

## I. IDENTIFIKAČNÍ ÚDAJE

<b>Název práce:</b>	An application for healthcare taxonomy translation
<b>Jméno autora:</b>	
<b>Typ práce:</b>	
<b>Fakulta/ústav:</b>	
<b>Katedra/ústav:</b>	
<b>Oponent práce:</b>	
<b>Pracoviště oponenta práce:</b>	

## II. HODNOCENÍ JEDNOTLIVÝCH KRITÉRIÍ

### Zadání

*Hodnocení náročnosti zadání závěrečné práce.*

Aplikace mi přišla jednoduchá – zobrazení několika dat z databáze a volání jedné REST service. Sice obsahuje část v Javě a část v Javascriptu, nicméně mi obě připadají jednoduché.

### Splnění zadání

*Posudte, zda předložená závěrečná práce splňuje zadání. V komentáři případně uveďte body zadání, které nebyly zcela splněny, nebo zda je práce oproti zadání rozšířena. Nebylo-li zadání zcela splněno, pokuste se posoudit závažnost, dopady a případně i příčiny jednotlivých nedostatků.*

Zadání bylo splněno.

### Zvolený postup řešení

*Posudte, zda student zvolil správný postup nebo metody řešení.*

Pojetí i implementace jsou přímočaré.

### Odborná úroveň

*Posudte úroveň odbornosti závěrečné práce, využití znalostí získaných studiem a z odborné literatury, využití podkladů a dat získaných z praxe.*

Kapitola 2 je ve stavu první verze před kompletním přepsáním. Nespecifikuje ani cíle, ani nepopisuje technické aspekty, jak uvádí nadpisy podkapitol; jakkoliv je tato kombinace podivná.

Uživatelské role jsou popsány v obrázku 2.1 a posléze v kapitole 4.2 jsou načrtnuty role 1 až 3, kde první si cca odpovídají, další nikoliv a 3. role raději ani nemá jméno. Předpokládám, že vše má vysvětlit obrázek 4.2. Takže obrázek dobrý, text, prosím, kompletně přepsat. Aha, situaci osvětluje o hodně později obrázek 5.3. Obrázek 2.1 vlastně není use-case, ale jen čáry mezi skupinami uživatelů a doménami.

Stavové diagramy, ač jednoduché, oceňuji.

Kapitola 5.2.4 – já jsem porozuměl už prvnímu odstavci, nebylo třeba opakovat.

Databázové schéma je jasné, ještě by potřebovalo vylepšit, mnoho spojek se kříží zbytečně.

Ohledně enumů v databázi nemáte pravdu. Pokud nemáte tabulku s enumy, můžete mít v odpovídajícím fieldu nesmysly a tudíž nevalidní databázi. Jestli jsou uloženy jako idčka nebo textové řetězce, je jedno.

### Formální a jazyková úroveň, rozsah práce

*Posudte správnost používání formálních zápisů obsažených v práci. Posudte typografickou a jazykovou stránku.*

Překlep na třetí řádce českého abstraktu ukazuje na nedbalou práci, celkem 4 překlepy v abstraktu mě v tom utvrzovaly.

Práce se nečte dobře.

"The aim of this thesis" je uprostřed kapitoly 1 po popisu historie. Zavedení dvou podkapitol by usnadnilo orientaci. Anebo popis struktury práce, která kopíruje obsah, úplně zrušit.

Autor přeskakuje z tématu na téma, odstavec začne jednou myšlenkou a jinou skončí. Příklad: poslední odstavec před kapitolou 2.1, případně první odstavec kapitoly 2.1.

Špatná angličtina, mnoho odstavců jsem musel číst několikrát, abych pochopil smysl. Příklad: odstavec „Using a relational database...“

Retrospektiva na konci je zajímavý prvek a je i zajímavá na čtení.

### **Výběr zdrojů, korektnost citací**

*Vyjádřete se k aktivitě studenta při získávání a využívání studijních materiálů k řešení závěrečné práce. Charakterizujte výběr pramenů. Posuďte, zda student využil všechny relevantní zdroje. 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.*

Výběr zdrojů je odpovídající a jsou velmi vhodně citované.

### **Další komentáře a hodnocení**

*Vyjádřete se k úrovni dosažených hlavních výsledků závěrečné práce, např. k úrovni teoretických výsledků, nebo k úrovni funkčnosti technického nebo programového vytvořeného řešení, publikačním výstupům, experimentální zručnosti apod.*

Doporučuji si přečíst 12 factor app – <https://12factor.net/>, obzvlášť kapitolu III. Config

\* url serveru `api.bp.fireup.studio` je hardcodované ve zdrojácích jak v JS, tak v Javě

Unit testy testují hlavně funkčnost Mockito a jestli, když něco do entity uložím, tak to zase dostanu zpět. Za mě úplně zbytečný kód který nic nekontroluje. Žádný zajímavý test jsem nenašel (ale určitě tam nějaký bude).

100 % coverage je falešný pocit bezpečí. Za nejrizikovější považuju komunikaci s REST serverem – jak testujete tuto část?

Odkaz na Figma se soukromým projektem je zbytečný.

Testovací scénáře by měly být konkrétní: "Change name" -- na jaký řetězec? Kde je ověření, že se test povedl? Takhle se testuje jen, že aplikace nespada, ne, že funguje.

Nejasné kroky: "Apply any flags that fit to the translation"

Tracked metrics -- všude stejné a kompletně zbytečné

Pokud jde jen o uživatelské UX testy – lidé z IT určitě nejsou vhodnými kandidáty. Normální uživatelské testy by byly vhodnější.

Použití transakcí si dovoluji označit za šílené – a dost možná, v kombinaci s použitím MySQL, za jeden z důvodů pomalého zpracování. Ve třídách označených jako `@Service` transakce nepoužíváte, takže každé volání Repository vytvoří vlastní transakci, je tedy extrémně neefektivní.

Příkladně metoda `ImportJobService.importTerms()` startuje a ukončuje transakci v každé z těchto metod:

- `findByld`
- `save`
- `save`
- `importRefset`

- processResult
- finishImport

Pravděpodobně i v dalších, protože jde o entity:

- importJobMapper.toDto()
- importJob.getTranslationSet()

Takže jedno volání funkce nastartuje a dokončí 5-7 transakcí, ve kterých je MySQL tradičně dost slabá.

Používáte třídy s public data membery?

Použití JWT považuji za nevhodné, pokud pro něj nemáte využití, tedy neslouží mezi dvěma microservisy pro bezpečný a ověřený přenos dat. U klienta po přihlášení stačí úplně cookie a REST rozhraní s informacemi o uživateli. Bude to rychlejší, jednodušší a vyhnete se problémům zmíněným v práci.

Při použití zmíněných technologií bych očekával velkou rychlost. U aplikace, která pouze poskytuje data bez nějakého zpracování, mě to překvapuje.

### III. CELKOVÉ HODNOCENÍ, OTÁZKY K OBHAJOBĚ, NÁVRH KLASIFIKACE

*Shrňte aspekty závěrečné práce, které nejvíce ovlivnily Vaše celkové hodnocení. Uvedte případné otázky, které by měl student zodpovědět při obhajobě závěrečné práce před komisí.*

U předložená práce předpokládám, že splňuje zadání vedoucího, nic nenasvědčuje opaku. Nicméně přestože aplikace je jednoduchá, objevil jsem několik míst, které se mi hodně nelíbí. Text práce je za mě dost špatný. Věřím, že student je schopen podat mnohem lepší výkon a předvede ho v diplomové práci.

Předloženou závěrečnou práci hodnotím klasifikačním stupněm

Otázky:

- 1) Která role vytváří „ratings“? V práci jsem to nenašel, čekal bych to v use-case diagramu 4.2 nebo 4.3. Není to zmíněno ani ve funkčních požadavcích (jen jejich moderování).
- 2) Jak budete postupovat, až bude potřeba změnit strukturu databáze, například přidáte atribut a zrušíte jiný? Předpokládám, že jedna běží na Vašem stroji, jedna v testovacím prostředí, jedna na produkci a několik dalších u vašich kolegů, minimálně u vedoucího práce.
- 3) Asynchronní metoda určitě zlepší odezvu aplikace. Jak se uživatel dozví její výsledek?
- 4) Používáte MySQL, ale to je pouze API. Ve skutečnosti používáte MyISAM nebo InnoDB? A proč zrovna tuto?

Datum:

Podpis: Ing. Petr Aubrecht, Ph.D.