POSUDEK BAKALÁŘSKÉ PRÁCE

Autor:	Jan Mádle
Název:	Systém pro tvorbu rozvrhu a rezervaci prostor
Posudek vypracoval:	oponent práce RNDr. Ondřej Žára

Bakalářská práce popisuje problematiku rozvrhování a následně realizuje webovou aplikaci pro on-line přístup k rozvrhovacímu systému. Tento je realizován komerčním solverem Gurobi, pro který autor práce sestavil matematický model typu *0-1 ILP*. Serverová část aplikace je psána v Node.js, pro komunikaci se solverem se používá Python API, data jsou ukládána do relační databáze.

Text práce je na výborné typografické úrovni a jazykové nedostatky (anglismy a zejména chybná práce s interpunkcí) se vyskytují v přiměřeném množství. Zdrojový kód je vybaven základním množstvím komentářů.

Výsledná aplikace je funkční, i když její uživatelské rozhraní je velmi prosté a pro náhodného uživatele neintuitivní.

K technické stránce práce mám několik drobných výhrad:

1. **Problematická volba komunikace na serverové straně.** Pro implementaci serverové strany aplikace bylo zvoleno prostředí Node.js, nicméně komunikace se solverem je k dispozici v jazyce Python. Tento externí program s aplikací komunikuje pomocí textových souborů s pevnými názvy, což považuji za velmi nevhodné. Jedná se o prostor pro souběh při paralelním zpracování. Jedním z řešení by bylo generovat názvy souborů náhodně (a předávat je jako parametry); ještě přímočařejší by ovšem bylo implementovat serverovou část aplikace taktéž v Pythonu (např. pomocí frameworků Flask, Django či Tornado).

2. **Nevhodný návrh REST API.** Implementované serverové metody používají HTTP GET a POST nahodile, bez pevné logiky. Ne-idempotentní metody by vždy měly využívat POST (kvůli nepopakovatelnosti volání). Dále řada metod přijímá potenciálně velké množství parametrů, ale jsou realizovány jako HTTP GET, což přináší riziko kolize s maximální délkou požadavku (např. `/updateAvailabilitiesRequest` či `/updatePreferencesRequest`).

3. **Nedostatečné zabezpečení aplikace.** Autor věnoval úsilí korektní implementaci přihlášení, ukládání hesel a obraně proti XSS. Taktéž v rámci obrany před CSRF používá tzv. *SameSite Cookies*. Tato volba je ovšem problematická, protože podpora použité technologie je velmi slabá a útočník se tak může velmi snadno vydávat za přihlášeného uživatele (majícího starší webový prohlížeč) a s jeho autorizací libovolně aplikaci ovládat.

   Zároveň bylo použito nastavení `SameSite=Strict`, které znemožňuje vytvářet odkazy do aplikace pro přihlášeného uživatele. V praxi by bylo vhodnější využít režimu `SameSite=Lax`, nicméně tím by se otevřelo riziko CSRF pro ty metody, které jsou nevhodně implementovány metodou HTTP GET (viz předchozí bod).

Výše uvedené body považuji za výtky, které ovlivňují hodnocení jinak výborného výsledku. Proto navrhuji bakalářskou práci ohodnotit známkou **C – dobře**.

V Praze dne 27. května 2019 RNDr. Ondřej Žára