

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Implementace evidence docházky pro elektronický docházkový systém Mendelovy univerzity v Brně

Bakalářská práce

Vedoucí práce:
Ing. Oldřich Faldík, Ph.D.

Andrey Chernenko

Brno 2023

Poděkování

Chtěl bych poděkovat svému vedoucímu Oldřichu Faldíkovi, Ph.D., za jeho vedení a rady při vypracování této bakalářské práce. Jeho odborné znalosti a rady byly pro mě velkou pomocí. Děkuji také rodině a přátelům za jejich podporu během mého studia.

Čestné prohlášení

Prohlašuji, že jsem práci **Implementace evidence docházky pro elektronický docházkový systém Mendelovy univerzity v Brně** vypracoval samostatně a veškeré použité prameny a informace uvádím v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a v souladu s platnou Směrnicí o zveřejňování závěrečných prací.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

Brno 21.9.2023

.....
podpis

Abstract

This bachelor thesis deals with the implementation of a reports component, which is a part of attendance records in the Smart Mendelu environment. The thesis also contains a general overview of the implementation of microservices and examines the benefits and shortcomings of using a microservices architecture. The solution was developed using the object-oriented programming language Java and its Spring Boot framework, along with JavaScript and its Vue.js framework to create a reactive user interface.

Abstrakt

Tato bakalářská práce se zabývá implementací komponenty sestavy, která je součástí evidence docházky v prostředí Smart Mendelu. Práce dále obsahuje obecný přehled problematiky implementace mikroslužeb a zkoumá přínosy a nedostatky použití architektury mikroslužeb. Řešení bylo vytvořeno s využitím objektově orientovaného programovacího jazyka Java a jeho frameworku Spring Boot, spolu s JavaScriptem a jeho frameworkem Vue.js pro vytvoření reaktivního uživatelského rozhraní.

Obsah

1	Úvod	6
1.1	Úvod do práce	6
1.2	Cíl práce	6
2	Současný stav evidence docházky	7
2.1	Architektura	7
2.2	Návrh	7
2.3	Bezpečnost	7
2.4	Výhody použití nových technologií	8
2.5	Současné trendy vývoje webových aplikací	8
3	Postup implementace	9
3.1	Backendová část	9
3.2	Frontendová část	9

1 Úvod

1.1 Úvod do práce

Evidence docházky je důležitou součástí správy lidských zdrojů ve většině organizací různé velikosti. Elektronické docházkové systémy se staly běžnou cestou pro správu docházky, protože poskytují vysokou úroveň automatizace a přesnosti. Volba určitých nástrojů a architektur pro sestavení a zprovoznění aplikací má své výhody a nevýhody.

1.2 Cíl práce

Cílem této práce je definovat problémy spojené s podstatou implementace evidence docházky a vymežit funkční a nefunkční požadavky, které by tyto problémy pokryly. Během implementace využít výhod mikroservisní architektury na backendové straně a architektury jednostranových aplikací na frontendové straně. Budou představeny různé metody a technologie, které se používají při vývoji informačních systémů.

2 Současný stav evidence docházky

2.1 Architektura

Komponenta ED, která je využívána v současné době v produkčním prostředí, je postavena na monolitické architektuře a napsána v skriptovacím jazyce PHP. Tento přístup zahrnuje:

- Jednu základnu kódu
- Jednu databázi pro všechna data
- Jeden nebo několik serveru, na kterých běží instance aplikace
- Jeden bod vstupu pro všechny požadavky (index.php)

Výhody tohoto přístupu jsou v tom, že za prvé je jednoduchá a pochopitelná na začátku vývoje. Za druhé je dobře škálovatelná v rámci jednoho zařízení, protože základná kódu se nachází na jednom místě. Ze stejného důvodu je snadno jí nasazovat, testovat a je proto nákladově efektivní. Evidence docházky není jedinou komponentou, která běží na serveru, spolu s ní neoddělitelně běží i další částí aplikace. Během času kódová základna informačního systému docházky se bude rozšiřovat, částí kódu se stanou na sobě více závislé. Kvůli tomu ladění a oprava chyb, testování jednotlivých částí kódu, nasazení budou probíhat obtížněji, protože se zvyšuje riziko poškodit program na jiném místě. Při škálování monolitického systému také nastává problém protože pokud bude třeba oddělit a přenést na jiné zařízení určitou část systému, to prakticky nebude uskutečnitelné.

2.2 Návrh

O vizuálním designu aplikace. Body k zohlednění:

- Jasnost a Srozumitelnost
- Přehlednost
- Responzivita

2.3 Bezpečnost

Body k zohlednění:

- Zranitelnosti použitých knihoven
- Validace Vstupů
- Ochrana proti CSRF, XSS, SQL Injection
- Šifrování

2.4 Výhody použití nových technologií

Podle nefunkčních požadavku popsaných v předchozí kapitole vyhovující architekturou na straně serveru bude architektura mikroslužeb. Podle Newmana (2015) mikroslužby jsou přístupem distribučních systémů které propagují použití dobře rozdělených služeb se samostatnými životními cykly, pro vzájemnou spolupráci. Mikroslužby mají několik výhod oproti monolitické architektuře, mezi nimi patří:

2.5 Současné trendy vývoje webových aplikací

(ne)krátký přehled současných architektur a principů postavení aplikací

- Škálovatelnost, díky které částí aplikaci se dá spustit nezávislé a na různých zařízeních.
- Odolnost: pokud z určitých důvodů komponenta přestane fungovat, ostatní komponenty zůstanou pokračovat v běhu.
- Znovupoužitelnost: v případě nadměrné zátěže na určitou komponentu lze tuto komponentu duplikovat a provést load balancing.
- Testování: jednotlivé mikroslužby je možné testovat nezávisle na ostatních.
- Jednoduchost vnímání a pochopení programu: v menší komponentě lze snadněji sledovat závislosti kódu.

3 Postup implementace

3.1 Backendová část

Popís struktury Spring Boot aplikace: endpointy, napojení na databazi, použité nástroje/balíčky

3.2 Frontendová část

Popís struktury Vue.js aplikace: implementace vizualních komponent (tabulky, uživatelský panel), autentikace, použité nástroje/balíčky.

Poznámky:

- Konfigurace adapteru pro identity management nástroj Keycloak, vlastní plugin Auth pro Vue
- Cross-site request forgery <https://security.stackexchange.com/questions/20187/oauth2-cross-site-request-forgery-and-state-parameter>
- Balík vue-router pro routing v SPA aplikacích, princip routování u spa aplikací
- State management plugin Vuex pro framework Vue