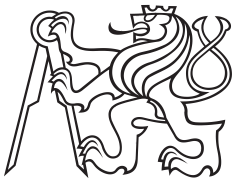
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Systém pro správu přístupů v Centru znalostního managementu

Kateřina Dvořáková

Vedoucí: Ing. Pavel Náplava, Ph.D.
Studijní program: Otevřená informatika
Specializace: Software
Květen 2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Dvořáková** Jméno: **Kateřina** Osobní číslo: **499208**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Specializace: **Software**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Systém pro správu přístupů v Centru znalostního managementu

Název bakalářské práce anglicky:

Access Management System in the Center for Knowledge Management

Pokyny pro vypracování:

Analyzujte prostředí Centra znalostního managementu (CZM), zaměřte se na uživatelské přístupy k systémům, místnostem atd. Navrhněte systém, který správu těchto přístupů zjednoduší. Postupujte následujícím způsobem:
1) Popište CZM, jeho fungování, používané systémy atd.
2) Popište způsob přidělování/odebírání přístupových práv, zaměřte se na jeho silné a slabé stránky
3) Analyzujte možnosti automatizace řízení přístupových práv od jejich evidence, přes automatické přiřazení k vybraným systémům až po automatické odebrání
4) Navrhněte systém, který bude na základě Vašich analýz správu přístupů podporovat
5) Navržený systém implementujte
6) Funkčnost systému ověřte prostřednictvím uživatelských testů

Seznam doporučené literatury:

[1] Spojujeme výuku s praxí. [online]. Centrum znalostního managementu FEL ČVUT. Dostupné z: https://czm.fel.cvut.cz/cs/.
[2] The Camunda Platform 7 Manual. [online]. Dostupé z https://docs.camunda.org/manual/7.20/.
[3] Why Spring? [online]. Dostupné z: https://spring.io/why-spring/.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Náplava, Ph.D. Centrum znalostního managementu FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **05.02.2024** Termín odevzdání bakalářské práce: **24.05.2024**

Platnost zadání bakalářské práce: **21.09.2025**

Ing. Pavel Náplava, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. Datum převzetí zadání

Podpis studentky

Poděkování

V první řadě děkuji svému vedoucímu práce panu Ing. Pavlu Náplavovi, Ph.D. za jeho cenné rady a vřelý přístup.

Dále bych chtěla vyjádřit vděčnost svým nejbližším a přátelům, kteří mě během studia podporovali.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškerou použitou literaturu.

V Praze, 24. května 2024

Abstrakt

Tato práce se zabývá automatizací správy uživatelských přístupů v Centru znalostního managementu FEL ČVUT. Hlavním cílem je snížit administrativní zátěž spojenou s přijímáním nových stážistů a zjednodušit proces přidělování přístupů pomocí automatizovaných postupů.

První část práce se zaměřuje na analýzu stávajícího systému správy přístupů v Centru znalostního managementu a následně na návrh nového systému, který optimalizuje správu přístupů. V další části práce je navržený systém implementován do integrační platformy HUB.FEL, která je v Centru znalostního managementu využívána i pro další projekty. Dále je nový systém podroben uživatelskému testování, které zahrnuje jak testování ze strany administrátorů, kteří přidělují přístupy stážistům v Centru znalostního managementu, tak testování ze strany uživatelů, kteří získávají přístupová práva a vyplňují k nim potřebné údaje.

Klíčová slova: správa přístupů, automatizace procesů, Spring Boot, Camunda, GitLab

Vedoucí: Ing. Pavel Náplava, Ph.D.

Abstract

This thesis deals with the automation of user access management in the Center for Knowledge Management FEE CTU. The main objective is to reduce the administrative burden associated with the recruitment of new interns and to simplify the process of access granting using automated procedures.

The first part of the thesis focuses on the analysis of the existing access management system in the Center for Knowledge Management and then on the design of a new system that optimizes access management. In the next part of the thesis, the proposed system is implemented in the HUB.FEL integration platform, which is also used in the Center for Knowledge Management for other projects. Furthermore, the new system is subjected to user testing, which includes both testing by administrators, who assign accesses to interns in the Center for Knowledge Management, and testing by users, who obtain access rights and fill in the necessary data for them.

Keywords: access management, process automatization, Spring Boot, Camunda, GitLab

Title translation: Access Management System in the Center for Knowledge Management

Obsah

Obrázky

Kapitola 1

Úvod

Tato práce se nejprve věnuje analýze prostředí Centra znalostního managementu FEL ČVUT s důrazem na přidělování a odebírání uživatelských přístupů k systémům. Na základě analýzy jsou popsány možnosti optimalizace řízení přístupových práv. Tyto možnosti zahrnují systematickou evidenci práv, automatizaci přidělování a odebírání přístupových práv tam, kde je to možné, a spouštění pracovních postupů v případech, kdy není automatizace proveditelná. Dále je navržen systém, který usnadní správu přístupů v souladu s provedenými analýzami. Následně je první verze tohoto systému implementována. V poslední fázi je systém podroben uživatelskému testování.

Motivací pro tento projekt je mé působení v CZM, kde přidělování a odebírání přístupových práv k systémům a službám je často prováděno manuálně. Tato ruční práce je zbytečně zdlouhavá, protože některé procesy by bylo možné zautomatizovat nebo alespoň podpořit proces jejich správy. Současně chybí systematická evidenční struktura, která by poskytovala jasný přehled o tom, kdo a kdy přístup přidal nebo odebral.

Cílem je zjednodušit správu uživatelských přístupů, evidovat a zautomatizovat přidělování a odebírání některých přístupových práv.

Kapitola 2

Situační analýza

2.1 Centrum znalostního managementu FEL ČVUT

Centrum znalostního managementu (CZM) je univerzitní centrum, které mimo jiné zapojuje studenty a absolventy ČVUT do svého stážového programu, kde jim poskytuje příležitost podílet se na rozvoji fakultních i externích IT projektů.

CZM se specializuje na analýzu a návrh technologických řešení, procesní mapování a optimalizaci, elektronizaci procesů, školení a implementaci informačních systémů. Mezi další aktivity CZM patří realizace workshopů a také se zapojuje do výuky [1].

Jelikož správa přístupů, která je předmětem této práce, se v CZM týká především stážového programu, budu se mu v následující části věnovat.

2.1.1 Fungování stážového programu

Nově přijatý stážista nastupuje do CZM na stážovou pozici či projekt, který byl dohodnut s HR (Human Resources) manažerem. Před nástupem na stáž je nutné, aby HR manažer zařídil pro stážistu přístupy do kanceláří, všech systémů a služeb, které bude pro danou pozici nebo projekt využívat.

Dále mají stážistovy přístupy na starosti mentoři, kterými jsou buď vedoucí projektu, pod kterým stážista je, nebo garant pozice, kterou nový stážista zastává.

Stážisté mají možnost si v CZM vyzkoušet více pozic a projektů. To při přechodu na jinou pozici způsobuje, že HR manažer a mentoři musí zajistit přístupy do dalších systémů používaných novou pozicí/projektem. V případě odchodu z pozice/projektu jsou HR manažerem nebo mentory odebrány přístupy ze systémů, které stážista nebude pro svou práci potřebovat.

Po ukončení stáže je důležité správně řídit proces odebrání přístupů stážistům, přičemž je třeba vzít v úvahu kontinuitu projektů. Není nezbytné odebrat všechny přístupy, zejména pokud stážista vykonával seniorní pozici, aby bylo možné udržovat spojení i po ukončení stáže.

2.1.2 Stážové pozice centra

Stážová pozice CZM je pracovní příležitost pro studenty, kde mohou získat praktické zkušenosti v oblasti IT a byznysu a pracovat na reálných projektech. V rámci CZM je jich v nabídce několik [1]:

- **Analytik**
- **DevOps engineer/SysAdmin**
- **Front-end vývojář**
- **Java vývojář**
- **PHP vývojář**
- **PR/Marketing specialista**
- **Projektový manažer**
- **Tester engineer**
- **UX/UI designér**

2.1.3 Projekty centra

CZM nabízí řadu projektů, na nichž má stážista možnost aktivně spolupracovat. Jedná se většinou o fakultní aplikace, mezi které patří:

- **Eprocesy** řeší elektronizaci procesů na fakultě.
- **Evaluace zaměstnanců** slouží k pravidelnému ročnímu hodnocení akademických a vědeckých pracovníků.
- **Felsight** nabízí především správu studijních aktivit a povinností, plánování schůzek, sledování změny v rozvrhu, vyhledávání informací o předmětech, učitelích a místnostech.
- **Hodnocení doktorandů** umožňuje zaznamenávat hodnocení doktoranda za uplynulý semestr.
- **HUB.FEL** je platforma, do které je integrováno několik služeb, které jsou poskytovány fakultě, mezi ně patří Evaluace zaměstnanců a Hodnocení doktorandů. Platformu je plánováno rozšířit o funkcionality, které nabízí Eprocesy a Témata.
- **Kometa** vypočítává podíl pedagogického výkonu jednotlivých kateder, na základě kterého jsou pak rozdělovány finance.
- **Moodle** je využíván jak studenty, tak i učiteli ke sdílení materiálů, psaní testů, odevzdávání úkolů a dalším činnostem souvisejícím s výukou.
- **Témata** podporují vypisování semestrálních, disertačních a týmových projektů.
- **Navigate FEL** poskytuje navigační služby prostřednictvím webu a navigačních kiosků umístěných u vstupů do budov fakulty.

2.1.4 Systémy a služby využívány centrem

CZM využívá několik systémů a služeb, které fungují jako nástroje pro usnadnění organizace práce, podporu efektivní komunikace v týmu a také pro vývoj a následnou správu aplikací. Řada z nich je určena pro použití pouze konkrétními pracovními pozicemi či výhradně pro projekty, které vyžadují specifickou funkcionalitu. Ve výčtu níže jsou popsány systémy, jež jsou využívány všemi stážisty:

- **GitLab** umožňuje vývoj software, jeho verzování, dokumentaci, testování, nasazení a údržbu [2]. CZM využívá fakultní GitLab pro správu svých projektů, které jsou přístupné pouze členům podílejících se na daném projektu.
- **Google Calendar** poskytuje centru několik kalendářů, které slouží k rezervaci kanceláří a zobrazování pracovního plánu stážistů. Každá stážová pozice má svůj vlastní kalendář. Stážista tedy zaznamenává svůj harmonogram do odpovídajícího kalendáře své stážové pozici.
- **Sharepoint** poskytuje prostor pro tvorbu a sdílení dokumentů. ČVUT využívá tuto platformu, přičemž CZM disponuje svým úložištěm pod Fakultou elektrotechnickou.
- **Slack** slouží jako nástroj pro komunikaci mezi jednotlivými stážisty a v rámci týmu, zároveň umožňuje distribuci informací, které mají vztah k celému centru. CZM využívá neplacenou verzi.
- **SysPass** je databáze citlivých údajů jakou jsou například přístupová data do databází nebo tokeny. Díky tomuto systému není nutné, aby administrátor hesla posílal běžnými online komunikačními prostředky (jako je například Slack) nebo bylo nutné si je pamatovat. V SysPassu jsou hesla rozdělena do různých skupin a uživatelé mají přístup pouze k heslům ve skupině, která jim byla přidělena. To zajišťuje, že každý uživatel vidí pouze relevantní hesla.
- **Teams** je komunikační platforma pro on-line schůzku. CZM jej využívá pro schůzky v případě, kdy se nemohou všichni zúčastnit prezenčně.
- **VPN** umožňuje vzdálený přístup k interním systémům CZM.
- **Wi-Fi v kancelářích CZM** poskytuje bezdrátové připojení k internetu v kancelářích CZM a také přístup k interním systémům CZM.

Existují další systémy, které jsou využívány pouze pro určité pozice nebo projekty, jako je Adobe XD, Figma, Grafana, HelpDesk, Kibana a SonarQube. V této práci se budu primárně zabývat systémy, které používá většina stážistů.

2.1.5 Přidělování přístupových práv v centru

Pro nově přijatého stážistu existuje spustitelný proces Nástup nového stážisty 2.1.6 v nástroji Workflow Center od IBM [3]. Přidělování přístupů během stáže pro konkrétní projekt či pozici funguje tak, že stážista požádá o přístup svého mentora, nebo přímo zodpovědnou osobu za systém. Nejčastěji se přístup komunikuje skrze Slack nebo manuálně vytvářené GitLab issue.

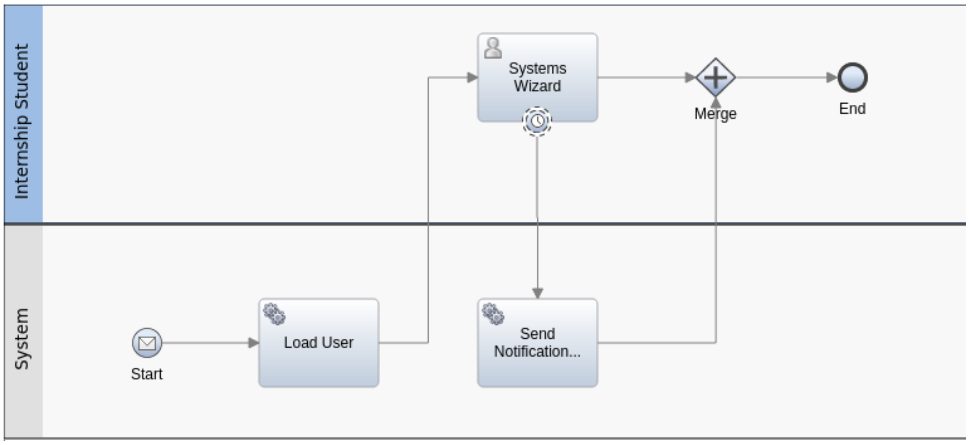
2.1.6 AS-IS Proces Nástup nového stážisty

HR manažer v procesním portálu poskytovaným IBM [4] na adrese bpm.cvut.cz zadá uživatelské jméno stážisty, pro kterého chce spustit proces Nástup nového stážisty zobrazený na obrázku 2.1. Systém získá e-mailovou adresu nového stážisty. Poté se spustí podproces zobrazený na obrázku 2.2, ve kterém systém zašle stážistovi odkaz na formulář v procesním portálu, kde vyplní údaje zobrazené na obrázku 2.3. Poté systém zašle e-mailem žádost o zajištění přístupových údajů k VPN a Wi-Fi hlavnímu systémovému administrátorovi. Ten většinou tuto žádost skrz GitLab issue předává kolegovi, který přístupy zařídí.

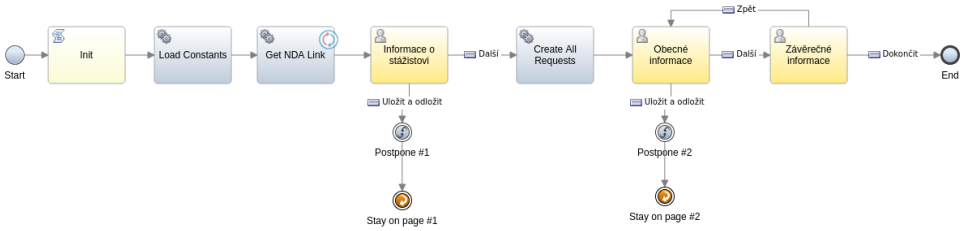
Mezitím HR manažer:

- poskytne vedoucímu CZM údaje o stážistovi, jemuž zajistí roli CZM v rámci ČVUT, díky které bude mít stážista přístup do kanceláří CZM a Sharepoint úložiště CZM;
- manuálně přidá stážistu do CZM workspace ve Slack;
- manuálně přidá stážistu do projektu CZM v GitLab;
- manuálně přidá stážistu do Google kalendářů CZM.
- manuálně přidá stážistu do týmu CZM v Teams.

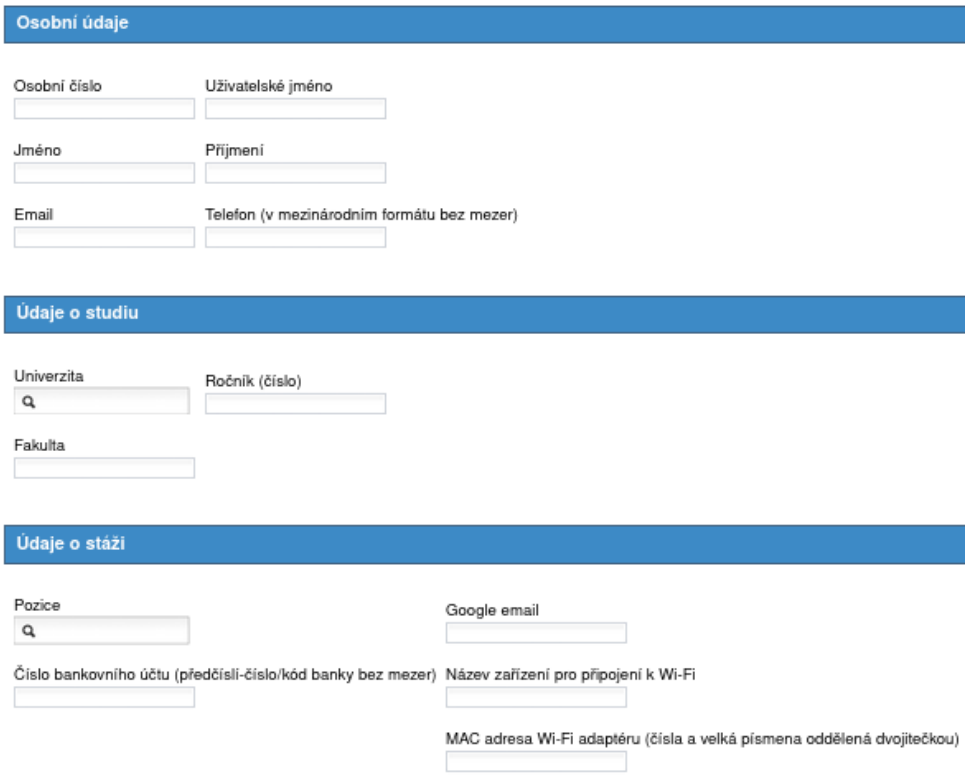
Pokud stážista studuje na jiné fakultě než elektrotechnické, musí vedoucímu CZM poskytnout číslo bankovního účtu pro účely vyplacení stipendia. Proces je dokončen, pokud stážista potvrdí pomocí tlačítka, že má již přístupy do všech konkrétních systémů jak lze vidět na obrázku 2.4.



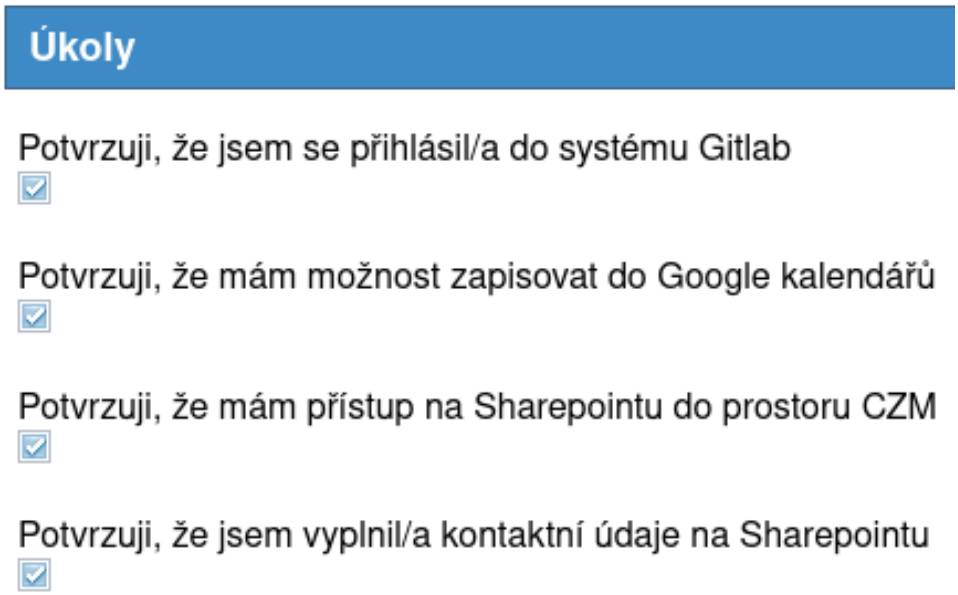
Obrázek 2.1: Hlavní proces Nástup nového stážisty v IBM Workflow Center



Obrázek 2.2: Podproces Systems Wizard v IBM Workflow Center



Obrázek 2.3: Požadované údaje nového stážisty



Obrázek 2.4: Výčet požadovaných přístupů pro nového stážistu

Tento proces byl navržen před několika lety a v současné době není v CZM k dispozici nikdo, kdo má zkušenost s výše zmíněním nástrojem Workflow Center od IBM. Zároveň se ani další projekty pomocí tohoto nástroje nevyvíjejí a neudržuje se tak nezbytné know-how pro provoz nebo rozvoj existujícího procesu.

2.1.7 Odebírání přístupových práv v centru

Odebírání přístupových práv stážistům, kteří odcházejí z CZM, není řízeno jednotným procesem, jelikož je potřeba udržet s odcházejícími stážisty kontakt, neboť jsou často nejlépe orientováni v některém konkrétním projektu. Většinou jsou odebírána pouze práva k citlivým systémům, jako je například SysPass, zatímco přístup ke komunikačním platformám zůstává zachován.

V případě, že uživatel potřebuje odebrat práva stážistovi k nějakému konkrétnímu systému, musí se obrátit na odpovědnou osobu, která je oprávněna tato práva odebírat.

2.1.8 Správa přístupových práv k systémům centra

Tato část je věnována procesům a postupům přidělování a odebírání přístupových práv k systémům popsaným v sekci 2.1.4, které jsou společně využívány všemi stážisty.

- **GitLab** Správce projektu CZM v GitLab přidává nové stážisty nebo je odebírá. Role uživatelů v podprojektech jsou následně přidělovány nebo odebírány osobou s odpovídajícími oprávněními
- **Google Calendar** Správce CZM kalendářů přidává nové stážisty do každého kalendáře zvlášť. Pro odebrání přístupu z kalendářů musí správce odebrat uživatele z každého kalendáře individuálně.
- **Sharepoint** Stážista získá přístup k úložišti CZM v rámci ČVUT poté, co mu je v databázi ČVUT přiřazena role člena CZM "B-13393-CLEN". Odebrání přístupu k úložišti CZM následuje v případě, že je stážistovi odstraněna tato role.
- **Slack** Správce CZM workspace přidává nové stážisty ve Slack, který poté automaticky zašle na e-mail novému členovi pozvánku, pomocí které se do CZM workspace připojí. Po přidání má stážista dostupné všechny kanály. Pro odebrání stážisty musí správce Slack deaktivovat účet stážisty. Tím je odebrán ze všech komunikačních kanálů a nemůže se již znovu přihlásit.
- **SysPass** Správce SysPass pro nového stážistovi v SysPass vytvoří účet, vygeneruje pro něj heslo, které zašle stážistovi přes Slack. Pro odebrání přístupu správce odstraní účet ze SysPass.
- **Teams** Správce CZM týmu přidává a odebírá manuálně stážisty v Teams. Stážista má po přidání přístup do všech kanálů týmu CZM.

Tato kapitola ukazuje, že proces přidělování a odebírání přístupů je komplexní a zahrnuje úlohy HR manažera, mentorů a správců jednotlivých systémů. Obsahuje řadu manuálních opakujících se kroků.

Následující kapitola se bude věnovat problémům spojeným se správou přístupů a jejich řešení.

Kapitola 3

Možnosti optimalizace správy přístupových práv

Tato kapitola se zaměřuje na identifikaci problémů správy přístupových práv vyplývajících z kapitoly 2 a představuje možná řešení těchto problémů.

3.1 Identifikace problematických oblastí

1. **Manuální procesy při přidělování a odebírání přístupů:** Přidělování a odebírání přístupů vyžaduje několik manuálních kroků, což zpomaluje celkový proces.

2. **Decentralizovaná správa přístupů:** Informace o přidělení přístupů jsou rozptýlené na různých místech, což znesnadňuje celkový přehled a dohled nad tím, kdo má přístup k jakým systémům.

3. **Nemožnost dohledat historii přidělených přístupů:** Některé přístupy jsou zařizovány prostřednictvím komunikačních platforem, jako je Slack, což komplikuje sledování historie přidělených práv. Navíc v bezplatné verzi Slack jsou zprávy automaticky mazány po 90 dnech, což znesnadňuje zpětné zjištění, kdo a kdy přístup přidělil nebo odebral.

4. **Nedostatek dokumentace o požadavcích na přístupy:** HR manažeři a mentoři si musejí pamatovat, jaké přístupy jsou potřebné pro jednotlivé projekty a pozice, protože tato informace není dostatečně zdo-

kumentována. Tento nedostatek dokumentace zvyšuje administrativní zátěž a riziko chyb.

5. **Absence odborníků ohrožuje schopnost opravy procesu nástupu nového stážisty:** Vzhledem k tomu, že v centru chybí odborníci na nástroj, ve kterém je proces implementován, nebude možné v případě selhání procesu zajistit jeho opravu.

6. **Neefektivita procesu nástupu nového stážisty** V procesu Nástup nového stážisty 2.1.6 se vyskytuje několik neefektivních kroků, které proces zpomalují:

   a. Stážista vyplňuje údaje o studiu, které lze získat pomocí KOSapi poskytující mimo jiné údaje o studiu studenta [5].

   b. Typ pozice nový stážista vyplňuje sám, přestože ji finálně před přijetím potvrzuje HR manažer. Tedy pozici pro proces ve výsledku neurčuje HR manažer, jak tomu v realitě je.

   c. Hlavní systémový administrátor musí manuálně předávat žádost o VPN a Wi-Fi svému kolegovi.

   d. Po vyplnění požadovaných údajů stážistou musí HR manažer a systémový administrátor zařídit několik přístupů manuálně.

   e. HR manažer posílá vedoucímu CZM zprávu o nově přijatém stážistovi s žádostí o zařízení role CZM.

3.2 Navrhované možnosti optimalizace

V sekci 3.1 jsou identifikovány jednotlivé problémy spojené se správou přístupových práv a pro každý z nich je v této části navrženo řešení.

1. **Automatizace procesů** Zavedení automatizovaných workflow pro přidělování a odebírání přístupů je detailně popsáno v sekci 3.2.1.

2. **Centralizace správy přístupových práv:** Implementace systému pro správu přístupových práv, který by umožnil jednotné přidělování a odebrání přístupů pro všechny systémy. Tento systém by centralizoval správu uživatelů a umožnil lepší přehled o aktuálních přístupech.

3. **Dohledatelná historie přidělených přístupů:** Zavedení systému, který ukládá a udržuje historii přidělených přístupů, umožní snadnější zpětné zjištění, kdo a kdy přístup přidělil nebo odebral. Tento systém by měl být integrován do existujících komunikačních platforem nebo správy přístupů.

4. **Dokumentace požadavků na přístupy:** Vytvoření centralizovaného dokumentu nebo databáze, která obsahuje informace o požadavcích na přístupy pro jednotlivé projekty a pozice, sníží administrativní zátěž a riziko chyb. Tato dokumentace by měla být přístupná a aktualizovatelná pro všechny relevantní osoby.

5. **Zavedení nového procesu Nástup nového stážisty:** CZM nyní pro management procesů používá technologii Camunda [6], která je podobná nástroji Workflow Center od IBM. Vzhledem k tomu, že Camunda je již implementována v CZM pro projekty jako Evaluace zaměstnanců a Hodnocení doktorandů, nemá smysl zavádět jiný nástroj pro správu procesů a lze proces nástupu nového stážisty 2.1.6 pomocí tohoto nástroje implementovat a monitorovat.

6. **Zefektivnění procesu nástupu nového stážisty:** Proces 2.1.6 lze zefektivnit v následujících krocích:

    a. Na základě zadaného ČVUT uživatelského jména nového stážisty proces získá pomocí KOSapi údaje o studiu.

    b. Pozici pro proces vyplňuje HR manažer.

    c. Po zadání MAC adresy stážistou se automaticky vytvoří GitLab issue pro přístupy k Wi-Fi a VPN.

    d. Automatizace přidělení přístupů, která je podrobněji popsána v sekci 3.2.1.

    e. Po vyplnění osobních údajů stážistou se automaticky odešle vedoucímu CZM žádost o zařízení CZM role pro nového stážistu.

Tyto návrhy mohou pomoci zlepšit efektivitu a spolehlivost správy přístupů a procesů v centru.

3.2.1 Automatizace správy přístupů k systémům CZM

Některé systémy, které CZM využívá, vystavují API (Application Programming Interface) poskytující operace pro správu uživatelských přístupů. Pokud některé systémy neumožňují automatizaci pomocí API, lze alespoň zefektivnit postup tak, že bude pro zodpovědné osoby vytvořen úkol s požadavkem na manuální přidělení nebo odebrání přístupu. Vzhledem k tomu, že CZM používá systém GitLab, který disponuje API pro správu issue [7], bude tento úkol zadán jako GitLab issue.

Níže jsou rozepsány možnosti automatizace přístupů k systémům CZM popsaným v sekci 2.1.8:

- **GitLab:** Pro GitLab lze spravovat členy skupin a projektů CZM pomocí GitLab API [7].
- **Sharepoint:** Získání role CZM pro přístup do Sharepointu lze urychlit odesláním notifikace vedoucímu CZM.
- **Slack:** Slack vystavuje API umožňující přidání a odebrání členů pomocí oprávnění, které je dostupné pouze pro předplatné "Enterprise Grid"[8]. CZM však používá neplacenou verzi, takže tato funkcionalita není k dispozici. Správce CZM workspace ale může vygenerovat odkaz pro pozvání uživatele, který může využít až 400 lidí [9]. Tento odkaz lze použít pro optimalizaci přidělení přístupu tím, že se automaticky odešle uživateli žádajícímu přístup do CZM workspace ve Slack.
- **SysPass:** SysPass API nenabízí operace pro změnu uživatelů [10]. Lze tedy alespoň využít automatické vytvoření GitLab issue, které proces přidání nebo odebrání přístupu usnadní.
- **Teams:** Pro správu členů týmu v Teams musí vlastník týmu získat token v Azure Active Directory, kterým se autorizuje pro Microsoft Graph API. Pomocí tohoto API může do týmu přidávat a odebírat z něj členy organizace, pod kterou tým patří [11].

V této kapitole jsem prozkoumala možnosti optimalizace správy uživatelských přístupů. Ukázalo se, že v některých případech je možné využít API pro automatizaci procesů, zatímco v jiných případech není tato možnost dostupná. Efektivním řešením by mohl být implementovat systém, který je detailně představen v následující kapitole.

Kapitola 4

Návrh systému pro správu přístupů

Tato kapitola se věnuje návrhu systému pro správu přístupů v CZM na základě popisu fungování správy přístupů z kapitoly 2 a následné analýze týkající se optimalizace správy přístupů provedené v kapitole 3.

4.1 Funkční požadavky

Systém by měl vyhovovat následujícím funkčním požadavkům:

1. Systém umožní administrátorům (HR manažeři, mentoři) přidělit přístup stážistovi pro určité skupiny.
2. Systém umožní administrátorům změnit přístup stážistovi, je-li změna přístupu možná.
3. Systém umožní administrátorům odebrat určitý přístup stážistovi.
4. Systém uchovává záznamy o přidělení a odebrání přístupů.
5. Systém umožní administrátorům spuštění procesu nástupu nového stážisty.
6. Systém umožní při spuštění procesu nástupu nového stážisty výběr přístupů, které stážista potřebuje.

7. Systém umožní změnu administrátorů, kteří mohou spouštět proces nástupu nového stážisty.
8. Systém umožní vedoucímu nějaké skupiny přidělit či odebrat přístup stážistovi k systémům v rámci dané skupiny.

4.2 Nefunkční požadavky

1. Systém by měl využívat již používané technologie v CZM pro jeho snadnější implementaci a následnou údržbu.
2. Systém udržuje informace o tom, které projekty nebo pozice (dále označované jako skupiny) vyžadují specifické přístupy.
3. Systém je snadno použitelný pro stážisty, kteří poskytují dodatečné údaje pro přístup.
4. Systém je snadno použitelný pro uživatele, kteří budou spravovat přístupová práva.
5. Systém by měl být schopen integrace s existujícími systémy poskytujícími API pro správu přístupů.

4.3 Diagram tříd

Pro návrh diagramu tříd systému pro správu přístupů, který vyhovuje výše zmíněným uvedeným požadavkům, navrhuji následující datovou strukturu reprezentovanou diagramem tříd 4.1:

- **User:** V kontextu CZM je uživatel člen CZM, který potřebuje nějaké přístupy (UserAccess) k systémům (System). Může se jednat jak o běžného stážistu, který nespravuje přístupy, tak o administrátora, který přístupy spravuje.
- **Group:** Tato třída definuje skupinu (pozici nebo projekt), která může obsahovat několik členů reprezentovaných třídou GroupRole. Skupina může poskytovat pro své členy přístupy (UserAccess) k systémům (System), ke kterým může přistupovat daná skupina. Existují dva typy skupin: POSITION a PROJECT.

- **GroupRole:** Tato třída vyjadřuje členství nějakého uživatele (User) v nějaké skupině (Group). Dostupné typy členství ve skupině jsou následující:
  - ADMIN: vyjadřuje správce skupiny, tedy HR manažera nebo mentora
  - USER: jedná se běžného člena skupiny bez možnosti zásahu do skupiny, tedy o nějakého stážistu pracujícím na nějaké pozici nebo projektu.
  - MANAGER: tento typ existuje pro identifikaci vedoucího CZM.

  Role vždy jsou spojeny s určitou skupinou (Group). Pokud jde o skupinu typu GENERAL a roli typu ADMIN nebo MANAGER, uživatel disponuje nejvyššími oprávněními v systému, jak je dále popsáno v případech užití pro roli ADMIN 4.4. Tato třída také uchovává informace o vytvoření a odstranění role, včetně času a uživatele, který provedl změnu. Role je aktivní, pokud datum odebrání je prázdné.

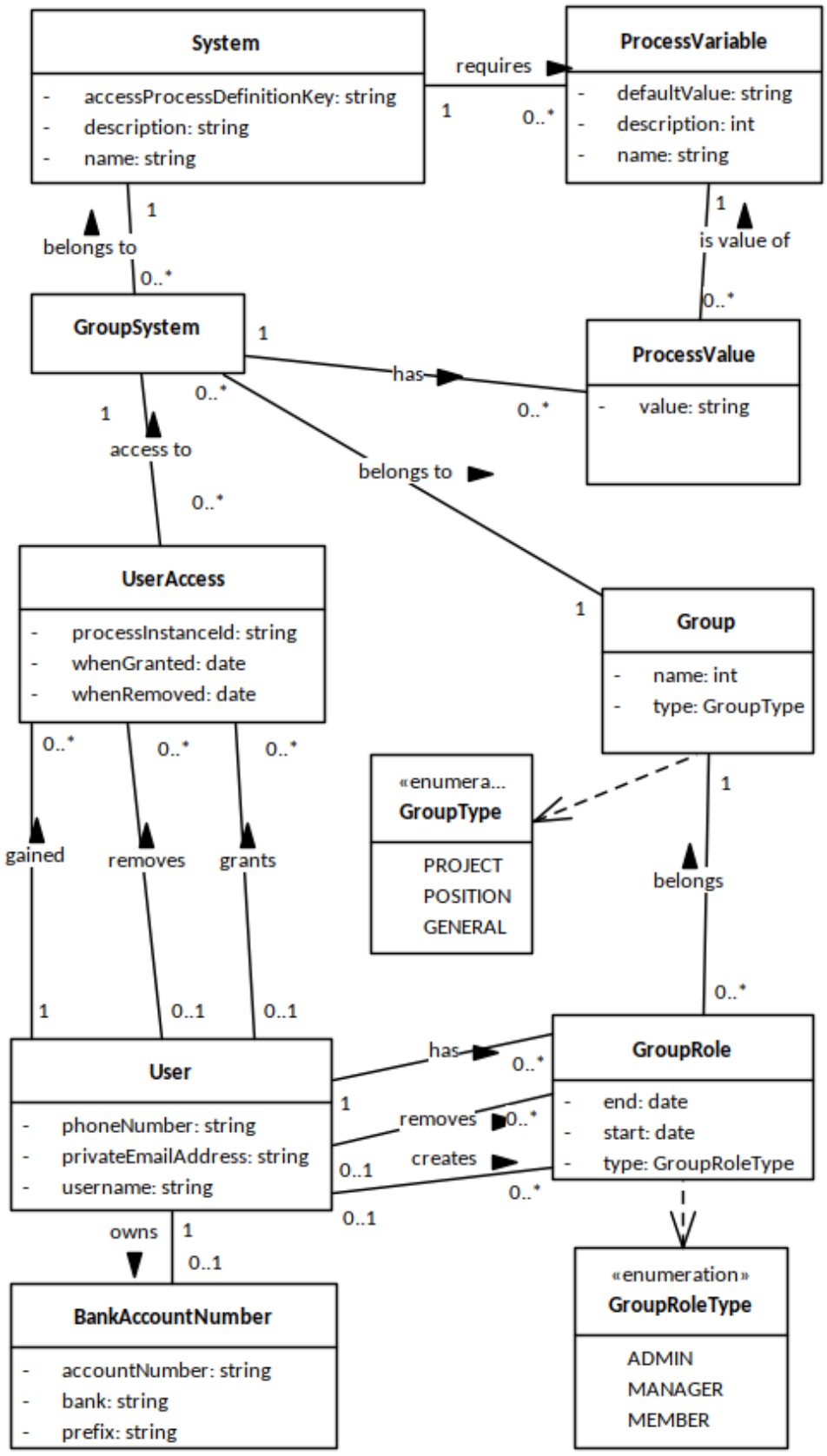
- **System:** Znázorňuje systém, který se v CZM využívá. K systému může existovat proces, který spravuje přístup k němu. Procesy jsou podrobně rozebrány v sekci 4.5.

- **UserAccess:** Tato třída uchovává informace o tom, kdy a kým byl uživateli přidělen nebo odebrán přístup k nějakému systému (System) v rámci skupiny (Group).

- **ProcessVariable a ProcessValue** Pro vysvětlení potřeby těchto tříd uvedu příklad: Skupina (Group) chce spravovat přístup k systému (System) GitLab. Takových skupin je v CZM několik. A proto je třeba nějak definovat, pro jakou GitLab skupinu bude přístup pomocí procesu spravován. Toho lze docílit tak, že systém pro spuštění procesu správy přístupu bude vyžadovat procesní proměnnou (ProcessVariable) s názvem `groupOrProjectId` a skupina (Group) pomocí třídy ProcessValue definuje hodnoty pro tuto procesní proměnnou. která patří k nějaké skupině (Group) a procesní proměnné (ProcessVariable). Díky tomuto přístupu správce skupiny nemusí při přidělování přístupu specifikovat pro jaké GitLab ID projektu nebo skupiny se má spustit proces správy přístupu ke GitLab, který je detailně popsán v sekci 4.5.2.

- **GroupSystem:** Tato třída značí, že skupina (Group) používá systém (System) a tedy správci dané skupiny mají právo přidělovat a odebírat přístup k danému systému.

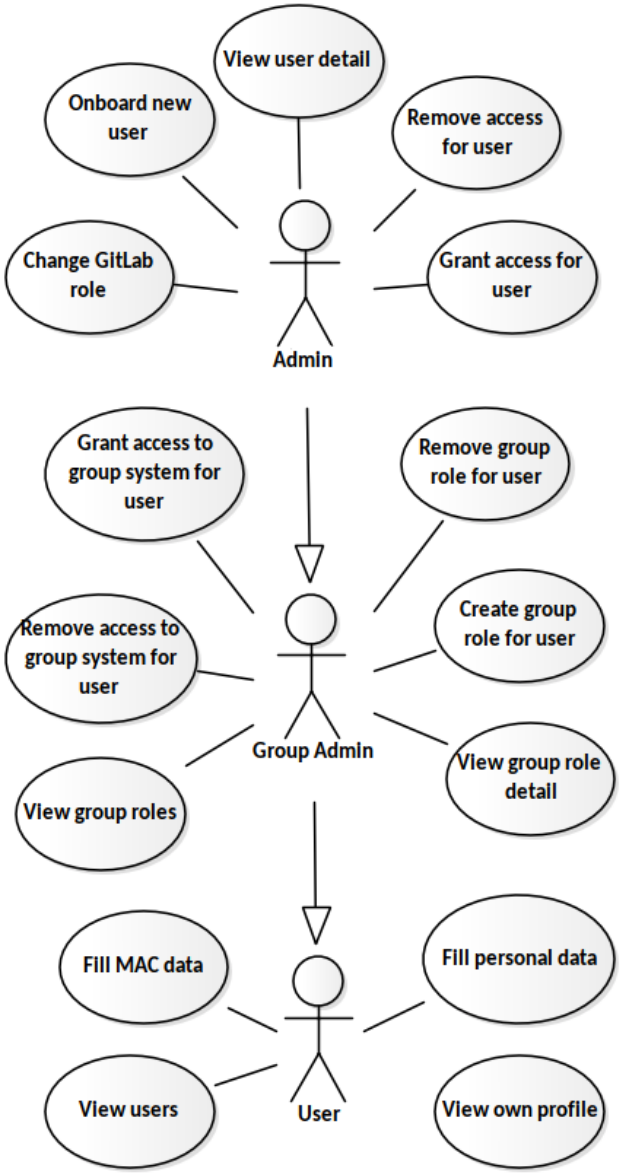
- **BankAccountNumber:** Třída vyjadřuje číslo bankovního účtu uživatele (User), které je poskytnuto v rámci procesu nástupu nového stážisty 2.1.6.



Obrázek 4.1: Diagram tříd

4.4 Případy užití

Operace, které lze v systému provádět v rámci rolí hlavního administrátora (Admin), správce skupiny (Group Admin) a běžného uživatele (User) jsou vizualizovány v diagramu případů užití 4.2.



Obrázek 4.2: Diagram případů užití

4.4.1 Role

- **Admin** je nejprivilegovanější role. V kontextu CZM se jedná o HR manažera a vedoucího CZM. V kontextu diagramu tříd se jedná o uživatele, kteří mají aktivní roli typu ADMIN nebo MANAGER pro skupinu typu GENERAL. Tato role má také oprávnění provádět akce, které mohou vykonávat Group Admin nebo User.
- **Group Admin** je odpovědný za správu skupin, které mu byly přiděleny. Tato role má možnost provádět akce v role User. V prostředí CZM je Group Admin ekvivalentem mentora a v diagramu tříd je označen jako aktivní role typu ADMIN.
- **User**. Tato role označuje stážistu CZM, v diagramu tříd je znázorněna třídou User.

4.4.2 Scénáře a obrazovky

V následující sekci popíšu scénáře, které splňují funkční požadavky, a u některých scénářů nastíním pomocí obrazovek, jak jednotlivé situace uživatel v systému uvidí.

Pro každý požadavek systém zkontroluje, že má uživatel oprávnění vykonávat akci.

Onboard new user

Admin chce přidat nového stážistu do systému a přidělit mu přístupy:

1. Systém získá seznam všech skupin (Group), jeho systémů (System), požadovaných procesních vstupů (ProcessVariable a ProcessValue) a zobrazí formulář pro přidání uživatele 4.3.
2. Admin zadá uživatelské jméno nového stážisty. Může pro něj zvolit více stážových pozic, projektů. Dále může zvolit přístupy, které chce uživateli přidělit. Formulář potvrdí.
3. Systém zkontroluje pomocí user-service, že zadaný stážista v databázi uživatelů ještě není a jeho uživatelské jméno v rámci ČVUT existuje.

4. Systém spustí proces Nástup nového stážisty 4.5.1 se zadaným uživatelským jménem.

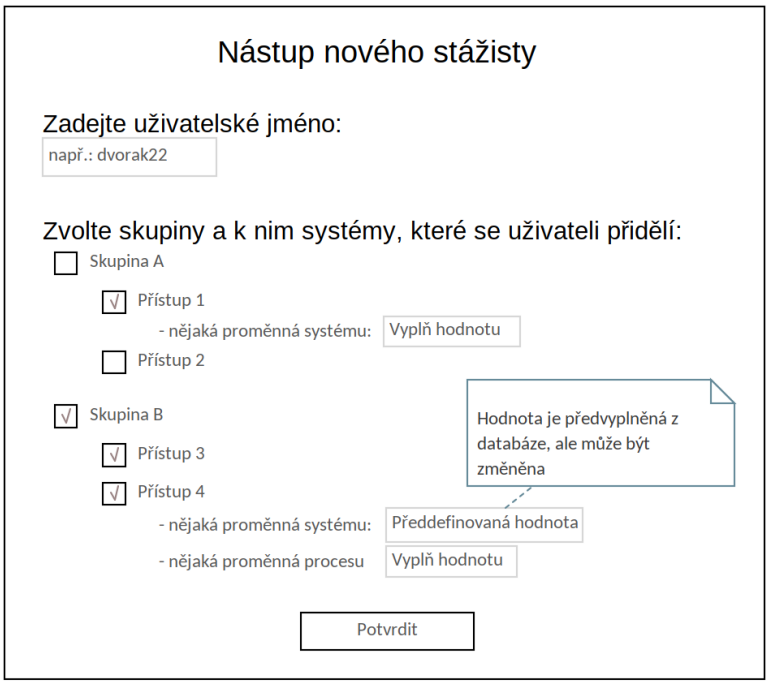
5. Systém vytvoří v databázi nového uživatele (User) a novou roli typu USER v rámci skupiny (Group) GENERAL.

6. Systém dále pokračuje scénářem pro přidělení přístupu pro zvolené skupiny 4.4.2.

Tento scénář podporuje funkční požadavky 5 a 6.

Změny oproti stávajícímu procesu nástupu nového stážisty 2.1.6:

- Administrátor může zvolit, zda se pro nového stážistu vytvoří úkol na vyplnění údajů pro přístup k CZM síti.
- Po vyplnění údajů pro přístup k CZM síti je automaticky vytvořené GitLab issue s požadavkem na přidělení přístupu.
- Po vyplnění osobních údajů je automaticky odesláno upozornění vedoucímu CZM s požadavkem na zařízení CZM role v rámci ČVUT.



Obrázek 4.3: Formulář pro přidání stážisty

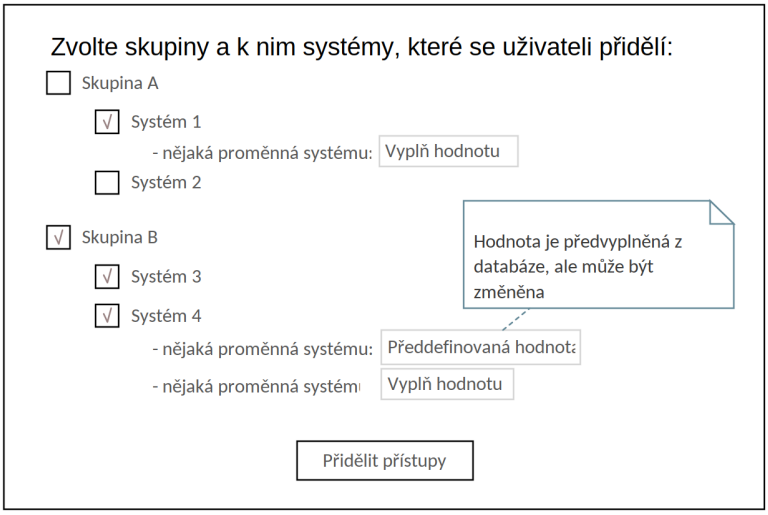
Grant access for user

Admin chce přidělit přístupy stážistovi:

1. Systém zobrazí formulář 4.4 se seznamem všech skupin a k nim jednotlivé spravované systémy a požadované proměnné pro spuštění procesu správy přístupu.
2. Admin formulář vyplní.
3. Systém pro každý vybraný systém provede následující:
   a. Systém vytvoří novou Group Role pro stážistu.
   b. Systém pro vybraný systém skupiny spustí proces (procesy popsány v sekci 4.5, který má id `accessProcessDefinitionKey`. Pokud proces běží, systém uloží do databáze záznam o přidělení přístupu (UserAccess), který se váže na zadaný systém (System) a skupinu (Group). Dále pro přístup nastaví datum přidělení a uživatele, co přístup přidělil.

Může se stát, že uživateli byl přidělen přístup mimo systém, tuto situaci řeší každý proces ze sekce 4.5 zvlášť. Pokud systém dokáže zjistit, že byl přístup přidělen, ale neexistuje proces, který by správu řídil, tak se vytvoří nový proces, aby mohl být přístup spravován.

Tento scénář podporuje funkční požadavek 1.



Obrázek 4.4: Formulář pro přidělení přístupů

Remove access for user

Admin chce odebrat přístup stážistovi:

1. Systém nastaví vybranému přístupu (UserAccess), kdy byl přístup odebrán a kým.
2. Systém předá procesu, který se váže na odebíraný přístup (identifikátor procesu je hodnota atributu `processInstanceId`), že má pokračovat v odebrání přístupu.

Tento scénář podporuje funkční požadavek 3.

Change GitLab role

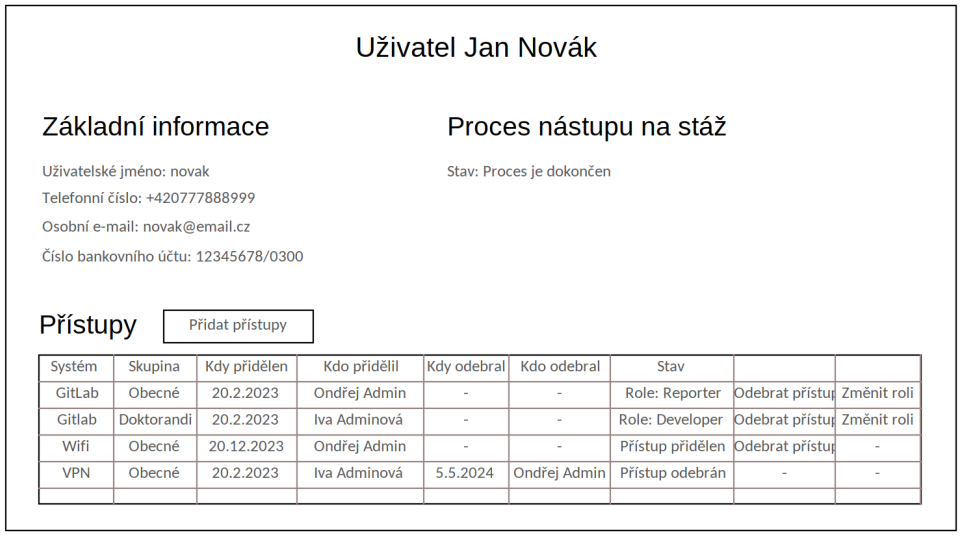
Admin chce změnit u přístupu ke GitLab roli:

1. Systém adminovi zobarzí dialogové okno se seznamem GitLab rolí.
2. Admin zvolí roli a potvrdí.
3. Systém získá procesní instanci na základě vybraného přístupu (UserAccess) a předá jí zprávu s požadavkem na změnu role. Změna role je detailněji popsána v procesu 4.5.2

Tento scénář podporuje funkční požadavek 2.

User detail

Admin si může zobrazit uživatelovy základní údaje a seznam jeho přístupů, jak je vidět na obrazovce 4.5.



Obrázek 4.5: Detail uživatele

Fill MAC data

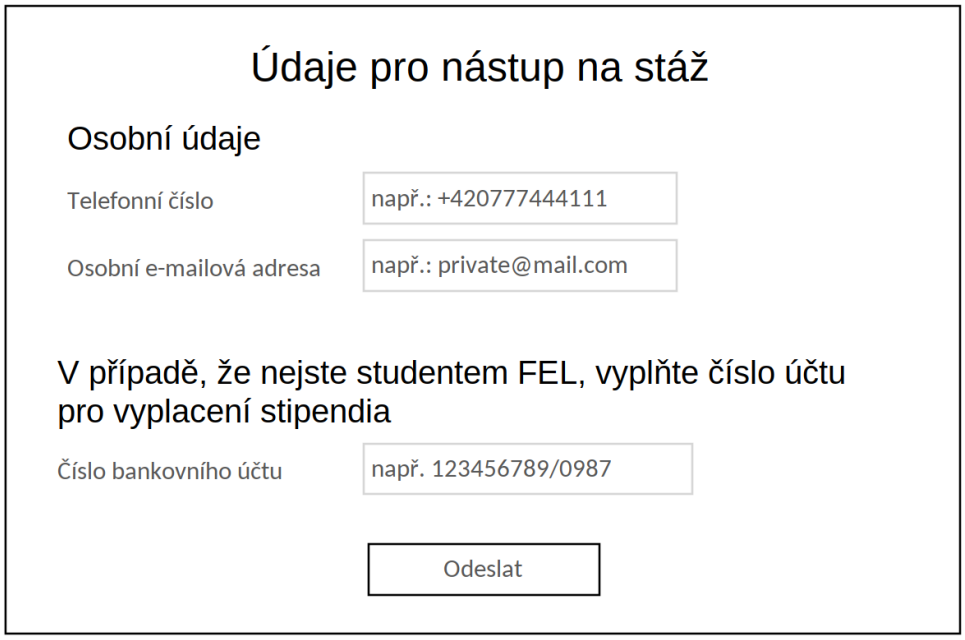
Uživatel chce vyplnit formulář pro přístup k CZM síti:

1. Systém zobrazí formulář pro přístup k CZM síti 4.6.
2. Uživatel formulář vyplní.
3. Systém vyplněná data předá procesům, které na ně čekají. Proces je detailně popsán v sekci 4.5.5

Obrázek 4.6: Formulář pro přístup k VPN a Wi-Fi

Fill personal data

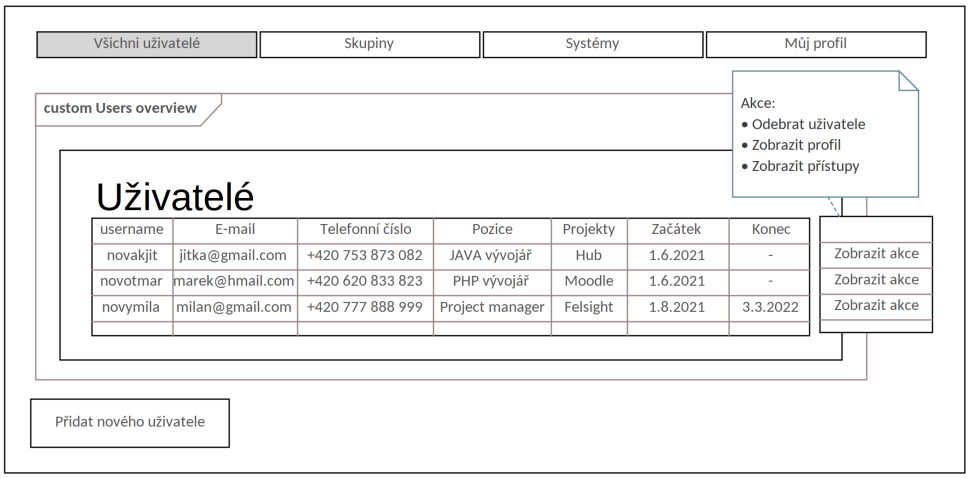
Tento formulář nový stážista vyplňuje v rámci procesu Nástupu nového stážisty 4.5.1



Obrázek 4.7: Formulář pro nového stážistu

4.4.3 View users

Všichni uživatelé si mohou zobrazit všechny uživatele systému, jak je zobrazeno na obrázku 4.8. Pouze uživatelé s rolí Admin mohou zobrazit detail uživatele.



Obrázek 4.8: Seznam uživatelů

4.4.4 Scénáře týkající se skupiny

Pokud je uživatel správcem nějakých skupin, může si je prohlédnout v tabulce s názvy skupin a zobrazit detailní informace o skupinách včetně jejich členů. Přidávat a odebírat role skupiny může pouze Admin či Group Admin skupiny. Tyto funkcionality podporují funkční požadavek 7.

4.5 Procesy systému

Již bylo zmíněno v možnostech optimalizace 3.2, že CZM využívá technologii Camunda [6] pro správu procesů. Rozhodla jsem se tedy procesy, které podporují správu přístupů v CZM, definovat v této technologii. Tím se splní nefunkční požadavek 1 ohledně využití již používaných technologií v CZM. Camunda nabízí pro modelování procesů Camunda Modeler.

V případě, že proces vyhodí výjimku, proces se navrátí do kroku, odkud začíná transakce. Takže pokud transakce započala od začátku procesu, proces se neuloží do Camunda databáze.

Když systém získává uživatelské přístupy pro frontend, ověří pro všechny procesy daného uživatele, zda některý z nich čeká na zprávu `STATE_CHECK`.

Pokud proces čeká na tuto zprávu, v dalším kroku aktualizuje stav přes servisní úkol.

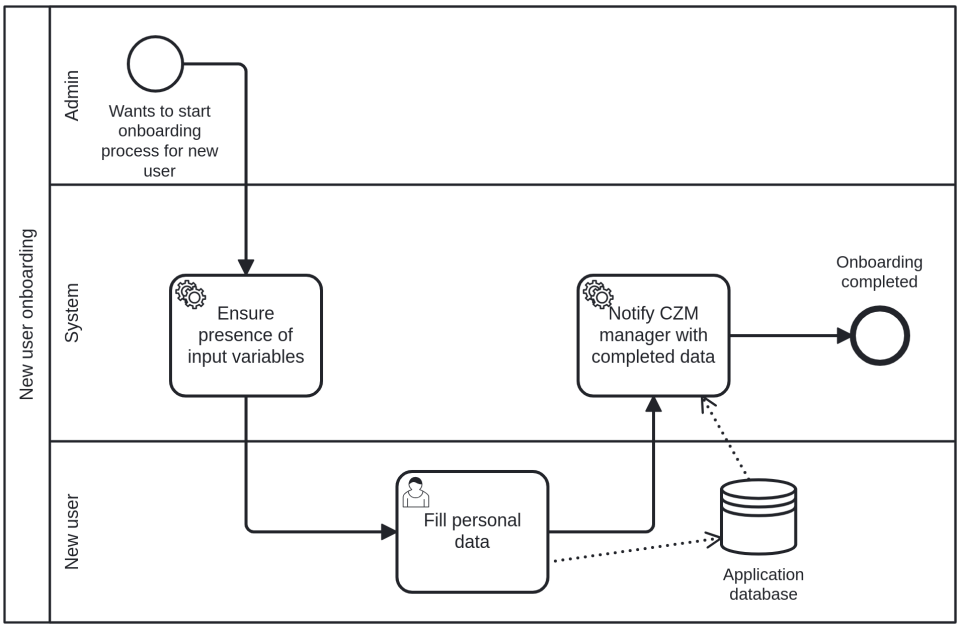
4.5.1 Proces Nástup nového stážisty

Proces Nástupu nového stážisty zobrazený na obrázku 4.9 je spuštěn ze scénáře 4.4.2 a probíhá následovně:

1. Proces zkontroluje, že jsou přítomné procesní proměnné accessFor, která vyjadřuje username uživatele, který nastupuje. Dále je nezbytná proměnná `onboardingUser`, která drží uživatelské jméno uživatele zahajujícího proces nástupu.

2. Proces zašle pomocí notifikační služby 5.1.1 upozornění na formulář s osobními údaji k vyplnění uživateli s uživatelským jménem `accessFor` a vytvoří pro něj uživatelský úkol s identifikátorem `FILL_PERSONAL_DATA` a čeká se na dokončení úkolu.

3. Proces zašle pomocí notifikační služby 5.1.1 vedoucímu CZM upozornění, aby zajistil novému uživateli roli CZM v rámci ČVUT. Pokud uživatel vyplnil bankovní údaje, jsou zaslány společně s upozorněním.



Obrázek 4.9: Proces Nástup nového stážisty

4.5.2 Správa přístupu ke GitLab

Proces Správa přístupu ke GitLab 4.10 pomocí endpointů poskytovaných GitLab API 6.1.1 automaticky přidělí přístup k určité skupině či projektu v GitLab s určitou rolí a následně na základě typu požadavku přístup odebere, změní úroveň přístupu, či aktualizuje jeho stav, neboť přístup je možné změnit i mimo GitLab API. Pokud je již spuštěný jeden proces, který spravuje přístup uživatele k nějaké GitLab skupině nebo projektu, není možné mít spuštěný další takový a změny v přístupu provádí již existující proces.

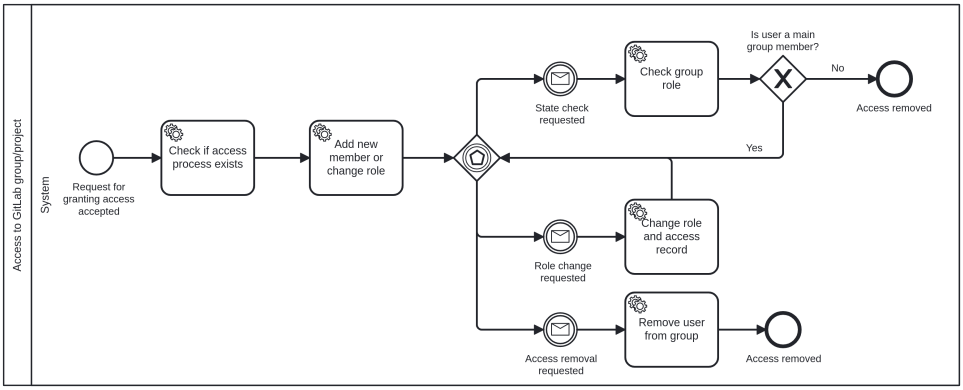
Popis procesu

Proměnná `isGroup`, která je zadána na začátku procesu, určuje, zda bude proces manipulovat s endpointy spravujícími skupinu nebo projekt.

1. **Check if access process exists:** Proces zkontroluje, zda na vstupu přišly proměnné `roleNumber`, `groupOrProjectId`, `isGroup`. Jinak vyhodí výjimku. Pokud již existuje běžící proces, který se týká stejného uživatele `accessFor` a zároveň projektu nebo skupiny `groupOrProjectId`, tak proces vyhodí výjimku. protože je proces od začátku v transakci, proces nebude uložen do Camunda databáze.
2. **Add new member or change role:** Proces se pokusí přidat uživatele do dané skupiny/projektu. Pokud je již uživatel ve skupině/projektu přidaný, což proces odhalí odchycením výjimky `FeignException.Conflict`, proces změní roli daného uživatele pro danou skupinu/projekt. V případě že uživatel danou roli již má, proces vyhodí výjimku. Pokud nebyla do dokončení tohoto kroku odchycena nějaká výjimka, proces není uložen do databáze.
3. Dále proces čeká na jednu z událostí:
   - **Check group role:** Proces získá stav členství pro danou skupinu/projekt. Pokud uživatel je členem, proces nastaví aktuální úroveň přístupu do procesní proměnné `accessState`. Pokud není členem, proces získá členství hlavní skupiny. V případě že není její členem, proces nastaví stav přístupu na odebraný a proces skončí. V opačném případě proces uloží do accessState úroveň přístupu hlavní skupiny a proces se vrací do kroku 3.
   - **Change role and access record:** Proces se pokusí změnit úroveň přístupu, nastaví kdo a kdy přístup přidělil a vrací se do kroku 3.

Tyto operace jsou v transakci spravované Springboot frameworkem a pokud se vyskytne výjimka, operace se neprovedou.

- **Remove user from group:** Proces odebere uživatele z dané skupiny/projektu a nastaví stav přístupu na odebraný. Pokud se vyskytne nějaká výjimka, proces se vrací do kroku 3.



Obrázek 4.10: Proces Přidělení a odebrání přístupu pomocí GitLab issue

4.5.3 Vytvoření GitLab issue

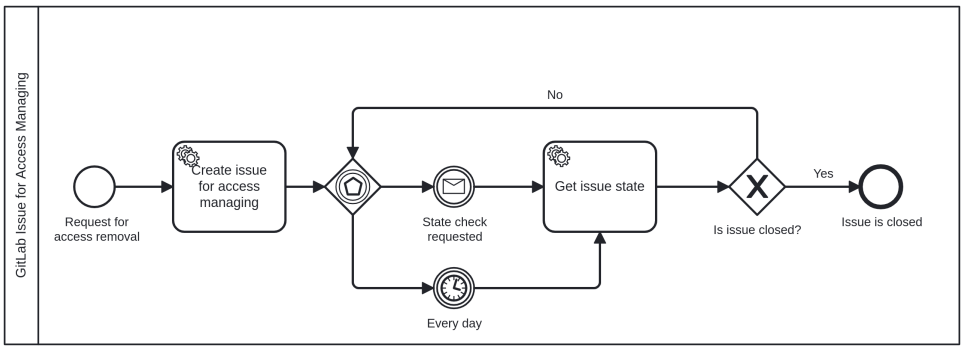
Tento proces se používá, když je potřeba odebrat nebo přidělit přístup na základě požadavku sepsaného v GitLab issue. K úspěšnému provedení procesu jsou nutné následující vstupní proměnné:

- `type` určuje, zda bude issue vytvořeno pro přidělení či odebrání přístupu.
- `issueTemplateName` je název šablony issue, která se použije pro vytvoření issue.
- `projectIdForIssue` je identifikátor projektu, ve kterém se hledá šablona issue a následně se v něm vytvoří issue.

Dále může přijít proměnná `additionalIssueContent`, která vyjadřuje dodatečný obsah, který je přidán nakonec obsahu issue.

Průběh procesu

1. **Create issue for granting access:** Proces pomocí metody `getProjectIssueTemplate` 6.1.3 načte obsah šablony issue pro projekt s identifikátorem `projectIdForIssue`. Název šablony issue je definovaný hodnotou `issueTemplateName`. Poté vytvoří nové issue pomocí metody `createIssue` 6.1.3 s názvem title a upraveným obsahem.
2. Dále proces čeká na jednu z událostí:
   - **State check requested:** Proces čeká na zprávu typu `STATE_CHECK`. Poté pomocí metody `getProjectIssue` 6.1.3 získá stav issue. Pokud je issue uzavřeno, je přístup považován za přidělený. Jinak se proces vrací do kroku 2.
   - **Every day:** Proces každý den kontroluje, zda je issue uzavřené. Pokud není, vrací se do kroku 2.



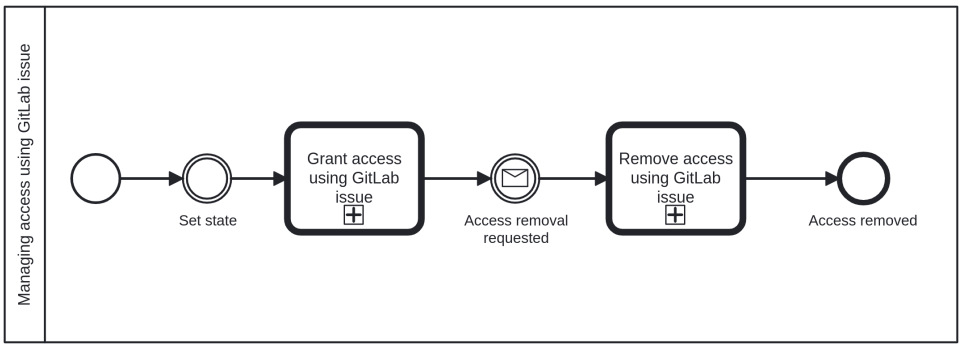
Obrázek 4.11: Proces Vytvoření GitLab issue

4.5.4 Správa přístupu pomocí GitLab issue

Pro ty systémy, ke kterým zodpovědná osoba jak přiřazuje, tak odebírá přístup manuálně, bude možné využít proces 4.12, který vytvoří v GitLab issue požadavek.

- **Set state:** Proces nastaví stav přístupu `accessState`, že se vytváří GitLab issue pro přidělení přístupu.
- **Grant access using GitLab issue:** Proces spustí proces Vytvoření GitLab issue 4.5.3 se vstupem `type`, který má hodnotu `granting` a také mu předá všechny své procesní proměnné.

- **Access removal requested:** Proces čeká na zprávu `ACCESS_REMOVAL_REQUESTED`, kdy je požadováno odebrání přístupu.
- **Remove access using GitLab issue:** Proces spustí proces Vytvoření GitLab issue 4.5.3 se vstupem `type`, který má hodnotu `removal` a předá mu všechny své procesní proměnné.



Obrázek 4.12: Proces Přidělení a odebrání přístupu pomocí GitLab issue

4.5.5 Správa přístupu k CZM síti

Proces pro správu přístupu k CZM síti 4.13 slouží k získání uživatelovy MAC adresy a názvu zařízení, které chce mít přístup k CZM síti. Následně je s těmito daty vytvořeno GitLab issue, ve kterém je požadováno přidělení přístupu pro dané zařízení. Eventuálně je vytvořeno issue pro odebrání přístupu. Tento proces se využívá pro získání CZM Wi-Fi anebo VPN.

Pro úspěšné provedení procesu jsou nezbytné následující vstupní proměnné `issueTemplateName` a `projectIdForIssue`, které jsou použity pro spuštění procesu Vytvoření GitLab issue 4.5.3.

Průběh procesu

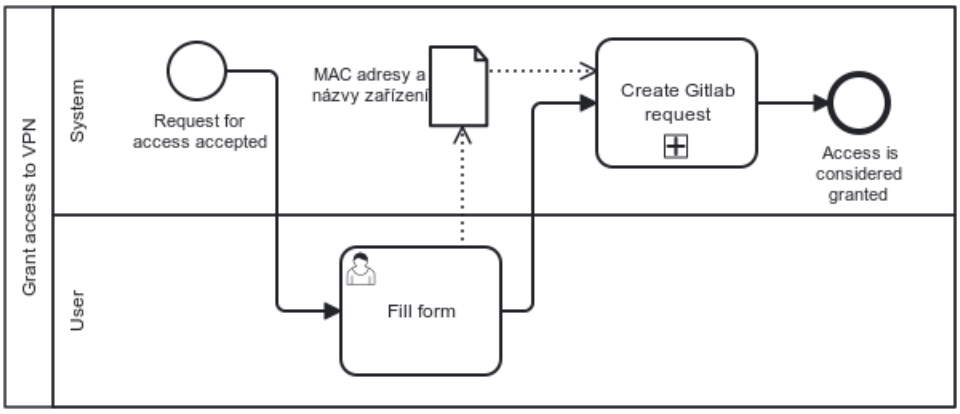
1. **Find task for filling MAC addresses:** Proces zjistí, zda existuje uživatelský úkol pro žádajícího uživatele `accessFor` s identifikátorem `FILL_MAC_DATA`.
2. **Wait for MAC data:** Pokud takový úkol existuje, proces čeká na zprávu `MAC_DATA_RECEIVED`, se kterou přicházějí MAC data.

3. **Fill form with mac addresses:** Pokud krok 1 vyhodnotil, že takový uživatelský úkol neexistuje, proces odešle pomocí notifikační služby 5.1.1 notifikaci, kde upozorňuje na vyplnění formuláře pro MAC data. Následně proces vytvoří pro žádajícího uživatele uživatelský úkol s identifikátorem `FILL_MAC_DATA`. Jakmile je uživatelský úkol dokončen, proces předá proměnnou `macInputList` procesům čekajícím na zprávu `MAC_DATA_RECEIVED` a mající stejnou hodnotu `accessFor`.

4. **Grant access using GitLab issue:** Proces spustí proces Vytvoření GitLab issue 4.5.3 se vstupy `type`, který má hodnotu `granting`, a s `additionalIssueContent`, který má stejnou hodnotu jako proměnná `macInputList`. Rovněž procesu pro vytvoření issue předá všechny své procesní proměnné.

5. **Access removal requested:** Proces čeká na zprávu `ACCESS_REMOVAL_REQUESTED`, kdy je požadováno odebrání přístupu.

6. **Remove access using GitLab issue:** Proces spustí proces Vytvoření GitLab issue 4.5.3 se vstupy `type`, který má hodnotu `removal`, a s `additionalIssueContent`, který má stejnou hodnotu jako proměnná `macInputList`. Rovněž procesu pro vytvoření issue předá všechny své procesní proměnné.



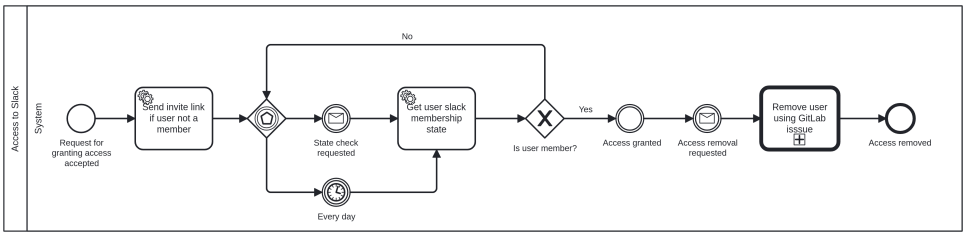
Obrázek 4.13: Proces Správa přístupu k CZM síti

4.5.6 Správa přístupu ke Slack

Proces 4.14 zašle uživateli pozvánku do Slack workspace a na základě požadavku vytvoří GitLab issue pro manuální odebrání přístupu.

Průběh procesu

1. **Send invite link if user not a member:** Proces zkontroluje, zda je uživatel členem Slack pomocí metody `findByEmail` 6.2.1. Pokud není členem, zašle uživateli upozornění s pozvánkou do Slack použitím notifikační služby 5.1.1.
2. Dále proces čeká na jednu z událostí:
   - **State check requested:** Proces čeká na zprávu typu `STATE_CHECK`. Poté pomocí metody `findByEmail` 6.2.1 zjistí, zda je uživatel členem Slack Workspace. Pokud uživatel uživatel členem je, přístup je považován za přidělený. V opačném případě se proces vrací do kroku 2.
   - **Every day:** Proces každý den kontroluje, zda je uživatel členem Slack Workspace. Pokud není, vrací se do kroku 2.
3. **Access granted:** Proces nastaví stav přístupu na `ACCESS_GRANTED`.
4. **Access removal requested:** Proces čeká na zprávu `ACCESS_REMOVAL_REQUESTED`, kdy je požadováno odebrání přístupu.
5. **Remove access using GitLab issue:** Proces spustí proces Vytvoření GitLab issue 4.5.3 se vstupem `type` mající hodnotu `removal` a svými procesními proměnnými.



Obrázek 4.14: Proces Správa přístupu ke Slack

4.6 Výchozí data

V systému pro správu přístupů je třeba zadefinovat, ke kterým externím systémům bude spravován přístup. Tato data nebude možné upravovat uživatelsky, protože při integrace nového externího systému bude třeba doimplementovat jeho integraci, proto bude vždy třeba zásah vývojáře. Z toho důvodu je třeba data do databáze přidat před spuštěním systému.

V databázi jsou 4 skupiny (Group), které představují reálné projekty/pozice a k nim náleží systémy (System), ke kterým může správce skupiny spravovat přístup. Pro některé systémy skupiny bylo ještě třeba definovat procesní proměnné (ProcessVariable), aby byl správně spuštěn proces správy přístupu. Jedná se o tyto skupiny a jejich systémy:

- Obecná skupina: tato skupina obsahuje systémy/služby, ke kterým potřebují přístup všichni stážisté:
  - VPN: přístup spravuje proces 4.5.5
  - Wi-Fi: přístup spravuje proces 4.5.5
  - Slack: přístup spravuje proces 4.5.6
  - GitLab skupina: přístup spravuje proces 4.5.2
- projekt HUB
  - GitLab skupina: přístup spravuje proces 4.5.2
- projekt Moodle
  - GitLab skupina: Přístup spravuje proces 4.5.2
  - Moodle systémy: Přístup k Moodle systémům spravuje proces 4.5.4
- pozice PR
  - GitLab projekt: Přístup spravuje proces 4.5.2

V této kapitole jsem se věnovala podrobnému technickému návrhu systému, který zjednodušuje a automatizuje správu přístupů v CZM dle analýz z předchozích kapitol 2 a 3.

Kapitola 5

Integrace do platformy HUB.FEL

Tato kapitola se zabývá technickým popisem platformy HUB.FEL a integrací Systému pro správu přístupů v CZM do platformy. Kapitola čerpá mimo jiné z diplomové práce *Integrační platforma FEL Hub* [12].

Detailní popis a pochopení toho, jak platforma funguje a co nabízí, byl nezbytný ke zvážení možností, jestli a jak systém pro správu přístupů do platformy integrovat. Bylo třeba pochopit zejména technologické detaily toho, jak je platforma postavená a s jakými částmi platformy by mohla být nově vznikající aplikace spojená.

5.1 Popis platformy HUB.FEL

Platforma HUB.FEL (někdy označovaná i jako FEL Hub) představuje mikroservisní prostředí vyvíjené Centrem znalostního managementu. Součástí platformy HUB.FEL je velká část CZM projektů, například Evaluace zaměstnanců nebo Hodnocení doktorandů. Platforma HUB.FEL je na frontendové části implementována v React.js [13]. Backend platformy je tvořen z Java/Kotlin microservices, které spolu komunikují výhradně přes rozhraní REST na úrovni meziservisní komunikace a pro vystavování dat frontendu výhradně přes rozhraní GraphQL.

Původní architektura platformy HUB.FEL obsahovala jako vstupní bod

backendu službu tzv. endpoint-service, ve které vývojáři manuálně mapovali GraphQL queries a mutace na REST volání, která byla dále směřována na služby dalších projektů. Vývoj tohoto návrhu se ukázal jako neefektivní; při jakékoli změně REST rozhraní nějaké služby bylo třeba provádět i úpravy na endpoint-service, což je o to komplikovanější, že do ní zasahovali vývojáři více týmů.

Platforma HUB.FEL tedy přešla na novou architekturu, kdy je endpoint-service nahrazována GraphQL gateway, v jejímž jádru používá knihovnu @apollo/gateway [14]. Gateway dynamicky registruje backendové služby vystavující GraphQL, které se přes service discovery eureka server [15] připojují a při změně GraphQL schématu není třeba manuálního zásahu do gatewaye.

5.1.1 Notifikační služba

Pro sjednocení notifikačního systému napříč celou platformou existuje centrální notifikační služba. Ostatní služby HUB.FEL mohou přes REST API zasílat požadavky na odeslání notifikací této službě, která následně zajistí jejich správné zobrazení na frontendu a případně odeslání upozornění uživateli emailem.

5.1.2 User-service

Na platformě HUB.FEL existuje také mikroslužba poskytující jednotný přístup do registru uživatelů, který je na FEL využíván a zjednodušuje tak všem projektům na platformě práci s daty o studentech a akademických pracovnících.

5.1.3 Kos-service

HUB.FEL obsahuje také mikroslužbu poskytující API rozhraní nad daty o studiu, což je v projektu systému pro správu přístupů třeba při získávání studijních dat nového stážisty.

5.1.4 LibCommon

Služba *czm-management-service* je postavená na CZM knihovně LibCommon, která pro službu zajišťuje základní funkcionalitu v rámci platformy:

1. **Monitoring, tracing, logging.** Pro monitorovací nástroje vystavuje endpoint s daty o stavu služby, aktuálním využití paměti a dalších parametrech potřebných pro dlouhodobý provoz aplikace. Upravuje jednotný loggovací formát pro snadnější zobrazení v Kibaně [16] a odesílá tracing data, pro snadnější hledání provozních chyb napříč celou microservisní architekturou.
2. **Service discovery.** Připojení na eureka server, skrz který služba získává dynamické IP adresy pro volání na ostatní mikroslužby, zároveň v metadatech připojení k eureka serveru služba indikuje svou žádost o dynamické připojení do graphQL gatewaye 5.1.
3. **Ověření uživatele.** Na platformě HUB.FEL je používán OAuth JWT, které jsou ověřovány knihovnou LibCommon. V případě volání na další služby pak LibCommon poskytuje konfiguraci pro správné přeposlání autorizační a impersonační hlavičky, aby byl požadavek správně validován i v dalších službách.
4. **Impersonace.** LibCommon poskytuje funkce pro snadnou detekci aktuálně přihlášeného uživatele, která bere v úvahu i impersonační žádost, jejíž validitu ověří.

5.1.5 Nasazení služby do HUB.FEL

V neposlední řadě je pro produkčně funkční aplikaci třeba, aby byla provozována na stabilním prostředí odděleném od prostředí na vývoj a testování pro zajištění co největší stability a testovatelnosti.

Platforma HUB.FEL využívá pro všechny projekty tři prostředí:

1. **Vývojové prostředí** používané na vývoj, je možné se k němu připojovat tzv. hybridním vývojem, pokud vývojář potřebuje svou lokálně vyvíjenou službu ladit spolu s dalšími částmi platformy.
2. **Testovací prostředí** je určeno především k internímu testování aplikace.

3. **Produkční prostředí** je poskytováno uživatelům.

Pro nasazení na tato prostředí služba *czm-management* používá Gitlab CI/CD [17]. Nasazovací pipeline službu zabalí do Docker image, který je nasazen do Docker Swarm [18], který je na platformě používán.

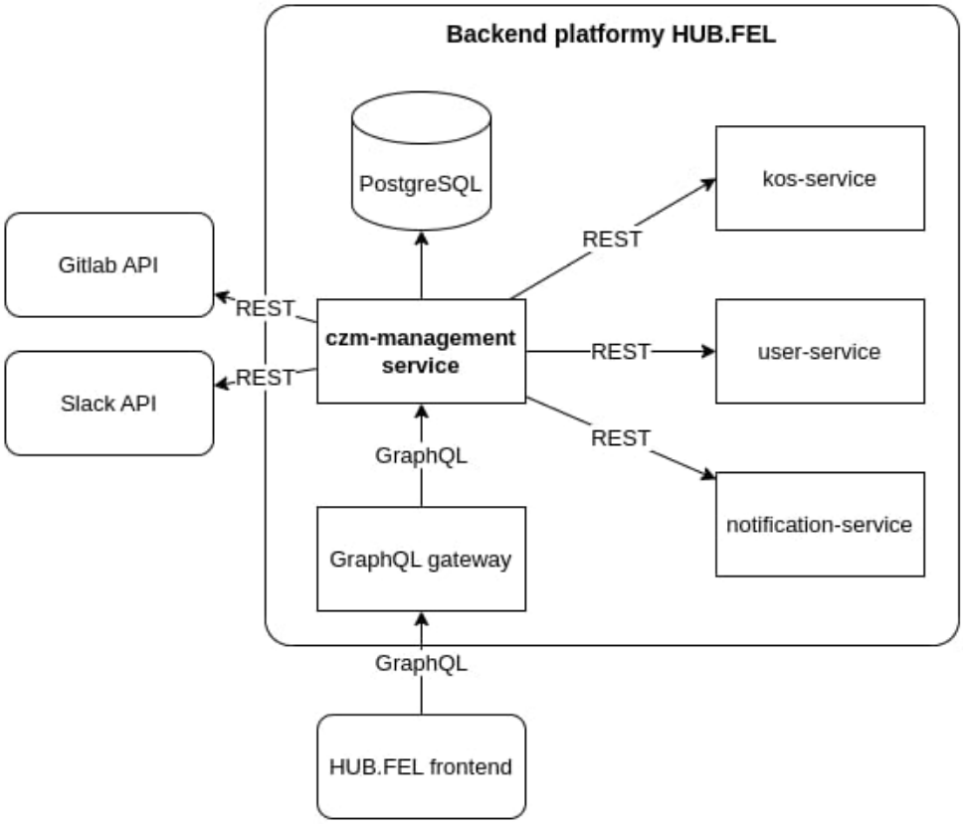
5.2 Důvody pro integraci do platformy HUB.FEL

Vzhledem k povaze platformy představené v 5.1 a nefunkčnímu požadavku 4.2 na snadnější udržitelnost a údržbu v rámci CZM dává smysl systém pro správu přístupů implementovat do platformy HUB.FEL. Dalším argumentem pro začlenění do platformy jsou i standardy a funkcionality, které platforma sama o sobě nabízí a které usnadní vývoj, popsané v 5.1. Systém pro správu přístupů bude dále provozován a udržován stážisty v CZM, kteří vyvíjejí další části platformy HUB.FEL popsané v 2.1.3, technologické know-how se tedy bude přenášet mezi projekty.

5.3 Popis integrace služby do platformy

Systém pro správu přístupů je implementován jako samostatná mikroslužba s názvem *czm-management-service*. Tato služba vystavuje GraphQL rozhraní, které se připojuje ke GraphQL gateway a je poskytováno frontendové části. Frontend je vytvořen v React.js, což umožňuje využití existujících komponent a dosažení jednotného designu s ostatními částmi platformy, které jsou rovněž postaveny na React.js, viz schéma integrace 5.1.

Služba *czm-management-service* je implementována v Javě pomocí frameworku Spring Boot [19] s využitím buildovacího nástroje Maven [20]. Data jsou ukládána do PostgreSQL databáze [21]. Tyto technologie jsou rovněž využívány v dalších službách integrovaných do platformy.



Obrázek 5.1: Schéma integrace služby do platformy HUB.FEL

Závěrem tedy lze říct, že po zvážení technologických možností platformy HUB.FEL se integrace do ní jeví jako rozumná a výhodná volba pro implementaci i dlouhodobou údržbu v CZM.

Kapitola 6

Využití externích API

Tato kapitola se zaměřuje na integraci a využití externích API v aplikaci, konkrétně GitLab API a Slack API. Pro komunikaci s těmito API jsem použila OpenFeign klienta, který nabízí jednodušší implementaci ve srovnání s WebClientem. [22]

6.1 Integrace GitLab API

GitLab API je pro aplikaci využíváno pro správu uživatelů, skupin, projektů a issues.

6.1.1 Operace pro správu členů skupiny a projektu

Při testování procesu přidávání uživatelů do skupin jsem zaznamenala situaci, kdy po přidání uživatele do hlavní skupiny byla jeho role automaticky propagována do všech podskupin a podprojektů a pro jakoukoli podskupinu/podprojekt není možné roli pouze změnit, je nutné uživatele do podskupiny/podprojektu explicitně přidat s požadovanou rolí a následně je možné úroveň přístupu měnit. Proto při přidávání uživatele do skupiny/projektu se nejprve pokusím uživatele přidat, pokud je uživatel již členem, změním mu jeho roli.

Pro správu členů skupiny a projektu byly implementovány následující operace:

- `addUserToGroup(groupId, userId, accessLevel)`: Metoda pro přidání uživatele do skupiny s určeným přístupovým úrovněm.
- `addUserToProject(projectId, userId, accessLevel)`: Metoda pro přidání uživatele do projektu s určeným přístupovým úrovněm.
- `editGroupMember(groupId, userId, accessLevel)`: Metoda pro úpravu úrovně přístupu uživatele ve skupině.
- `editProjectMember(projectId, userId, accessLevel)`: Metoda pro úpravu úrovně přístupu uživatele v projektu.
- `removeUserFromGroup(groupId, userId)`: Metoda pro odstranění uživatele ze skupiny.
- `removeUserFromProject(projectId, userId)`: Metoda pro odstranění uživatele z projektu.
- `getGroupMemberByUserId(groupId, userId)`: Metoda pro získání člena skupiny podle ID uživatele.
- `getProjectMemberByUserId(projectId, userId)`: Metoda pro získání člena projektu podle ID uživatele.

6.1.2 Operace pro získání informací o uživatelích a šablonách issue

Pro správnou identifikaci uživatelů v GitLab API je nezbytné získat jejich uživatelské ID. Kvůli integraci s ČVUT Single Sign-On (SSO) identitou, která je používána k přihlašování uživatelů do GitLabu, je třeba využít uživatelského jména k získání odpovídajícího uživatelského ID pomocí metody:

- `getUsersByUsername(username)`: Metoda pro získání seznamu uživatelů podle uživatelského jména.

6.1.3 Operace pro správu issue

- `getProjectIssueTemplate(projectId, issueTemplateName)`: Metoda pro získání šablony issue na základě názvu šablony pro daný projekt.
- `createIssue(projectId, title, description)`: Metoda pro vytvoření nového issue pro daný projekt s daným nadpisem a popisem.
- `getProjectIssue(projectId, issueIid)`: Metoda pro získání detailů issue v projektu.

6.2 Integrace Slack API

Slack API je zde využito pro zjištění stavu členství uživatele.

6.2.1 Operace pro vyhledávání uživatele podle e-mailu

Pro vyhledávání uživatele podle e-mailové adresy byla implementována následující operace:

- `findByEmail(email)`: Metoda pro vyhledání uživatele podle e-mailové adresy.

6.3 Testování integrací

Pro účely testování jsem vytvořila testovací prostředí jak pro GitLab, tak pro Slack. Testovací GitLab skupina napodobuje produkční prostředí, aby nedocházelo k narušení reálných dat. Stejně tak je vytvořen testovací workspace pro Slack, aby se zabránilo vlivu testování na produkční prostředí.

Kapitola 7

Uživatelské rozhraní systému

Uživatelské rozhraní je implementováno v React.js, jak již bylo zmíněno v kapitole 5.1 s využitím existujících komponent, které využívají i další služby integrované do HUB.FEL.

7.1 Přehled uživatelů systému a přidělování přístupu

Výsledný přehled uživatelů lze vidět na obrázku 7.1. Odsud může administrátor kliknout na tlačítko pro přemístění se na formulář pro nástup nového stážisty 7.2.

Další obrazovkou, na kterou může administrátor z přehledu uživatelů přejít, je detail uživatele 7.3. Odsud může administrátor přejít na formulář pro přidělení přístupů 7.4 uživateli.

7.2 Formuláře

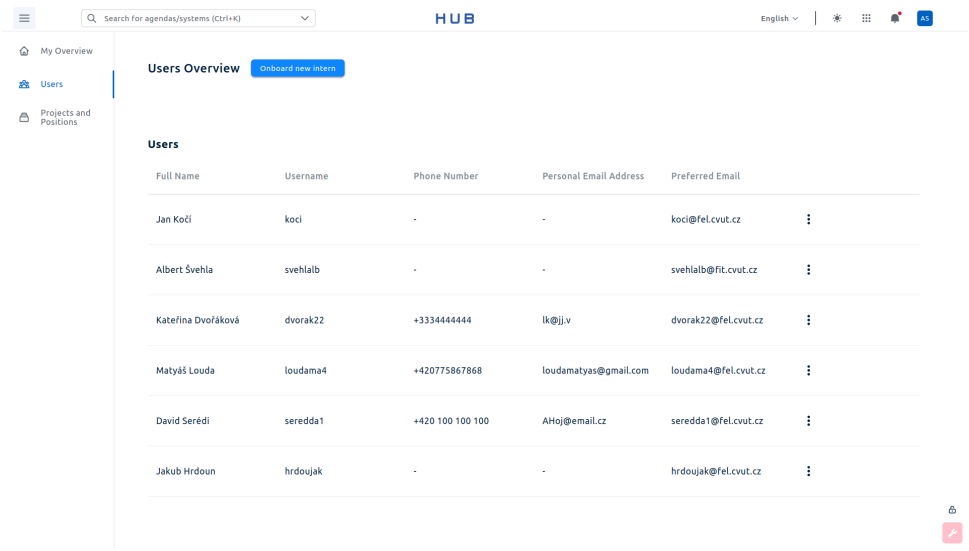
V případě, že se uživateli systém odešle notifikace přes notification-service 5.1.1, zobrazí se mu v přehledu notifikací 7.5, který je spravován platformou

HUB.FEL 5.1. Zmíněný seznam notifikací ukazuje ta upozornění, která odesílá systém pro správu přístupů.

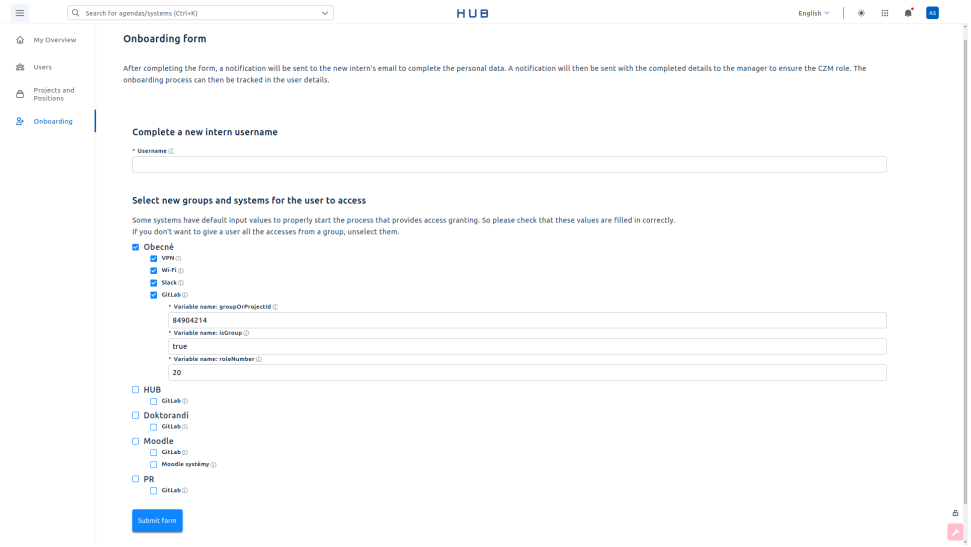
Po zobrazení notifikace ohledně nástupu na stáž se zobrazí formulář pro vyplnění osobních údajů 7.6, který dokončí úkol v procesu nástupu nového stážisty 4.5.1. Notifikace o vyplnění údajů pro přístup k CZM síti uživatele přesměruje na formulář 7.7. Po odeslání údajů je dokončen uživatelský krok `FILL_MAC_DATA` v tomto procesu 4.5.5. Notifikace s pozvánkou do Slack uživateli zobrazí Slack webovou stránku, přes kterou se může zaregistrovat do CZM Slack workspace.

7.3 Správa skupin

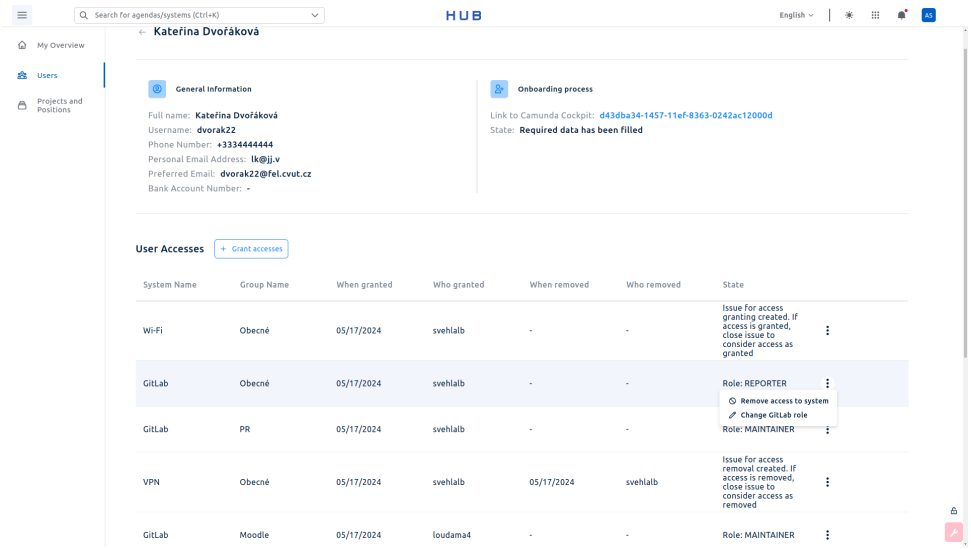
Přehled skupin spravovaných uživatelem je vidět na obrazovce 7.8. Z tohoto přehledu správce skupiny může jít na detail skupiny 7.9, kde může přidat pomocí dialogového okna 7.10 novou roli skupiny nebo odebrat roli skupiny. Poté správce skupiny může jít na detail role 7.11 a vidět přístupy role pouze pro danou skupinu. Z tohoto detailu správce skupiny může přejít na formulář pro přidělení přístupů k systémům dané skupiny 7.12 pro danou roli.



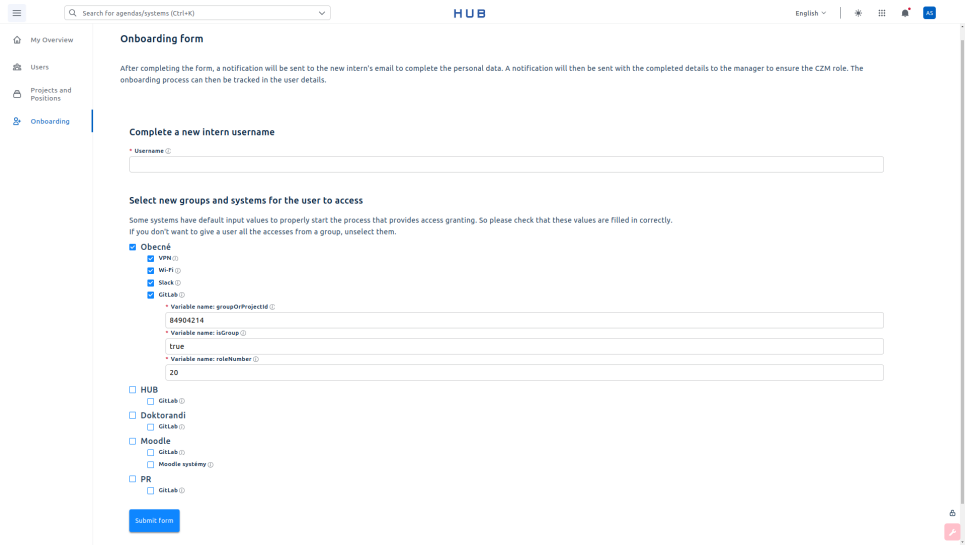
Obrázek 7.1: Přehled uživatelů



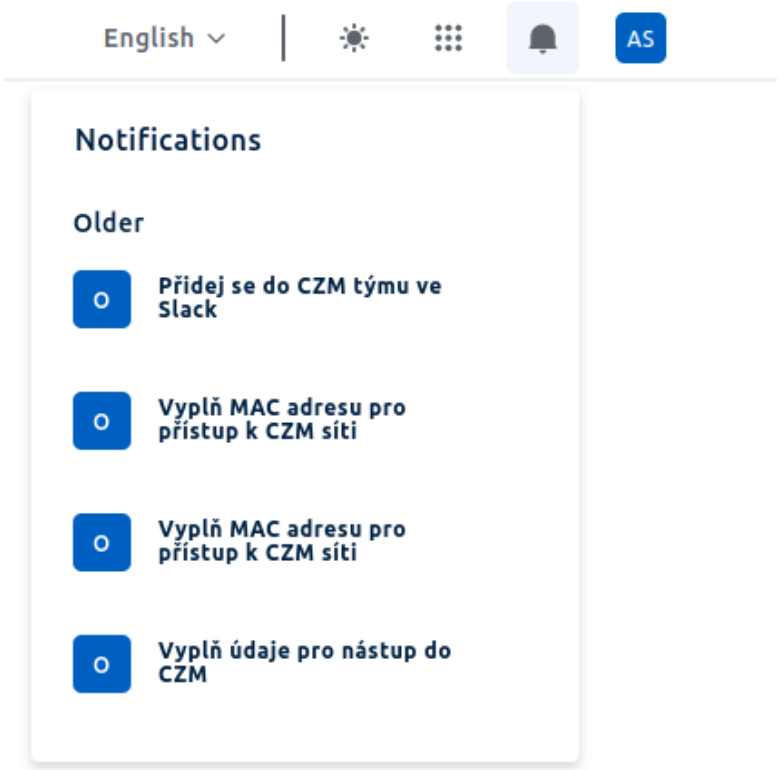
Obrázek 7.2: Formulář pro nástup nového stážisty



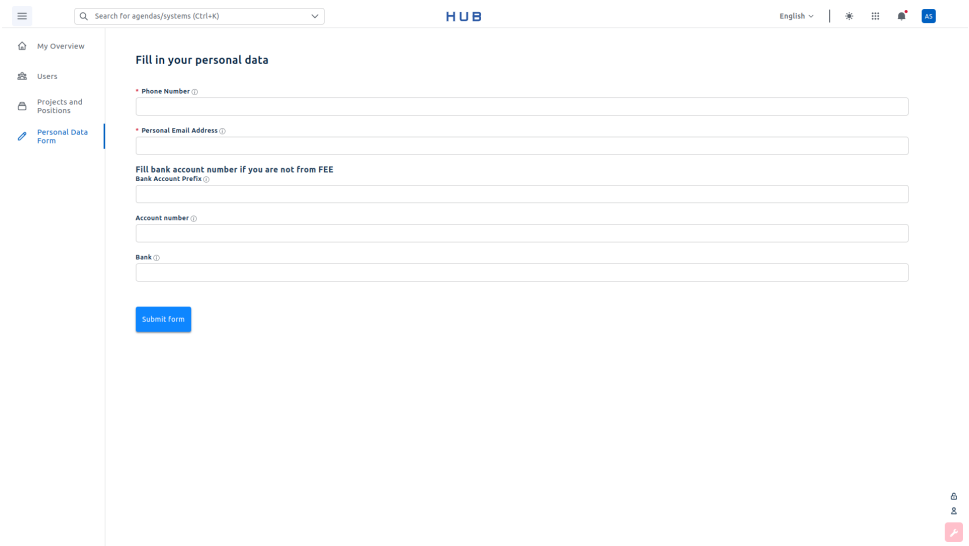
Obrázek 7.3: Detail uživatele



Obrázek 7.4: Formulář pro přidělení přístupů

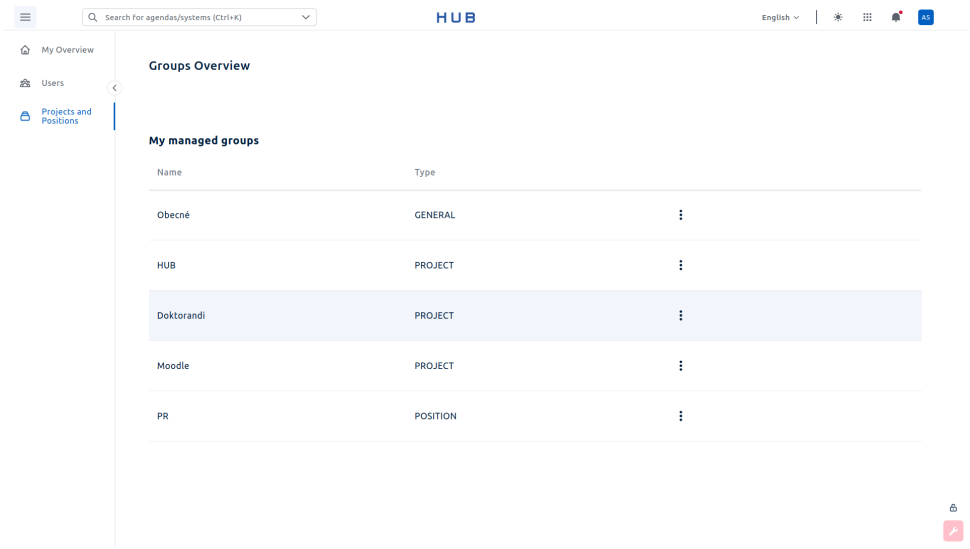


Obrázek 7.5: Seznam notifikací

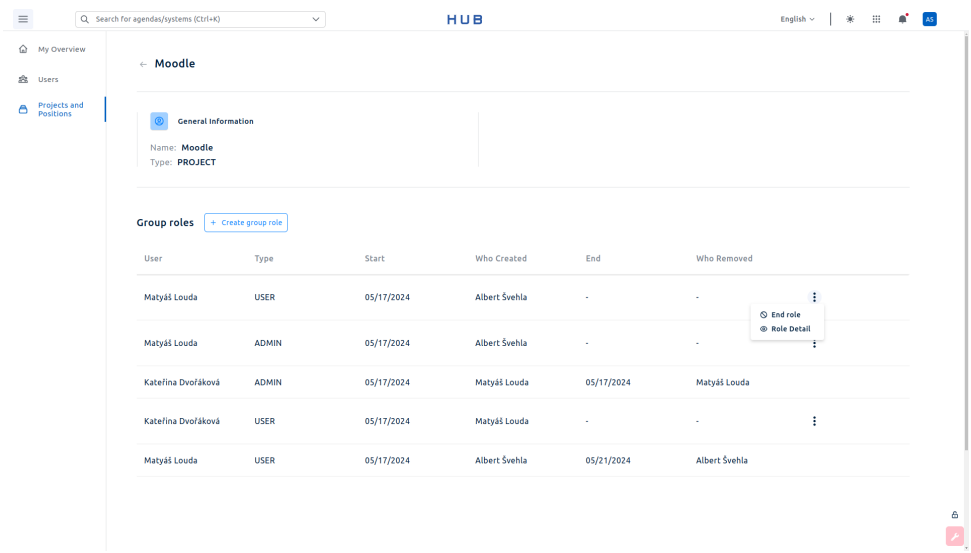


Obrázek 7.6: Formulář pro vyplnění osobních údajů

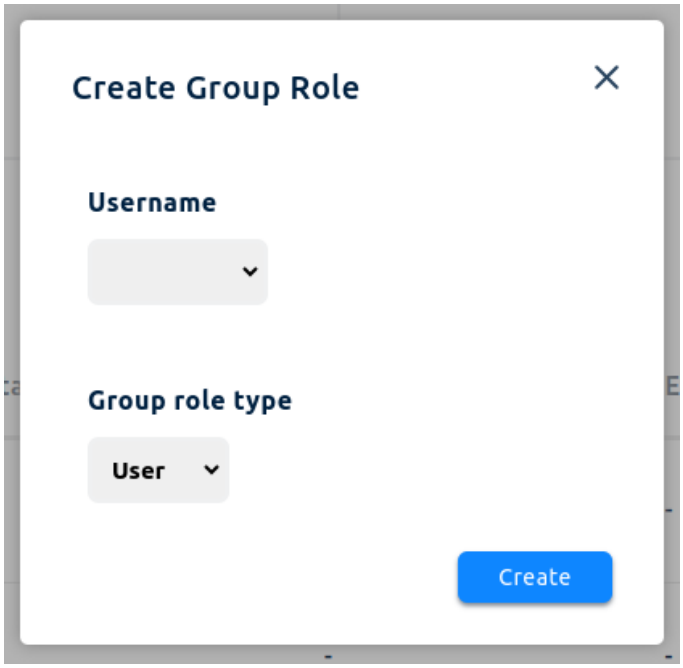
Obrázek 7.7: Formulář pro vyplnění MAC adresy a názvu zařízení



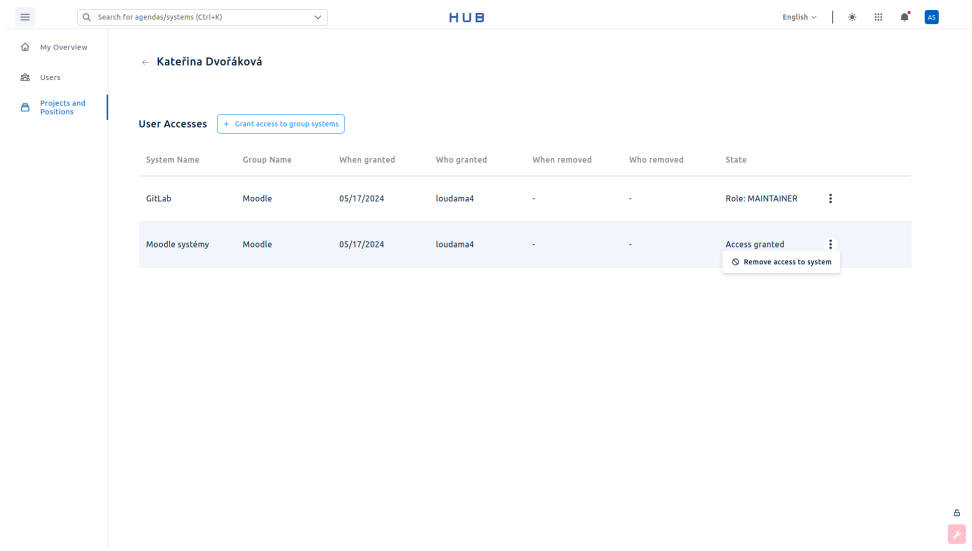
Obrázek 7.8: Přehled skupin



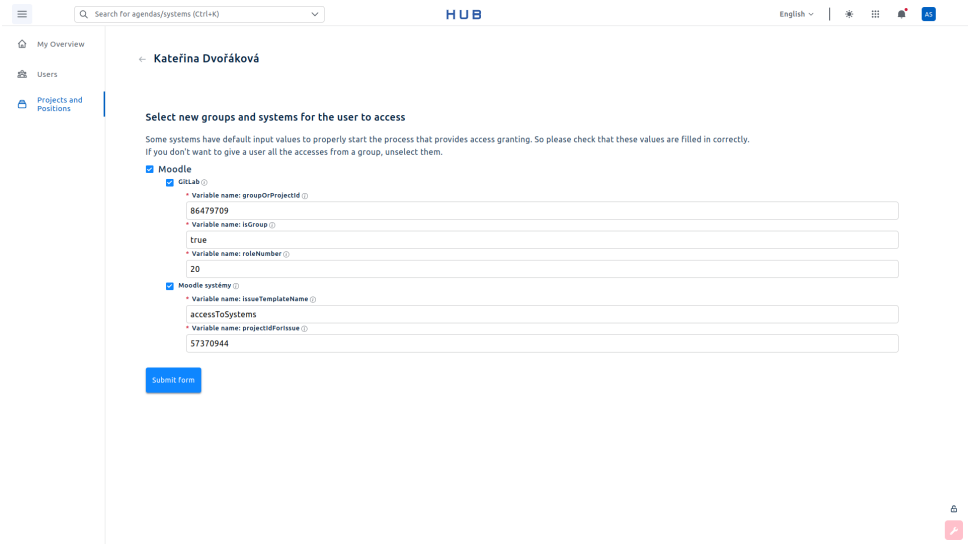
Obrázek 7.9: Detail skupiny



Obrázek 7.10: Vytvoření role pro skupinu



Obrázek 7.11: Detail role skupiny



Obrázek 7.12: Formulář pro přidělení přístupů k systémům skupiny

Kapitola 8

Uživatelské testování systému

Testování probíhalo ve více scénářích, které vycházejí ze sekce 4.4.2. Někteří uživatelé byli v roli administrátora aplikace nebo v roli běžného uživatele bez administrátorských práv. Testování se účastnilo celkem 5 stážistů, mezi kterými byl i HR manažer a mentor. Testování proběhlo s výchozími daty v systému 4.6.

8.1 Příprava testovacího prostředí

Backendová část aplikace byla nasazena ve vývojovém prostředí HUB.FEL, zatímco frontend běžel lokálně na mém zařízení. Do vývojového prostředí jsem se přihlásila jako uživatel s oprávněním impersonace 5.1.4, protože ne všichni testeři mají možnost se přihlásit svým vlastním uživatelským jménem. Díky tomu mohli testeři využívat tohoto uživatele k impersonaci a testování pod různými uživatelskými účty.

8.2 Průběh testování v roli administrátora systému

Během následujícího testování chceme zjistit, zdali správně funguje přidání stážisty administrátorem, který následně stážistů uvidí v přehledu uživatelů 7.1. Stážistovi jsou dále přidány přístupy, které by administrátor měl také

vidět v detailu uživatele 7.3. Při odebrání přístupu chceme vidět, že se změní stav přístupu.

V databázi aplikace je jeden uživatel v roli administrátora Obecné skupiny. Tento uživatel prošel následujícími scénáři:

8.2.1 Scénář 1: Správa stážistů a přístupů

1. Přidání nového stážisty do systému s přidělením různých přístupů.
2. Kontrola stavu přístupů v aplikaci a jejich aktuálního stavu v externích systémech.
3. Kontrola vytvoření issues pro žádost o přidělení přístupu.
4. Odebrání některých přístupů uživateli.
5. Přidělení přístupů existujícímu uživateli.

8.2.2 Scénář 2: Delegování administrátorských povinností

1. Povýšení existujícího uživatele na administrátora konkrétní skupiny.
2. Impersonace za nového administrátora a přidělení přístupů patřících dané skupině uživateli.
3. Odebrání přiděleného přístupu.
4. Opětovné přihlášení jako administrátor Obecné skupiny a odebrání administrátorské role nově povýšenému administrátorovi.

Po každé změně přístupu uživatel kontroloval, zda je stav přístupu stejný jak v aplikaci, tak přímo v externích systémech.

8.3 Průběh testování v roli uživatele systému

Před samotným testováním se uživatel registrovat přes svou ČVUT identitu do gitlab.com, aby se mohla otestovat funkčnost správy přístupu do GitLab. Poté jsem uživatele přidala do formuláře pomocí formuláře pro Nástup nového stážisty 7.2 a přidělila některé přístupy, aby uživatel mohl být v roli nového stážisty. Poté se uživatel řídil tímto scénářem.

1. Podívat se na notifikace v platformě, které obsahovaly tato upozornění 7.5, a řídit se jejich popisem.
2. Podívat se na svůj profil a stav svých přístupů.
3. Zkontrolovat ve Slack a GitLab testovacích prostředí, že stav přístupů odpovídá.

Dále jsem uživatele uvedla do situace, kdy přechází na jiný projekt a přidělila jsem mu nové přístupy. Uživatel poté opět zkontroloval, zda stav v aplikaci odpovídá stavu přímo v systémech.

8.4 Výsledky testování

Po dokončení testování jsem se každého uživatele zeptala na několik otázek týkajících se funkčnosti, uživatelské přívětivosti, návrhů na zlepšení a porovnání s aktuálním stavem.

Uživatelé hodnotili funkčnost systému na škále od 1 do 10 průměrnou hodnotou 8,8. Níže jsou uvedeny nedostatky a návrhy, které byly během testování zjištěny:

- Když byly spuštěny 2 procesy Správa přístupu k CZM síti 4.5.5 pro získání VPN a Wi-Fi, tak se v seznamu notifikací objevily dvě notifikace pro vyplnění formuláře. Tento formulář však stačí vyplnit pouze jednou. Přestože to neovlivnilo funkčnost systému, bylo to pro uživatele matoucí a měla by se zobrazovat pouze jedna notifikace.

- Ve formuláři pro vyplnění MAC adresy nebylo možné do názvu zařízení vložit pomlčku.
- Uživatelům chyběl úvodní návod, jak používat HUB.
- Uživatelé postrádali přehled splněných úkolů.
- Chyběl překlad do češtiny.
- Vysvětlivky byly příliš malé a hůře čitelné.
- Administrátorovi chybělo tlačítko pro odebrání všech přístupu najednou.
- Při přidělování přístupu do GitLab administrátorovi chyběl výběr role pomocí select boxu.

8.5 Závěr testování

Zpětná vazba od testujících uživatelů byla pozitivní. Získala jsem cenné poznatky pro vylepšení systému. Celkově se jednalo hlavně o vizuální nedostatky, které neovlivňovaly funkčnost systému. Tyto nedostatky zatím nebyly opraveny. Aplikace je však plně použitelná.

Kapitola 9

Závěr

V analýze Centra znalostního managementu jsem se zaměřila především na přidělování přístupů v rámci nástupu nového stážisty a také na správu přístupu během celého období stáže. V rámci analýzy byly popsány možnosti optimalizace řízení přístupových práv. Navrhovaná opatření zahrnují systematickou evidenci, automatická přiřazování a odebírání práv a také automatické vytváření požadavků na přidělení či odebrání přístupu. Tyto postupy směřovaly k eliminaci některých manuálních úkonů a zvýšení efektivity správy přístupů. V další fázi práce jsem navrhla uživatelské rozhraní a funkcionality systému, které vycházejí z provedených analýz. Dále byl navržený systém implementován do platformy HUB.FEL a uživatelsky otestován.

Díky uživatelskému testování jsem si ověřila, že implementováním systému se zrychluje proces nástupu nového stážisty, dále systém přináší evidenci přístupových práv. Také se snížil počet manuálních kroků při přidělování a odebírání přístupů díky integracím API některých systémů, které CZM využívá.

9.1 Výhled do budoucna

S ohledem na pozitivní výsledky uživatelského testování je plánováno pokračovat ve vývoji a zdokonalování systému v rámci stáže CZM. Níže jsou uvedeny doporučené kroky pro další postup.

9.1.1 Oprava chyb a nedostatků při testování

Nedostatky, které byly zmíněny v sekci 8.4, jsou minoritní a jejich zapracování nebude časově náročné. Jedná se především o frontendové úpravy.

9.1.2 Automatizace dalších systémů

V současném systému je správa přístupů zjednodušena automatickým vytvářením požadavků na přidělení či odebrání přístupu.

Do budoucna CZM plánuje integrovat Kibana API [23] stejným způsobem, jako jsou již integrovány GitLab API a Slack API, jak je popsáno v kapitole 6. Rovněž hodláme automatizovat přidělování a odebírání přístupů k Wi-Fi a VPN.

Dalším krokem pro zlepšení systému je konzultace s administrátory CZM Teams skupiny ohledně získání tokenu pro komunikaci s endpointy spravujícími členy v Teams, což by umožnilo automatizovat přístupy k CZM Teams skupině.

Dále se nabízí využití Google Calendar API pro automatické přidávání a odebírání oprávnění uživatelům k CZM kalendářům [17].

Literatura

[1] *Spojujeme výuku s praxí.* [online]. Centrum znalostního managementu FEL ČVUT. [vid. 15. 10. 2023]. Dostupné z: https://czm.fel.cvut.cz/cs/.

[2] *Platform.* [online]. GitLab B.V. [vid. 18. 10. 2023]. Dostupné z: https://about.gitlab.com/platform/.

[3] *Workflow Center - IBM Documentation.* [online]. [vid. 8. 1. 2024]. Dostupné z: https://www.ibm.com/docs/vi/dbaoc?topic=tools-workflow-center.

[4] *Process Portal - IBM Documentation.* [online]. [vid. 8. 1. 2024]. Dostupné z: https://www.ibm.com/docs/vi/dbaoc?topic=tools-process-portal.

[5] *KOSapi.* [online]. [vid. 11. 1. 2024]. Dostupné z: https://kosapi.fit.cvut.cz/projects/kosapi/wiki.

[6] *The Universal Process Orchestrator.* [online]. Camunda © 2024. [vid. 15. 1. 2024]. Dostupné z: https://camunda.com/.

[7] *Group and project members API.* [online]. [vid. 20. 1. 2024]. Dostupné z: https://docs.gitlab.com/ee/api/members.html.

[8] *admin.users:write permission scope.* [online]. ©2024 Slack Technologies, LLC, a Salesforce company. [vid. 28. 1. 2024]. Dostupné z: https://api.slack.com/scopes/admin.users:write.

[9] *Invite new members to your workspace.* [online]. ©2024 Slack Technologies, LLC, a Salesforce company. [vid. 28.

1. 2024]. Dostupné z: https://slack.com/help/articles/201330256-Invite-new-members-to-your-workspace#share-an-invitation-link.

[10] *API*. [online]. [vid. 29. 1. 2024]. Dostupné z: https://syspass-doc.readthedocs.io/en/3.1/application/api.html#methods.

[11] *Authentication and authorization basics - Microsoft Graph*. [online]. © Microsoft 2024. [vid. 2. 2. 2024]. Dostupné z: https://learn.microsoft.com/en-us/graph/auth/auth-concepts.

[12] KOVÁŘ, Adam. *Integrační platforma FEL Hub*. Diplomová práce ČVUT FEL Praha 2023.

[13] *React*. [online]. [vid. 8. 2. 2024]. Dostupné z: https://react.dev.

[14] *The gateway*. [online]. © 2024 Apollo Graph Inc. [vid. 8. 2. 2024]. Dostupné z: https://www.apollographql.com/docs/federation/v1/gateway/.

[15] *2. Service Discovery: Eureka Server*. [online]. © 2024 Apollo Graph Inc. [vid. 8. 2. 2024]. Dostupné z: https://cloud.spring.io/spring-cloud-netflix/multi/multi_spring-cloud-eureka-server.html.

[16] *Kibana: Explore, Visualize, Discover Data*. [online]. © 2024. Elasticsearch B.V. [vid. 17. 2. 2024]. Dostupné z: https://www.elastic.co/kibana.

[17] *Acl: insert | Google Calendar | Google for Developers*. [online]. [vid. 2. 3. 2024]. Dostupné z: https://developers.google.com/calendar/api/v3/reference/acl/insert.

[18] *Swarm mode overview*. [online]. Copyright © 2013-2024 Docker Inc. [vid. 6. 3. 2024]. Dostupné z: https://docs.docker.com/engine/swarm/.

[19] *Spring Boot3.2.5*. [online]. Copyright © 2005 - 2024 Broadcom. [vid. 6. 3. 2024]. Dostupné z: https://spring.io/projects/spring-boot.

[20] *Welcome to Apache Maven*. [online]. © 2002–2024 The Apache Software Foundation. [vid. 13. 3. 2024]. Dostupné z: https://maven.apache.org/.

[21] *PostgreSQL*. [online]. Copyright © 1996-2024 The PostgreSQL Global Development Group. [vid. 21. 3. 2024]. Dostupné z: https://www.postgresql.org/.

[22] *Spring Boot FeignClient vs. WebClient*. [online]. Vlad Fernoaga. [vid. 22. 3. 2024]. Dostupné z: https://www.baeldung.com/spring-boot-feignclient-vs-webclient.

[23] *Security APIs: Elasticsearch Guide [8.13]*. [online]. © 2024. Elasticsearch B.V. [vid. 28. 3. 2024]. Dostupné z: https://www.elastic.co/guide/en/elasticsearch/reference/current/security-api.html.

Příloha A

Použité zkratky

API Application programming interface

CI/CD Continuous integration and continuous delivery

CZM Centrum znalostního managementu

HR Human resources

JWT JSON Web Token

MAC Media Access Control

PR Public relations

REST Representational state transfer

SSO Single sign-on

VPN Virtual private network

Příloha B

Zdrojové kódy

Všechny zdrojové kódy implementovaného systému jsou dostupné na fakultním GitLabu. Pokud máte zájem o přístup, obraťte se na autora.

Backendová část aplikace https://gitlab.fel.cvut.cz/czm/hub/czm-management/czm-management-service/-/tree/bachelor-thesis

CI/CD konfigurace: https://gitlab.fel.cvut.cz/czm/hub/czm-management/deployments/-/tree/bachelor-thesis

Frontendová část aplikace: https://gitlab.fel.cvut.cz/czm/hub/frontend/frontend-base/-/tree/czm-management-service