



Posudek oponenta závěrečné práce

Oponent práce: Ing. Pavel Štěpán
Student: Stela Augustínová
Název práce: Informační systém pro házenkářské sportovní kluby
Obor / specializace: Webové a softwarové inženýrství, zaměření Webové inženýrství
Vytvořeno dne: 6. června 2023

Hodnotící kritéria

1. Splnění zadání

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

Práce splňuje zadání v plném rozsahu – byla vytvořena a zdokumentována požadovaná webová aplikace.

2. Písemná část práce

85 / 100 (B)

Autorka v práci popisuje analýzu řešeného problému (včetně srovnání s podobnými existujícími řešeními), dále se zabývá implementací (s částmi, týkajícími se servera, klienta a použité databáze). Pokračuje popisem testování a navrhuje i případná další rozšíření. Je zde tedy k dispozici kompletní popis aplikace a jejího vývoje.

Práce je provedena značně pečlivě – obsahuje množství obrázků, tabulek i výpisů kódu a rozsah je zcela dostačující. Nemohu posoudit jazykovou stránku – práce je psána ve slovenštině (což není můj rodný jazyk). Je však dobře čitelná a srozumitelná. Po technické stránce obsahuje několik překlepů, které však nenarušují srozumitelnost.

Jen několik poznámek k obsahu: dost nepřehledně je porovnávána 2 a 3 vrstvá architektura. Např. MVC je (převážně) architektura serverové části a thin/thick klient plus smart/dumb server se naopak týká způsobu rozdělení aplikace mezi klienta a servera.

Také popis MVC (a příslušný graf) nepovažuji za příliš přesný. V obrázku není zdůrazněno, že jednotlivé části (M, V i C) mohou být ve více exemplářích. Ale hlavně, popisky jsou matoucí: od view do controlleru vede šipka s popisem "sends input from user". Ve skutečnosti vstupy od uživatele (z browseru) zpracovává controller. A take šipka od modelu k view s popisem "updates" neodpovídá MVC. Ve skutečnosti controller (a v

některých implementacích i view) požaduje data (či provedení určitých operací – update, delete, ...) po view a ne naopak.

Škoda je, že autorka neuvedla podrobnější popis Dependency Injection (DI), když jiné – a známější – technologie popisuje detailněji. O DI se totiž jen krátce zmiňuje v souvislosti s Angular, ale dnes se tato technologie velmi často používá. Použitím DI se údajně zpružňuje tvorba a údržba aplikací (osobně si myslím, že se též velmi znehledňuje struktura a fungování aplikace – zastánci DI by tvrdili pravý opak). V každé, případě by bylo vhodné uvést o DI více detailů.

Naopak hezky je popsán např. HTTP protocol a přesný je popis databázového modelu.

Je přiložena i obsáhlá bibliografie (většinou ve formě webových odkazů, protože v práci se používají moderní technologie).

3. Nepísemná část, přílohy

90/100 (A)

Autorka vytvořila fungující aplikaci a k její realizaci použila mnoho aktuálních technologií - framework Spring (a jeho nastavby) pro serverovou část, Angular (a jeho nastavby) pro klienta a další technologie.

Také je zajímavé, že autorka dokáže zpracovat statistiky, které jsou jí k dispozici jen ve formě HTML (což je dost neobvyklý datový zdroj, i když zde jde o stránky s pravidelnou strukturou).

4. Hodnocení výsledků, jejich využitelnost

85/100 (B)

O použitelnosti jakéhokoli software rozhodne jedině praxe (proto tuto otázku v hodnocení považuji za zcela nesmyslnou). Kolik již bylo nadějných aplikací, které naprosto propadly a naopak těch (dle odborníků) "nic moc", které mají ohromný úspěch.

Pokud mám "hádat z křišťálové koule", tak bych řekl, že jestliže zatím slouží pro organizaci činnosti sportovního oddílu tři různé aplikace, tak tento program již jenom tím, že je nahradí, může zajistit významné zlepšení situace. Navíc poskytuje další funkčnost.

Otázkou ovšem je to, že pokud je jazykem uživatelského rozhraní angličtina (jak ukazují i snímky obrazovek), jak si s tím "fyzikálníci" poradí – zejména při případném nasazení v menších oddílech.

Je tu ale ještě další otázka, která se však netýká jenom této práce, ale většiny webových aplikací. Aplikace byla vytvořena použitím Javy, frameworku Spring a jeho nastaveb, JavaScriptu (resp. TypeScriptu), Angularu a jeho nastaveb a dalších technologií. Toto vše autorka prakticky zvládla (za což je jí třeba vyslovit uznání), ale jaká bude budoucnost aplikace s hlediska údržby a dalšího rozvoje? Pokud je bude provádět autorka, je vše v pořádku. Ale pokud ne, je provozovatel nucen sehnat programátora se všemi těmito znalostmi, nebo si opatřit tým, jehož členové tyto technologie znají (čímž se vše značně prodraží). Za mnohem vhodnější považuji přístup firmy Microsoft, kde je server i klient (ve formě Blazoru) vytvářen stejnou technologií.

Vzhledem k výše uvedenému hodnotím tento bod jen s největším sebezapřením.

Celkové hodnocení

85 /100 (B)

Ve výše uvedených bodech jsem uvedl, že autorka vytvořila fungující aplikaci, kterou posala dobře srozumitelným způsobem. K práci mám jen menší výhrady, proto navrhuji uvedené hodnocení.

Otázky k obhajobě

- byly prováděny ještě další testy aplikace kromě uživatelského testování?
- je zvažována budoucí podpora jiných jazyků (a jak by to bylo náročné)?
- jak se předpokládá údržba a rozvoj aplikace? Bude je provádět autorka, nebo někdo jiný?
- jaký je (zhruba) předpokládaný roční rozsah dat v databázi (včetně statistik)?

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.