



# Posudek oponenta závěrečné práce

Oponent práce: Ing. Oldřich Malec  
Student: Alena Ježková  
Název práce: Webová aplikace - Builder  
Obor / specializace: Webové a softwarové inženýrství, zaměření Softwarové inženýrství  
Vytvořeno dne: 12. června 2023

## Hodnotící kritéria

### 1. Splnění zadání

- [1] zadání splněno
- [2] zadání splněno s menšími výhradami
- ▶ [3] zadání splněno s většími výhradami
- [4] zadání nesplněno

Není kompletně splněn bod 2 - více popsáno v sekci Hodnocení výsledků.

Není řádně splněn bod 6 - více popsáno na konci sekce Písemná část práce.

### 2. Písemná část práce

48/100 (F)

Text má potenciál na to být kvalitní, práce se zdroji je v pořádku, líbí se mi například jak je každá kapitola uvedena textem v kurzívě, dobře jsou rozepsané požadavky a případy užití.

Členění textu je také příjemné, i když někdy je sekcí až moc a mohly by jednotlivé sekce být trochu delší. Níže popsané chyby ale text velmi shazují. Celkově text působí jako rozpracované zápisky, které je před vydáním potřeba dokončit.

Nesmyslné konstrukce vět, například:

- \* "Výstupem této části je být aktuální seznam požadavků." (Cíle práce)
- \* "Zajímavým případem, bylo zjištění, že při generování modelu musí být vždy alespoň jeden atribut." (3.6.3)
- \* "Pro informace o běhu aplikace, o problémech, které vznikly a podobně slouží logování." (3.10)

Špatná práce s odkazy na ukázky kódu, obrázky či přílohy:

- \* "Pro vytváření modelů byly použity příkazy: 3.1." - lepší by bylo třeba "... příkazy, které naleznete v ukázce kódu 3.1." (3.6.3),
- \* "V nástroji Postman byla vytvořena struktura dotazů odpovídajících dotazům definovaným v API 2.4.1, jak je vidět na 4.1 a 4.2." - lepší by bylo třeba "... definovaným v

kapitole 2.4.1, jak je vidět na obrázcích 4.1 a 4.2." (4.2.1),

\* podobně na několika dalších místech.

Drobné chyby:

\* "electron", "android" a "gitlab" by měly být s velkými písmeny,

\* čísla v místě, kde by bylo lepší být číslovky: "Objektově orientované programování se vyznačuje 4 pilíři" (1.6.1.2), nebo "produkt byl zakoupen pro užívání na 2 pobočkách" (F9 v 1.6.3.3),

\* "MoSCow" namísto "MoSCoW",

\* opakující se slovo "vyjmenovány" (1.6.3.2), opakující se sousloví "Cílem kapitoly" (Cíle práce) a "koncové body" (2.4.1),

\* nadbytečné čárky,

\* překlepy ("jsoou"),

\* běžně mi anglické výrazy v česky psané práci nevadí, zejména jde li o zaběhnuté termíny, ale "defaultní hodnoty" by opravdu mohly být "výchozí hodnoty",

\* výpis kódu 3.2 přetéká mimo okraje prostoru pro kód i samotné stránky.

Faktické chyby / věty zkreslující realitu:

\* "REST je spojen s HTTP a je datově orientovaný a implementuje CRUD" - namísto "je spojen" bych napsal spíše "využívá" nebo "staví na" HTTP. Rozhodně ale nelze říct, že implementuje CRUD. REST je styl architektury softwaru, který tedy sám o sobě nic neimplementuje. Teprve RESTful API, tedy konkrétní implementace API dle vzoru REST, může současně "implementovat" i CRUD, přesto bych spíše zvolil obrat "umožňuje provádět CRUD akce" (2.4.1).

\* "Jako varianta se jevil jazyk TypeScript, který běží v prostředí Node.js" - ne, Typescript nikde nativně neběží, ale musí být nejprve transpilován do Javascriptu (3.7.2).

\* "Pro implementaci polymorfismu v jazyku JavaScript slouží rozšíření ES6" - jednak ES6 je spíše verze než rozšíření, a dále polymorfismus lze samozřejmě v prototypové dědičnosti implementovat i ve starší verzi než je ES6, ta pouze přidala syntaktický cukr, aby toto bylo snazší. (3.7.4)

Obsahové problémy:

\* Přejde mi nedostatečně vysvětleno, k čemu přesně projekt slouží - uvítal bych např. kapitolu ještě před Návrhem, která popíše celý projekt License Manager a roli komponenty Builder v tomto systému, nyní je to jen krátce popsáno v sekci 1.5 a dále si to může čtenář domýšlet z jednotlivých požadavků a případů užití.

\* V kapitole Analýza jsem se na mnoha místech zasekl a měl jsem pocit, že obsah patří spíše do kapitoly Návrh. Teprve zpětně jsem si uvědomil, že zde autorka popisuje verzi aplikace, která vznikla dříve a z které vychází pro svou bakalářskou práci. V úvodu kapitoly je tato skutečnost sice je uvedena, ale vzhledem k formátování a stylu textu jsem na to několikrát zapomněl a měl pocit, že je zde už navrhována nová aplikace.

\* Do kapitoly Návrh se plete implementace, jelikož je zde popisováno, jak něco bylo implementováno, a ne jak je to navrženo, aby to teprve bylo implementováno v budoucnu (tj. v kapitole Implementace) (sekce 2.5.1).

\* Často jsou popisovány dosti základní principy, které by se dalo očekávat, čtenář bude znát nebo ho nebudou zajímat: například výběr IDE nebo linteru není vzhledem k celkové délce textu zhruba 38 normostran to nejdůležitější, co se čtenář chce dozvědět, podobně jako popis, že "byly implementovány různé druhy asociací: vazba 1:M a vazba M:N" (3.6.5).

\* Sekce 3.7.4 končí nedokončenou otevřenou větou, které chybí minimálně předmět a tečka.

\* Prázdňá sekce 1.5.4 "Další služby" - nemá žádný textový obsah ani podsekce.

\* Zcela chybí závěr! Práce prakticky končí tím, že je popsáno "co je to integrační testování" - čtenář má pocit, že text musí nějak dál pokračovat; třeba informací, jak autorka integrační testy prováděla, jaké použila knihovny a nástroje apod., ale hlavně poté chybí i celkové shrnutí práce. Mám z toho pocit, jako kdyby byla práce odevzdána ve 23:59 v den termínu a konec se zkrátka nestihl napsat. Úvod a Cíle dokonce zmiňují i kapitolu "Budoucí vývoj", která ale vůbec v práci není!

\* Chybí i popis obsahu přiloženého média.

### 3. Nepísemná část, přílohy

74 /100 (C)

Nepísemná část textu:

\* Oceňuji přiložení activity diagramu pro složitější Use case (Obrázek 2.2), není zda však správně provedeno rozdělení na hlavní a alternativní scénář - namísto rozhodovacího uzlu je zde paralelní běh, takže dle tohoto diagramu dojde k oběma akcím: "obnovením licence" i "smazáním všech dat". Obdobný problém je pak v Activity diagramu na Obrázku 2.3.

\* Datový model a Stavový diagram na obrázcích 2.4 a 2.5 by mohly být lépe uspořádány, aby nemusel jejich text být tak malý - stačilo by pouze jinak rozmístit jejich obsah.

Přílohy na médiu:

\* Velmi kladně hodnotím přiloženou Postman kolekci pro testování.

\* V přílohách, konkrétně ve zdrojových souborech LaTeX jsem našel i kapitolu "Budoucí vývoj" a "Závěr", které v samotném hlavním textu chybí - zřejmě tedy nebyl includován zdrojový .tex soubor do hlavního zdrojového souboru.

\* Přiložena je také specifikace API ve formátu HTML stránky Swagger, uvítal bych však také podkladové soubory z kterých byla tato stránka vygenerována, tedy OpenAPI specifikaci ve formátu JSON či YAML, s kterými lze dále programově pracovat.

Zdrojový kód:

\* Jde o 2725 řádků kódu aplikace a 1464 řádků kódu unit testů a integračních testů

\* Je rozumně strukturovaný.

\* Oceňuji dokumentaci všech metod pomocí JSDoc.

\* Některé metody, které jsou nyní implementovány jako asynchronní, by klidně mohly být synchronní, což by mohlo zjednodušit kód v místech jejich volání. Například jde o funkci validateType v souboru src/main/app/factories/ForTypeServiceFactory.js

\* Chybí mi lepší handling chyb, většina asynchronních volání počítá pouze s úspěšným dokončením,

\* Místa zbyly debug výpisy do konzole.

\* Většina souborů s Unit testy a integračními testy obsahuje mnoho zakomentovaného kódu, bez komentáře, proč je kód zakomentovaný. Některé soubory jsou zakomentované celé - z celkových 1464 řádků kódu testů je jich zhruba 650 zakomentovaných!

### 4. Hodnocení výsledků, jejich využitelnost

80 /100 (B)

\* Celkově jde o složitější téma, zadáním byla tvorba poměrně nestandardní části systému, s čímž se autorka zdánlivě poradila. Využitelnost komponenty rozhodně je, avšak předtím bude potřeba dokončit jak integraci, tak komponentu samotnou.

\* Jedním z výsledků mohlo být i akceptační testování s vedoucím práce, v textu se píše "Akceptační testování nakonec nebylo provedeno.", ale není vysvětleno, proč.

\* Integrace s dalšími komponentami systému License Manager, na kterých pracovali jiní studenti, nebyla zcela naplněna. cituji text: "Vzhledem k tomu, že se spolupráce

rozcházel, nebylo možné sjednotit návrhy. V určité chvíli byla komunikace pozastavena a implementována dohodnutá část, vzhledem k časovým možnostem. Je tedy možné, že komponenty Builder a DM nebudou na sebe přímo navazovat přes vytvořené API a bude nutné tuto změnu provést na základě další domluvy" - závěrečné práce, které spoléhají na součinnost dalších prací jsou vždy rizikové, a nebudu zde proto snižovat hodnocení kvůli tomuto externímu vlivu, musím ale konstatovat, že cíl vzájemné integrace komponent byl z tohoto důvodu naplněn pouze částečně - jistá část API kompatibilní je, zbývá ale i část, která kompatibilní není.

## **Celkové hodnocení**

49 /100 (F)

Myslím si, že text práce by měl být řádně dokončen, dvě chybějící kapitoly jsou hlavní nedostatek, proč jsem byl nucen v celkovém hodnocení udělit stupeň F. Tyto kapitoly jsem sice nakonec našel ve zdrojových kódech, ale v textu zkrátka chybí.

## **Otázky k obhajobě**

- \* Proč neproběhlo akceptační testování?
- \* Text práce je nedokončený; mimo jiné chybí celá jedna kapitola a závěr. Proč?

## **Instrukce**

### **Splnění zadání**

Posudte, zda předložená ZP dostatečně a v souladu se zadáním obsahově vymezuje cíle, správně je formuluje a v dostatečné kvalitě naplňuje. V komentáři uveďte body zadání, které nebyly splněny, posudte závažnost, dopady a případně i příčiny jednotlivých nedostatků. Pokud zadání svou náročností vybočuje ze standardů pro daný typ práce nebo student případně vypracoval ZP nad rámec zadání, popište, jak se to projevilo na požadované kvalitě splnění zadání a jakým způsobem toto ovlivnilo výsledné hodnocení.

### **Písemná část práce**

Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále posudte, 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. Posudte správnost používání formálních zápisů obsažených v práci. Posudte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 52/2021, článek 3.

Posudte, zda student využil a správně citoval relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami. Zhodnoťte, zda převzatý software a jiná autorská díla, byly v ZP použity v souladu s licenčními podmínkami.

### **Nepísemná část, přílohy**

Dle charakteru práce se případně vyjádřete k nepísemné části ZP. Například: SW dílo – kvalita vytvořeného programu a vhodnost a přiměřenost technologií, které byly využité od vývoje až po nasazení. HW – funkční vzorek – použité technologie a nástroje, Výzkumná a experimentální práce – opakovatelnost experimentů.

### **Hodnocení výsledků, jejich využitelnost**

Dle charakteru práce zhodnoťte možnosti nasazení výsledků práce v praxi nebo uveďte, zda výsledky ZP rozšiřují již publikované známé výsledky nebo přinášející zcela nové poznatky.

### **Celkové hodnocení**

Shrňte stránky ZP, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení nemusí být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích. Obecně platí, že bezvadně splněné zadání je hodnoceno klasifikačním stupněm A.