Posudek oponenta závěrečné práce

České vysoké učení technické v Praze **Fakulta informačních technologií**

Student: Bc. Vít Steklý
Oponent práce: Ing. Jakub Jirůtka
Název práce: Knihovna pro správu autorizačních a autentifikačních údajů pro projekt psaný ve Spring frameworku
Obor: Webové a softwarové inženýrství

Datum vytvoření: 1. 2. 2017

Hodnotící kritérium:	*Způsob hodnocení - následující škálou 1 až 5:*
1. Náročnost a další komentář k zadání	*1=mimořádně náročné zadání,* *2=náročnější zadání,* ***3=průměrně náročné zadání,*** *4=lehčí, ale ještě dostatečně náročné zadání,* *5=nedostatečně náročné zadání*

Popis kritéria:
Podrobněji charakterizujte diplomovou (bakalářskou) práci a její případné návaznosti na předchozí nebo běžící projekty. Dále posuďte, čím je zadání této ZP náročné. (U obtížnější ZP lze dále tolerovat některé nedostatky, které by u ZP standardní obtížnosti tolerovány nebyly; a naopak u jednoduché ZP mohou být zjištěné nedostatky hodnoceny přísněji.)

Komentář:
Práce z praktického hlediska přímo navazuje na existující framework Spring Security, který poskytuje robustní a komplexní základ pro řešení autentizace a autorizace v aplikacích, včetně podpory OAuth 2.0 (v rámci Spring Security OAuth).

Autor v práci analyzuje dva rutinní procesy související se správou uživatelských účtů, které se řeší ve většině webových aplikací, a to potvrzení emailové adresy po registraci a reset hesla prostřednictvím tokenu zaslaného emailem. Dále správu uživatelových „zařízení“, což ve skutečnosti znamená pouze správu OAuth 2 klientů, resp. jimi vydaných tokenů, v rámci frameworku Spring Security OAuth.

První část spočívá v návrhu a implementaci knihovny pro zmíněný framework, která poskytne univerzálně použitelné a přizpůsobitelné řešení těchto dvou procesů. Druhá část spočívá v konfiguraci Spring Security OAuth, mírném rozšíření modelu tokenu o IP adresu, user-agent a časové informace, plus vytvoření RESTful API a jednoduchého uživatelského rozhraní pro výpis vydaných tokenů a možnost jejich zneplatnění.

Výsledná práce je nakonec poněkud prostší, než jak se jeví z textu zadání. Na druhou stranu je nutno dodat, že její splnění vyžadovalo důkladné pochopení velmi složitého frameworku, rovněž architektonických vzorů a konvencí v něm použitých. Nakonec tedy zadání (v kontextu toho, jak bylo pojato) hodnotím jako průměrně náročné.

Hodnotící kritérium:	*Způsob hodnocení - následující škálou 1 až 4:*
2. Splnění zadání	*1=zadání splněno,* ***2=zadání splněno s menšími výhradami,*** *3=zadání splněno s většími výhradami,* *4=zadání nesplněno*

Popis kritéria:
Posuďte, zda předložená ZP splňuje zadání. V komentáři uveďte body zadání, které nebyly zcela splněny, případně rozšíření ZP oproti původnímu zadání. Nebylo-li zadání zcela splněno, pokuste se posoudit závažnost, dopady a případně i příčiny jednotlivých nedostatků.

Komentář:
Rešeršní část byla splněna bez výhrad.

Implementované řešení se mírně odchýlilo od zadání. Předpokládám však, že k tomu došlo po domluvě s vedoucím práce.

Zadání požaduje uveřejnění implementované knihovny jako open-source. Také autor na několika místech v textu zmiňuje, že šlo o zásadní požadavek vedoucího práce, a v poslední kapitole popisujete plánovaný postup publikace na GitHubu. Přestože práce byla odevzdána před půl rokem, tak k uveřejnění knihovny stále nedošlo. Tento bod zadání tedy považuji za nesplněný.

Hodnotící kritérium:	*Způsob hodnocení - následující škálou 1 až 4:*
3. Rozsah písemné zprávy	***1=splňuje požadavky,*** *2=splňuje požadavky s menšími výhradami,* *3=splňuje požadavky s většími výhradami,* *4=nesplňuje požadavky*

Popis kritéria:
Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části.

Komentář:
Písemná zpráva je nadprůměrně rozsáhlá, čítá 63 stran. Požadavky kladené na diplomové práce tedy zcela splňuje.

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*

4. Věcná a logická úroveň práce	*75 (C)*

Popis kritéria:
Posuďte, zda předložená ZP je po věcné stránce v pořádku, případně vyskytují-li se v práci věcné chyby nebo nepřesnosti. Zhodnoťte dále logickou strukturu ZP, návaznosti jednotlivých kapitol a pochopitelnost textu pro čtenáře.

Komentář:
Práce má poměrně logickou strukturu a je přehledně členěna. Malou výhradu mám ke kapitole páté, jejíž podkapitola Licence patří spíše do rešeršní části.

V úvodu autor uvádí jako typický příklad důsledku špatně napsané klíčové části systému únik emailové komunikace premiéra Sobotky. Tento případ je sice mediálně atraktivní, avšak s popsaným problémem nijak nesouvisí, neboť v něm nebylo prokázáno zavinění technickou chybou v systému.

Rešeršní část (první kapitola) obsahuje mnoho technických nepřesností až chyb. Vzhledem k chabé jazykové stránce nedokáži posoudit, zda jsou způsobené neznalostí či pouze špatným vyjadřováním.

V prvním odstavci podkapitoly 1.1 autor chybně uvádí, že entity jsou reprezentovány třídami. Parafrázuje zde zdroj, ve kterém se ovšem (správně) píše o *instancích* tříd. Dále autor píše: „S tímto nápadem v roce 2001 přišel Gavin King.“ Pokud tím nápadem je myšleno ORM, tak toto tvrzení není pravdivé (G. King není autorem konceptu ORM). Pokud využití ORM entit jako modelu v MVC, tak nevím o zdroji, který by toto potvrzoval.

Způsob vysvětlení rozdílu mezi jazyky Groovy a Java v sekci 1.2.2.1 není moc fundovaný; neodpovídá tomu, co bych čekal od absolventa magisterského oboru Webové a softwarové inženýrství.

Vysvětlení procesu fungování OAuth, které najdeme v sekci 1.4.1, je přinejmenším nepřesné a silně zmatečné.
V následující sekci autor píše: „Je velmi důležité si pohlídat způsob připojení mezi jednotlivými částmi aplikace, *případně* posunout komunikační vrstvu na šifrovaný protokol HTTPS.“ Tuto formulaci považuji za chybnou, neboť využití TLS, resp. HTTPS, je RFC 6749 vyžadováno, nikoli pouze doporučeno. Bezpečnost OAuth silně závisí na šifrování na transportní vrstvě. Dále autor uvádí: „Celá bezpečnost tohoto protokolu spočívá v důvěryhodnost klientské aplikace.“ Toto tvrzení je nepravdivé. OAuth naopak řeší problém nedůvěryhodnosti aplikací třetích stran a poskytuje mechanismy pro minimalizaci s tím spojených rizik.

Nefunkční požadavky v sekci 2.2.2 obsahují bod „Bezpečnost generovaných potvrzovacích klíčů“. Toto není nikde dál rozvedeno, přitom na tom do velké míry závisí bezpečnost celého řešení. Očekával bych zmínku o způsobu generování klíčů, jejich délku apod.

Třetí kapitola je zpracována naopak velmi pečlivě, rozebírá implementaci do hloubky a diskutuje učiněná implementační rozhodnutí. Provází ji UML diagramy tříd a komunikační diagramy. Bohužel zde zcela chybí návrh a popis implementace druhé části práce, která se týká využití OAuth 2.0.

V páté kapitole musím vyzdvihnout přínosnou diskuzi o volbě vhodné licence, která uvádí i málo známé, avšak podstatné detaily.

Hlavní body uvedené v sekci 5.2.2 nejsou popisem GitFlow, jak autor tvrdí, ale spíše GitHub Flow, který je od jiného autora.

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
5. Formální úroveň práce	*60 (D)*

Popis kritéria:
Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 14/2015, článek 3.

Komentář:
Celou práci provází velmi špatná stylistika, kostrbatá až nesmyslná slovní spojení (mj. nadužívání slov „dochází k“ a „problematika“) a chybná interpunkce. Obsahuje také množství překlepů či záměn slov a několik duplicitních souvětí. Student se mohl vyhnout některým anglicismům, např. „servicy“.

Text je psaný v prvním rodě množného čísla, což je sice jeden z doporučovaných stylů pro závěrečné práce, avšak v kapitole Závěr působí jednoznačně jako „majestátní plurál“. To zřejmě souvisí s tím, že celá první část závěru je pojatá jako vyprávění, čím autor při studiu prošel, kterak potkal vedoucího své práce atp.

Po typografické stránce musím vytknout časté používání spojovníku namísto pomlčky a příliš velký font v ukázkách kódu oproti ostatnímu textu.

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
6. Práce se zdroji	*100 (A)*

Popis kritéria:
Vyjádřete se k aktivitě studenta při získávání a využívání studijních materiálů k řešení ZP. Charakterizujte výběr studijních pramenů. Posuďte, zda student využil všechny relevantní zdroje nebo zda se pokoušel řešit již vyřešené problémy. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků a úvah, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami.

Komentář:
Seznam zdrojů obsahuje celkem 47 položek. Bibliografické citace se zdají být v souladu s citační normou.

Zdrojové soubory, které autor převzal z existujících projektů, obsahují správnou licenční hlavičku a jméno původního autora.

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
7. Hodnocení výsledků, publikační výstupy a ocenění	*75 (C)*

Popis kritéria:
Vyjádřete se k úrovni dosažených hlavních výsledků ZP, např. k úrovni teoretických výsledků, nebo k úrovni a funkčnosti technického nebo programového vytvořeného řešení, apod. Případně také zhodnoťte, zda software nebo zdrojové texty, které nevytvořil sám student, byly v ZP použity v souladu s licenčními podmínkami a autorským právem. Popište případnou publikační činnost a získaná ocenění související s řešením této ZP.

Komentář:
Výstupem implementační části jsou následující.

1. Knihovna, která řeší proces ověření emailové adresy při registraci uživatele a reset hesel.

* Knihovna je poměrně pečlivě navržená a implementovaná. Jde zde vidět, že se autor velmi dobře seznámil s architekturou Spring Security a vzory v něm používanými.
* Překvapivě neumožňuje snadnou úpravu textu posílaných emailů pomocí šablon, text je vepsán přímo do třídy (tu je však možné nahradit vlastní implementací).
* Regulární výraz dlouhý přes 6 500 znaků (!) rozhodně nelze považovat za rozumné řešení pro validaci emailové adresy.
* Musím vytknout de facto absenci dokumentace kódu (JavaDoc), která by v některých třídách byla potřeba (komentáře automaticky generované IDE nepřidávají žádnou informační hodnotu).

2. Velmi jednoduchá aplikace demonstrující využití této knihovny.

* Aplikace umožňuje přihlášení uživatele ihned po registraci, před potvrzením emailové adresy, což je v rozporu se scénářem v sekci 2.3.2.
* V aktivačním emailu se generuje chybná URL pro potvrzení adresy.

3. Aplikace demonstrující správu uživatelových „zařízení“ pomocí OAuth 2.0.

* Aplikace poskytuje vlastní implementaci rozhraní TokenStore pro persistenci tokenů, ta je však chybná. Obnovovací (refresh) tokeny jsou ukládány pouze jako součást přístupových (access) tokenů a metoda pro ukládání obnovovacích tokenů je zakomentovaná, tedy nefunkční.
* To, co aplikace nazývá poněkud zvláštně „zařízeními“, jsou ve skutečnosti přístupové (access) tokeny. Při správné implementaci TokenStore by takto implementovaná revokace „zařízení“ neměla kýžený efekt.
* Aplikaci tedy nelze považovat za dobrou demonstraci dané funkcionality.

U obou demonstračních aplikací zcela chybí dokumentace postupu spuštění a především externích závislostí. Obě předpokládají běžící instanci MySQL, první aplikace navíc vyžaduje nahrání schématu do databáze. Potřebný SQL skript je sice na CD přiložen, avšak zcela mimo adresář vlastního projektu. Člověk bez znalosti daných technologií by nebyl schopen demonstrační aplikace spustit.

Kód obsahuje překlepy v názvech identifikátorů (např. „Messanger“, „validty“) a menší prohřešky proti uznávaným konvencím a pravidlům psaní kódu v jazyce Java.

Hodnotící kritérium:	*Způsob hodnocení - nehodnotí se*
8. Komentář o využitelnosti výsledků	

Popis kritéria:
Uveďte, zda hlavní výsledky ZP rozšiřují již publikované známé výsledky a/nebo přinášející zcela nové poznatky. Uveďte možnosti využití výsledků ZP v praxi.

Komentář:
Implementovaná knihovna by bezpochyby byla přínosná pro komunitu vývojářů využívající Spring Security. Bohužel však stále nedošlo k jejímu uveřejnění, přestože je to požadováno v zadání práce.

Hodnotící kritérium:	*Způsob hodnocení - nehodnotí se*
9. Otázky k obhajobě	

Popis kritéria:
Uveďte případné dotazy, které by měl student zodpovědět při obhajobě ZP před komisí (body oddělte odrážkami).

Otázky:
Co nastane v případě, kdy se uživatel zaregistruje, ale potvrzovací e-mail se po cestě ztratí nebo se k němu uživatel dostane až po vypršení platnosti tokenu?

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
10. Celkové hodnocení	*75 (C)*

Popis kritéria:
Shrňte stránky ZP studenta, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení **nemusí** být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích 1 až 9.

Text hodnocení:

Silnou stránkou práce je návrhová a implementační část věnující se procesu potvrzení emailové adresy a resetu hesla. Tyto procesy jsou sice samy o sobě triviální, avšak jejich zpracování v kontextu Spring Security v podobě obecně použitelné knihovny náročnost zvyšuje. Tuto část student zpracoval velmi pečlivě.

Naopak část týkající se OAuth 2.0 zpracoval pouze v podobě nepříliš povedené, jednoduché demonstrační aplikace. V textu je zcela opomenuta, vyjma obecné rešerší části.

Nejslabší stránkou práce je rešeršní část, která obsahuje řadu technických nepřesností, chyb či zmatených formulací, a dále jazyková úroveň práce. Autor zde neprokázal schopnost přesného technického vyjadřování a/nebo dobrou znalost některých ze zkoumaných technologií (zejm. OAuth).

Vzhledem k nepříliš vysoké náročnosti práce – soudě podle výstupů – a množství nalezených nedostatků, navrhuji známku C.

Podpis oponenta práce: