zdrojů bude rozdělen na určité periody (například měsíce či roky). Pro každou takovouto periodu bude vždy dána částka, kterou v ní lze čerpat.

V informačním systému jsou evidovány jednotlivé výplaty, a u každé takovéto výplaty je jasně dáno, která část je placena z kterého zdroje. Dále je možné do systému po jednotlivých měsících zadávat skutečně vyčerpané prostředky z daného zdroje (v konkrétní periodě) nezávisle na tom, jak byly tyto prostředky využity.

2.3 Výpočet dostupných prostředků

Pro plánování výplat je nutné vždy vědět, jaké prostředky jsou momentálně dostupné a lze s nimi tedy disponovat.

Dostupné prostředky k danému datu systém počítá tak, že nejprve zjistí, jaká je v periodě, do které datum náleží, celková částka určená rozpočtem daného zdroje. Od této částky odečte všechny záznamy o skutečně vyčerpaných prostředcích z této periody. Posledním krokem je pak odečtení všech naplánovaných výplat, které jsou datovány později, než je měsíc s posledním nenulovým údajem o skutečném čerpání.

Poslední část postupu by se mohla zdát podivná, nicméně má své opodstatnění. Pokud totiž za daný měsíc zadáme skutečně vyčerpanou částku, pak jsou v ní již zahrnuty vyplacené mzdy a nemusíme s nimi tedy počítat. Pokud však údaje o skutečném čerpání doposud nemáme, je nutné s výplatami počítat a pracujeme tedy s nejlepším možným odhadem, který máme k dispozici – tedy s naplánovanými výplatami.

2.4 Plánování výplat ze zdroje

V systému je zahrnut algoritmus po plánování výplat. Tento výpočet lze spustit pro konkrétní zdroj, nebo pro celý projekt, a to vždy od určitého data.

Postup pro výpočet z konkrétního zdroje vypadá ve zjednodušené podobě takto:

```
pro každý měsíc ve kterém je zdroj platný:
   pro každou osobu která tento měsíc na projektu pracuje:
      vypočti jakou má daná osoba dostat mzdu
      zjisti jaká část mzdy má být vyplacena právě z tohoto zdroje
      vynásobením dvou předchozích hodnot zjisti kolik se musí vyplatit
      pokud pro danou periodu zbývá ze zdroje dost peněz:
         vyplať ze zdroje potřebnou částku
      pokud dost peněz není:
         vyplať tolik, kolik lze
         zbytek vyplať z pomocného zdroje
```

2.5 Pomocný zdroj

Z posledního kroku je patrné, že aby všechny výpočety dávaly dohromady smysl, je nutné odněkud prostředky vyplatit alespoň virtuálně. Pro tento účel je v informačním systému u každého projektu pomocný (servisní) zdroj, u kterého není nutné mít nezáporný zůstatek a ze kterého jsou automaticky plánovány všechny výplaty, na které nejsou v jiných zdrojích prostředky.

Takovýto postup si můžeme dovolit proto, že se jedná pouze o plánování a tím umožním vedoucímu projektu či skupiny vidět, jaké prostředky chybí. Předpokládám pak, že odpovědný vedoucí údaje doladí a ve výsledku bude bilance pomocného zdroje nulová.

2.6 Plánování výplat z projektu

Kromě ručního spouštění plánovacího procesu pro jednotlivé zdroje lze v informačním systému provést i hromadné plánování pro všechny zdroje v daném projektu. Jedná se však pouze o zjednodušení pro uživatele, protože systém jen postupně provede naplánování pro všechny zdroje v daném projektu.

2.6.1 Přeplánování

Kromě běžného plánování lze spustit také přeplánování, který nejprve smaže všechny výplaty ze zdrojů daného projektu počínaje datem, od kdy se přeplánování provádí. Poté se provede plánování jako obvykle, nicméně s aktuálními daty (která při minulém spuštění nemusela být známa).

Kapitola 3
Návrh informačního systému

V této kapitole se budu informačnímu systému věnovat spíše z hlediska obecných požadavků, případů užití a datového modelu. Další kapitola se pak bude věnovat systému z hlediska implementace a použitých technologií.

3.1 Požadavky

Informační systém je navržen s ohledem na následující požadavky:

- má evidovat údaje o zaměstnancích, projektech a výplatách z těchto projektů
- u zaměstnanců eviduje zároveň i detaily týkající se jejich pracovní smlouvy
- citilivá data musí být šifrována
- musí být řešena uživatelská oprávnění tak, aby k údajům zaměstnanců měli přístup pouze jejich nadřízení (tedy vedoucí jejich projektů a skupin) a administrátoři systému
- systém musí fungovat autonomně, neboť se prozatím nepředpokládá napojení na jiné informační, účetní či přihlašovací systémy
- systém musí komunikovat v anglickém jazyce
- aplikace musí být dle zadání postavena na technologii Java Enterprise Edition 7. Tento požadavek jsem upřesnil tak, že musí být spustitelná na aplikačním serveru Glassfish 4.0.

3.2 Uživatelské role

V informačním systému existují čtyři uživatelské role:

- **Uživatel** – každý, kdo má přístup do informačního systému
- **Vedoucí projektu** – každá z osob pověřených vést určitý projekt, a má přístup k informacím, které jsou nějakým způsobem k danému projekt navázány
- **Vedoucí skupiny** (organizační vedoucí) – osoba pověřená vedením skupiny (oddělení), a tedy má přístup k informacím týkajícím se zaměstnanců v této skupině
- **Administrátor** – správce systému, který má přístup ke všem datům v systému a navíc může spravovat jednotlivé uživatele

Přiřazení rolí „Uživatel“ a „Administrátor“ k jednotlivým uživatelským účtům je evidováno ve speciální tabulce.

Zbývající dvě role jsou přiřazovány automaticky. K tomu je nutné, aby byl uživatelský účet svázán s osobou vedenou v databázi. Pak má tato osoba roli „Vedoucí projektu“ ve vztahu ke všem projektům, které vede, a obdobně roli „Vedoucí skupiny“ ve vztahu ke skupinám, které vede.

3.3 Případy užití

1. **Zobrazení seznamu osob a jejich detailů, smluv a platových výměrů**
   Role: Vedoucí skupiny, Vedoucí projektu, Administrátor
   Popis: Uživatel může zobrazit seznamy a podrobnosti o osobách, které mají vztah k jím vedeným skupinám nebo projektům. Může zároveň prohlížet i detaily jejich smluv a pracovních výměrů. Administrátor má přístup ke všem údajům.

2. **Administrace osob, smluv a platových výměrů**
   Role: Administrátor
   Popis: Administrátor může přidávat, upravovat a mazat jednotlivé osoby, jejich smlouvy a platební výměry.

3. **Zobrazení seznamu a detailů skupin**
   Role: Vedoucí skupiny, Administrátor
   Popis: Uživatel může zobrazit seznam a podrobnosti o jím vedených skupinách. Administrátor má přístup ke všem údajům.

4. **Administrace skupin**
   Role: Administrátor
   Popis: Administrátor může přidávat, upravovat a mazat skupiny.

5. **Zobrazení seznamu projektů, jejich detailů, k nim navázaných zdrojů a jejich period**
   Role: Vedoucí projektu, Administrátor
   Popis: Vedoucí projektů vidí podrobnosti o svých projektech, jejich zdrojích a jednotlivých period těchto zdrojů včetně finančních informací. Administrátor má přístup ke všem údajům.

6. **Administrace projektů**
   Role: Administrátor
   Popis: Administrátor může přidávat, upravovat a mazat projekty.

7. **Administrace zdrojů a jejich period**
   Role: Vedoucí projektu, Administrátor
   Popis: Vedoucí projektu mohou přidávat, upravovat a mazat zdroje k jejich projektům. K těmto zdrojům také mohou přidávat, upravovat a mazat jednotlivé periody. Administrátor může vše.

8. **Zadávání skutečně vyčerpaných zdrojů**
   Role: Vedoucí projektu, Administrátor
   Popis: Vedoucí projektu mohou zadávat skutečně vyčerpané finanční prostředky ze zdrojů svých projektů. Data se zadávájí k jednotlivým periodám, vždy souhrnně za kalendářní měsíc. Administrátor může vše.

9. **Přiřazování osob k projektům a zdrojům**
   Role: Vedoucí skupiny, Vedoucí projektu, Administrátor
   Popis: Vedoucí skupin a vedoucí projektů mohou přidělovat osoby, které jsou ve vztahu k jejich skupinám a projektům, k jednotlivým zdrojům, ze kterých budou

tyto osoby placeny. Administrátor může vše.

10. **Výpočet navrhovaných výplat**
    Role: Vedoucí projektu, Administrátor
    Popis: Vedoucí projektů mohou u svých projektů a zdrojů nechat předpočítat výplaty pro jednotlivé osoby. Administrátor může vše.

11. **Administrace výplat**
    Role: Vedoucí skupiny, Vedoucí projektu, Administrátor
    Popis: Vedoucí skupin a vedoucí projektů mohou přidávat, upravovat a mazat evidované (ať již vyplacené či teprve plánované) výplaty osob, které jsou ve vztahu k jejich skupinám a projektům. Administrátor může vše.

12. **Změna hesla**
    Role: Jakýkoliv uživatel
    Popis: Kterýkoliv uživateli může měnit své vlastní heslo.

13. **Administrace uživatelů**
    Role: Administrátor
    Popis: Administrátor může přidávat a mazat uživatele. Dále jim může měnit hesla a oprávnění.

14. **Zobrazení tarifní tabulky**
    Role: Jakýkoliv uživatel
    Popis: Kterýkoliv uživatel může prohlížet tarifní tabulku.

15. **Úprava tarifní tabulky**
    Role: Administrátor
    Popis: Administrátor může upravovat tarifní tabulku a spravovat její verze.

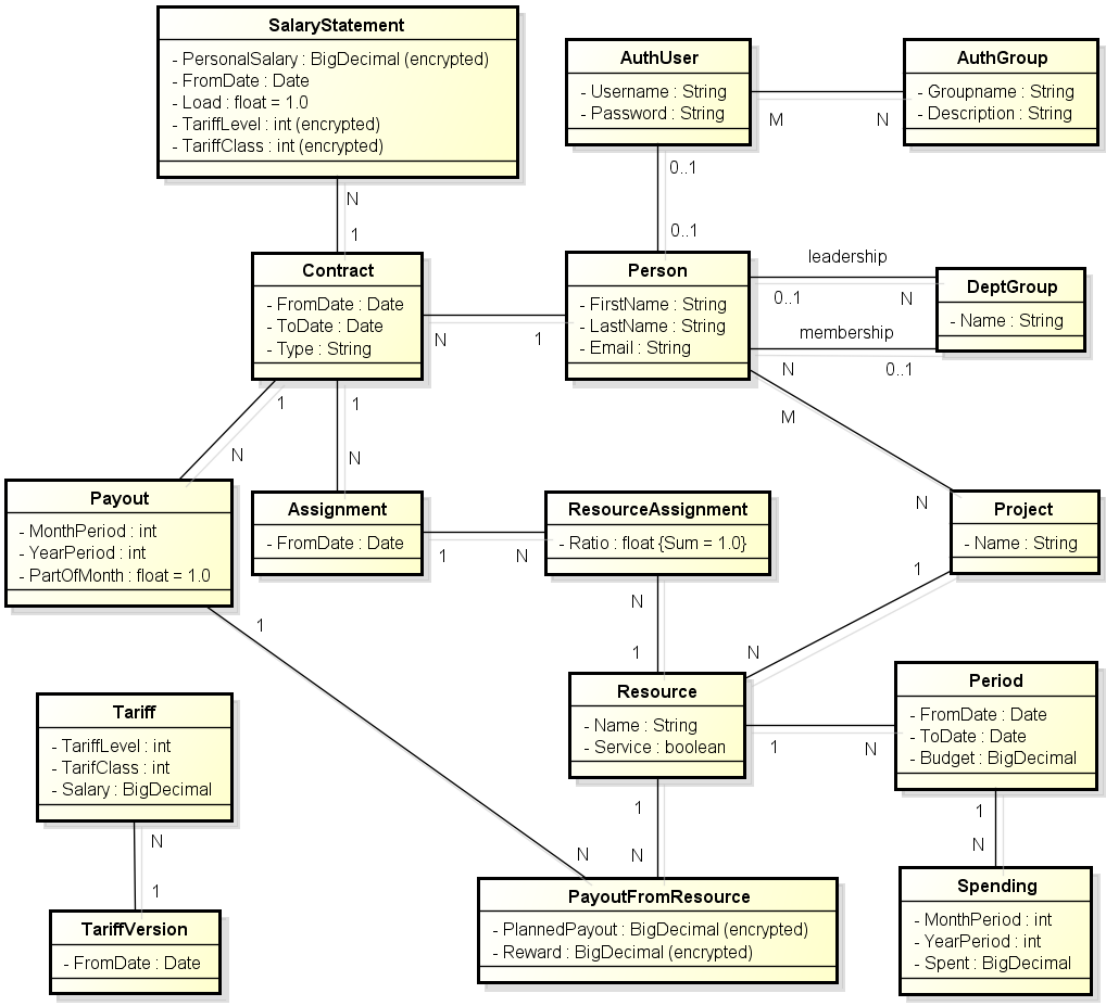
3.4 Datový model

Následující schéma představuje diagram datového modelu. V informačním systému jsou jednotlivé entity reprezentované třídami jazyka Java, a díky technologii JPA (Java Persistence API [4]) mohou být jednotlivé objekty mapovány na řádky v databázových tabulkách. Definice těchto tabulek, datové typy jednotlivých atributů a vhodné databázové dotazy jsou automaticky generovány tak, aby správně fungovaly na daném databázovém serveru. JPA také zajišťuje tvorbu pomocných tabulek, které nejsou v diagramu vidět – tedy spojovacích tabulek pro vztahy typu M:N.

Díky JPA je možné aplikaci provozovat téměř na jakékoliv relační databázi, ke které jsou k dispozici odpovídající JDBC ovladače.

Ještě bych rád zmínil jeden atribut, který jsem do diagramu nezahrnoval – `id`. Každý objekt má číselný identifikátor, podle kterého jej lze v databázi dohledat (výjimkou jsou tabulky `AuthString` a `AuthGroup`, kde jsou řádky identifikovány textovými řetězci). Číselné identifikátory jsou uložené v tzv. identity sloupci, ve kterém jsou hodnoty automaticky generovány databázovým serverem, což však vyžaduje, aby tuto operaci daný databázový server podporoval. Příslušnou funkcionalitu však nabízí většina dnes používaných databázových serverů.

V JPA existují i alternativní možnosti generování identifikátorů, lze například využít pomocnou tabulku a do ní ukládat, jaký identifikátor byl použit naposledy. Díky tomu víme, že další identifikátor v řadě je dosud nepoužitý. Identity sloupce jsem však použil proto, abych mohl do databáze vkládat nové záznamy i mimo aplikaci (například při nahrávání počátečních dat) a nemusel řešit, jaký mám záznamu přidělit identifikátor a co mám kde upravit tak, abych nezpůsobil nekonzistenci systému.



Obrázek 3.1. Datový model

3.4.1 Uživatelské účty a role

Údaje týkající se uživatelských účtů a jejich oprávnění jsou uloženy v tabulkách `AuthUser` (uživatelské účty) a `AuthGroup` (uživatelské role).

Jedná se o jediné dvě tabulky v databázi, v nichž identifikátorem (primárním klíčem) není číselná hodnota, ale textový řetězec (uživatelské jméno, resp. jméno skupiny odpovídající uživatelské roli). Pro tyto identifikátory jsem se rozhodl proto, že se podle nich v JPA velice snadno dohledávají konkrétní účty a zároveň je garantována jedinečnost daného uživatelského jména (resp. uživatelských rolí, nicméně tyto role jsou do systému automaticky přidány při prvním spuštění a nepočítá se s jejich úpravami). Zároveň jsou tyto dvě tabulky (společně s automaticky generovanou tabulkou, která definuje vztahy mezi jejich jednotlivými záznamy) uzpůsobeny tak, aby je bylo možné využít pro přihlašování přes autentizační realm aplikačního serveru.

Při vytvoření nového uživatele je mu automaticky přiřazena role „Uživatel“, která je nutná k přístupu do systému.

3.4.2 Osoby

Dalších několik tabulek se týká osob a jejich pracovních smluv. Konkrétně se jedná o tabulky `Person` (osoby), `Contract` (pracovní smlouvy) a `SalaryStatement` (platové výměry).

Osoby mohou mít několik smluv, které mají vždy dané počáteční datum (mohou mít i koncové, jinak se předpokládá, že daná smlouva je na dobu neurčitou, nebo stále platí). Ke každé smlouvě může existovat několik platových výměrů, které musí mít vždy dané počáteční datum. Platový výměr pozbývá svoji platnost automaticky v momentě, kdy začíná platit jiná verze výměru. Součástí platového výměru je atribut `load`, který uvádí velikost celkového pracovního úvazku zaměstnance.

Tabulka `SalaryStatement` je jedna ze dvou, ve které se nachází šifrované hodnoty (zde se jedná o výši osobního ohodnocení a zařazení do platové úrovně a třídy). V informačním systému se s nimi pracuje jako s běžnými číselnými hodnotami, nicméně před uložením do databáze jsou zašifrovány algoritmem AES-128 (Advanced Encryption Standard) a jsou v databázi ukládány jako pole bajtů.

3.4.3 Výzkumné kupiny

Výzkumné skupiny jsou ukládány do tabulky `Group`. Každá skupina může mít stanoveného vedoucího (nicméně není to nutné), který může zároveň vést i několik skupin. Sám však může každý člověk být členem maximálně jedné skupiny.

3.4.4 Projekty a zdroje

Detaily o projektu a jeho zdrojích se nacházejí v tabulkách `Project` (projekty), `Resource` (zdroje), `Period` (jednotlivé periody zdrojů) a `Spending` (výkazy o skutečně využitých prostředcích).

Ke každému projektu je navázán libovolný počet zdrojů, které tvoří rozpočet daného projektu, a také jeden pomocný zdroj pro účely plánování. Ke zdroji pak patří období (periody), přičemž každá perioda má jasně dané počáteční datum, konečné datum a rozpočet. Periody se nesmí překrývat a v ideálním případě na sebe přímo navazují.

Skutečně vyplacené prostředky jsou evidovány k periodám a lze je zadávat po měsících.

3.4.5 Přiřazení zdrojů k osobám

Aby bylo možné prostředky ze zdroje rozdělovat jednotlivým osobám, je nutné vědět, kdo z něj má být placen a jakou část svého úvazku danému projektu věnuje. Proto je zde tabulka `Assignment` (přiřazení) a pomocná tabulka `ResourceAssignment` (spojující jednotlivá přiřazení se zdroji).

Přiřazení fungují podobně jako dříve zmiňované platební výměry – mají dané datum počátku platnosti, a končí automaticky v momentě, kdy začne platit přiřazení nové. Každé přiřazení je prostřednictvím pomocné tabulky svázáno s jedním či více zdroji a zároveň určuje, jaká poměrná část úvazku daného pracovníka bude z tohoto zdroje hrazena.

U přiřazení je nutné, aby pokrylo celý úvazek daného zaměstnance a proto musí být splněna podmínka, že součet všech podílů na úvazku zaměstnance v rámci jednoho přiřazení musí být přesně 1.

Je možné, že v době přiřazování nemá vedoucí skupiny k dispozici všechny projekty, kterými bude zaměstnancův úvazek pokryt. V tom případě doporučuji založit pro tyto účely pomocný projekt se zdrojem a periodou s nulovým rozpočtem. V případě naplánování tohoto projektu dojde k naplánování mzdy zaměstnance z pomocného zdroje daného projektu, a tento zdroj tedy představuje dluh dané skupiny či oddělení.

3.4.6 Výplaty

Agenda výplat je reprezentována dvěma tabulkami: tabulkou `Payout` (výplaty) a pomocnou tabulkou `PayoutFromResource`, spojující výplaty se zdroji.

Pro každou smlouvu se každý měsíc vytvoří právě jedna výplata, která zároveň specifikuje, jak velká část měsíce se zaměstnanci započetla a na jak velký podíl ze mzdy stanovené platovým výměrem má tedy nárok. Tento atribut má využití v měsíci, kdy zaměstnanci začíná nebo končí pracovní smlouva a tudíž dostává výplatu jen za část měsíce.

Každá výplata je pak spojena s jedním či více zdroji, které ji financují. Ve spojovací tabulce je uvedena výplata (plánovaná) ze zdroje a případně také mimořádná odměna určená vedoucím projektu či výzkumné skupiny.

Částky vyplacené (případně naplánované) zaměstnancům ze zdrojů jsou v databázi, stejně jako v případě platových výměrů, ukládány šifrovaně, neboť se jedná o citlivé údaje.

3.4.7 Tarifní tabulka

Poslední částí datového modelu je tarifní tabulka. Její obsah je uložen ve tvou databázových tabulkách, `TariffVersion` (verze tarifní tabulky) a `Tariff` („políčka“ tarifní tabulky).

Tarifní tabulka (resp. její konkrétní verze) musí mít vždy dané datum odkdy platí, a její platnost končí automaticky v momentě, kdy začně platit nová verze. Ke každé verzi jsou pak navázány hodnoty uvedené v tarifní tabulce tak, že každá trojice <*verze tarifní tabulky, tarifní třída, tarifní stupeň*> je uložena jako samostatný záznam.

Při prvním spuštění je do databáze automaticky nahrána tarifní tabulka dle vnitřního mzdového předpisu ČVUT [5].

Kapitola 4
Implementace informačního systému

V této části práce bych rád popsal aplikaci z hlediska implementace a použitých technologií.

4.1 Použité technologie

Při implementaci informačního systému jsem použil dále popsané technologie.

4.1.1 Java 7

Informační systém je napsán v jazyce Java 7. Ta přináší oproti starším verzím některé novinky, z nichž nejvýznamnější jsou například podpora textových řetězců v konstrukci switch/case, nebo tzv. diamantový operátor. Ten umožňuje zkrácení některých zápisů při práci s generickými třídami, a díky němu lze například zápis

```
Map<Integer, List<Set<BigDecimal>>> map =
    new HashMap<Integer, List<Set<BigDecimal>>>();
```

zkrátit na mnohem kratší a čitelnější zápis

```
Map<Integer, List<Set<BigDecimal>>> map =
    new HashMap<>();
```

Teprve v období dokončování této práce se objevila Java 8 [6], která s sebou přináší například podporu lambda výrazů. Vzhledem k pokročilému stavu řešení této práce jsem však zůstal u verze 7.

4.1.2 Maven

Apache Maven je systém pro správu projektů v jazyce Java [7]. Nejčastěji se používá pro automatizaci buildu (tedy procesu, během něhož dochází k sestavení spustitelné verze programu), ale rozsah funkcí poskytovaných Mavenem je mnohem širší. Hlídá potřebné závislosti jednotlivých projektů, umožňuje spouštění testů, generování různých reportů či dokonce nasazování aplikací na servery.

Základem projektu v Mavenu je tzv. Project Object Model (v souboru `pom.xml`), ve kterém jsou definovány veškeré informace potřebné k provedení kompletního buildu projektu. Jsou zde tedy informace o projektu, jeho podprojektech a závislostech (ať již těch, které jsou potřeba pouze ke kompilaci či k testování, ale i těch, které se mají nakopírovat do výsledného souboru). Dále zde mohou být například informace o jednotlivých vývojářích nebo napojení na systémy pro správu verzí.

Nedílnou součástí Mavenu jsou přídavné moduly, tzv. pluginy. Ty obstarávají prakticky všechny činnosti, například kompilaci (`compiler`), „balení“ projektu do výsledných balíčků (`jar`, `war`, `ear`), či testování (`surefire`, `failsafe`). Kromě těchto základních pluginů jsou dostupné i další, umožňující například nasazení na různé aplikační servery či generování API dokumentace.

4.1.3 Java EE 7

Java Enterprise Edition (obvykle označovaná jako Java EE nebo dříve J2EE) [8] je sada specifikací API určená pro vývoj podnikových aplikací v jazyce Java. Na rozdíl od Java Standard Edition není její součástí implementace těchto API, ta je prováděna různými společnostmi či open-source projekty.

Součástí Javy EE je mnoho specifikací, z nichž v této práci využívám například:

- Java Persistence API (JPA)
- JavaServer Faces (JSF)
- Contexts and Dependency Injection (CDI)
- Enterprise JavaBeans (EJB)
- Bean Validation

4.1.4 Glassfish 4

Aplikace postavené na platformě Java EE lze provozovat na speciálních aplikačních serverech, z nichž nejznámější jsou například Glassfish (vyvíjený společností Oracle), WildFly (dříve známý jako JBoss, vyvíjený společností Red Hat) nebo IBM WebSphere.

Existují také servery nabízející pouze nejdůležitější součásti pro provoz webových aplikací, ale neobsahují složitější enterprise API. Příkladem takovýchto serverů jsou Apache Tomcat nebo Eclipse Jetty.

Glassfish 4 byl první aplikační server, který oficiálně podporoval Javu EE 7. V době, kdy jsem začal s vývojem informačního systému, byl zároveň jediný. Nyní je již k dispozici také WildFly 8.

4.1.5 PrimeFaces

PrimeFaces [9] je sada doplňkových komponent pro JSF (JavaServer Faces). Některé z nich lze použít jako náhradu za komponenty ze standardního JSF API (například tabulky, tlačítka či textová pole), mnoho z nich (například různé dialogy) však v původním JSF žádnou obdobu nemá.

Komponenty frameworku PrimeFaces ve velké míře využívají technologii AJAX, díky čemuž výsledné aplikace mohou být svižné a uživatelsky přívětivé. Vzhled všech komponent z PrimeFaces lze upravovat pomocí tzv. skinů.

4.1.6 PrettyFaces

PrettyFaces [10] je knihovna, která k JSF přidává tzv. hezké adresy (clean URL). Standardní adresy stránek v JSF nejsou příliš pěkné, a navíc dle mého názoru až příliš odhalují vnitřní strukturu aplikace (přinejmenším je z nich velice snadno čitelné, jak je strukturována složka s XHTML šablonami, díky čemuž je v kombinaci s případným špatným nastavením zabezpečení možné se snadno dostat ke zdrojovým kódům dané šablony).

Adresy v PrettyFaces se nastavují pomocí XML souboru, kde se pro každou stránku vytvoří záznam jako na následujícím příkladu:

```
<url-mapping id="login">
    <pattern value="/login" />
    <view-id value="/faces/login.xhtml" />
</url-mapping>
```

V případě uvedeného mapování se po zadání adresy `/login` zobrazí stejná stránka, jako by se zobrazila po zadání adresy `/faces/login.xhtml`. V rámci aplikace je možné

na tuto stránku odkazovat řetězcem `pretty:login`, který je před zobrazením nahrazen správnou adresou.

4.1.7 MySQL

Díky použití JPA by měl být informační systém schopen pracovat s daty uloženými v jakékoliv databázi, ke které je k dispozici JDBC ovladač. Pro účely vývoje a testování jsem se rozhodl použít databázový server MySQL, který je k dispozici bezplatně a je jedním z celosvětově nejpoužívanějších relačních databázových systémů.

4.2 Výběr vhodného frameworku

Relativně dlouhou dobu jsem strávil nad výběrem vhodného webového frameworku. Pro Javu existuje nespočet různých webových frameworků, v závěru jsem se však rozhodoval frameworky Vaadin [11] a JavaServer Faces (JSF) [12].

Vaadin slouží k vytváření moderních webových aplikací založených na technologii AJAX, přičemž výsledkem je stránka obsahující jeden velký skript v jazyce JavaScript a zbytek je ze serveru stahován za běhu. Výhodou je možnost webové rozhraní vytvářet velice podobným způsobem, jako například běžné okenní aplikace v prostředí Swing.

JSF je poněkud tradičnější framework tvořený komponentami vkládanými do XHTML stránky. Kromě základních komponent daných JSF specifikací (a implementovaných například projekty Mojarra nebo Apache MyFaces) existuje také mnoho projektů vyvíjejících doplňkové komponenty, jmenovitě například IceFaces nebo PrimeFaces.

Nakonec jsem se rozhodl pro použití JSF, neboť dle mého názoru nechává vývojářům více svobody a pouze vkládá části do dané XHTML šablony. Vaadin oproti tomu přebírá kontrolu nad uživatelským rozhraním a vývojáře poněkud omezuje. Dle mého názoru se mnoho moderních frameworků (nejen Javových) snaží vytvářet nad výsledným HTML kódem co největší abstrakci, zatímco já preferuji plný přístup k HTML kódu a frameworku dovolím jen to, co uznám za vhodné.

4.3 Uživatelské rozhraní

Jak již bylo zmíněno, uživatelské rozhraní je tvořeno převážně komponentami z frameworku PrimeFaces s využitím skinu vycházejícího z frontendového frameworku Bootstrap [13] (tento styl je používán v mnoha internetových aplikací a pro mnoho uživatelů je přívětivější, než klasický „systémový“ vzhled HTML komponent.

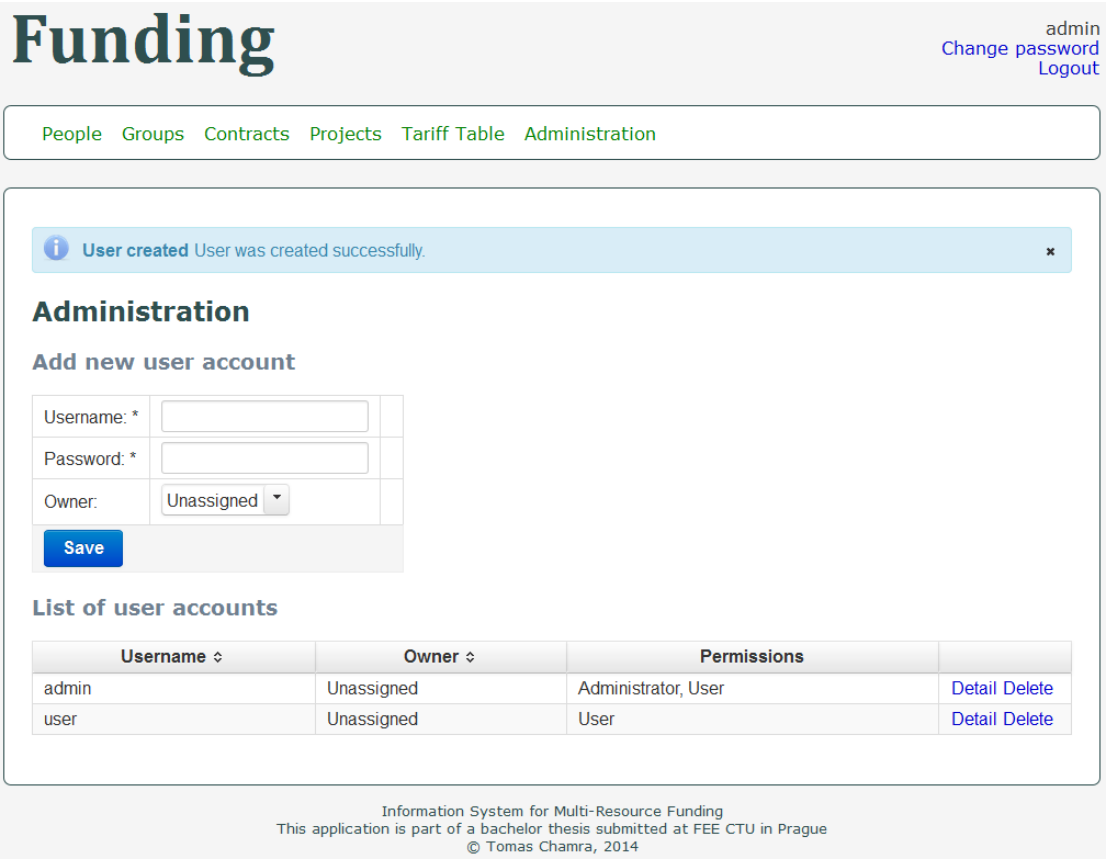
U polí pro zadávání emailové adresy jsem využil jednu z novinek v Java EE 7 – HTML 5 passthrough. Díky tomu je možné skrze JSF komponenty „protlačit“ vlastní atributy až do výsledného HTML kódu. U textového pole tak lze například změnit atribut `type` z `text` na některý z nových typů vstupních prvků, jako jsou `email` nebo `number`. Interpretace takovýchto vstupních polí zavisí na konkrétním webovém prohlížeči, například u vstupu typu `email` se na mobilních zařízeních obvykle použije jiné rozložení klávesnice.

4.4 Rozložení uživatelského rozhraní

V celé aplikaci je dodržován jednotný vzhled stránky:

- V levé horní části stránky je logo aplikace

- V pravé horní části stránky jsou informace o přihlášeném uživateli a odkazy na přihlášení, odhlášení a změnu hesla (v závislosti na tom, zda je uživatel přihlášen či nikoliv)
- Pod logem aplikace se nachází hlavní menu aplikace
- V hlavní části stránky je samotný obsah, v jehož horní části mohou být zobrazeny různé informační či chybové hlášky
- Ve spodní části stránky je pak patička s informacemi o aplikaci



Obrázek 4.1. Ukázka rozložení stránka - administrace uživatelů

4.4.1 Seznam obrazovek

Aplikace se ovládá prostřednictvím obrazovek uvedených v následujícím seznamu. U každé obrazovky je poznamenán také název souboru se šablonou, ze které je stránka vykreslována. Všechny tyto šablony fungují tak, že vkládají určité komponenty stránky do hlavní šablony v souboru `layout.xhtml`.

- Úvodní stránka (existují dvě verze – pro přihlášené a napřihlášené uživatele) – `dashboard.xhtml`
- Přihlášení – `login.xhtml`
- Seznam osob – `secure/people-list.xhtml`
  - Detail a úprava osoby se seznamem smluv – `secure/people-detail.xhtml`
  - Smazání osoby – `secure/people-delete.xhtml`
- Seznam smluv – `secure/contracts-list.xhtml`
  - Detail a úprava smlouvy se seznamem platových výměrů, přiřazení ke zdrojům a výplat – `secure/contracts-detail.xhtml`

    - Detail a úprava platového výměru – `secure/salarystatements-detail.xhtml`
    - Smazání platového výměru – `secure/salarystatements-delete.xhtml`
    - Detail a úprava přiřazení – `secure/assignments-detail.xhtml`
    - Smazání přiřazení – `secure/assignments-delete.xhtml`
    - Detail a úprava výplaty – `secure/payouts-detail.xhtml`
    - Smazání výplaty – `secure/payouts-delete.xhtml`
  - Smazání smlouvy – `secure/contracts-delete.xhtml`
- Seznam skupin – `secure/groups-list.xhtml`
  - Detail a úprava skupiny – `secure/groups-detail.xhtml`
  - Smazání skupiny – `secure/groups-delete.xhtml`
- Seznam projektů – `secure/projects-list.xhtml`
  - Detail a úprava projektu se seznamem zdrojů a možností naplánovat či přeplánovat projekt – `secure/projects-detail.xhtml`
    - Detail a úprava zdroje se seznamem period a možností naplánovat zdroj – `secure/resources-detail.xhtml`
      - Detail a úprava periody se zadáváním výkazů o skutečném čerpání – `secure/periods-detail.xhtml`
      - Smazání periody – `secure/periods-delete.xhtml`
    - Smazání zdroje – `secure/resources-delete.xhtml`
  - Smazání projektu – `secure/projects-delete.xhtml`
- Seznam verzí tarifní tabulky – `secure/tariff-list.xhtml`
  - Zobrazení a úprava tarifní tabulky – `secure/tariff-detail.xhtml`
  - Smazání verze tarifní tabulky – `secure/tariff-delete.xhtml`
- Adminstrace uživatelů – `secure/admin/admin-users.xhtml`
  - Detail uživatele – `secure/admin/admin-edituser.xhtml`
  - Smazání uživatele – `secure/admin/admin-deleteuser.xhtml`
- Změna hesla – `secure/password.xhtml`

4.5 Lokalizace

Celá aplikace je vytvořena tak, aby ji bylo možné snadno lokalizovat do jiných jazyků. Využívá k tomu standardní mechanismy platformy Java EE a veškeré texty jsou ukládány do speciálních souborů s příponou `.properties`. Pro lokalizaci do dalších jazyků pak stačí vytvořit kopie těchto souborů, přeložit je a správně je pojmenovat. Poté již zbývá jen změnit výchozí jazykové prostředí (locale) v konfiguračních souborech aplikace.

Jedinou součástí vyžadující samostatnou lokalizaci jsou některé komponenty z knihovny PrimeFaces. Ty mají například svůj vlastní kalendář v jazyce JavaScript a je nutné pro ně při načtení stránky nastavit správné popisky. To lze provést přidáním lokalizovaných textů do souboru `locale.js`, který je vkládán do každé stránky obsažené v informačním systému.

4.6 Zabezpečení

Přihlašování uživatelů a částečně i ověřování uživatelských oprávnění je řešeno přes autentizační realm v aplikačním serveru Glassfish. Ten je napojen na stejnou databázi, ve které uchovává data informační systém. V konfiguraci server je pak nastavena adresa přihlašovací stránky, a seznam oprávnění potřebných pro přístup k jednotlivým složkám či souborům. Pokud se je pak uživatel pokusí otevřít, je vyzván aby se přihlásil. V případě, že se přihlášený uživatel pokouší dostat na stránku, k níž nemá oprávnění (v případě tohoto informačního systému se jedná zejména o stránky určené pouze pro administrátory), zobrazí se mu hlášení o chybě a přístup mu nebude umožněn.

Data o platech zaměstnanců jsou do databáze ukládána v šifrované podobě, zde je použit algortimus AES-128. Šifrovací klíč (heslo) je uložen ve zdrojovém kódu aplikace, protože data musí být přístupná pro všechny uživatele s patřičnými oprávněními.

Přístupová hesla uživatelů jsou hashována algoritmem SHA-256.

Kapitola 5
Testování

Po dokončení implementace (nebo ideálně ještě v jejím průběhu) je nutné informační systém řádně otestovat a zjistit zda funguje tak, jak je od něj očekáváno.

5.1 Úrovně testování

Software lze testovat na několika úrovních. Zde se však různé zdroje velmi liší a každý rozděluje testy do trochu jiných úrovní, například dle [14] se používají tyto tři úrovně:

- **Unit testy** – testují malé části kódu (například jednotlivé funkce či metody)
- **Integrační testy** – testují větší celky složené například z několika metod a ověřují, zda dohromady fungují správně
- **Systémové testy** – testují aplikaci jako celek

5.2 Testování v Mavenu

Automatické testy lze během build procesu v Mavenu spouštět dvěma pluginy:

- **Surefire** – slouží pro spouštění unit testů
- **Failsafe** – slouží pro spouštění integračních testů

Hlavní rozdíl mezi těmito dvěma pluginy je fakt, že se testy provádí v různých fázích buildu. Surefire spouští unit testy po kompilaci jednotlivých tříd, a v případě selhání některého z nich build selže a už dále nepokračuje.

Naproti tomu Failsafe provádí testy až po „zabalení“ zkompilovaných a dalších souborů do výstupního archivu (například s příponou JAR, WAR, nebo EAR) a případné selhání některého z testů tedy vytvoření archivu nezabrání.

Oba pluginy umožňují spouštět testy vytvořené pomocí frameworků jUnit a TestNG.

5.3 Testování informačního systému

Při testování systému v této bakalářské práci jsem používal automatické a ruční testování.

Z automatických testů se jednalo o unit testy, nicméně bylo možné je použít pouze ve velmi omezené míře. Většina metod v mém informačním systému slouží buď jako prostředník mezi webovou stránkou a EJB, nebo se jedná přímo o EJB, která obvykle pracují s databází a testují se tedy obtížněji.

I tyto složitější metody se dají testovat, a to buď pomocí dočasných mock objektů, nebo s použitím pokročilejších testovacích frameworků, jako je například Arquillian. Ten k testování používá upravené verze aplikačních serverů a databázových serverů, a umožňuje v nich testovat i kompikovanější EJB pracující s databázemi. Po konzultaci s vedoucím bakalářské práce jsme však konstatovali, že použití těchto technik je mimo rozsah této bakalářské práce.

Většinu funkcionality informačního systému tedy testuji manuálně, pomocí systémových testů které jsem navrhl na základě jednotlivých případů užití.

5.4 Systémové testy

V této sekci jsem navrhl systémové testy, jejichž účelem je otestovat, zda informační systém pracuje správně v soulady s případy užití danými v části 3.3 této práce.

V případě, že je vstupní podmínkou testu přihlášený uživatel, je nutné tento test vykonat jako uživatel s odpovídajícími oprávněními, která jsou blíže popsána v případěch užití.

5.4.1 Přihlášení, odhlášení, změna hesla

1. **Přihlášení uživatele**
   Vstupní podmínky: Uživatel není přihlášen
   Postup: V pravé horní části úvodní stránky klikněte na odkaz „Click here to login“, vyplňte přihlašovací údaje a formulář odešlete.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přihlášení.

2. **Odhlášení uživatele**
   Vstupní podmínky: Uživatel je přihlášen
   Postup: V pravé horní části úvodní stránky klikněte na odkaz „Logout“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odhlášení.

3. **Změna vlastního hesla se správným zadáním původního hesla**
   Počáteční podmínky: Uživatel je přihlášen
   Postup: V pravé horní části úvodní stránky klikněte na odkaz „Change password“, vyplňte své dosavadní heslo a nové heslo.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změne hesla. Uživatel se může přihlásit s novým heslem.

4. **Změna vlastního hesla se špatným zadání původního hesla**
   Počáteční podmínky: Uživatel je přihlášen
   Postup: V pravé horní části úvodní stránky klikněte na odkaz „Change password“, vyplňte špatně své dosavadní heslo a nové heslo.
   Předpokládaný výstup: Zobrazí se chybová hláška.

5.4.2 Osoby

1. **Přidání nové osoby**
   Počáteční podmínky: Uživatel je přihlášen
   Postup: V menu klikněte na „People“ a na zobrazené stránce vyplňte formulář pro přidání nové osoby. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu osob se objeví nově vytvořená osoba.

2. **Úprava osoby**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna osoba
   Postup: Zobrazte detail kterékoliv osoby, změňte některý z údajů a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně údajů. V seznamu osob

a detailu dané osoby jsou nyní zobrazeny aktualizované údaje.

3. **Smazání osoby**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna osoba
   Postup: V seznamu osob klikněte na odkaz „Delete“ a operaci potvrďte.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu osob se již daná osoba nenachází.

5.4.3 Smlouvy

1. **Přidání nové smlouvy**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna osoba
   Postup: V menu klikněte na „Contracts“ a na zobrazené stránce vyplňte formulář pro přidání nové smlouvy a přiřaťe ji k existující osobě. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu smluv se objeví nově vytvořená smlouva.

2. **Úprava smlouvy**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna smlouva
   Postup: Zobrazte detail kterékoliv smlouvy, změňte některý z údajů a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně údajů. V seznamu smluv a detailu dané smlouvy jsou nyní zobrazeny aktualizované údaje.

3. **Smazání smlouvy**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna smlouva
   Postup: V seznamu smluv klikněte na odkaz „Delete“ a operaci potvrďte.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu smluv se již daná smlouva nenachází.

5.4.4 Platové výměry

1. **Přidání nového platového výměru**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna smlouva
   Postup: V detailu smlouvy vyplňte formulář pro přidání nového platového výměru a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu platových výměrů se objeví nově vytvořený platový výměr.

2. **Úprava platového výměru**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jeden platový výmě
   Postup: Zobrazte detail kterékoliv platového výměru, změňte některý z údajů a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně údajů. V seznamu platových výměrů a detailu daného platového výměru jsou nyní zobrazeny aktualizované údaje.

3. **Smazání platového výměru**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jeden platový výměr
   Postup: V seznamu platových výměrů k dané smlouvě klikněte na odkaz „Delete“ a operaci potvrďte.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu platových výměrů se již daný platový výměr nenachází.

5.4.5 Skupiny

1. **Přidání nové skupiny**
   Počáteční podmínky: Uživatel je přihlášen
   Postup: V menu klikněte na „Groups“ a na zobrazené stránce vyplňte formulář pro přidání nové skupiny. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu skupin se objeví nově vytvořená skupina.

2. **Úprava skupiny**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna skupina
   Postup: Zobrazte detail kterékoliv skupiny, změňte některý z údajů a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně údajů. V seznamu skupin a detailu dané skupiny jsou nyní zobrazeny aktualizované údaje.

3. **Smazání skupiny**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna skupina
   Postup: V seznamu skupin klikněte na odkaz „Delete“ a operaci potvrďte.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu skupin se již daná skupina nenachází.

5.4.6 Projekty

1. **Přidání nového projektu**
   Počáteční podmínky: Uživatel je přihlášen
   Postup: V menu klikněte na „Projects“ a na zobrazené stránce vyplňte formulář pro přidání nového projektu. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu projektů se objeví nově vytvořený projekt. Společně s tímto projektem se automaticky vytvoří servisní zdroj a v něm perioda.

2. **Úprava projektu**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jeden projekt
   Postup: Zobrazte detail kteréhokoliv projektu, změňte některý z údajů a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně údajů. V seznamu projektů a detailu daného projektu jsou nyní zobrazeny aktualizované údaje.

3. **Smazání projektu**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jeden projekt

Postup: V seznamu projektů klikněte na odkaz „Delete“ a operaci potvrďte.
Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu projektů se již daný projekt nenachází.

5.4.7 Zdroje

1. **Přidání nového zdroje**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jeden projekt
   Postup: V detailu projektu vyplňte formulář pro přidání nového zdroje. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu zdrojů se objeví nově vytvořený zdroj.

2. **Úprava zdroje**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jeden ne-servisní zdroj
   Postup: Zobrazte detail kteréhokoliv (ne-servisního) zdroje, změňte některý z údajů a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně údajů. V seznamu zdrojů a detailu daného zdroje jsou nyní zobrazeny aktualizované údaje.

3. **Smazání zdroje**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jeden ne-servisní zdroj
   Postup: V seznamu zdrojů klikněte na odkaz „Delete“ a operaci potvrďte.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu zdrojů se již daný zdroj nenachází.

5.4.8 Periody

1. **Přidání nové nepřekrývající se periody**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jeden ne-servisní zdroj
   Postup: V detailu zdroje vyplňte formulář pro přidání nové periody tak, aby se perioda časově nepřekrývala s jinou periodou. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu period se objeví nově vytvořená perioda a v detailech této periody vynulovaný výkaz o skutečném čerpání zdrojů.

2. **Přidání nové překrývající se periody**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jeden ne-servisní zdroj
   Postup: V detailu zdroje vyplňte formulář pro přidání nové periody tak, aby se perioda časově překrývala s jinou periodou. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se chybová hláška.

3. **Úprava periody**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna perioda
   Postup: Zobrazte detail kteréhokoliv periody ne-servisního zdroje, změňte některý z údajů a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně údajů. V seznamu period

a detailu dané periody jsou nyní zobrazeny aktualizované údaje. Výkaz o skutečném čerpání se změnil tak, aby obsahoval měsíce, které jsou zahrnuty v aktualizované periodě.

4. **Smazání periody**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna perioda
   Postup: V seznamu period klikněte na odkaz „Delete“ a operaci potvrďte.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu period se již daná perioda nenachází.

5.4.9 Výkazy o skutečném čerpání

1. **Zadání údaje o skutečném čerpání**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna perioda
   Postup: Zobrazte detail kteréhokoliv periody ne-servisního zdroje, v některém měsíci změňtě údaj o skutečném čerpání a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně údajů. V detailu periody se změní aktuální dostupná částka (available budget).

5.4.10 Přiřazování zdrojů k osobám

1. **Vytvoření nového přiřazení**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna smlouva
   Postup: Zobrazte detail kterékoliv smlouvy, vyplňte formulář pro přidání nového přiřazení (assignmentu) a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu přiřazení se objeví nově vytvořené přiřazení.

2. **Nastavení přiřazení (se správným poměrem)**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedno přiřazení a jeden ne-servisní zdroj
   Postup: Zobrazte detail přiřazení, přidejte do něj zdroj, nastavte mu poměr (ratio) na hodnotu 1.0 a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném uložení.

3. **Nastavení přiřazení (se špatným poměrem)**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedno přiřazení a jeden ne-servisní zdroj
   Postup: Zobrazte detail přiřazení, přidejte do něj zdroj, nastavte mu poměr (ratio) na hodnotu 0.5 a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o neúspěšném pokusu o uložení.

4. **Smazání přiřazení**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedno přiřazení
   Postup: V seznamu přiřazení klikněte na odkaz „Delete“ a operaci potvrďte.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu přiřazení se již dané přiřazení nenachází.

5.4.11 Plánování výplat

1. **Plánování výplat ze zdroje**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedno přiřazení mezi smlovou a zdrojem, k tomuto zdroji existuje perioda
   Postup: Zobrazte detail zdroje a klikněte na tlačíko „PLAN“.
   Předpokládaný výstup: Naplánují se výplaty z daného zdroje po dobu, po kterou je vždy první den měsíce pokryt některou z period. Dostupný zůstatek zdroje nebude záporný (pokud již dříve nebyl). Pokud ve zdroji nebylo dost finančních prostředků, pak byly tyto prostředky naplánovány z pomocného zdroje projektu, ke kterému zdroj patří.

2. **Plánování výplat z projektu**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedno přiřazení mezi smlovou a zdrojem, k tomuto zdroji existuje perioda
   Postup: Zobrazte detail projektu a klikněte na tlačíko „PLAN“.
   Předpokládaný výstup: Naplánují se výplaty pro všechny zdroje projektu. U jednotlivých zdrojů se předpokládá stejné chování jako v předchozím testu.

3. **Přeplánování výplat z projektu**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedno přiřazení mezi smlovou a zdrojem, k tomuto zdroji existuje perioda
   Postup: Zobrazte detail projektu, klikněte na tlačíko „RE-PLAN“ a operaci potvrďte.
   Předpokládaný výstup: Smažou se všechny budoucí výplaty ze zdrojů daného projektu. Dále se očekává stejné chování jako v předchozím testu.

5.4.12 Výplaty

1. **Manuální přidání nové výplaty**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna smlouva
   Postup: V detailu smlouvy vyplňte formulář pro přidání nové výplaty. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu výplat se objeví nově vytvořená výplata.

2. **Úprava výplaty**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna výplata
   Postup: Zobrazte detail kterékoliv výplaty, změňte její složení z údajů a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném uložení. V detail výplaty jsou nyní zobrazeny aktualizované údaje.

3. **Smazání výplaty**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna výplata
   Postup: V seznamu výplat klikněte na odkaz „Delete“ a operaci potvrďte.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu výplat se již daná výplata nenachází.

5.4.13 Administrace uživatelů

1. **Přidání nového uživatele nenavázaného na konkrétní osobu**
   Počáteční podmínky: Uživatel je přihlášen
   Postup: V menu klikněte na „Administration“ a na zobrazené stránce vyplňte formulář pro přidání nového uživatele. V Poli „Owner“ ponechte „Unassigned“. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu uživatelů se objeví nově vytvořený uživatel s rolí „User“.

2. **Přidání nového uživatele navázaného na konkrétní osobu**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna osoba bez uživatelského účtu
   Postup: V menu klikněte na „Administration“ a na zobrazené stránce vyplňte formulář pro přidání nového uživatele. V Poli „Owner“ zvolte některou osobu. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu uživatelů se objeví nově vytvořený uživatel s rolí „User“. Ve formuláři pro přidání nového uživatele již nelze zvolit osobu, k níž je navázán nově vytvořený uživatelský účet.

3. **Změna hesla uživatele**
   Počáteční podmínky: Uživatel je přihlášen
   Postup: Zobrazte detail uživatele, vyplňte nové heslo a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně. Uživatel se může přihlásit novým heslem.

4. **Změna oprávnění uživatele**
   Počáteční podmínky: Uživatel je přihlášen
   Postup: Zobrazte detail uživatele, změňte zvolená oprávnění a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně. V seznamu uživatelů a v detailu uživatele jsou vidět nové role.

5. **Smazání uživatele**
   Počáteční podmínky: Uživatel je přihlášen
   Postup: V seznamu uživatelů klikněte na odkaz „Delete“ a operaci potvrďte.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu uživatelů se již daný uživatel nenachází a nelze se za něj přihlásit. Pokud byl navázán na osobu, pak lze k této osobě vytvořit nový uživatelský účet.

5.4.14 Tarifní tabulka

1. **Přidání nové verze tarifní tabulky**
   Počáteční podmínky: Uživatel je přihlášen
   Postup: V menu klikněte na „Tariff Table“ a vyplňte formulář pro vytvořené nové verze. Poté klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném přidání. V seznamu verzí se objeví nově vytvořená verze.

2. **Úprava tarifní tabulky**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna verze tarifní tabulky
   Postup: Zobrazte kteroukoliv verzi tarifní tabulky, změňte hodnoty a klikněte na tlačítko „Save“.
   Předpokládaný výstup: Zobrazí se hláška o úspěšné změně. V tarifní tabulce jsou nyní zobrazeny aktualizované údaje.

3. **Smazání verze tarifní tabulky**
   Počáteční podmínky: Uživatel je přihlášen, existuje alespoň jedna verze tarifní tabulky
   Postup: V seznamu verzí tarifní tabulky klikněte na odkaz „Delete“ a operaci potvrďte.
   Předpokládaný výstup: Zobrazí se hláška o úspěšném odstranění. V seznamu verzí se již daná verze nenachází.

Kapitola 6
Možná rozšíření do budoucna

V této kapitole bych rád zmínil několik návrhů na funkcionalitu, kterou by systém mohl do budoucna obsahovat, nicméně prozatím nebyla z důvodu zadaného rozsahu práce implementována.

6.1 Dohody s hodinovou sazbou

Informační systém v podobě, v jaké je nyní, umí evidovat pouze klasické pracovní smlouvy s pevně danou měsíční mzdou dle tarifní tabulky (k níž se ještě přičítají osobní ohodnocení a případné odměny).

Někdy však zaměstnanci mohou pracovat na dohody konané mimo procovní poměr, přičemž zákoník práce [15] zná dvě takové:

- dohoda o provedení práce
- dohoda o pracovní činnosti

V těchto případech jsou zaměstnanci placeni jejich hodinovou sazbou a výsledná mzda se tedy počítá na základě počtu odpracovaných hodin v daném měsíci. Celou věc ještě může komplikovat fakt, že některé zdroje mají oddělené prostředky na financování pracovních smluv a dohod.

Možným řešením této situace je následující postup:

1. zadávání typu smlouvy upravit tak, aby bylo zřejmé, zda se jedná o měsíční nebo hodinovou sazbu
2. do platového výměru přidat hodinovou sazbu a vždy vyplnit pouze hodnoty, které jsou pro danou smlouvu smysluplné
3. do periody (a výkazu čerpání z periody) přidat prostředky určené na dohody
4. do výplaty ze zdroje přidat počet odpracovaných hodin
5. upravit plánování výplat tak, aby na základě typu smlouvy rozhodlo o správném postupu při výpočtech

Po patřičných úpravách by pak mělo být možné s dohodami počítat. Nicméně, tyto dohody bývají často na velice omezenou dobu a pro případné plánování výplat je nutné předem znát odhad počtu odpracovaných hodin.

6.1.1 Stipendia a příplatky za vedení

Z některých zdrojů mohou být vyplácena stipendia pro studenty a někteří pracovníci mohou pobírat nárokové příplatky (například příplatek za vedení). Tyto další složky mzdy je možné do systému přidat podobně jako v předchozím případě.

1. do platového výměru přidat výši stipendia a příplatku za vedení
2. do periody (a výkazu čerpání z periody) přidat prostředky určené na stipendia
3. do výplaty ze zdroje přidat pravidelné stipendium, mimořádné stipendium a příplatek za vedení

4. upravit plánování výplat tak, aby bylo možno plánovat i tyto složky mzdy (s výjimkou mimořádného stipendia, jež se plánuje pouze ručně, stejně jako v případě mimořádných odměn)

6.2 Import dat

Pokud se systém rozroste a bude evidovat výplaty většího množství zaměstnanců, může jejich zadávání být dosti časově náročné. Proto by bylo ideální data importovat například z účetního systému (nicméně, tato data jsou tak citlivá, že si o nich administrátoři všech informačních systémů – s výjimkou těch opravdu nejdůležitějších – mohou pravděpodobně nechat jen zdát).

Za předpokladu, že se podaří data získat (ideálně ve formátu XML, který je na takovéto přenosy mezi různými systémy pravděpodobně nejvhodnější), pak už zbývá jen doprogramovat modul, která příslušná data ze souboru načte a uloží do databáze.

Alternativní možností (a převděpodobně rychlejší) by bylo data například pomocí XSLT transformace převést do formátu SQL a do databáze je importovat přímo. Zde je však složitější odchytit případné nesrovnalosti v datech, a proto bych doporučoval je spíše načítat postupně a před vložením do databáze kontrolovat, zda je vše tak, jak má být.

6.3 Externí ověřování přihlašovacích údajů

Dalším logickým požadavkem by mohlo být napojení na již existující systémy sloužící pro ověřování přihlašovacích údajů, jako například adresářové služby (skrze protokol LDAP) nebo jiné SSO (single sign-on) systémy.

Ověřování přihlašovacích údajů je řešeno na úrovni aplikačního serveru Glassfish a pokud by tedy bylo nutné jen změnit server/technologii vůči které se údaje ověřují, pak stačí v administrační konzoli Glassfishe upravit nastavení autentizačního realmu.

V dnešní době také existují některé SSO systémy, které z bezpečnostních důvodů vyžadují zadávání údajů přímo do aplikace na serverech jejich provozovatele. Informační systém by v takovém případě bylo nutné upravit tak, aby uživatele na příslušný server přesměroval a po přihlášení zpracoval údaje, které od serveru obdrží.

Kapitola 7
Závěr

V této práci jsem se věnoval problematice vícezdrojového financování, jak je praktikováno na elektrotechnické fakultě ČVUT. Nejprve jsem objasnil, jaké jsou vazby mezi jednotlivými zaměstnanci, výzkumnými skupinami, projekty a jejich zdroji. Dále jsem zavedl způsob skládání výplat tak, aby mohly být dopředu naplánovány.

Navrhl jsem informační systém pro správu vícezdrojového financování, který eviduje všechny potřebné informace nutné pro plánování jednotlivých výplat. Definoval jsem případy užití tohoto informačního systému a vytvořil jeho datový model.

Navržený systém jsem implementoval pomocí technologie Java Enterprise Edition 7 a připravil jej k provozu na aplikačním serveru Glassfish 4. Při práci mne mírně zdržely chyby v implementaci některých komponent platformy Java EE a mírně zmatečné změny, které byly ve specifikacích této platformy v nedávné době provedeny. Všechny tyto překážky se mi však podařilo překonat.

Implementovaný systém jsem následně otestoval pomocí systémových testů, které jsem k tomuto účelu navrhl. Vzhledem k obtížným možnostem testování kódu využívajícího součásti Java EE se mi v rámci této práce nepodařilo ve větší míře využít automatizované jUnit testy. V práci jsem však zmínil možné způsoby provádění těchto testů.

Na závěr práce jsem navrhl několik možných rozšíření informačního systému, která by v budoucnu mohla být implementována. V příloze práce jsem pak uvedl návod, jak informační systém zprovoznit.

Literatura

[1] Antonio Goncalves. *Beginning Java EE 7*. Apress, New York City, NY, USA, 2013.

[2] Josh Juneau. *Java EE 7 Recipes: A Problem-Solution Approach*. Apress, New York City, NY, USA, 2013.

[3] Arun Gupta. *Java EE 7 Essentials*. O'Reilly Media, Sebastopol, CA, USA, 2013.

[4] Java Persistence 2.1 specification (JSR 338). Citováno dne 21.5.2014. `https://jcp.org/en/jsr/detail?id=338`.

[5] Vnitřní mzdový předpis ČVUT v Praze ze dne 29. května 2013.

[6] Oracle: What's New in JDK 8. Citováno dne 21.5.2014. `http://www.oracle.com/technetwork/java/javase/8-whats-new-2157071.html`.

[7] Sonatype. *Maven: The Definitive Guide*. O'Reilly Media, Sebastopol, CA, USA, 2008.

[8] The Java EE 7 Tutorial. Citováno dne 21.5.2014. `http://docs.oracle.com/javaee/7/tutorial/doc/home.htm`.

[9] PrimeFaces Showcase. Citováno dne 21.5.2014. `http://www.primefaces.org/showcase`.

[10] PrettyFaces. Citováno dne 21.5.2014. `http://ocpsoft.org/prettyfaces`.

[11] Book of Vaadin. Citováno dne 21.5.2014. `https://vaadin.com/book`.

[12] JavaServer Faces 2.2 specification (JSR 344). Citováno dne 21.5.2014. `https://jcp.org/en/jsr/detail?id=344`.

[13] Bootstrap. Citováno dne 21.5.2014. `http://getbootstrap.com`.

[14] Pierre Bourque and Richard E. Fairley. *Guide to the Software Engineering Body of Knowledge, Version 3.0*. IEEE Computer Society, 2014.

[15] Zákon č. 262/2006 Sb.

[16] GoJava: JDBC security realm with glassfish and jsf. Citováno dne 21.5.2014. `http://jugojava.blogspot.com/2011/02/jdbc-security-realm-with-glassfish-and.html`.

Příloha A
Návod k instalaci

Ke zprovoznění informačního systému je nutné mít nainstalován tento software:

- JDK 1.7 (v době psaní této bakalářské práce již byl vydán JDK 1.8, nicméně ten není v prozatím oficiálně podporován v Glassfishi 4.0)
- Glassfish 4.0
- databázový server kompatibilní s JPA (například MySQL)
- Maven 3 (pro kompilaci aplikace ze zdrojových kódů)

A.1 Kompilace

Pokud nemáte k dispozici zkompilovanou aplikaci (v souboru s příponou EAR), je nutné ji ručně zkompilovat. Pokud nemáte nainstalovaný systém Maven, nainstalujte jej. Poté přejděte do složky se zdrojovými kódy aplikace a vykonejte příkaz `mvn verify`, čímž se aplikace zkompiluje a provedou se všechny testy. Pokud vše proběhlo v pořádku, pak se ve složce `funding-ear/target` nachází soubor `funding-ear-1.0.ear`, který lze nasadit na server.

A.2 Nastavení databáze

Nyní je potřeba nastavit údaje o databázi, do níž bude informační systém ukládat svá data. Glassfish sám o sobě se k databázi připojit neumí, potřebuje k tomu mít JDBC ovladač (v případě MySQL se jmenuje MySQL Connector/J). První tedy tento ovladač (v souboru s příponou JAR) nakopírujte do složky `glassfish/lib`.

Spusťte Glassfish příkazem `asadmin start-domain` a otevřte administrační konzoli běžící standardně na portu 4848. Pokud jste neměnili nastavení zabezpečení, pak by k přístupu ze stejného počítače, na kterém server běží, neměly být požadovány žádné přihlašovací údaje.

V pravém menu zvolte `Resources -> JDBC -> JDBC Connection Pools` a tlačítkem `New...` vytvořte nové databázové spojení podle následujícího vzoru (zvolte však svůj databázový systém):

Pool Name	FundingPool
Resource Type	javax.sql.DataSource
Database Driver Vendor	MySql

Tabulka A.1. Nastavení JDBC poolu

Po kliknutí na tlačítko `Next` se zobrazí další formulář, v němž na řádku `Ping` zaškrtněte `Enabled` (díky čemuž se při pokračování automaticky otestuje spojení s databází) a ve spodní části (Additional Properties) doplňte údaje o konkrétním databázovém serveru, zde jsou parametry závislé na použitém databázovém ovladači (v tabulce jsou opět uvedeny parametry pro MySQL server).

portNumber	číslo portu, například 3306
databaseName	jméno databáze, například funding
serverName	adresa databázového serveru, například localhost
user	přihlašovací jméno k databázovému serveru
password	heslo k databázovému serveru, nesmí být prázdné
characterEncoding	UTF-8

Tabulka A.2. Nastavení JDBC poolu - Advanced Properties

Je možné, že se tabulka automaticky předvyplní několika stovkami parametrů, pak doporučuji je všechny smazat a potřebné řádky přidat ručně.

Po uložení tlačítkem `Finish` by se mělo zobrazit potvrzení o úspěšném pingu, čímž je spojení vytvořeno.

Dalším krokem je vytvoření databázového zdroje, díky čemuž bude možné k databázi přistupovat z aplikací. V menu tedy zvolte `Resources -> JDBC -> JDBC Resources` vytvořte nový zdroj s parametry dle tabulky a uložte jej.

JNDI Name	jdbc/funding
Pool Name	FundingPool

Tabulka A.3. Nastavení JDBC zdroje

A.3 Autentizace

Přihlašování užívatelů a kontrolu oprávnění nechává aplikace provádět Glassfish, a proto je nutné nastavit způsob ověřování přihlašovacích údajů. V menu tedy zvolte `Configurations -> server-config -> Security -> Realms` a založte nový autentizační realm. Paremetry zadejte přesně tak, jak jsou uvedeny v následující tabulce a nastavení uložte.

Realm Name	funding-realm
Class Name	com.sun.enterprise.security.ee.auth.realm.jdbc.JDBCRealm
JAAS Context	jdbcRealm
JNDI	jdbc/funding
User Table	authuser
User Name Column	username
Password Column	password
Group Table	authusergroup
Group Name Column	groupname
Password Enc. Alg.	SHA-256

Tabulka A.4. Nastavení autentizačního realmu

A.4 Nahrání aplikace na server

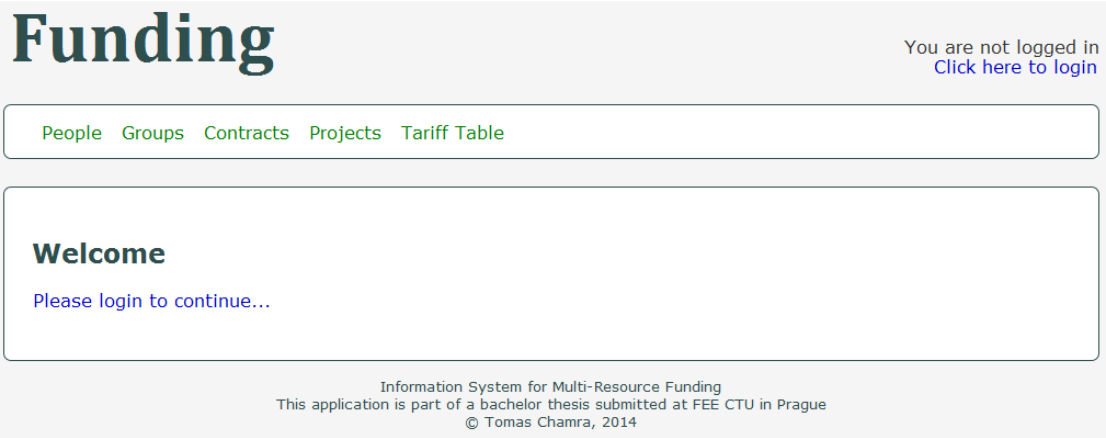
Posledním krokem je nahrání aplikace na server. V administrační konzoli Glassfishe zvolte možnost `Applications` a stiskněte tlačítko `Deploy`. V zobrazeném formuláři nahrajte EAR soubor s aplikací a stísknutím `OK` aplikaci spusťte. Tato operace bude pravděpodobně trvat několik desítek sekund.

Alternativní možností je nahrání aplikace z příkazové řádky následujícím příkazem:

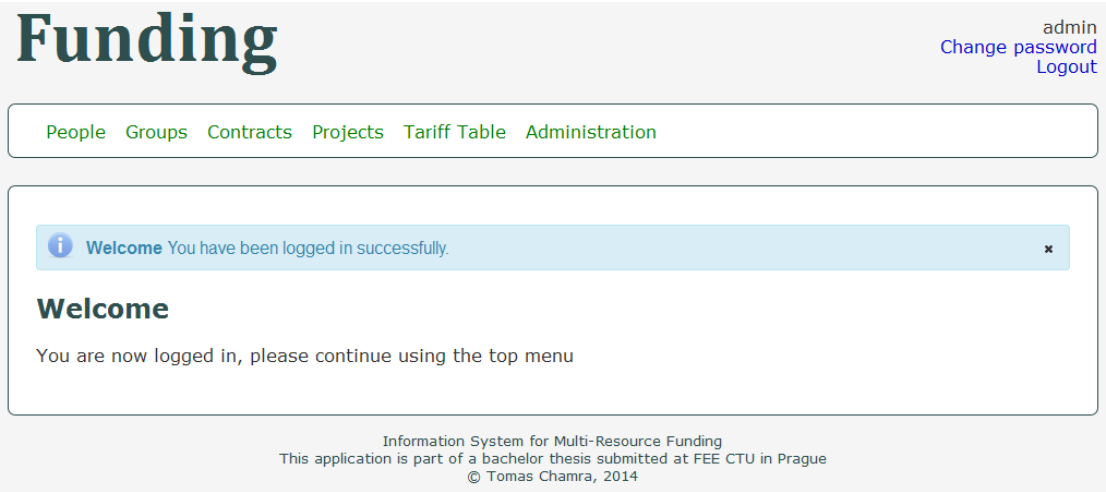
```
asadmin deploy funding-ear-1.0.ear
```

A.5 První přihlášení

Po dokončení můžete otevřít adresu `http://localhost:8080/funding-web`, kde aplikace běží (za předpokladu, že máte standardně nastavený Glassfish běžící na Vašem stroji). Pro první přihlášení do systému použijte přihlašovací jméno `admin` a heslo `heslo`. Doporučuji ihned po přihlášení heslo změnit pomocí odkazu `Change password` v pravé horní části stránky.



Obrázek A.1. Úvodní stránka informačního systému před přihlášením



Obrázek A.2. Úvodní stránka informačního systému po přihlášení

Příloha B
Protokol o provedení testů

Aplikace byla otestována dne 21. května 2014.

B.1 Unit testy

Všechny unit testy byly úspěšně provedeny pluginem Surefire.

B.2 Systémové testy

Systémové testy byly provedeny na aplikačním serveru Glassfish 4.0, databázovém serveru MySQL 5.6 a s použitím internetového prohlížeče Mozilla Firefox 29.

Všechny systémové testy byly provedeny úspěšně.

Přihlášení uživatele	OK
Odhlášení uživatele	OK
Změna vlastního hesla se správným zadáním původního hesla	OK
Změna vlastního hesla se špatným zadání původního hesla	OK
Přidání nové osoby	OK
Úprava osoby	OK
Smazání osoby	OK
Přidání nové smlouvy	OK
Úprava smlouvy	OK
Smazání smlouvy	OK
Přidání nového platového výměru	OK
Úprava platového výměru	OK
Smazání platového výměru	OK
Přidání nové skupiny	OK
Úprava skupiny	OK
Smazání skupiny	OK
Přidání nového projektu	OK
Úprava projektu	OK
Smazání projektu	OK
Přidání nového zdroje	OK
Úprava zdroje	OK
Smazání zdroje	OK
Přidání nové nepřekrývající se periody	OK
Přidání nové překrývající se periody	OK
Úprava periody	OK
Smazání periody	OK

Tabulka B.5. Provedené systémové testy, první část

Zadání údaje o skutečném čerpání	OK
Vytvoření nového přiřazení	OK
Nastavení přiřazení (se správným poměrem)	OK
Nastavení přiřazení (se špatným poměrem)	OK
Smazání přiřazení	OK
Plánování výplat ze zdroje	OK
Plánování výplat z projektu	OK
Přeplánování výplat z projektu	OK
Manuální přidání nové výplaty	OK
Úprava výplaty	OK
Smazání výplaty	OK
Přidání nového uživatele nenavázaného na konkrétní osobu	OK
Přidání nového uživatele navázaného na konkrétní osobu	OK
Změna hesla uživatele	OK
Změna oprávnění uživatele	OK
Smazání uživatele	OK
Přidání nové verze tarifní tabulky	OK
Úprava tarifní tabulky	OK
Smazání verze tarifní tabulky	OK

Tabulka B.6. Provedené systémové testy, druhá část

Příloha C
Obsah přiloženého CD

Nedílnou součástí této bakalářské práce je CD s následujícím obsahem:

bakalarskaprace.pdf	elektronická verze této práce ve formátu PDF
funding/	zdrojové kódy informačního systému
funding-ear-1.0.ear	zkompilovaná verze informačního systému
tex/	soubory v TeXu, ze kterých byla tato práce vygenerována

Tabulka C.7. Obsah přiloženého CD

Příloha D
Zkratky

AES	Advanced Encryption Standard
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CDI	Contexts and Dependency Injection
EJB	Enterprise JavaBeans
HTML	HyperText Markup Language
JDBC	Java Database Connectivity
JDK	Java Development Kit
JPA	Java Persistence API
JSF	JavaServer Faces
LDAP	Lightweight Directory Access Protocol
POM	Project Object Model
SHA	Secure Hash Algorithm
SQL	Structured Query Language
SSO	Single sign-on
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations