



Posudek oponenta závěrečné práce

Oponent práce:	Ing. Marek Suchánek
Student:	Bc. Jan Přívratský
Název práce:	MassSpecBlocks: Databáze sekvencí a stavebních bloků mikrobiálních metabolitů pro analýzu hmotnostních spekter
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Vytvořeno dne:	8. května 2021

Hodnotící kritéria

1. Splnění zadání

► [1] zadání splněno

[2] zadání splněno s menšími výhradami

[3] zadání splněno s většími výhradami

[4] zadání nesplněno

Veškeré body zadání byly úspěšně splněny. V rámci práce je navržena a implementována aplikace MassSpecBlocks, která je reimplementací a rozšířením původní aplikace. Funkční (uživatelské účty, třídění záznamů, oprávnění uživatelů) i nefunkční (určený framework pro backend i frontend, zefektivnění, změna DB) požadavky stanovené zadáním jsou v práci popsány a naplněny v implementaci. Aplikace je rovněž otestována (včetně automatického spouštění testů na CI) a nasazena.

2. Písemná část práce

55 / 100 (E)

Struktura práce z mého pohledu není pro čtenáře úplně vhodná. V práci chybí úvod, pravděpodobně je nahrazen pouze kapitolou „Cíl práce“, kde už se částečně předbíhá a popisuje se výsledek. Alespoň krátký úvod do problematiky, vysvětlení souvislostí a popsání domény, kterou aplikace (původní i nová) podporuje, by bylo více než vhodné. Celkově je text práce popisem změn oproti původní aplikaci. Lepší by byl popis současného stavu, jaké má problémy, pak čistě novou aplikaci, a nakonec ve zhodnocení například porovnání. Některé odstavce a sekce jsou hůře čitelné nevhodným členěním textu (odstavec téměř přes celou stránku). Pro práci v češtině jsou zbytečně užity některé anglické pojmy (např. „routování“, „správa dependencies“ apod.). V odborné práci by bylo dobré se vyhnout neformálním formulacím typu „Každá vteřinka dobrá“ (navíc správně vteřina není jednotka času ale úhlu). Podobně kapitola „Co tedy aplikace umí?“ by mohla být pojmenována jako „Popis výsledné aplikace“. Zdůvodnění použití Symfony a zmínka jednoho konkurenčního frameworku v sekci 2.9 práci degraduje (stačilo jen zdůvodnit, že

je stanoven v zadání práce, a raději jej lépe popsat). Nicméně všechny důležité části jsou v textu popsány a navazují na sebe dle tradičního procesu vývoje software.

V textu jsou pouze občasné překlepy a například chybějící čárky. Z typografického hlediska jsou použity spojovníky místo pomlček (např. str. 1, 7 a 33), chybný formát datumu (str. 29), obsah místy přetéká za okraj (např. strana 7, 24 a 35 nebo obrázky 3.1 a 3.2 na str. 13 a 14 – když už tak by měly „přetékat symetricky“). Dalším problémem je nekonzistence stylů písma, především použití strojového (monospace) písma, některé složky, soubory, URL, třídy, metody a další jsou tímto písmem, jindy ne, a to i v rámci jednoho odstavce. Algoritmy 1 a 3 (což jsou spíše ukázky kódu – SQL dotazy), by měly použít pro čitelnost zvýraznění syntaxe. Algoritmus 2 by dával spíše smysl jako normální číslovaný seznam, navíc by poté byl i správně použit ve větě s dvojtečkou. Poslední výtku mám k názvům a psaní velkých písmen. Velké množství názvů má chybně a v rámci práce často i nekonzistentně použita velká písmena, například: Mysql (MySQL), Github (GitHub), javascript (JavaScript), Typescript (TypeScript), Pubchem (PubChem), svg (SVG), tanimoto (Tanimoto – název, navíc podle osoby) a další. Citování převzatých informací je v souladu s pravidly a zvyklostmi, je použito celkem 48 zdrojů.

3. Nepísemná část, přílohy

95 /100 (A)

Hlavní nepísemnou přílohou je aplikace rozdělená na backend a frontend část. Po prozkoumání repozitářů se zdrojovými kódy obou částí mohu konstatovat, že aplikace je dobře navržena, kód je přehledný a pěkně strukturovaný v souladu se konvencemi pro použité technologie. Aplikace je také kvalitně zdokumentována, a to jak z pohledu uživatele, tak i pro lokální vývoj a produkční nasazení. Jedinou výtku mám ke konfiguraci frontend aplikace přes soubor `src/constant/Constants.ts`, vhodnější by bylo konfiguraci předávat jinak, typicky JS/JSON config souborem. Pro repozitáře je také nastavení CI pro build a spouštění testů, navíc je nastavený i Dependabot pro zjišťování aktualizací závislostí. Uživatelské rozhraní je účelné, intuitivní a přehledné. Je škoda, že ukázky zajímavých částí kódu implementace (a testů) nejsou obsaženy v práci, určitě by to výrazně pomohlo.

4. Hodnocení výsledků, jejich využitelnost

100 /100 (A)

Aplikace je nasazená a lze ji snadno podle návodu používat. V tomto spatřuji nejvýraznější kladnou stránku práce. Ačkoliv nejsem expert z domény, věřím, že aplikace svými funkcemi je uživatelům užitečná. Vylepšení oproti původní aplikaci jsou zřejmá a výrazně usnadňují práci. Autor také navrhuje další možnosti rozvoje do budoucna.

Celkové hodnocení

80 /100 (B)

Práci hodnotím jako velmi dobrou. Výsledná aplikace je kvalitně zpracovaná a její přínosy v oblasti přírodních věd jsou zřejmé. Bohužel písemná část práce takové kvality nedosahuje viz komentáře výše.

Otázky k obhajobě

- Proč jste zvolil adresy API s `/rest` namísto běžnějšího `/api`? Čekáte, že backend bude využíván jinými aplikacemi nebo jen Vaším frontendem?
- V práci zmiňujete paralelizaci, jakým způsobem je jí dosaženo (na úrovni kódu)? Je

využit preemptivní nebo kooperativní multitasking? V čem tkví zrychlení pomocí paralelizace (čekání na odpověď z externího API, zpracování odpovědi, ...)?

- U aplikace SmilesDrawer (dříve vytvořená aplikace autorem ZP, použitá novou aplikací) se z historických důvodů používá Travis CI. Nebylo by lepší zmigrovat také na GitHub Actions z pohledu udržitelnosti? Jaká by byla náročnost či případné překážky?

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. 26/2017, 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.