

POSUDEK OPONENTA DIPLOMOVÉ PRÁCE

Student: Nesteš Filip

Oponent: Ing. Prítrský Tomáš

Studijní program: **Informační technologie**

Studijní obor/Specializace: **Softwarové inženýrství**

Akademický rok: **2023/2024**

Téma diplomové práce: **Moderní architektury webových aplikací s mikrofrontendy**

Hodnocení práce:

Študent v teoretickej časti diplomovej práce rieši dôvody vzniku microservice architektúry a z nej vyplývajúcu microfrontend architektúru. Treba povedať, že táto téma je veľmi aktuálna a priamo súvisí s Cloud technológiami.

K teoretickej časti:

- kapitola 1.1.1: Obrázok 3 a 4 je rovnaký, pričom v 3 mal zobrazit' MVC.

- kapitola 2.1.9, "*Na druhej strane mikroslužby bežia nezávisle od konkrétneho hardvéru či platformy a to ušetrí organizáciám náklady na drahé inovácie.*" Nie je pravda, aj monolit môže bežať nezávisle od HW (v kontajneri).

- kapitoly 3.2.5, 3.3.1 - spomína nezávislosť komponentov a potom predpokladá potrebu zdieľania stavov, či už pomocou Redux, alebo Mobx. Ak 2 komponenty potrebujú zdieľať stav, mali by sme zvážiť, či tieto netreba spojiť do jedného (sú závislé). Komunikácii medzi komponentami sa v komplexných aplikáciách nevyhneme, ale tento problém sa dá riešiť pomocou event alebo callback mechanizmu.

- kapitola 3.4.3, "*Vynikajúca škálovateľnosť: MPA môže obsahovať neobmedzený počet stránok, čo je ideálne pre online obchody a spravodajské portály*" - SPA môže tiež obsahovať neobmedzený počet 'stránok'

Celkovo študent správne vysvetlil kedy, prečo a ako používame microfrontend architektúru, a naopak, kedy MF nie je vhodné riešenie.

Ku praktickej časti:

Študent splnil zadanie diplomovej témy a naimplementoval aplikáciu, ktorá využíva microfrontend architektúru. Navyše si to skomplikoval tým, že použil komponenty napísané v 4 rôznych JavaScript frameworkoch. Tu narazil na problémy komunikácie medzi takýmito komponentami – preto v reálnom svete aj pri použití MF, koordinujeme vývoj jednotlivých biznis modulov tak, aby používali rovnaké frameworky a ich verzie (pokiaľ možno). Avšak študent demonštroval, že aj takéto komponenty dokážu medzi sebou komunikovať a toto aj naimplementoval.

Aj keď takýto prístup nie je optimálny v reálnom svete, demonštruje výhody microfrontend architektúry.

Viac pozornosti by si zaslúžil ModuleFederation plugin, a ako prispieva k optimalizácii zdrojov posielaných na webového klienta. Napríklad by ma zaujímalo, keď použijem závislosť na rovnakú knižnicu v package.json všetkých 4 modulov, či sa táto do webového klienta načíta len raz, alebo viac krát.



Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Študent myslel aj na prípad, kedy načítanie microfrontendu zlyhá. Tento stav vyriešil tak, že miesto daného MF komponentu sa na jeho mieste zobrazí OopsComponent. Toto je kritické pre robustnosť distribuovaného systému.

Dotazy k obhajobe:

1. Prečo sa rozhodol použiť 4 rôzne JS frameworky?
2. Prečo je vhodné, aby boli microfrontend komponenty nezávislé?
3. Prečo predpokladá zdieľanie stavu medzi takýmito komponentami?

Celkové hodnotení práce:

Známku uvede oponent dle svého uvážení dle klasifikační stupnice ECTS:

A – výborně, B – velmi dobře, C – dobře, D – uspokojivě, E – dostatečně, F – nedostatečně.

Stupeň F znamená též „nedoporučuji práci k obhajobě“.

Předloženou diplomovou práci doporučuji k obhajobě a navrhuji hodnocení

B - velmi dobře.

V případě hodnocení stupněm „F – nedostatečně“ uveďte do připomínek a slovního vyjádření hlavní nedostatky práce a důvody tohoto hodnocení.

Datum

22.5.2024

Podpis oponenta diplomové práce