

Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce Martin Baroš
Název práce BrickSnoop: Optimizer of LEGO® Brick Orders
Rok odevzdání 2024
Studijní program Informatika
Specializace Programování a vývoj software

Autor posudku doc. RNDr. Martin Kruliš, Ph.D. **Role** Oponent
Pracoviště Katedra distribuovaných a spolehlivých systémů

Prosím vyplňte hodnocení křížkem u každého kritéria. Hodnocení *OK* označuje práci, která kritérium vhodným způsobem splňuje. Hodnocení *lepší* a *horší* označují splnění nad a pod rámec obvyklý pro bakalářskou práci, hodnocení *nevyhovuje* označuje práci, která by neměla být obhájena. Hodnocení v případě potřeby doplňte komentářem. Komentář prosím doplňte všude, kde je hodnocení jiné než *OK*.

K celé práci	lepší	OK	horší	nevyhovuje
Obtížnost zadání	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Splnění zadání	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Rozsah práce ... <i>textová i implementační část, zohlednění náročnosti</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p>Komentář Práce představuje software řešící poměrně známý Internet Shopping Optimisation Problem (ISOP), tedy optimální nákup více položek z více e-shopů, tematicky zasazený do komunity nakupující LEGO kostičky. Interface je řešený formou single-page webové aplikace, backend je založený na serverless technologiích od Google (Firebase a Google Cloud Workflows). Backend, který řeší samotný optimalizační problém, implementuje dva známé aproximační algoritmy (hledový výběr a simulované žíhání).</p> <p>- Hlavní krok, který se vyžaduje od uživatele, je upload seznamu nabídek z různých serverů, aby mohla proběhnout jejich optimalizace. Vzhledem k tomu, že se předpokládá, že nabídky jsou dostupné online, bych očekával, že tento krok bude proveden (polo)-automaticky, tedy stažení nabídek z vybraných serverů přes API nebo pomocí scrapingu. Je jasné, že realizace této funkcionality by výrazně zvýšila pracnost, nicméně tato možnost mohla být alespoň zahrnuta do analýzy a mohla proběhnout přípravu vhodného rozhraní/modulu uvnitř aplikace (např. s triviální prototypovou implementací). Analogicky výstupy jsou k dispozici pouze ke stažení ve formátu JSON (bez jakékoli vizualizace).</p> <p>- Autor v analýze volil mezi tradičním (server-based) a cloudovým (server-less) řešením, přičemž volba padla na cloud. Argumenty pro výběr jsou spíše zástupné, pro použití cloudu nevidím reálný důvod (snad jen, že může být zajímavé si takovou věc vyzkoušet v rámci studia). Důsledkem pak bohužel je, že aplikaci nelze plnohodnotně testovat bez použití Google cloud služeb.</p> <p>Práce zcela splňuje zadání, implementace používá moderní technologie a zavedené SW-inženýrské postupy (až na absenci testování zmíněnou dále).</p>				

Textová část práce

	lepší	OK	horší	nevyhovuje
Formální úprava ... <i>jazyková úroveň, typografická úroveň, citace</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Struktura textu ... <i>kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Analýza	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Vývojová dokumentace	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Uživatelská dokumentace	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Komentář Text je psán kvalitní angličtinou, je rozumně strukturovaný a obsahuje vesměs všechny očekávané části. Hlavním problémem je úroveň detailu -- některé klíčové věci (optimalizační algoritmy) jsou popsány nedostatečně, jiné (autentizace, UI design) jsou až přehnaně detailní. Jako nejzásadnější vnímám následující připomínky:

- Autor uvažuje jen hladový algoritmus a metody obecné optimalizace (genetické algoritmy, Ant colony a simulované žíhání), které jsou založené na pravděpodobnosti a neposkytují žádné záruky. ISOP je svoji podstatou blíže problému batohu a ceny jednotlivých položek (a dopravy) budou mít pravděpodobně velmi omezený celočíselný rozsah, takže se nabízí přirozená otázka, zda by neexistoval vhodnější algoritmus založený např. na celočíselném počítání, nebo by to nebylo možné řešit dynamickým programováním. (Tj. analýzu optimalizací vnímám jako nedostatečnou.)

- Popis aproximačních algoritmů, které byly implementovány, by měl být výrazně podrobnější -- např. hladový algoritmus lze použít s různými variantami heuristik, což v popisu zcela chybí. Autor se to snaží částečně napravit v sekci 6.5.3, ale stále nedostatečně. Obecný optimalizační postup celé aplikace, který je v zásadě přímočarý, je popsán podrobně, ale klíčové algoritmy velmi vágně. Očekával bych přinejmenším hrubý nástin pseudokódem a jasný odkaz do zdrojového kódu, kde lze nalézt detaily. Simulované žíhání navíc (dle textu) vyžaduje řadu parametrů, které autor ladil empiricky. Tyto parametry ani proces jejich hledání není v textu vůbec popsán.

- Poměrně hodně prostoru je věnováno procesu registrace a přihlášení (funkční požadavky, sekce 4). Vzhledem k tomu, že se jedná o vyřešený problém by bylo vhodnější to zestručnit.

- Některé kroky analýzy obsahují podivnou argumentaci. Např. volba cloudu je podivně vysvětlena (neočekávám, že by bylo aplikaci nutné dynamicky rychle škálovat na velký počet uživatelů). U popisu PHP se autor dopouští argumentačních faulů (např. odkaz na výkon PHP je poněkud podivný, když autor nakonec vybírá Python, který je znatelně pomalejší). Nicméně výběr použitých technologií (až na zmíněnou volbu cloudu), ale nelze objektivně rozporovat.

- Je dobře, že se autor zamyslel nad realizací UI a jeho vzhledem do větší hloubky. Na druhou stranu je úroveň detailu, se kterou je popsán výběr vizualizačního frameworku nebo dokonce barev a typografie (sekce 5.1 a 5.2), velmi přehnaná. Pokud by snad bylo žádoucí, dělat tuto analýzu na této úrovni, tak je navíc značně nedotažená, protože prezentovat pouze 3 možné frameworky (navíc bez jasného kritéria výběru) je zoufale málo. Dále se tímto vytváří ještě větší kontrast vůči nepřiliš detailnímu popisu optimalizačních algoritmů, což budí dojem, že barvy použité v UI jsou důležitější, než algoritmus řešící hlavní problém. Podobně přehnaná je většina zbytku sekce 5 -- nemá např. smysl věnovat 2 stránky tomu, jak má vypadat "about page" a přihlašování, když se to v principu vůbec neliší od aplikací tohoto typu.

- Pojetí uživatelské dokumentace považuji za nevhodné. Autor probírá krok za krokem jednotlivé obrazovky a poměrně detailně popisuje jejich části. Vzhledem k tomu, že UI je triviální, a že očekávaným čtenářem bakalářské práce je informatik, je tento text zcela zbytečný. Pokud by autor snad chtěl vytvořit dokumentaci pro děti (což nepochybně může být cílová skupina), musel by zvolit jiný postup i jiný formát (nehledě na lepší rozhraní, protože ruční upload/download JSON souborů není úplně vhodný pro počítačové laiky). Místo toho by se měl autor podrobněji zaměřit na související kroky (např. jak získat jiná než vzorová data) a také na interpretaci výsledků (je trochu divné, že tři nabízené algoritmy dávají velmi odlišné výsledné ceny na vzorových datech).

- V závěru autor vyhodnocuje vyvinutou aplikaci na základě funkčních požadavků (což je správně), avšak chybí také vyhodnocení efektivity použitých optimalizačních algoritmů (což je poměrně zásadní nedostatek) a další očekávané evaluační postupy (unit testy, UI testy, CI, nebo alespoň testovací scénáře).

- Appendix A je zcela bezcenný, adresářovou strukturu si umí čtenář snadno zobrazit sám. Smysl by to mělo, pokud by jednotlivé adresáře (příp. klíčové soubory) byly doplněné popisem jejich účelu, aby se čtenář rychle zorientoval. Podobně Appendix B pouze zabírá místo -- příklady zdrojových kódů mohly být klidně umístěny do elektronické přílohy, zejména pak když postrádají jakýkoli podrobnější popis.

Implementační část práce

	lepší	OK	horší	nevyhovuje
Kvalita návrhu ... <i>architektura, struktury a algoritmy, použité technologie</i>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kvalita zpracování ... <i>jmenné konvence, formátování, komentáře, testování</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Stabilita implementace	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Komentář Přiloženou aplikaci není možné jednoduše testovat lokálně (což je důsledek ne zcela vhodného výběru backend řešení). Autor poskytuje pro účely nasazené demo prostředí na vlastním serveru. Při testování se aplikace jevila stabilně, neodhalil jsem žádné zásadní problémy. UI je navrženo moderně s ohledem na jednoduché použití (v rámci zvoleného řešení, výtka s nutností uploadovat vstupní data jako soubory je platná). Několik drobných poznámek:

- Některé operace v UI (např. uložení změn v profilu) můžou trvat i více než 5 vteřin během kterých není jasné, zda byla operace zahájena a čeká se na její ukončení (nezobrazuje se žádný indikátor, jen se zakáže příslušné tlačítko). Podobně stránka upřesnění algoritmu (customization) reaguje poměrně těžkopádně (zejména výběr obchodů, které chce uživatel vyloučit).
- Výsledky lze pouze exportovat do JSONu, očekával bych alespoň nějaký náhled (např. tabulkou) přímo v prohlížeči. Místo toho autor nabízí cizí online nástroj na zobrazování JSONu, což vypadá, jako by tuto funkci nestihl dokončit.
- Při pokusu znovu spustit vyhodnocení se musí znovu zcela vyplnit formulář konfigurace, přičemž předchozí konfiguraci nelze ani číst.

Zdrojové kódy strukturou a designem odpovídají zvyklostem ve vybraných technologiích a rozsah odpovídá bakalářské práci. Níže uvedené připomínky jsou opět drobnějšího charakteru:

- Zdrojové kódy v TypeScriptu obsahují minimum komentářů. U React komponent to lze pochopit, u sdílených funkcí nebo dokonce deklarácí interfaces už ne. Situace je naštěstí výrazně lepší v Pythonu, kde autor pravidelně používá dokumentační komentáře u funkcí/tříd.
- Již při letném průchodu kódem odhalil linter nemalé množství drobných nedostatků. To nutně nemusí znamenat, že software obsahuje chyby, ale je dobrým zvykem tyto nedostatky odstranit (podobně jako v Jave nebo C++ není slušné bezdůvodně ignorovat warningy kompilátoru).
- Příloha k práci obsahuje zazipované instance git repozitářů včetně pracovních .git podadresářů (které zabírají 210 z celkových 220MiB). Tento fakt nemá sice vliv na hodnocení práce, ale doporučuji autorovi, aby si na podobné excesy dával v budoucnu pozor (nevypadá to profesionálně).

Asi nejzásadnější výtka směřuje k testování, protože práce zcela postrádá jakoukoli formu automatických testů. Optimalizační backend mohl být celkem jednoduše testován externě jako black-box nebo pomocí unit testů. Aplikace mohla být otestována alespoň jako celek skrz UI např. použitím Selenia.

Celkové hodnocení Velmi dobře
Práci navrhuji na zvláštní ocenění Ne

Datum 29. srpna 2024

Podpis