



Posudek oponenta závěrečné práce

Oponent práce:	doc. Ing. Robert Pergl, Ph.D.
Student:	Alisher Nurmatov
Název práce:	Jednodušší navazování mezioborové spolupráce ve vědecké komunitě díky efektivnímu zobrazování vzájemných vztahů
Obor / specializace:	Informační systémy a management
Vytvořeno dne:	7. června 2024

Hodnotící kritéria

1. Splnění zadání

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

Zadání práce směřuje k vytvoření nástroje pro zefektivnění agendy navazování mezioborové spolupráce ve vědecké komunitě. Sestává z analytické, návrhové-implentační a manažerské části. Vzhledem k této šíři jej lze hodnotit jako náročnější.

Zadání bylo celkově splněno ve všech bodech, níže rozebírám své připomínky, které směřují zejména k bodu 2 a 4.

2. Písemná část práce

60/100 (D)

Práce má spíše nadprůměrný rozsah 57 stránek (s přílohami a bibliografií 71). Všechny části jsou hodnotné.

Kapitola 2 Existující vizualizační aplikace by měla být rozdělena na dvě části: nástroje pro analýzu vztahů na základě záznamů o vědeckých publikacích a obecné nástroje pro vizualizaci grafů. V práci je představen pouze jeden nástroj z každé kategorie (a tyto kategorie nejsou řádně vymezeny). V první kategorii by mohly být dále např. CiteSpace a VOSviewer a v druhé kategorii pak dále např. Cytoscape, Pajek či Neo4j Bloom. Následné srovnání by pak mělo respektovat specifika těchto dvou kategorií a mělo by být detailnější, zejména z hlediska funkcí, vstupů, obsahu a přehlednosti výstupů, apod. U aplikace Connected Papers dále chybí uvedení, že se jedná o webovou aplikaci.

Doménový model by bylo vhodné umístit až za případy užití, které jsou součástí analýzy požadavků. Doménový model je dost zjednodušený: neobsahuje atributy a je též málo reifikovaný -- není z něj patrný doménový pojem spoluautorství (UC4). Po doménovém modelu by měl následovat implementační model, který již může být zjednodušený (odrážet implementační rozhodnutí).

Pro zobrazení struktury systému na obr. 5.1. by bylo vhodné držet se zavedené notace UML komponentového diagramu. Části věnující se implementaci a testování jsou popsány dostatečně podrobně. V textu se ovšem hovoří o "jednotkových testech", ukázka uvedená v textu je však v zásadě jediným realizovaným netriviálním jednotkovým testem na backendu i frontendu. Ačkoliv u bakalářské práce, zejména ISM zaměření, je určitě akceptovatelná i jen takováto ukázka této důležité softwarově-inženýrské dovednosti, konstatování "byly využity jednotkové testy, které pomohly ověřit funkčnost jednotlivých funkcí" je (mírně řečeno) zavádějící.

Na pěkné úrovni detailu jsou pak zpracovány kapitoly 7 Ekonomický přínos a 8 Další rozvoj a studie proveditelnosti, kde autor uvažoval řadu různých aspektů. K dokonalosti bych uvítal trochu více podložené odhady časové pracnosti rozšíření funkcí, které jsou uvedeny pouze jako sumární expertní odhad počtu hodin. U analýzy nákladů na technologické zdroje a infrastrukturu je poté uvažován pouze Amazon Web Services a autor se spokojil s konstatováním "Vizualizační nástroj není nijak vázán na poskytovatele AWS. Byly využité virtuální servery, které lze nakoupit i od jiných poskytovatelů. Ceny by byly velmi podobné". Zde bych uvítal, kdyby autor alternativně alespoň zmínil a nejlépe i konkrétněji prezentoval, např. Microsoft Azure a zejména rozpracoval i jiné modely provozu než IaaS.

Jazykově je práce na slušné úrovni z hlediska gramatického i formulačního, typograficky je též v pořádku, čitelnost zvyšují obrázky, schémata, diagramy, tabulky a ukázky kódu.

Práce obsahuje spíše nadprůměrný počet položek literatury (40), které jsou ovšem až na jeden zdroj jen online články typu blog. Autorovi zřejmě unikl princip citování odborných zdrojů, nejvíce je to znát u citování online článku shrnující Nielsenovy heuristiky, který na konci "na talíři" přináší reference na původní renomované zdroje.

Zdroje jsou náležitě používány v textu, představený existující vizualizační nástroj Connected Papers ovšem postrádá citaci. Resp. reference je v seznamu literatury a je použita u obr. 2.1, očekával bych ji však také hned v začátku popisu, podobně jako třeba u 2.2 Gephi. Chybí též další reference na použité technologie TypeScript (4.3), Ant design (5.2.1.2) a řadu dalších.

3. Nepísemná část, přílohy

70/100 (C)

Soubory README obsahují všechny potřebné instrukce pro zprovoznění serveru i klienta. Systém se mi podařilo zprovoznit a vyzkoušet na svém počítači až s pomocí studenta -- z důvodu problému s CORS, který ovšem při nasazení na serveru nebude již činit problém. Vizualizace je funkční a naplňuje očekávání.

Frontend využívá již zastaralý Create React App, v důsledku čehož je řada závislostí starších verzí, což přináší mj. bezpečnostní rizika (v současné době indikováno 12 závažných rizik). Pro další rozvoj a nasazení je určitě vhodné přejít na aktuální alternativu např. Vite či Next.js. Kompilace frontendu též obsahuje některé (sice neškodné) warningy (nevyužitě proměnné apod.), které by ale bylo vhodné začistit.

Jako zásadnější vidím, že v textu není popsána potřebná struktura CSV souboru, ani postup, jakým jej exportovat ze systému V3S. Při pokusu nahrát soubor s nevyhovující strukturou se pouze zobrazí "Error uploading file. Please try again.", tedy bez vysvětlení, co je špatně. Pro otestování ale bylo možné využít přiložený příklad, z něj lze dovodit i potřebnou strukturu. Nakonec se mně podařilo i provést export z KOSu svých publikací a vyzkoušet na něm. V rámci toho se ukázal jeden nedostatek -- parser seznamu autorů nepočítá s "et al.", takže někteří autoři jsou v grafu zdvojeni (jednou bez "et al." a podruhé s ním). Toto by ale podle vyjádření autora mělo být v další verzi odstraněno.

Kód backendu není příliš dobře strukturován: soubor index.ts obsahuje inicializaci serveru, router i pomocné rutiny. Vzhledem k rozsahu kódu to v současnosti není velký problém, do budoucna bude ale třeba refaktoring. Z kódu je patrné, že je možné pomocí systémové proměnné nastavit PORT serveru, tato možnost však není zdokumentována. Adresu serveru nelze nastavit, což může být problém při některých typech nasazení, např. do Kubernetes. Ještě horší je to se systémovými parametry Neo4J databáze, které jsou přímo natvrdo v kódu v konstantách. Rozumím, že cílem bylo vytvořit pouze prototyp, tyto věci ale nezaberou příliš mnoho času a výrazně pomohou do budoucna.

Backend čítá okolo 300 řádků kódu, frontend 439 řádků TypeScript a 45 řádků CSS, tedy z hlediska kódování nelze hovořit o příliš rozsáhlé práci, ale systém splňuje zadání a vzhledem k těžišti v oboru ISM je to jistě dostatečné.

4. Hodnocení výsledků, jejich využitelnost

90/100 (A)

Výsledné řešení je funkční, byť má své mouchy, ale přináší kýžený vizuální přehled spoluprací na publikacích. Jako vizuálně-orientovanému člověku je mi velmi sympatické a vítám ho. Autor demonstroval potenciál řešení a jistě poslouží jako dobrý základ pro další rozvoj. Zcela ideální by mně přišlo jej dotáhnout (třeba v navazujících pracech) do dvou podob:

1. Přímá integrace s V3S
2. Nezávislý nástroj s možností importu i dalších datových zdrojů (např. Semantic Scholar).

Celkové hodnocení

70/100 (C)

Zajímavá práce, která přináší výsledky použitelné do budoucna. Rešeršní část je slabší, implementační část nijak neoslňuje, ISM část práce je ale slušná, což je pozitivní vzhledem k oboru autora. Autor celkově prokázal schopnost odborné práce na bakalářské úrovni a práci doporučuji k obhajobě.

Otázky k obhajobě

1. Které metody existují k odhadu pracnosti vývoje software? Proč jste žádnou z nich nepoužil?
2. Pro hostování jste uvažoval pouze o pronájmu virtuálních serverů, tedy IaaS. Proč jste neuvažoval i model PaaS či FaaS? Nemohly by nakonec vyjít výhodněji?

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.