



Posudek vedoucího diplomové práce

Jméno studenta: Bc. Pavel Křivda

Téma práce: Automatizované vyhodnocování studentských prací s využitím Git

Téma a cíle diplomové práce

Cílem diplomové práce je navrhnout a vytvořit prostředí, které umožní automatické testování a vyhodnocování studentských prací, které jsou odevzdány do kolaboračního Git serveru.

Použité metody v diplomové práci

Diplomant ve své práci využil zejména znalosti z oblasti pokročilých programovacích technik, programování webových a databázových aplikací a návrhu systémů.

Co diplomant při vypracování diplomové práce vytvořil

V teoretické části diplomové práce se diplomant nejdříve věnuje přehledu souvisejících webových technologií a v následně ve 2. kapitole představuje problematiku verzovacího systému Git a souvisejících kolaboračních serverů. Dále je zde provedena rešerše dostupných řešení, které je možné použít pro realizaci vlastního cíle diplomové práce. 3. a 4. kapitola se dále věnují vybraným technologiím použitým pro návrh a implementaci praktické části diplomové práce.

Kapitolou 5 pak začíná vlastní návrh řešení systému pro odevzdávání prací, student se podrobně věnuje analýze plánovaného použití systému, formalizuje požadavky a případy užití. V následující 6. kapitole je proveden vlastní návrh systému a v 7. kapitole jsou shrnuty základní implementační fakta, včetně představení vlastní řídicí aplikace v kapitole 7.4. Poslední 8. kapitola popisuje konfiguraci a instalaci systému do provozu.

Prokázání správnosti navrženého řešení

Systém se při testování podařilo nasadit, spustit a provést několik základních testů, které zahrnovaly vytvoření projektu, přihlášení studenta k projektu, odevzdání práce a její vyhodnocení. Systém pracoval korektně. Systém během testování nebyl testován v cílovém prostředí a v očekávaném zatížení.

Splnění zadaných cílů diplomové práce

Výsledek práce splňuje základní zadání práce a umožňuje automatizované vyhodnocování studentských prací.

Hodnocení textu diplomové práce z hlediska jeho kvality, struktury, srozumitelnosti, jazykové a typografické úrovně

Práce je po formální stránce poměrně dobře strukturována, ačkoliv některá teoretická témata jsou uvedena dříve, než je to relevantní. V práci je věnováno velké množství prostoru popisu použitých technologií oproti vlastnímu systému. V práci se na několik místech vyskytují drobné jazykové chyby a obtížněji pochopitelné formulace.

Jak byla vyhodnocena kontrola textu DP (případně zdrojových kódů softwaru) pomocí systému pro odhalování plagiátů mezi závěrečnými pracemi?

Samotný text práce vykazuje maximální shodu 8 % s textem bakalářské práce diplomanta, ze které převzal několik částí a necitoval je korektním způsobem, žádná další významnější shoda nebyla identifikována. Práce není plagiátem.

Další nejasnosti a otázky:

- Nefunkční požadavky 2, 3 (str. 46), 7 (str. 47) jsou vágní a bez stanovení měřitelných metrik není nijak prokazatelné, že byly naplněny. S ohledem na téma práce a cílové nasazení aplikace jsou tyto požadavky poměrně zásadní.
- Navržený systém vyžaduje pro řídicí aplikaci databázi, která uchovává duplicitní údaje jako GitLab – autentizační údaje – které mohou být získány ze systému GitLab, resp. využito OAuth2 identity provider API, popis projektu (zadání úlohy) – může být například uloženo v podobě README.md přímo v repozitáři daného projektu.
- Sekce Správa projektů v řídicí aplikaci je velmi nepřehledná, jelikož kombinuje zadání projektů a vytvořené forký jednotlivých studentů.
- Sekce Přehled s výsledky postrádá rychlé filtrování podle předmětů a projektů pro rychlý náhled.
- Sekce Přehled a její načítání je extrémně pomalé, i při jednotkách projektů načítání sekce trvá velmi dlouhou dobu. Vzhledem k implementaci, která v dané části nevyužívá cachování a realizuje velké množství API požadavků na systém GitLab je pomalost očekávatelná. Nicméně to je v přímém rozporu s nefunkčními požadavky na výkon a škálovatelnost.

Shrnutí

Celkově je možné konstatovat, že zadání práce bylo splněno a navržený systém je funkční a v omezené míře použitelný. Systém nicméně sám nesplňuje nefunkční požadavky týkající se výkonu a škálování, které stanovil sám diplomant a jeho použití ve větším měřítku počtu studentů a zadání povede k výraznému zvýšení doby odezvy systému. V kombinaci s některými dalšími nedostatky uvedenými výše by takový systém byl poměrně málo uživatelsky přívětivý.

Otázky k obhajobě:

1. Jakým způsobem by bylo možné doplnit cachování výsledků, aby byly čtecí operace z řídicí aplikace vykonávány bez nutnosti kontaktovat GitLab API. Uveďte základní použitelné techniky/způsoby cachování a vhodné formy perzistence/externích db/cachovacích nástrojů, který by bylo možné použít pro uchovávání cache.

Doporučení práce k obhajobě: ano

Navržený klasifikační stupeň: C