

— Posudek oponenta bakalářské práce —

Název: Vizualizace meteorologických a environmentálních dat
geodetické observatoře Pecný
Student: Lukáš Běloch
Vedoucí: doc. Ing. Jakub Kostecký, Ph.D.
Fakulta: FSv ČVUT v Praze
Katedra: Katedra geomatiky
Oponent: Ing. Martin Landa, Ph.D.
Pracoviště oponenta: Katedra geomatiky, FSv ČVUT v Praze

Zadání. (*Hodnocení náročnosti zadání závěrečné práce.*)

Zadání považuji s ohledem na požadavky kladené na bakalářskou práci za náročnější. Jeho směřování je čistě geoinformatické, což vyžadovalo po studentovi dostudovat celou škálu konceptů a technologií nutnou pro návrh moderních webových aplikací.

Splnění zadání. (*Posuďte, zda předložená závěrečná práce splňuje zadání. V komentáři případně uveďte body zadání, které nebyly zcela splněny, nebo zda je práce oproti zadání rozšířena.*)

Zadání považuji za splněné. Student se ujal řešení problému svědomitě, některé aspekty přesahují očekávání, např. pokus o průběžnou integraci. Student se dostal od návrhu aplikace přes její implementaci až po nasazení v prostředí produkčního serveru.

Zvolený postup řešení. (*Posuďte, zda student zvolil správný postup nebo metody řešení.*)

Prezentované řešení považuji vzhledem k tomu, že student neměl předchozí zkušenost s návrhem webových aplikací, za komplexně pojaté přesahující požadavky kladené na bakalářskou práci. Kromě vývoje backendu a frontendu aplikace se student věnoval návrhu REST API.

Diskutabilní je implementace backendu v programovacím jazyce JavaScript. Na druhou stranu je to v posledních letech jakýsi trend, který přináší výhody i nevýhody.

Odborná úroveň. (*Posuďte úroveň odbornosti závěrečné práce, využití znalostí získaných studiem a z odborné literatury, využití podkladů. Posuďte též schopnost studenta vnímat řešenou problematiku v širších souvislostech a aplikovat inženýrský přístup při řešení.*)

Některé pojmy nejsou v textu blíže vysvětleny, což snižuje jeho pochopitelnost. Jako příklad lze uvést metody GET a POST (str. 21), které jsou pro fungování aplikace (komunikace mezi frontendem a backendem) podstatné. Velmi pozitivně hodnotím přiložené vývojové diagramy. Zcela nad rámecem práce je využití průběžné integrace, i když končí spíše řadou otázek a nejasností. Kladně hodnotím také studentovu snahu o diverzitu použitých programovacích jazyků. Testovací aplikace, zmíněná v kap. 5.1.1, byla napsána v C++.

Snahu o široký záběr práce reprezentuje kap. 5.3.2, popisující výsledky testování aplikace na vzorku uživatelů. Je ovšem otázka, zda je vzorek pěti uživatelů dostatečně reprezentativní.

Formální a jazyková úroveň, srozumitelnost práce. (*Posud'te správnost používání formálních zápisů obsažených v práci. Posud'te typografickou a jazykovou stránku práce a její celkovou srozumitelnost.*)

Kvalitu jazykové stránky práce snižují neobratné konstrukce. Jako příklad můžeme uvést „a je vyvíjen především na rychlost dotazů“ (str. 12) či „dostat z databáze“ (str. 21). Někdy se dostáváme až na hranici srozumitelnosti, např. „Tabulky vyhrajeme pomocí metody GET“ (str. 21). Detailem jsou potom drobné překlepy typu „GitHab“ (str. 31).

Práce je koncipována srozumitelně. Tomu napomáhá logické řazení kapitol od realizace, přes implementaci, až po testování.

Výběr zdrojů, korektnost citací. (*Vyjádřete se k aktivitě studenta při získávání a využívání studijních materiálů k řešení závěrečné práce. Posud'te výběr pramenů. Ověřte, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi.*)

Výčet použité literatury je poměrně bohatý, i když je většina zdrojů „online“. Citace jsou nicméně korektní a v souladu s citačními zvyklostmi.

Celkové hodnocení, otázky k obhajobě, návrh klasifikace.

Úroveň práce odpovídá požadavkům kladeným na bakalářskou práci. Její kvalitu v tomto kontextu nesnižuje ani fakt, že má povahu spíše technické dokumentace. Na druhou stranu je to práce srozumitelná i pro čtenáře, který nemá žádnou zkušenost s návrhem webových aplikací. Oceňuji snahu autora pojmout řešení komplexně. Od návrhu přes realizaci, nasazení aplikace v produkčním prostředí, až po testování. To, že není výsledek dokonalý odpovídá zkušenostem autora a v žádném případě nesnižuje kvalitu práce. Doufám, že se bude autor projektu věnovat i nadále a napraví drobné (např. prázdná stránka [O projektu](#)), ale i větší nedostatky (nefunkční tlačítko *Uložit*, na stránce [Týdenní data](#) např. funguje pouze při zadání počátečního a koncového data¹). Stálo by také za úvahu aplikaci nasadit v izolovaném prostředí jako je např. Docker kontejner.

Jako oponent práce mám následující dotazy:

1. Jaký je rozdíl mezi metodou GET a POST? Uved'te příklady.
2. Jak přesně funguje navržené REST API? Uved'te příklady.
3. Jak jste testoval průběžnou integraci, když oba uvedené Git repositáře obsahují pouze jeden commit z 18.května?
4. Z jakého důvodu používáte pro uložení hodnot data a času datový typ `varchar(255)`, viz obr. 3.2?
5. Co Vás vedlo k nasazení JavaScriptu i na straně backendu?
6. Proč jste používal PuTTY? (kap. 4.4, jinde v textu zmiňujete operační systém Ubuntu)

¹Testováno ve webovém prohlížeči Firefox (OS Debian GNU/Linux) dne 14.6.2020

7. Řešil jste otázku bezpečnosti? (CI na produkčním serveru, privátní klíč vložen do GitHubu.)
8. Zkoušel jste chování backendu na úmyslně poškozených datech (např. jiný než předpokládaný počet položek v řádku a pod.)?
9. Jakým způsobem informujete správce o havarii importu do DB?
10. S jak velkými soubory pracujete při importu do DB? V kap. 5.3.1. zmiňujete problém s operační pamětí.
11. Kde je testovací C++ aplikace? V Git repositáři jsem ji nenašel.

Přes drobné výtky hodnotím předloženou závěrečnou práci klasifikačním stupněm

– **A (výborně)** –

V Praze dne 15. června 2020

.....
Ing. Martin Landa, Ph.D.
Fakulta stavební, ČVUT v Praze