

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Zdeněk B a l á k

Studijní program: Kybernetika a robotika (bakalářský)

Obor: Robotika

Název tématu: Otevření portálu Spolupráce s průmyslem pro více univerzit

Pokyny pro vypracování:

1. Seznamte se s portálem Spolupráce s průmyslem, který je využíván Fakultou informačních technologií Českého vysokého učení technického v Praze.
2. Navrhněte a implementujte rozšíření tohoto portálu, které umožní používání jedné instance aplikace více univerzitami a umožní sdílení zdrojů (studenti, experti, průmysloví partneři).
3. Provedte audit dat a datových zdrojů a obohatte doménový model aplikace tak, aby podporoval zapojení více univerzit.
4. Připravte seznam nutných technických úkonů, které musí univerzita provést, aby mohla být zapojena do portálu. Dále se soustředte na umožnění nastavení úrovně sdílení výše uvedených zdrojů jednotlivých univerzit.
5. Držte se zásad pro návrh a realizaci softwarového díla. Zdrojový kód řádně zdokumentujte a otestujte.

Seznam odborné literatury:

- [1] Fišer Aleš - Posílení zpětné vazby z průmyslu na univerzity - Praha 2014
- [2] JSR 168: Portler Specification 2014. Dostupné z: <http://jcp.org/en/jsr/detail?id=168>
- [3] JSR 286: Portler Specification 2.0. 2014. Dostupné z: <http://jcp.org/en/jsr/detail?id=286>
- [4] Liferay - homepage. 2014. Dostupné z: <https://www.liferay.com/>

Vedoucí bakalářské práce: Ing. Aleš Fišer

Platnost zadání: do konce letního semestru 2015/2016

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 5. 2. 2015

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ
KATEDRA KYBERNETIKY



Bakalářská práce

Otevření portálu Spolupráce s průmyslem pro více univerzit

Zdeněk Balák

Vedoucí práce: Ing. Aleš Fišer

22. května 2015

Poděkování

Rád bych touto cestou poděkoval Ing. Aleši Fišerovi za cenné rady nejen při psaní této práce, ale hlavně po celou dobu mého působení na Fakultě informačních technologií - ČVUT.

Prohlášení

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 22. května 2015

.....

České vysoké učení technické v Praze
Fakulta elektrotechnická

© 2015 Zdeněk Balák. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě elektrotechnické. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Balák, Zdeněk. *Otevření portálu Spolupráce s průmyslem pro více univerzit*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, 2015.

Abstrakt

Tato práce pojednává o technickém zapojení partnerských fakult do portálu Spolupráce s průmyslem.

Uvádí přístupy ke spolupráci s průmyslem ostatních fakult a popisuje jakým způsobem je pojata spolupráce v portálu SSP. Dále popisuje jednotlivé uživatelské role, jejich funkce a seznamuje čtenáře se základními technologiemi používanými k vývoji a provozu této aplikace.

Jako další práce uvádí seznam úprav potřebných k technickému zapojení partnerských fakult do portálu.

V závěru sepisuje kroky, které budou sloužit jako návod pro fakulty, které se chystají zapojit do této aplikace

Klíčová slova Spolupráce s průmyslem, Liferay, portál, portlet, rest API, Java EE

Abstract

This thesis discuss about technical realization of including fellow faculties into portal Cooperation with Industry.

It presents other faculties' methods of cooperating with industry. Thesis Describes how is this problem dealt with in CWI and presents application's user roles and lists their purpose. It also describes basics technologies used to develop and operate this application.

Next it lists technical changes needed to be processed in order to enable including more than one faculty into the portal.

Thesis ends with step-by-step manual for faculties that are trying to join CWI.

Keywords Cooperation with industry, Liferay, portal, portlets, rest API, Java EE

Obsah

Úvod	1
Cíle práce	1
Členění práce	2
1 Teoretická část	3
1.1 Rešerše	3
1.2 O portálu Spolupráce s průmyslem	5
1.3 Technologie	7
2 Analýza	15
2.1 Formální požadavky	15
2.2 Struktura projektu	15
3 Návrh a realizace řešení	19
3.1 Obohacení datového modelu	19
3.2 Data pro SSP potřebná	23
3.3 Wireframes	24
3.4 Restové API	24
3.5 Technické roky, které musí fakulta vykonat	26
4 Testování	29
4.1 Jednotkový test	29
4.2 Integrovaný test	30
Závěr	31
Literatura	33
A Seznam použitých zkratk	35
B Specifikace JSON objektů	37

C Wireframes	41
C.1 Správa fakult	41
C.2 Správa fakulty	42
C.3 Profil fakulty	43
D Nové pohledy	45
D.1 Správa fakult	45
D.2 Správa fakulty	47
D.3 Profil fakulty	48
D.4 Vybírání schopností	49
D.5 Veřejný profil fakulty	50
D.6 Mapování schopností na předmět	50
E Obsah příloženého CD	51

Seznam obrázků

2.1	Schéma projektu SSP	16
3.1	Logický datový model SSP	20
3.2	Obohacený logický datový model SSP	21
3.3	Návrh BI	22
3.4	Návrh UDB	23
C.1	Wireframe - Správa fakult	41
C.2	Wireframe - Správa fakulty	42
C.3	Wireframe - Profil fakulty	43
D.1	GUI - Správa fakult	45
D.2	GUI - Vytvoření nové univerzity	46
D.3	GUI - Vytvoření nové fakulty	46
D.4	GUI - Správa fakulty	47
D.5	GUI - Profil fakulty	48
D.6	GUI - Vybírání schopností	49
D.7	GUI - Veřejný profil fakulty	50
D.8	GUI - Mapování schopností na předmět	50

Úvod

Portál Spolupráce s průmyslem vznikl v létě roku 2013. První nasazení proběhlo v zimním semestru téhož roku. Pilotní verze SSP byla spuštěna pouze pro studenty doktorského studia a měla za úkol zmapovat požadavky studentů a průmyslových partnerů na vylepšení.

V roce 2014 přibyla do portálu nová grafika a poté byl portál zpřístupněn všem studentům FIT - ČVUT.

V současné době portál hostí přes 800 studentů a téměř 100 průmyslových partnerů. Dokončených zadání je bez mála 20 a dalších více než 20 je v procesu vypracovávání.

V poslední době se také vyjednává vstup mimouniverzitních fakult do SSP, stejně jako zpřístupnění portálu pro více fakult ČVUT. Aplikace ale nebyla navržena, aby rozlišovala příslušnosti k fakultám a s tím spojené sdílení zdrojů¹.

Tato práce se zabývá právě touto problematikou a klade si za úkol technickou realizaci zapojení více fakult do SSP a přehlednou dokumentaci pro připojující se fakulty.

Cíle práce

- Audit datových zdrojů BI modulu, odstranění redundance
- Obohacení datového zdroje aplikace, aby podporoval zapojení více fakult
- Úprava rozhraní a implementace tak, aby bylo respektováno nových skutečností²
- Sepsání technických kroků nutných vykonat před vstupem fakulty do portálu

¹Různé fakulty mohou mít různé množiny spolupracujících průmyslových partnerů

²Př.: Student již nebude moci vidět všechna zadání, ale pouze zadání určená pro jeho fakultu

Členění práce

Teoretická část - Tato část se zabývá mapováním spolupráce ostatních fakult. Popisuje základní myšlenku portálu SSP a seznamuje s uživatelskými rolemi projektu. Dále popisuje použité technologie.

Analýza - Definuje funkční a nefunkční požadavky, popisuje strukturu projektu.

Návrh a realizace řešení - Tato část se zaměřuje na obohacení datových struktur, navrhuje nová grafická uživatelská rozhraní, vypisuje restové služby, ať už poskytované aplikací nebo fakultou. V závěru vypisuje technické kroky, podle kterých může fakulta postupovat při jejím zapojení do SSP.

Testování - Tato kapitola popisuje způsoby testování, obsahuje ukázkou jednotkového a integračního testu.

Teoretická část

1.1 Rešerše

Hlavním cílem této práce je rozšířit portál Spolupráce s průmyslem tak, aby bylo možné jeho užívání více fakultami. Proto je důležité zjistit, na jaké úrovni spolupracují s průmyslem ostatní významné fakulty s podobným zaměřením jako FIT - ČVUT.

1.1.1 Fakulta elektrotechnická - České vysoké učení technické

Na internetových stránkách fakulty se nachází sekce „Firmy a média“ s podsekcí „Nabídka spolupráce pro firmy“. [1] Zde se lze dozvědět, že fakulta rozlišuje následující způsoby spolupráce:

- Nabídka spolupráce s katedrami a laboratořemi
spolupráce s externími firmami za účelem prezentace: značky, produktu, personální politiky (lidských zdrojů), vědeckých a výzkumných aktivit
- Možnosti spolupráce v oblasti reklamy, PR a HR

Dále vyvěšuje fakulta na svých stránkách pracovní nabídky pro své studenty a absolventy [2]

1.1.2 Fakulta informačních technologií - Vysoké učení technického v Brně

Na internetových stránkách fakulty se nachází přímo sekce „Spolupráce“ [3] Fakulta rozlišuje následující formy spolupráce:

- Společná příprava výzkumných projektů národních i mezinárodních
Fakulta řeší výzkumné projekty národní (MPO, MV, GAČR apod.) i mezinárodních (FP7 EU a FP7-ARTEMIS EU).

1. TEORETICKÁ ČÁST

- Spolupráce ve výuce studentů

Vedení závěrečných prací, hostování přednášek, vlastní předměty připravující studenty k praxi

- Zakázky, služby a licence v oblasti informačních technologií

Fakulta přijímá zakázky v oblasti informačních technologií - zajímavé z pohledu výzkumu informačních technologií. Podle potřeby poskytuje licence na výsledky a software, které má k dispozici.

- Hosting laboratoří a výzkumných pracovišť

Pro případy dlouhodobé spolupráce je možné zřídit laboratoř přímo v prostorách fakulty pro lepší komunikaci se zaměstnanci fakulty a hlavně se studenty

Stejně jako Fakulta elektrotechnická - ČVUT, vyvěšuje nabídky pracovních pozic pro své studenty a absolventy [4]

1.1.3 Fakulta elektrotechnická - Západočeská univerzita v Plzni

V rámci projektu „Partnerství v elektrotechnice a strojírenství“ vznikl na fakultě portál „Virtuální kontaktní místo“ [5] Podnikům, firmám a dalším institucím nabízí spolupráci v následujících oblastech:

- řešení technologických, technických a dalších problémů praxe
závěrečné práce a smluvní výzkum
- Spolupráce v oblasti přípravy absolventů na jejich uplatnění v praxi
- Zapojení do výzkumných a vývojových programů financovaných z veřejných zdrojů
- Vědecko-technické služby, expertízy, posudky
- Realizaci kurzů profesního vzdělávání podle potřeb firem, školení, semináře
- Propagace firmy při veřejných akcích pořádaných fakultou anebo s účastí fakulty

Dále fakulta nabízí nabídky brigád a zaměstnání pro své studenty a absolventy [6]

1.1.4 Fakulta informatiky a statistiky - Vysoká škola ekonomická v Praze

Fakulta nabízí průmyslovým partnerům podporu v následujících odvětvích: [7]

- kvantitativní metody

postupy spojené s nasazováním statistických, demografických metod a metod operačního výzkumu pro podporu řízení aktivit a procesů organizací – včetně vyhodnocování investic a projektů

- informatika organizací.

podpora aktivit řízení podnikové informatiky a inovativního využívání informačních technologií ve všech oblastech řízení organizace

1.1.5 Závěr řešerše

Všechny fakulty se snaží spolupracovat s průmyslem různými způsoby, mapovat aktuální pracovní nabídky. Fakulta elektrotechnická na Západočeské univerzitě má dokonce portál, který shromažďuje veškeré informace o spolupráci fakulty. Portál Spolupráce s průmyslem velice usnadňuje spolupráci mezi studenty a průmyslem. Zároveň se snaží využít této spolupráce v rámci výuky. Všechna data o spolupráci se zachovávají, čímž se shromažďují informace o aktuálním dění v praxi. Rozšíření portálu SSP na více fakult by zajistilo větší zdroje na jeho další rozvoj. Dále by se rozšířilo spektrum a hlavně počet řešitelů, což by mělo mít za následek větší aktivitu ze strany průmyslu.

1.2 O portálu Spolupráce s průmyslem

Portál Spolupráce s průmyslem je webová aplikace dostupná všem studentům ČVUT - Fakulty informačních technologií. Jejím cílem je efektivně propojit „spolupráci studentů s průmyslem“ s jejich působením na fakultě.

Někteří studenti pracují na částečný úvazek mimo fakultu na zadáních mnohdy vhodných jako zadání semestrálních prací předmětů, které studují, a přesto musí navíc vypracovávat smyšlené zadání, kde jediná motivace spočívá ve splnění předmětu.

Někteří studenti nemají dost vysoké sebevědomí nebo zkušenosti na to, ucházet se o pozici, či jen jednorázový úkol, kde by získali potřebnou praxi pro budoucí zaměstnání.

Pro oba případy je tu SSP, která spojuje studenty, průmyslové partnery a učitele. Výhoda spočívá v tom, že učitelé mohou vypsaná zadání průmyslových partnerů schvalovat do svých předmětů jako semestrální nebo dokonce závěrečnou práci a studenti mají možnost požádat učitele o radu při řešení těchto

prací. Zároveň fakulta pomůže studentům s právní stránkou spolupráce. Postará se, aby studenti dostali předem domluvenou částku, a zaštituje i sepsání smluv.

1.2.1 Základní role v SSP

1.2.1.1 Za průmysl:

- Referent průmyslového partnera

Referent průmyslového partnera spravuje profil společnosti. Přidává a odebírá zadavatele společnosti. Typicky bývá každý referent za instituci zároveň i zadavatel. Společnost může mít více referentů.

- Zadavatel

Zadavatel vytváří zadání společnosti, definuje požadované výstupy, zdroje pro řešitele, časovou náročnost, finanční odměnu, termíny přihlášení a odevzdání. Dále definuje jednotlivé role z množiny: řešitel, programátor, manažer, analytik, datový vědec, datový inženýr a nastaví jejich požadované schopnosti (programování, matematika, komunikace, . . .) v rozmezí 1-5 hvězdiček.

K jednotlivým zadáním přijímá nebo odmítá řešitele na základě jejich profilu popřípadě osobní schůzky. Má také možnost nominovat ke svému zadání studenty, kteří mají svůj profil nastavený jako veřejný nebo se kterými již v minulosti spolupracoval.

Při ukončení spolupráce má možnost ohodnotit spolupráci se studenty, čímž ovlivní jejich profil.

1.2.1.2 Za univerzitu:

- Student

Student vystupuje v SSP jako řešitel. Má možnost slovně se ohodnotit na svém profilu a nastavit subjektivní hodnocení svých schopností (nehodnoceno - 5 hvězdiček). Mimo to se každému studentovi vypočítává hodnocení jeho schopností na základě jeho studijních výsledků a hodnocení, které získá od zadavatele po dokončení zadání. Tomuto hodnocení říkáme garantované/objektivní.

- Učitel

Učitel reprezentuje v SSP vazbu mezi spoluprací a studiem. Každý učitel může ke svým předmětům schvalovat zadání, která se tematicky shodují s náplní předmětu a nabízet je jako semestrální, popřípadě i závěrečnou práci. Toto schválení může provést z vlastní iniciativy nebo jako odpověď na nominaci předmětu k zadání, kterou může provést jakýkoliv uživatel SSP.

Dále je jeho úkolem poskytovat studentům řešícím jim schválené zadání konzultace, které jim pomohou úspěšně vypracovat řešení.

- Referent za univerzitu

Referent za univerzitu zastává funkci administrátora, přidává studenty, průmyslové partnery, experty, předměty a jejich učitele. Dohlíží na dodržování termínů, vyplácení odměn, zařizuje veškerou administrativu spojenou s využíváním SSP (NDA, smlouvy, ...)

- Expert

Expert je zaměstnanec fakulty placený za odbornou pomoc průmyslovým partnerům. Měl by mít přehled o vyučovaných předmětech fakulty, o schopnostech studentů a o tom, jak by mělo vypadat zadání, aby nalákalo co možná nejvíce řešitelů.

1.3 Technologie

SSP je vyvíjen na portálu Liferay 6.1.2 do něj se instalují portlety psané v JSF. Závislost mezi objekty je řízena frameworkem Spring. Backend je psaný v javě a s postgresovou databází komunikuje pomocí mapování Hibernate. Jednotlivé technologie jsou dále podrobněji popsány.

1.3.1 Liferay

Liferay je open-source portál založený na jazyce java. Je distribuovaný pod licencí GNU Lesser General Public Licence [8]. Portál je webová aplikace, která svému uživateli poskytuje jednotným způsobem a centralizovaně informace z různých zdrojů, které uživatele zajímají nebo se ho nějakým způsobem týkají.

Portál je hostitelem pro další webové aplikace zvané „portlety“. Může zobrazovat různé portlety pro uživatele s různým oprávněním.

Liferay dále mimo jiné zajišťuje správu uživatelských účtů, jednotný vzhled pomocí theme a zabezpečení.

1.3.2 Portlet

Java portlety jsou webové komponenty umožňující integraci webových aplikací a portálů. Portlety jsou používány jako výměnné komponenty uživatelského rozhraní poskytující prezentační vrstvu pro informační systém. Součinnost portálů a portletů je zajišťována pomocí API, které je definováno v Java Portlet Specification. Díky tomuto API mohou být portlety spuštěny na jakémkoliv portálu založeném na Java EE bez ohledu na jeho implementaci. [9]

Dynamicky generovaný obsah portletu se nazývá fragment, je psán ve značkovacím jazyce (HTML, XHTML, WML). Portlet může generovat rozdílný

fragment pro různé uživatele (př.: Studentovi se na stránce pro vytvoření zadání místo formuláře pro vytvoření zadání zobrazí „Nepovolený přístup na stránku“).

Životní cyklus portletu:

- Portlet vzniká metodou `init()` - je volána hned po vytvoření instance portálu, specifikuje například inicializační parametry
- Implementuje metodu `render()` - generuje obsah podle aktuálního stavu portletu
- Implementuje metodu `processAction()` - volána na základě interakce uživatele (kliknutí na odkaz, ...)
- Implementuje metodu `processEvent()` - reaguje na událost vyvolanou jiným portletem
- Implementuje metodu `processResource()` - slouží k renderování menší části stránky (př.: ajaxová volání)
- Zaniká metodou `destroy()` - dochází k uvolňování prostředků a zdrojů

1.3.3 Java Server Faces

Java Server Faces je framework, který je součástí Javy EE. JSF umožňuje elegantně oddělit uživatelské rozhraní od aplikační logiky. Uživatelské rozhraní se definuje pomocí speciálních XML tagů. Těm se předávají data z aplikační logiky pomocí Java beanů.

- Příklad managed bean můžete vidět v ukázce 1.1
- Příklad XML tagů můžete vidět v ukázce 1.2

V SSP dále tyto beany komunikují s backendem aplikace, kde dochází k předávání dat s databází

JSF definuje následující životnosti bean:

- `SessionScope`
Tato instance žije po celou relaci uživatele se serverem
- `RequestScope`
Tato instance žije pouze na jeden dotaz
- `ApplicationScope`
Tato instance žije od jejího prvního zavolání jakýmkoliv uživatelem po smazání webové aplikace

- ViewScope

Tato instance žije od jednoho načtení webové stránky po druhé. V ssp je tento scope nejvíce využívaný, protože dokáže uchovávat informace při různých ajaxových volání pro jednu návštěvu stránky a zároveň nealokuje paměť na déle než právě na tu jednu návštěvu

```
@ManagedBean(name = "bean")
@RequestScoped
public class AssignmentsBean implements PortletPageBean {

    private List<AssignmentPreviewDto> assignments;

    public List<AssignmentPreviewDto> getAssignments() {
        return assignments;
    }

    @Override
    public Status isUserAuthorized() {
        // init assignmentPreviewDtos
        return Status.forAuthorized(CommonUtil.getSecurityHelper()
            .isReferentForIndustry());
    }

    public String createUrl(Long id) {
        final String url = "...initURL"
        return url;
    }
}
```

Ukázka kódu 1.1: Příklad managed bean

Z ukázky je možné vidět, že beana načte data z backendu (v tomto případě seznam náhledů zadání), která následně využívá xhtml soubor pro předání informací uživateli.

```
<pcc:secureContent portletPageBean="#{bean}">
  <h1>#{languages[ 'filteredAssignmentsTitle ' ]}</h1>
  <h:form class="form-horizontal">
    <ui:repeat value="#{bean.assignments}" var="assignment">
      <h3 class="title">
        <h:outputLink value="#{bean.createUrl(assignment.id)}">
          #{assignment.title} - #{assignment.institutionName}
        </h:outputLink>
      </h3>
    </ui:repeat>
  </h:form>
</pcc:secureContent>
```

Ukázka kódu 1.2: Příklad XHTML

1.3.4 Objektově relační mapování - Hibernate

Objektově relační mapování je v softwarovém inženýrství technika, která zajišťuje automatickou konverzi dat mezi relační databází a objektově orientovaným jazykem.

Java Persistence API je standard jazyka java, který umožňuje ORM. JPA bylo definováno v rámci JSR 220. [10] Hibernate je jednou z implementací JPA.

Entita 1.3 je objekt, který reprezentuje data v databázi. Typicky reprezentuje jednu tabulku v databázi a každá její instance reprezentuje jeden řádek té tabulky. Musí být anotována anotací Entity. Entity mezi sebou mají vztah:

- OneToOne - Instance má referenci na jednu třídu
- OneToMany - Instance má referenci na více tříd
- ManyToOne - Více instancí má referenci na jednu třídu
- ManyToMany - Více instancí má referenci na více tříd

Tyto vztahy mohou být jak jednostranné, tak oboustranné.

Do databáze se poté přistupuje pomocí instance třídy EntityManager 1.4, která má tyto základní metody:

- persist(entity)
Uloží entitu/řádek do databáze
- remove(entity)
Vymaže entitu/řádek z databáze
- find(class, id)
Najde v databázi řádek s primárním klíčem (id) a vrátí jeho instanci

```
@Entity
public class Assignee extends Trackable implements Tagable {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long assigneeId;

    @OneToOne
    @JoinColumn(name = "sspUserId")
    private SspUser sspUser;

    @OneToMany(mappedBy = "assignee")
    private List<SponsorCooperationEvaluation>
        sponsorCooperationEvaluations;

    @OneToMany(mappedBy = "assignee")
```



```

private List<AssigneeSkill> assigneeSkills;

@OneToMany
@JoinColumn(name = "assigneeId")
private List<AssigneeText> assigneeTexts;

//getters and setters
}

```

Ukázka kódu 1.3: Příklad entity

Z ukázky je vidět, že entita Assignee je propojena s entitou SspUser přes sloupec sspUserId. Tato struktura přesně kopíruje strukturu databáze, kde tabulka assignee obsahuje cizí klíč sspuser_id. Obdobně to funguje u všech atributů.

```

@PersistenceContext
private EntityManager entityManager;

@Override
@Transactional
public void createExpert(String username, Long facultyId) {
    final ExpertFaculty expertFaculty = new ExpertFaculty();
    final Faculty faculty =
        entityManager.find(Faculty.class, facultyId);
    expertFaculty.setFaculty(faculty);
    entityManager.persist(expertFaculty);
    final Expert expert = new Expert();
    expert.setDeleted(false);
    expertFaculty.setExpert(expert);
    entityManager.persist(expertText);
    entityManager.persist(expert);
}

```

Ukázka kódu 1.4: Příklad EntityManager

Z ukázky je možné vidět, jak se používají dříve zmíněné metody entity manageru, konkrétně find() a persist()

Další způsob přístupu do databáze, který často používáme je pomocí Hibernate criteria API. 1.5 Tento způsob podporuje rozmanitější filtrování a je vhodný zejména pro vyhledávání více záznamů nebo chceme-li filtrovat podle něčeho jiného než primárního klíče.

```

@Override
@Transactional
public List<AssigneeToAdminDto> getAllAssignee(Long facultyId)
{
    final Criteria criteria = hibernateCriteriaCreator.
        createCriteria(Assignee.class, "assignee");

    criteria.createAlias("assignee.sspUser", "sspUser");
}

```

```
criteria.createAlias("assignee.assigneeFaculties", "
    assigneeFaculty");
criteria.createAlias("assigneeFaculty.faculty", "faculty")
;

criteria.add(Restrictions.eq("assignee.deleted", false));
criteria.add(Restrictions.eq("faculty.facultyId",
    facultyId));

criteria.addOrder(Order.asc("sspUser.username"));

final List<Assignee> assignees = criteria.list();
return convert(assignees);
}
```

Ukázka kódu 1.5: Příklad hibernate criteria API

1.3.5 Dependency Injection a Spring Framework

Pokud je požadováno v objektu využívat služby jiného objektu, musí se hostící objekt postarat o jeho životní cyklus. Ve větších aplikacích by ale inicializace cizích objektů přímo uvnitř objektu způsobovala chaos. Obzvláště jsou-li objekty hojně provázány. Navíc pro změnu implementace rozhraní je nutné přepsat inicializaci v každém objektu zvlášť.

Pro tyto případy je vhodné použít techniku Dependency injection, která přesune zodpovědnost za vytvoření a provázání objektů z aplikace na framework. Díky tomu se dají lehce zaměňovat implementace rozhraní bez nutnosti zasahovat do kódu. Také programátorovi ulehčuje testování a přináší pořádek do životního cyklu objektů.

V projektu SSP se k tomuto používá framework Spring. [11] Pro definici závislostí a životního cyklu objektů se používají XML soubory. 1.7

Níže lze vidět ukázkou vkládání závislostí pomocí anotace Autowired. Za povšimnutí stojí, že se rozhraní HibernateCriteriaCreator a SspUserConverter nemusí inicializovat v kódu, ale o všechno se stará kontejner podle instrukcí z xml konfiguračního souboru.

```
public class ExpertServiceImpl implements ExpertService {

    @Autowired
    private HibernateCriteriaCreator hibernateCriteriaCreator;

    @Autowired
    private SspUserConverter sspUserConverter;

    @Override
    @Transactional
    public List<SspUserDto> findAllForReferent(Long facultyId) {
        final Criteria criteria = hibernateCriteriaCreator.
            createCriteria(SspUser.class, "sspUser");
        //filter, join, ...
    }
}
```

```
        final List<SspUser> list = criteria.list();  
        return sspUserConverter.convert(list);  
    }  
}
```

Ukázka kódu 1.6: Příklad definice závislosti

```
<beans>  
  <bean class="cz.cvut.fit.ssp.domain.service.  
    HibernateCriteriaCreator"/>  
  <bean class="cz.cvut.fit.ssp.domain.service.form.  
    ExpertServiceImpl"/>  
  <bean class="cz.cvut.fit.ssp.domain.converter.  
    SspUserConverterImpl"/>  
</beans>
```

Ukázka kódu 1.7: Příklad XML konfiguračního souboru

Analýza

2.1 Formální požadavky

2.1.1 Funkční

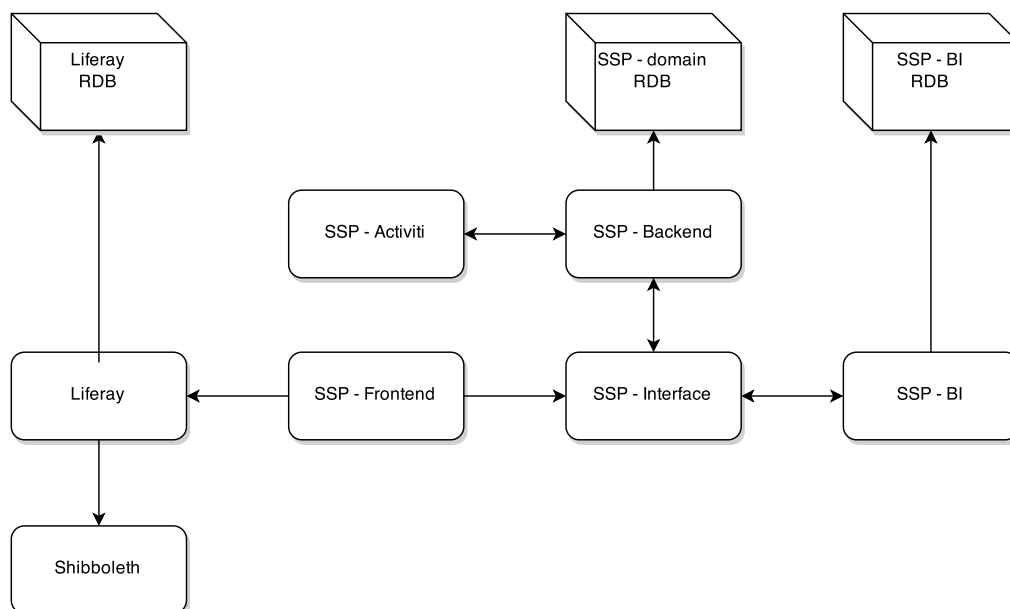
- Administrátor má možnost vytvářet nové fakulty
- Aplikace bude umožňovat nastavit vazby mezi univerzitou a průmyslovým partnerem
- Systém nebude zobrazovat uživatelům fakulty žádné informace průmyslových partnerů, který s danou fakultou nespolečně spolupracují
- Průmyslový partner bude mít možnost zveřejnit zadání pro menší počet fakult než s kterými spolupracuje
- Student bude mít možnost patřit k více fakultám
- Systém umožňuje fakultě spravovat svůj profil
- Systém umožňuje fakultě nahrávat záznamy o svých předmětech hromadně přes restová API

2.1.2 Nefunkční

- Implementace v jazyku Java a respektování technologií, které používá portál Spolupráce s průmyslem.
- Portlety lze nasadit na Liferay v. 6.1.2

2.2 Struktura projektu

Pro navržení správného řešení je nutné seznámit se se strukturou projektu SSP. Tato struktura je naznačena schématem 2.2.



Obrázek 2.1: Schéma projektu SSP

2.2.1 SSP - Frontend

Frontend projektu obsahuje portlety, které jsou instalovány do Liferaye. Je psán podle již zmíněných standardů JSF. 1.3.3

2.2.2 SSP - Interface

Vrstva interface definuje jednak objekty typu Data transfer object 2.1, druhak definuje rozhraní 2.2 mezi jednotlivými moduly (frontend - backend, backend - bi).

```
public class FacultyDto {  
    private Long facultyId;  
    private String facultyName;  
    private String facultyAbbrev;  
    //getters and setters
```

```
}
```

Ukázka kódu 2.1: Příklad DTO

```
public interface ExpertAdminService {  
  
    void createExpert(String username, String createdBy, Long  
        facultyId);  
  
    void deleteExpert(String username, String createdBy, Long  
        facultyId);  
  
}
```

Ukázka kódu 2.2: Příklad rozhraní

2.2.3 SSP - Backend

Vrstva backend implementuje rozhraní definované v SSP - Interface. Komunikuje s databází za použití ORM frameworku Hibernate. Dále komunikuje s procesním strojem Activity. Metody komunikující s databází anotujeme anotací Transactional, to zaručí, že všechny změny provedené touto metodou proběhnou v jedné transakci, což zajišťuje konzistenci dat. V případě jakékoli chyby se zavolá metoda rollback a všechny změny se vrátí do původního stavu. Ukázku této komunikace můžeme vidět výše viz.: 1.4 a 1.5

2.2.4 SSP - BI

Modul BI (Business Intelligence) uchovává ve své oddělené databázi vypočítané hodnocení schopností studentů, jejich vhodnost k zadání a data k těmto výpočtům potřebná, což jsou: informace o jejich studiu, výsledky z absolvovaných předmětů a mapování předmětů na schopnost. Tato data potom poskytuje dalším vrstvám.

Modul BI používá ke komunikaci s databází stejných technologií jako SSP - Backend.

Návrh a realizace řešení

Tato kapitola se věnuje rozboru datové struktury. Definuje, jaká data musí fakulta dodat. Dále se zabývá návrhem nových pohledů, které vzniknou pro administrátora a správce jednotlivých fakult.

3.1 Obohacení datového modelu

Jako první je potřeba zmapovat datovou strukturu. Jelikož se v minulosti nemusela řešit příslušnost k fakultě, je zřejmé, že se bude stávající struktura muset obohatit.

3.1.1 SSP - Backend

Z podmnožiny³ datového modelu 3.1.1 jdou vyčíst vazby mezi hlavními entitami SSP a jejich multiplicita. Z požadavků fakult a z analýzy vyplývá, že je potřeba tento model obohatit o následující entity:

- Administrátor

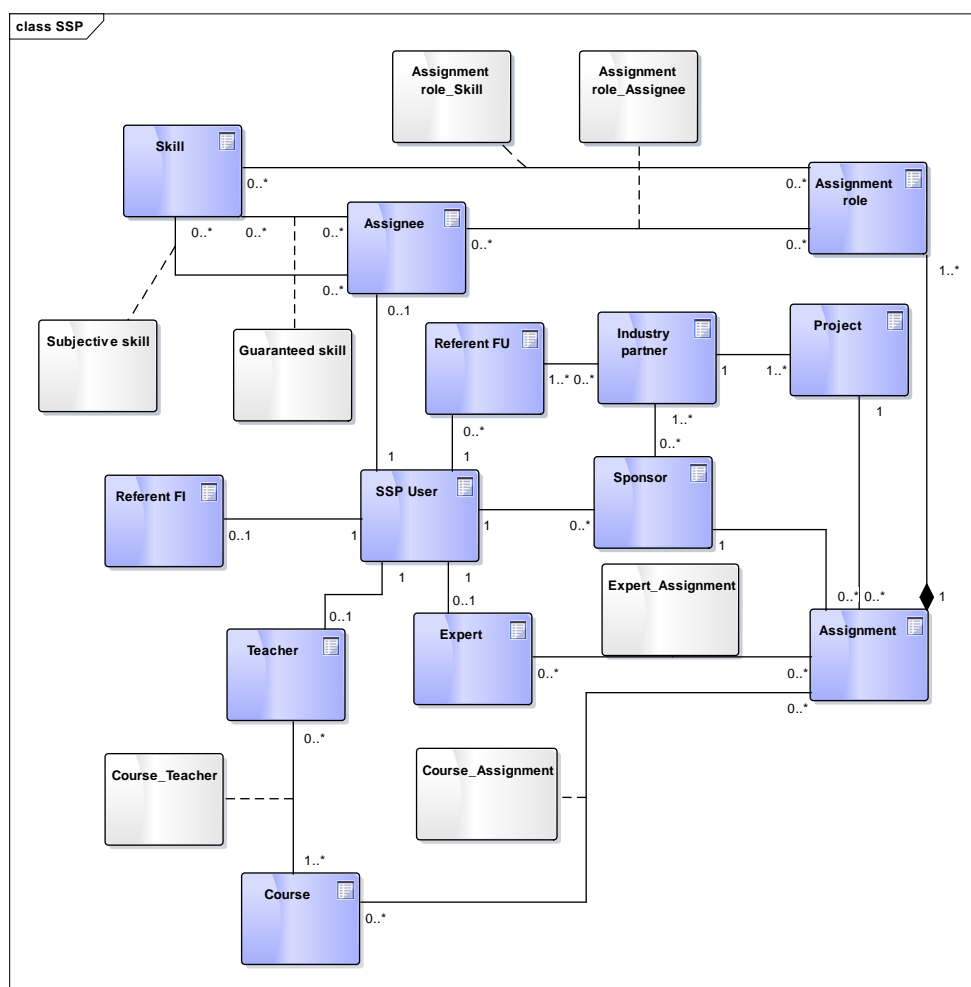
Roli administrátora do této chvíle zastával referent za univerzitu. Nyní bude ale referent za univerzitu zastávat roli referenta na úrovni fakulty. Musí proto vzniknout nová role Administrátor, který se bude starat například o vytváření nových průmyslových partnerů, vytváření nových fakult nebo o schvalování odměn.

- Univerzita

Entita univerzita nebude mít nijak zvlášť velký význam. Ponese informaci o názvu a zkratce univerzity a budou se k ní vázat fakulty.

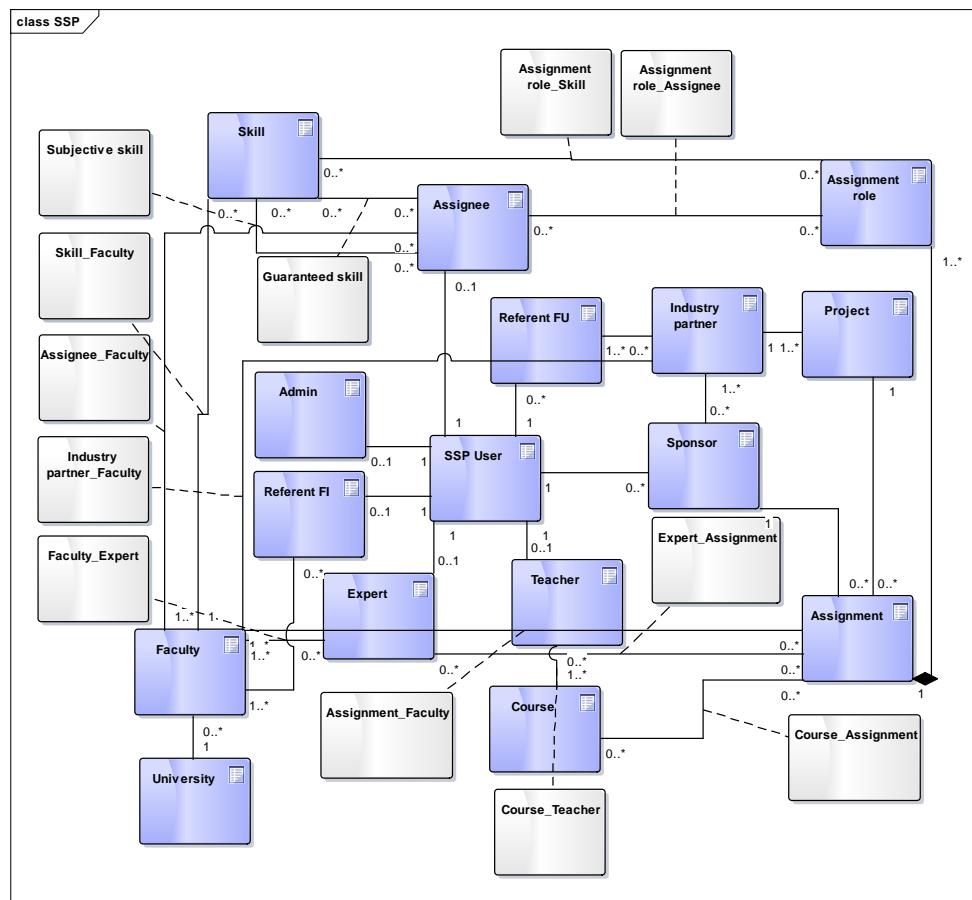
³Kompletní datový model ssp obsahuje přes 100 entit, proto není možné ho zobrazit přehledně celý

3. NÁVRH A REALIZACE ŘEŠENÍ



Obrázek 3.1: Logický datový model SSP

- **Fakulta**
Fakulta bude uchovávat informace, které budou tvořit její profil. Dále se k ní budou vázat jednotlivé role s příslušností k fakultě.
- **Průmyslový partner - Fakulta**
Tato entita bude definovat množinu průmyslových partnerů s kterými fakulta spolupracuje. Dále bude uchovávat informace o úrovni partnerství (vyšší úrovně mají například možnost žádat od fakulty experta)
- **Schopnost - Fakulta**
Každá fakulta si bude moci vybrat podmnožinu schopností, které SSP nabízí. Pouze tyto schopnosti si její studenti budou moci ohodnotit.



Obrázek 3.2: Obohacený logický datový model SSP

- Zadání - Fakulta

Tento vztah bude uchovávat informaci o spolupráci právě na tomto zadání. I když průmyslový partner s fakultou spolupracuje, může mít důvod vytvořit některá zadání pouze pro podmnožinu svých fakult. Dále specifikuje, kteří studenti (myšleno kterých fakult) budou o vzniku zadání informováni.

- Expert - Fakulta

Expert může být zaměstnancem více fakult. Jinak tato entita zatím nenesé žádnou jinou informaci.

- Řešitel - Fakulta

Řešitel může být studentem více fakult. Jinak tato entita zatím nenesé žádnou jinou informaci.

3. NÁVRH A REALIZACE ŘEŠENÍ

Mimo to je model potřeba obohatit o následující vazby:

- Předmět - Fakulta
Tato vazba bude nejenom družít předmět k fakultě, ale zároveň i učitele.
- Referent za univerzitu - Fakulta

3.1.2 SSP - BI

Další datová struktura se nachází v modulu BI. Tato struktura vznikla v rané fázi projektu, přičemž data se do ní nahrávají z datového skladu fakulty informačních technologií. K nahrávání dat z onoho datového skladu se využívá jeden pohled, který vznikl propojením všech dat, které BI potřebuje. Z těchto důvodů trvá nahrání veškerých potřebných dat, vypočítání garantovaných schopností a vypočítání vhodnosti k zadání až několik hodin.

To se do budoucna nemůže tolerovat, proto v rámci analýzy bylo vyho-
věno požadavku rozšířit pohledy do datového skladu na potřebné tabulky jednotlivě, což kompletní nahrávání a vypočítávání dat urychlilo na cca jednu minutu.

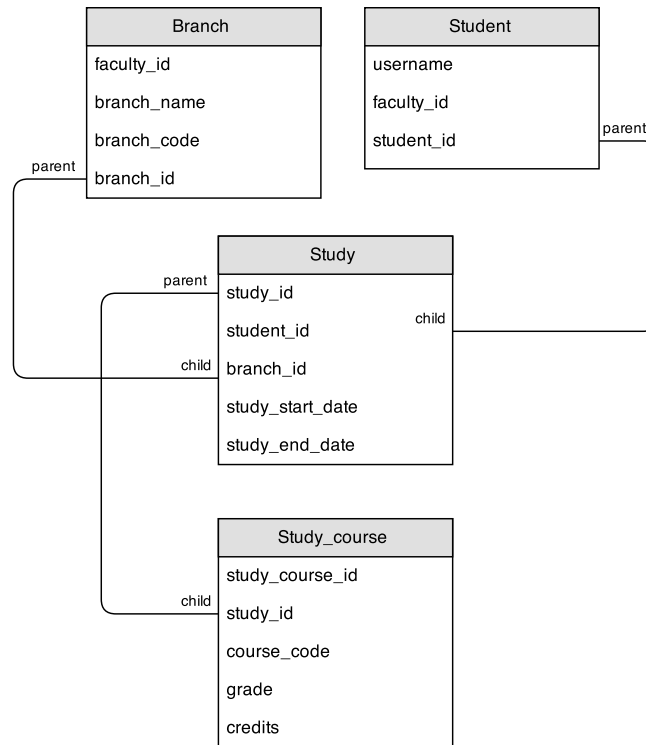
Do budoucna je ale potřeba vytvořit datovou strukturu, která bude umožňovat fakultám hromadné nahrávání dat přes restové služby. Modul BI se proto rozdělí na BI a na UDB.

BI - bude uchovávat mapování předmětů na schopnosti, garantované schopnosti studentů a vhodnosti studentů k zadání.

Student assignment	Student skill	Course skill
username	username	course_code
faculty_id	faculty_id	skill_id
assignment_id	skill_id	faculty_id
value	value	value

Obrázek 3.3: Návrh BI

UDB - bude uchovávat informace o studiu a studijních výsledcích jednotlivých studentů.



Obrázek 3.4: Návrh UDB

3.2 Data pro SSP potřebná

Před zapojením fakulty do SSP je nutné specifikovat jaká data budou od fakulty vyžadována.

3.2.1 Nepostradatelná

Mezi nepostradatelná data patří všechna data specifikována v modulu UDB. Tato data se využívají k posílení zpětné vazby z průmyslu na univerzitu. [12]

3.2.2 Potřebná

Data potřebná jsou data specifikovaná v modulu BI. Vypočítává se podle nich vhodnost studentů k zadání a garantované schopnosti studentů. Nepřítomnost těchto dat ovšem nijak neohroží chod aplikace, pouze studenti dané fakulty

nebudou mít žádné garantované schopnosti, čili nebudou doporučováni k zadání.

3.2.3 Nepovinná

Mezi data nepovinná patří jména a emaily uživatelů. Na FIT - ČVUT jsou tato data získávána přes restové služby z usermap API.

Kvůli požadavkům na co nejmenší počet restových služeb se ovšem nabízí možnost uchovávat jméno a email přímo v databázi SSP. To bude mít nevýhodu v tom, že se budou data uchovávat na více místech a pokud se tyto údaje změní, budou se muset měnit na všech místech.

Další možnost je uchovávat tyto údaje v liferayi, kde se mohou využít i pro jiné aplikace.

Každá fakulta si tedy bude moci vybrat, jakým způsobem bude tato data skladovat. Aplikace je potom bude volat v pořadí: datový sklad SSP, Liferay, restové API.

3.3 Wireframes

Po přidání fakult do systému vzniká potřeba rozšířit i grafické uživatelské rozhraní. K tomuto účelu slouží wireframes. Ty fungují jako předloha pro rozmístění funkčních komponent na stránce. Nejedná se ovšem o grafický návrh. Konkrétní wireframes se nachází v příloze C této práce. Hotové pohledy vytvořené podle této předlohy jsou zdokumentovány rovněž v příloze D této práce.

3.4 Restové API

Restové služby jsou jednak používány k získání dat pro moduly BI a UDB, které se jiným způsobem získat nedají, ale také je to možnost, jak nahrát větší množství předmětů a jejich mapování na schopnosti, které se dále dají vyplnit přes uživatelské rozhraní z pozice referenta za fakultu.

Pro každou fakultu bude vygenerován token, který bude fakultu identifikovat při komunikaci přes restové služby.

Každá žádost respektive odpověď je kódována v JSON formátu.

Základní URL restové služby je `http://hostname:port/api/api-version/resourceName`, kde:

hostname = adresa, kde je aplikace přístupná

port = port, kde je aplikace přístupná

api-version = verze používaného API

resource-name = zdroj, se kterým budeme manipulovat

Zbytek URI krom resource-name se pro jednotlivé fakulty bude uchovávat v databázi a pro aplikaci bude specifikována při vstupu fakulty do SSP.

3.4.1 Vystavené SSP

`/courses?token={token}`

- GET - vrátí všechny předměty dané fakulty
 - 200 - seznam předmětů úspěšně vrácen
 - 401 - nezadán nebo špatně zadaný token
- PUT - nahraje celou kolekci předmětů (stávající se smažou)
 - 200 - akce byla úspěšně dokončena
 - 401 - nezadán nebo špatně zadaný token
 - 400 - chyba ve formátu žádosti

`/skillcourses?token={token}`

- GET - vrátí všechny mapování schopností na předměty dané fakulty
 - 200 - seznam předmětů úspěšně vrácen
 - 401 - nezadán nebo špatně zadaný token
- PUT - nahraje celou kolekci mapování skillů na předměty (stávající se smažou)
 - 200 - akce byla úspěšně dokončena
 - 401 - nezadán nebo špatně zadaný token
 - 400 - chyba ve formátu žádosti

3.4.2 Požadované od fakulty

`/branch`

- GET - vrátí všechny obory dané fakulty
 - 200 - seznam oborů úspěšně vrácen

`/study`

- GET - vrátí všechna studia studentů dané fakulty
 - 200 - seznam studií úspěšně vrácen

/study/{username}

- GET - vrátí všechny studia studenta s daným username
 - 200 - studia studenta úspěšně nalezena a vrácena
 - 400 - neplatný username

/studycourse

- GET - vrátí všechny studijní výsledky studentů dané fakulty
 - 200 - seznam studijních výsledků úspěšně vrácen

/studycourse/{username}

- GET - vrátí všechny absolvované předměty studenta s daným username
 - 200 - předměty studenta úspěšně nalezeny a vráceny
 - 400 - neplatný username

/username/fullname

- GET - vrátí celé jméno studenta s unikátním username
 - 200 - jméno úspěšně nalezeno a vráceno
 - 400 - nevalidní username

/username/email

- GET - vrátí email studenta s unikátním username
 - 200 - email úspěšně nalezen a vrácen
 - 400 - nevalidní username

3.5 Technické roky, které musí fakulta vykonat

1. Určit kdo bude spravovat profil fakulty a přidávat do portálu nové uživatele fakulty
2. Určit, kdo bude v portálu vystupovat jako expert fakulty.⁴
3. Připravit data o svých studentech ve formátu navrženém v modulu UDB
4. Vystavit restová API, které aplikace vyžaduje ve formátu popsaném touto prací

⁴Expert by měl mít přehled o všech předmětech fakulty zapojených do aplikace a umět určit, jaká zadání se k předmětu vážou

3.5. Technické roky, které musí fakulta vykonat

5. Získat od FIT - ČVUT autorizační token
6. Oznámit FIT - ČVUT na jaké adrese budou dostupné restové služby fakulty

Testování

V projektu se používá několik druhů testování:

Jednotkové testy - testy, které testují funkčnost jednoho objektu neboli jednotky

Integrační testy - testy které testují integraci více objektů dohromady

Uživatelské testování - uživatelské testy se dělají před nasazováním na produkci, aby se zkontrolovaly textové a funkční chyby. Dále slouží k otestování intuitivnosti grafického uživatelského rozhraní

4.1 Jednotkový test

Zde lze vidět ukázkou jednotkového testu. Nejprve se vytvoří univerzita, uloží do databáze a poté se otestuje, jestli lze univerzitu se správnými parametry v databázi nalézt.

```
@org.junit.Test
public void testCreatingUniversity() {
    final Long universityId = universityAdminService.
        createUniversity("Ceske_vysoke_uceni_techicke", "
        Czech_Technical_University", "CVUT", "CTU", "balakzde"
        );
    final UniversityDto university = databaseHelper.
        findUniversity(universityId);
    Assert.assertEquals(university.getUniversityAbbrev(), "
        CVUT");
    Assert.assertEquals(university.getUniversityName(), "Ceske
        _vysoke_uceni_techicke");
}
```

Ukázka kódu 4.1: Příklad jednotkového testu

4.2 Integrační test

Zde lze vidět ukázkou integračního testu. Nejprve se vytvoří univerzita, poté fakulta patřící do této univerzity. Dále se uložená data načtou hned dvěma způsoby. Když se ukáže, že data načtená souhlasí s daty uloženými ověří se tím funkčnost hned tří objektu. (universityAdminService, facultyAdminService a myFacultyService)

```
@org.junit.Test
public void testCreatingFaculty() {
    final Long universityId = universityAdminService.
        createUniversity("Ceske_vysoke_uceni_technicke", "
        Czech_Technical_University", "CVUT", "CTU", "balakzde"
        );
    final Long facultyId = facultyAdminService.createFaculty(
        universityId, "Fakulta_elektrotechnicka", "Faculty_of_
        Electrical_Engineering", "FEL", "FEE", "balakzde");
    try {
        final FacultyAdminDto faculty = facultyAdminService.
            findFaculty(facultyId);
        Assert.assertEquals(faculty.getFacultyName(), "Fakulta
        _elektrotechnicka");
        Assert.assertEquals(faculty.getFacultyAbbrev(), "FEL")
            ;
        Assert.assertEquals(faculty.getFacultyNameEn(), "
        Faculty_of_Electrical_Engineering");
        Assert.assertEquals(faculty.getFacultyAbbrevEn(), "FEE
        ");
        Assert.assertEquals(faculty.getUniversityId(),
            universityId);
    } catch (NonexistentFacultyException ignore) {
    }

    try {
        final MyFacultyProfileDto myFaculty = myFacultyService
            .findMyFaculty(facultyId);
        Assert.assertEquals(myFaculty.getFacultyName(), "
        Fakulta_elektrotechnicka");
    } catch (NonexistentFacultyException ignore) {
    }
}
```

Ukázka kódu 4.2: Příklad integračního testu

Závěr

Cílem práce bylo navržení a realizace změn vedoucích k možnosti technického zapojení více fakult do portálu Spolupráce s průmyslem

Nejprve bylo nutné zmapovat požadavky fakult, zajímajících se o portál SSP, a také požadavky přímo FIT-ČVUT. Dále bylo důležité zmapovat data potřebná přímo pro chod aplikace.

Z těchto požadavků potom vyplynula struktura obohacení datových zdrojů aplikace. A způsob získávání těchto dat.

Pozměněna musela být všechna volání objektů ovlivněných změnou datových zdrojů, aby se aplikace chovala podle funkčních požadavků.

Dále bylo zapotřebí vytvořit nová uživatelská rozhraní, která vyplynula z přidání hodnot aplikace. Tato uživatelská rozhraní se nejprve navrhovala jako wireframes a poté byla implementována v souladu s tématem celé aplikace.

Nakonec vznikl seznam restových API, které budou umožňovat předávání dat mezi fakultou a aplikací.

Plány do budoucna

Do budoucna se ještě chystá rozdělit modul BI na dva celky (modul UDB a modul BI), jako bylo popsáno v této práci. Tento návrh nebyl doposud realizován, jelikož čeká na schválení vedením FIT. Jakmile tato nová struktura vznikne bude možné také implementovat volání definovaných restových služeb.

Literatura

- [1] Nabídka spolupráce pro firmy - ČVUT - Fakulta elektrotechnická. Naposledy navštíveno 21.5.2015. Dostupné z: <https://www.fel.cvut.cz/cz/vz/spoluprace-s-firmami.html>
- [2] Pracovní příležitosti - ČVUT - Fakulta elektrotechnická. Naposledy navštíveno 21.5.2015. Dostupné z: <https://www.fel.cvut.cz/cz/psps/pracovniprilezitosti.html>
- [3] Spolupráce FIT-VUT. Naposledy navštíveno 21.5.2015. Dostupné z: <http://www.fit.vutbr.cz/spoluprace.php>
- [4] Archív elektronické konference FIT-VUT. Naposledy navštíveno 21.5.2015. Dostupné z: <http://www.fit.vutbr.cz/news/nabidky/>
- [5] Partnerství v elektrotechnice FEL-ZCU. Naposledy navštíveno 21.5.2015. Dostupné z: <http://partnerstvi.fel.zcu.cz/>
- [6] Nabídky brigád a zaměstnání FEL-ZCU. Naposledy navštíveno 21.5.2015. Dostupné z: http://fel.zcu.cz/study/nabidky_prace.html
- [7] Spolupráce s praxí FIS-VSE. Naposledy navštíveno 21.5.2015. Dostupné z: <http://fis.vse.cz/spoluprace/spoluprace-s-praxi/>
- [8] GNU Lesser General Public License v2.1. Naposledy navštíveno 21.5.2015. Dostupné z: <http://www.gnu.org/licenses/lgpl-2.1.html>
- [9] Portlet - ScienceGraph.net. Naposledy navštíveno 21.5.2015. Dostupné z: <http://cs.sciencegraph.net/wiki/Portlet>
- [10] JSR 220: Enterprise JavaBeans™ 3.0. Naposledy navštíveno 21.5.2015. Dostupné z: <https://jcp.org/en/jsr/detail?id=220>
- [11] Spring Framework. Naposledy navštíveno 21.5.2015. Dostupné z: <http://projects.spring.io/spring-framework/>

LITERATURA

- [12] Aleš, F.: *Diplomová práce - Posílení zpětné vazby z průmyslu na univerzity*. 2014.

Seznam použitých zkratek

API Application Programming Interface

BI Business Intelligence

ČVUT České vysoké učení technické

DTO Data transfer object

EU Evropská unie

FIT Fakulta informačních technologií

GAČR Grantová agentura České republiky

GUI Graphical user interface

HR Human resources

HTML HyperText Markup Language

Java EE Java Enterprise Edition

JPA Java persistence API

JSF Java server faces

JSON JavaScript Object Notation

MPO Ministerstvo průmyslu a obchodu

MV Ministerstvo vnitra

NDA Non-disclosure agreement

ORM Objektové relační mapování

PR Public relations

A. SEZNAM POUŽITÝCH ZKRATEK

RDB Relační databáze

REST Representational State Transfer

SSP Spolupráce s průmyslem

UDB University data bank

WML Wireless Markup Language

XHTML Extensible HyperText Markup Language

XML Extensible markup language

Specifikace JSON objektů

Pro komunikaci přes restové služby se v tomto případě používají JSON objekty. Pro možnost správné implementace je potřebné znát jejich strukturu.

Předmět_Schopnost

```
"type": "object",
"properties": {
  "courseCode": {
    "type": "string"
  },
  "skillId": {
    "type": "number"
  },
  "value": {
    "type": "number"
    "maximum": 1,
    "minimum": 0
  },
},
"required": ["courseCode", "skillId", "value"]
```

Ukázka kódu B.1: Struktura JSON objektu CourseSkill

Předmět

```
"type": "object",
"properties": {
  "courseCode": {
    "type": "string"
  },
  "courseName": {
    "type": "string"
  },
  "courseNameEn": {
    "type": "string"
  },
  "courseAnnotation": {
    "type": "string"
  },
  "courseAnnotationEn": {
    "type": "string"
  },
  "courseLecturesOutline": {
    "type": "string"
  },
  "courseLecturesOutlineEn": {
    "type": "string"
  },
  "seminarOutline": {
    "type": "string"
  },
  "seminarOutlineEn": {
    "type": "string"
  },
  "openForAssignments": {
    "type": "boolean"
  }
},
"required": ["courseCode"]
```

Ukázka kódu B.2: Struktura JSON objektu Course

Obor

```
"type": "object",
"properties": {
  "branchCode": {
    "type": "string"
  },
  "branchName": {
    "type": "string"
  }
},
"required": ["branchCode", "branchName"]
```

Ukázka kódu B.3: Struktura JSON objektu Branch

Studium

```
"type": "object",
"properties": {
  "branchCode": {
    "type": "string"
  },
  "username": {
    "type": "string"
  },
  "startDate": {
    "description": "yyyy.MM.dd",
    "type": "string"
  },
  "endDate": {
    "description": "yyyy.MM.dd",
    "type": "string"
  }
},
"required": ["branchCode", "username"]
```

Ukázka kódu B.4: Struktura JSON objektu Study

Studium_Předmět

```
"type": "object",
"properties": {
  "branchCode": {
    "type": "string"
  },
  "username": {
    "type": "string"
  },
  "courseCode": {
    "type": "string"
  },
  "grade": {
    "description": "from group 1 1.5 2 2.
                    5 3 4",
    "type": "number"
  },
  "credits": {
    "type": "number"
  }
},
"required": ["branchCode", "username", "courseCode", "
             grade", "credits"]
```

Ukázka kódu B.5: Struktura JSON objektu StudyCourse

Wireframes

C.1 Správa fakult

Tato stránka bude sloužit k vytváření nových univerzit a fakult administrátorem. Definiuje zde název a zkratku univerzity a to v českém i anglickém jazyce. U fakulty poté data obdobná, přičemž u fakulty navíc určuje k jaké univerzitě fakulta patří.

Správa fakult

Seznam univerzit:

ČVUT - České vysoké učení technické v Praze	detail
---	---------------

VŠE - Vysoká škola ekonomická v Praze	detail
---------------------------------------	---------------

Přidat univerzitu

Seznam fakult:

ČVUT - Fakulta Informačních technologií	detail
---	---------------

ČVUT - Fakulta elektrotechnická	detail
---------------------------------	---------------

VŠE - Fakulta informatiky a statistiky	detail
--	---------------

Přidat fakultu

Obrázek C.1: Wireframe - Správa fakult

C.2 Správa fakulty

Tato stránka bude sloužit k editaci základních údajů fakulty a k přiřazování Referentů. Opět se jedná o stránku pro administrátora.

Správa fakulty

Univerzita	<input type="text" value="ČVUT"/>
Název	<input type="text" value="Fakulta informačních technologií"/>
Zkratka	<input type="text" value="FIT"/>
Název en	<input type="text" value="Faculty of Informatic Technology"/>
Zkratka en	<input type="text" value="FIT"/>

Uložit

Referenti:

Aleš Fišer	odebrat
------------	----------------

Pavel Kordík	odebrat
--------------	----------------

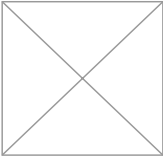
Přidat referenta

Obrázek C.2: Wireframe - Správa fakulty

C.3 Profil fakulty

Tato stránka bude sloužit pro editaci profilu fakulty. Dále se zde bude určovat podmnožina schopností, kterou bude fakulta využívat. Spravovat jí bude referent té fakulty.

Profil Fakulty

Název fakulty	<input type="text" value="Fakulta informačních technologií"/>
Zkratka	<input type="text" value="FIT"/>
Logo	
Popis cz	<p>Informatika je obor budoucnosti. Informační a komunikační technologie prorůstají do všech oblastí lidské činnosti. Zpracování informací je stále více klíčem k pochopení světa, ve kterém žijeme i nás, lidí samotných. Dobrých odborníků, kteří rozumí stále složitějším počítačovým systémům je stále nedostatek</p>
Popis en	<p>Computer Science is a beautiful and dynamically developing field which is an integral part of all spheres of our society. It changes our way of life dramatically, it opens new possibilities, and it has a dramatic impact on everything we do.</p>

Používané schopnosti

Uložit

Obrázek C.3: Wireframe - Profil fakulty

Nové pohledy

Tato stránka slouží k vytváření nových univerzit a fakult. Definuje zde název a zkratku univerzity a to v českém i anglickém jazyce. U fakulty poté data obdoba, přičemž u fakulty navíc určuje k jaké univerzitě fakulta patří.

D.1 Správa fakult

Správa fakult

Všechny university

ČVUT - České vysoké učení technické v Praze

Detail univerzity →

VŠE - Vysoká škola ekonomická

Detail univerzity →

Vytvořit novou univerzitu +

Všechny fakulty

FIS - Fakulta informatiky a statistiky

Detail fakulty →

FIT - Fakulta informačních technologií

Detail fakulty →

Vytvořit novou fakultu +

Obrázek D.1: GUI - Správa fakult

Nová univerzita X

Jméno univerzity

Jméno univerzity EN

Zkratka univerzity

Zkratka univerzity EN

Obrázek D.2: GUI - Vytvoření nové univerzity

Nová fakulta X

Univerzita

Jméno fakulty

Jméno fakulty EN

Zkratka

Zkratka EN

Obrázek D.3: GUI - Vytvoření nové fakulty

D.2 Správa fakulty

Tato stránka slouží k editaci základních údajů fakulty a k přiřazování Referentů. Jedná se o stránku pro administrátora

Správa fakulty

Univerzita	ČVUT - České vysoké učení technické v Praze
Jméno fakulty	Fakulta elektrotechnická
Jméno fakulty EN	Faculty of Electrical Engineering
Zkratka	FEL
Zkratka EN	FEE

[Uložit](#)

Referenti

Test Test (fiserale)	Smazat Referenta
Tibor Szolár (borovto1)	Smazat Referenta

[Přidat nové referenty](#)

Obrázek D.4: GUI - Správa fakulty


D.3 Profil fakulty

Tato stránka slouží pro editaci profilu fakulty. Dále se zde určuje podmnožina schopností, kterou bude fakulta využívat. Spravuje jí referent té fakulty.

Profil fakulty

[ČVUT - FIT](#) [VŠE - FIS](#) [ČVUT - FEL](#)

Jméno fakulty

Logo fakulty 

Popis fakulty

Popis fakulty EN

Štítky

Kontakt

Telefon	<input type="text" value="+420 224 359 826"/>	<input type="button" value="X"/>
WWW	<input type="text" value="www.fit.cvut.cz"/>	<input type="button" value="X"/>
Email	<input type="text" value="ssp@fit.cvut.cz"/>	<input type="button" value="X"/>
<input type="button" value="--- Přidat kontakt ---"/>		

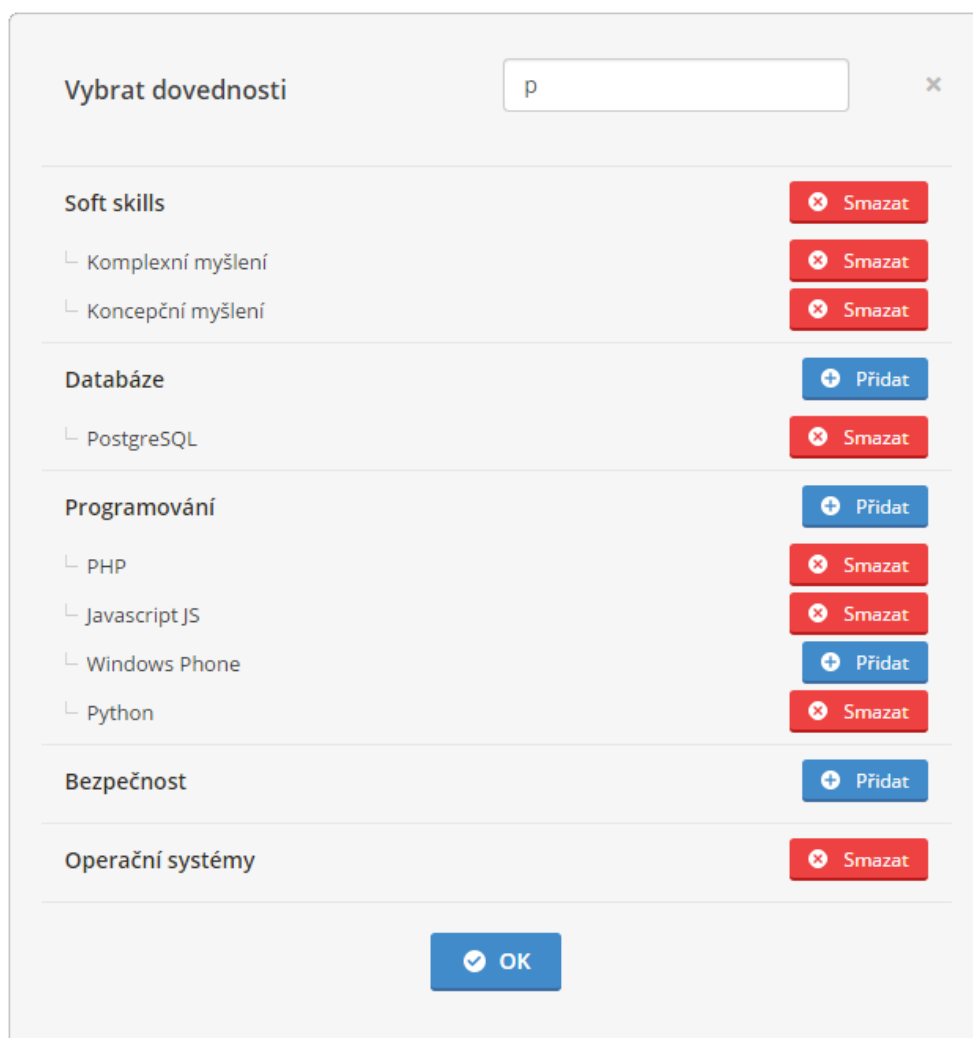
Používané schopnosti

Uložit změny

Obrázek D.5: GUI - Profil fakulty

D.4 Vybírání schopností

Tento dialog slouží jednak k určení schopností, které bude fakulta využívat. Dále pak slouží pro přidávání schopností k předmětu.



Obrázek D.6: GUI - Vybírání schopností

D.5 Veřejný profil fakulty


CTU - Faculty of Information Technology

Java
C++
FIT
CVUT
Programming
Big data

Computer Science is a beautiful and dynamically developing field which is an integral part of all spheres of our society. It changes our way of life dramatically, it opens new possibilities, and it has a dramatic impact on everything we do.

The production volume and employee count of ICT business is comparable to the automotive industry, and grows continuously. The number of ICT professionals in the Czech Republic and in the world is still insufficient.

By the way, establishing a new company is easier in ICT than in the automotive industry. Many of our students established such companies during studies. All you need is a clever idea for an innovative application, web service, etc.



Contacts

✉ ssp@fit.cvut.cz

☎ +420 224 359 826

www.fit.cvut.cz

Obrázek D.7: GUI - Veřejný profil fakulty

D.6 Mapování schopností na předmět

Mapování schopnosti na předmět určuje vztah mezi předmětem a schopností v procentech. Tento vztah ovlivní úroveň souvisejících schopností studentů, kteří tento předmět absolvovali

Související dovednosti

Název dovednosti	Vliv		
Datamining	<input type="text" value="100"/>	0%	100%
Grafika	<input type="text" value="100"/>	0%	100%
XML Sedna	<input type="text" value="50"/>	0%	100%
Kritičnost	<input type="text" value="50"/>	0%	100%
Databáze	<input type="text" value="100"/>	0%	100%

✓ Přidat schopnosti

Reset ✕

Uložit 📄

Obrázek D.8: GUI - Mapování schopností na předmět

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
├─ impl.....	zdrojové kódy projektu
├─ thesis	zdrojová forma práce ve formátu L ^A T _E X
text	text práce
├─ BP_Balak_Zdenek_2015.pdf	text práce ve formátu PDF
├─ BP_Balak_Zdenek_2015.ps	text práce ve formátu PS