

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra počítačové grafiky a interakce



Diplomová práce

Integrační portál pro sdílení a zálohování dat

Bc. Petr Strnad

Vedoucí práce: Ing. Ondřej Macek

Studijní program: Otevřená informatika

Obor: Softwarové inženýrství

prosinec 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Petr Strnad**

Studijní program: Otevřená informatika
Obor: Softwarové inženýrství

Název tématu: **Integrační server pro sdílení a zálohování dat**

Pokyny pro vypracování:

V rámci grantu CESNET Integrace datových služeb vědecko-výukové skupiny vytvořte integrační portál. Portál bude integrovat služby datového úložiště pro sdílení souborů a úložiště pro zálohování a archivaci. Koncovým uživatelům nabídne server REST rozhraní. Další požadavky identifikujte na základě dokumentace grantu a potřeb uživatelů. Při implementaci se soustřeďte i na udržitelnost vývoje.

Očekávané výstupy práce:

1. Přehled požadavků na integrační portál.
2. Návrh architektury integračního portálu a implementace co největší části požadavků s ohledem na udržitelnost vývoje (viz bod 3).
3. Kód bude řádně dokumentován a vhodně otestován pomocí unit a integračních testů.

Seznam odborné literatury:

1. CESNET. CESNET DATOVÁ ÚLOŽIŠTĚ. <http://www.cesnet.cz/sluzby/datova-uloziste/>
2. ALFRESCO. RESTful API reference. <http://docs.alfresco.com/4.2/references/RESTful-intro.html>

Vedoucí: Ing. Ondřej Macek

Platnost zadání: do konce letního semestru 2015/2016



prof. Ing. Jiří Žára, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 4. 11. 2014

Poděkování

Děkuji vedoucímu práce Ing. Ondřeji Mackovi, Ph.D. a Ing. Pavlovi Strnadovi, Ph.D. za konzultace a cenné rady při vývoji integračního portálu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 2.1.2015

Abstract

The aim of this project is to create a system, which would integrate data storage services for uploading, sharing and archiving files, and other systems from the Faculty of Electrical Engineering at the Czech Technical University. The result of this work is an analysis of technologies required for implementing the integration portal, list of functional requirements, implementation of the foundation of the system, and verification of its functionality.

Abstrakt

Cílem projektu je vytvoření systému pro integraci služeb datových úložišť pro nahrávání, sdílení a archivaci souborů, a systémů Fakulty elektrotechnické na ČVUT. Výsledkem práce je analýza technologií nutných pro realizaci systému integračního portálu, seznam požadavků na funkčnost, implementace základu systému a ověření jeho funkčnosti.

Obsah

Kapitola 1 Úvod.....	1
1.1 Kontext projektu.....	2
Kapitola 2 Analýza a návrh řešení.....	3
2.1 Základní popis architektury.....	3
2.1.1 Archivace a zálohování.....	3
2.1.2 Sdílení.....	4
2.1.3 Služby a systémy FEL ČVUT.....	4
2.2 Popis požadavků na integrační portál.....	6
2.2.1 Funkční požadavky.....	6
2.2.1.1 Funkční požadavky na uživatelské účty.....	6
2.2.1.2 Funkční požadavky na organizační jednotky.....	7
2.2.1.3 Funkční požadavky na domovskou složku a sdílení.....	7
2.2.1.4 Funkční požadavky na archivaci a zálohování.....	8
2.2.2 Obecné požadavky.....	8
2.3 Případy užití (use case).....	9
2.3.1 Aktéři.....	9
2.3.2 Případy užití uživatelských účtů.....	10
2.3.2.1 Mapování na funkční požadavky.....	12
2.3.2.2 Příklad užití UC1 – přihlášení přes FELid.....	13
2.3.2.3 Příklad užití UC2 – přihlášení přes integrační portál.....	13
2.3.2.4 Příklad užití UC3 – vytvoření účtu pro externistu.....	13
2.3.2.5 Příklad užití UC4 – smazání účtu externisty.....	14
2.3.2.6 Příklad užití UC5 – vytvoření uživatelské role.....	14
2.3.2.7 Příklad užití UC6 – smazání uživatelské role.....	15
2.3.2.8 Příklad užití UC7 – přiřazení role uživateli.....	15
2.3.2.9 Příklad užití UC8 – odebrání role uživateli.....	16
2.3.2.10 Příklad užití UC9 – zobrazení seznamu uživ. rolí.....	16
2.3.2.11 Příklad užití UC10 – zobrazení seznamu uživatelů.....	17
2.3.3 Případy užití organizačních jednotek.....	17
2.3.3.1 Mapování na funkční požadavky.....	18
2.3.3.2 Příklad užití UC11 – přidání uživatele do org. jednotky.....	19
2.3.3.3 Příklad užití UC12 – odebrání uživatele z org. jednotky.....	20
2.3.3.4 Příklad užití UC13 – zobrazení využití kvóty org. jednotky.....	20
2.3.3.5 Příklad užití UC14 – zobrazení využití místa uživatelem v org. jednotce.....	20
2.3.3.6 Příklad užití UC15 – zobrazení využití místa dat. úložiště.....	21
2.3.3.7 Příklad užití UC16 – přiřazení práv správce org. jednotky.....	21
2.3.3.8 Příklad užití UC17 – vytvoření org. jednotky.....	22
2.3.3.9 Příklad užití UC18 – smazání org. jednotky.....	22
2.3.3.10 Příklad užití UC19 – nastavení kvóty org. jednotky.....	23
2.3.3.11 Příklad užití UC20 – vybrání org. jednotky.....	23
2.3.4 Případy užití domovské složky a sdílení.....	24
2.3.4.1 Mapování na funkční požadavky.....	26
2.3.4.2 Příklad užití UC21 – vytvoření složky.....	27
2.3.4.3 Příklad užití UC22 – nahrání souboru.....	27
2.3.4.4 Příklad užití UC23 – smazání složky.....	27
2.3.4.5 Příklad užití UC25 – přesunutí složky.....	28
2.3.4.6 Příklad užití UC29 – sdílení složky.....	28
2.3.4.7 Příklad užití UC32 – zobrazení souboru v domovské složce.....	29

2.3.4.8 Příklad užití UC33 – zobrazení sdílené složky.....	29
2.3.4.9 Příklad užití UC35 – přesunout složku do archivu.....	29
2.3.5 Případy užití archivace a zálohování.....	30
2.3.5.1 Mapování na funkční požadavky.....	32
2.3.5.2 Příklad užití UC51 – přepnutí stavu souboru v archivu.....	32
2.4 Integrované služby.....	33
2.4.1 FELid.....	33
2.4.2 KOSapi.....	34
2.4.3 CESNET.....	34
2.4.3.1 Hierarchický model ukládání dat.....	34
2.4.3.1.1 Příkazy HSM.....	35
2.4.3.1.2 Seznam stavů archivace.....	36
2.4.4 Alfresco.....	36
Kapitola 3 Implementace.....	39
3.1 Implementované požadavky.....	39
3.2 Použité technologie, knihovny a API.....	40
3.2.1 REST.....	41
3.2.2 Maven.....	41
3.2.3 Spring.....	41
3.2.3.1 Spring Core.....	42
3.2.3.2 Spring Web a Spring WebMVC.....	42
3.2.3.3 Spring Data Access.....	42
3.2.3.4 Spring Security.....	42
3.2.4 OAuth.....	43
3.2.5 OpenCMIS.....	43
3.2.5.1 Hierarchická struktura složek v Alfrescu.....	43
3.3 Balíčky integračního portálu.....	44
3.3.1 Balíček controller.....	45
3.3.2 Balíček validator.....	45
3.3.3 Balíček representation.....	45
3.3.4 Balíček service.....	46
3.3.5 Balíček dao.....	46
3.3.6 Balíček model.....	46
3.3.7 Balíček cmis.....	46
3.3.8 Balíček cesnet.....	47
3.3.9 Balíček filter.....	47
Kapitola 4 Popis API integračního portálu.....	49
4.1 Přihlášení.....	49
4.2 Administrace uživatelů.....	52
4.3 Organizační skupina.....	53
4.4 Domovská složka uživatele.....	53
4.5 Složka se soubory sdílenými s uživatelem.....	54
4.6 Archiv.....	54
Kapitola 5 Ověření výsledku.....	55
5.1 Grafický klient.....	55
5.2 Jednoduchý REST klient.....	56
5.3 Testování.....	58
5.3.1 Uživatelské účty.....	58
5.3.2 Domovská složka.....	59
5.3.3 Sdílení.....	60

5.3.4 Archiv.....	60
5.3.4.1 Čas potřebný pro migraci souborů.....	61
5.4 Shrnutí.....	63
Kapitola 6 Závěr.....	65
6.1 Budoucí práce.....	65
6.2 Shrnutí.....	66
Použité zdroje.....	69

Seznam ilustrací

Obr. 2.1: Základní nástin architektury integračního portálu.....	5
Obr. 2.2: Diagram hierarchie aktérů.....	10
Obr. 2.3: Diagram případů užití pro uživatelské účty.....	11
Obr. 2.4: Diagram případů užití pro org. jednotky.....	18
Obr. 2.5: Diagram případů užití pro domovskou složku a sdílení.....	25
Obr. 2.6: Diagram případů užití pro archiv a zálohování.....	31
Obr. 3.1: Hierarchie uživatelské složky v Alfrescu.....	44
Obr. 3.2: Diagram balíčků integračního portálu.....	44
Obr. 5.1: Grafické uživatelské rozhraní.....	55
Obr. 5.2: Jednoduchý REST klient.....	57
Obr. 5.3: Graf času potřebného pro přepnutí souboru do offline stavu.....	62

Seznam tabulek

Tabulka 2.1: Splnění funkčních požadavků případy užití pro uživ. účty.....	12
Tabulka 2.2: Splnění funkčních požadavků případy užití pro org. jednotky.....	19
Tabulka 2.3: Splnění funkčních požadavků případy užití pro sdílení.....	26
Tabulka 2.4: Splnění funkčních požadavků případy užití pro archiv.....	32
Tabulka 4.1: Požadavek pro přihlášení GET /oauth/token.....	51
Tabulka 4.2: Požadavek pro vygenerování access tokenu prostřednictvím refresh tokenu.....	51
Tabulka 4.3: Odpověď na požadavek GET /oauth/token.....	51
Tabulka 5.1: Čas potřebný pro přepnutí stavu souborů podle jejich velikosti.....	62

Kapitola 1 Úvod

Akademičtí pracovníci a studenti ČVUT mezi sebou často potřebují sdílet data, ke kterým chtějí mít kdykoli snadný přístup. Navíc také vyžadují možnost archivovat i velké soubory, aniž by byli omezováni nedostatkem volného místa.

Existuje spousta služeb, které umožňují ukládání a sdílení souborů. Kupříkladu **Disk Google** umožňuje snadné nahrávání a sdílení souborů s ostatními uživateli [16]. **Microsoft OneDrive** umožňuje navíc integraci s produkty Microsoft Office [17].

Problém se zmíněnými službami ovšem tkví ve faktu, že žádná z nich neumožňuje přímou integraci s libovolnými dalšími systémy ČVUT. Disk Google je sice propojen se systémem **FELid**, tedy globální autentizační a autorizační systém pro aplikace v síti Fakulty elektrotechnické na ČVUT [14], nicméně ostatní systémy nepodporuje. Přitom dalším často potřebným systémem je **KOSapi**, který poskytuje přístup k vybrané části dat v databázi uživatelů KOS [15], a který není možné integrovat ani do Disku Google, ani do žádné jiné poskytované služby.

Z těchto faktů tedy plyne, že je nutné vytvořit nový systém, který bude integrovat jednotlivé služby pro sdílení dat, a navíc umožní provázání se systémy ČVUT. Takovýto systém navíc může vzniknout v rámci grantu **HS: 13144 / 830 / 8301442C** pro integraci datových služeb **CESNET** pro vědecko-výukové skupiny. CESNET je sdružení, zabývající se výzkumem a vývojem informačních a komunikačních technologií [1]. Vytvořený systém nad CESNETem bude tedy integrovat ČVUT systémy (jako je FELid a KOSapi) s úložným prostorem, který univerzitě poskytne sdružení CESNET. Výsledkem bude ucelený integrační portál pro ukládání a sdílení dat, který bude dostupný všem akademikům Fakulty elektrotechnické na ČVUT.

Cílem diplomové práce je sestavení požadavků na takovýto integrační portál, návrh architektury a implementace co největší části požadavků. Vzhledem k rozsahu práce a časovému omezení se již během zadání práce počítalo s tím, že projekt nebude zcela dokončen. Hotová část tedy bude předána k dokončení dalšímu studentovi.

1.1 Kontext projektu

Integrační portál bude komunikovat se systémy ČVUT, dále s úložišti dat, a navíc bude muset svým uživatelům poskytovat intuitivní uživatelské rozhraní. Práce na integračním portálu byla tedy rozdělena do dvou projektů.

První částí je tzv. **backend**, tedy serverová část. Ta bude nasazena na server a bude se starat o integraci služeb ČVUT a datových úložišť. Bude poskytovat aplikační rozhraní (**API**), jehož prostřednictvím mohou klienti přistupovat k těmto službám. Tato diplomová práce se zabývá právě serverovou částí.

Druhou částí je tzv. **frontend**, tedy klientská část, která bude zobrazována v internetových prohlížečích uživatelů. Ta bude využívat API poskytované serverovou částí, a umožní uživatelům používat tyto služby prostřednictvím intuitivního grafického rozhraní. Touto částí se zabývá diplomová práce Kateřiny Hašlarové.

Hlavní využití integračního portálu tkví ve sdílení a archivaci souborů. V rámci sdílení souborů bude uživateli umožněno, aby ke svým souborům umožnil přístup i jiným uživatelům. Naproti tomu požadavky na archivaci jsou mírně odlišné, neboť uživatelé potřebují mít uložena velká množství dat, ke kterým přistupují jen velmi zřídka. Archivace tedy vyžaduje daleko větší datové úložiště, které však nemusí být tak rychlé, jako v případě sdílení dat.

Kromě sdílení a archivace dat je však dalším cílem i poskytnutí služeb pro administraci. Správci integračního portálu by tedy měli být schopni monitorovat a spravovat datové úložiště a uživatelské účty.

Kapitola 2 Analýza a návrh řešení

2.1 Základní popis architektury

Jak již bylo nastíněno v úvodu, tato diplomová práce se nezabývá uživatelským rozhraním, které je vytvářeno v rámci související diplomové práce Kateřiny Hašlarové, nýbrž pouze samotnou serverovou částí integračního portálu.

Serverová část má integrovat služby datových úložišť pro sdílení, zálohování a archivaci souborů, a dále služby ČVUT jako je **KOSapi** a **FELid**. Jelikož se však integrací se službami ČVUT zabývají již další studentské projekty, bylo rozhodnuto, že hlavním zaměřením práce budou datové služby. Integrace služeb jako je KOSapi a FELid tedy proběhne až v navazující práci. Celkový přehled architektury zachycuje obr. 2.1: Základní nástin architektury integračního portálu.

2.1.1 Archivace a zálohování

Pro zálohování a archivaci souborů byla v rámci **CESNET** grantu HS: 13144 / 830 / 8301442C „integrace datových služeb pro vědecko-výukové skupiny“ vybrána právě úložiště CESNETu.

Datová úložiště CESNETu poskytují dostatečný prostor pro úložiště souborů integračního portálu (v době psaní této práce 400 + 900 GB v základní virtuální organizaci „storage“ [2], s možností vytvoření vlastní organizace s daleko větším limitem pro univerzitu). Přístup k těmto úložištím je realizován prostřednictvím SFTP (Secure File Transfer Protocol [20]).

Dále je možné se k těmto serverům připojit i prostřednictvím SSH (Secure Shell [19]). Prostřednictvím SSH může integrační portál kontrolovat, které soubory jsou ukládány na pomalá, vysokokapacitní úložiště, a které jsou udržovány na rychlejších médiích.

2.1.2 Sdílení

Pro sdílení souborů bylo vybráno **Alfresco** [25], což je open source **ECM** systém (Enterprise Content Management - systém pro vytváření, správu, zabezpečení, prohlížení a distribuci dat [24]).

Mezi hlavní výhody Alfresca patří fakt, že implementuje **CMIS** (Content Management Interoperability Services – standard pro komunikaci s úložišti dat [3]). Díky tomu je možné v integračním portálu přímo použít klientskou verzi tohoto API bez nutnosti implementace komunikace se samotným Alfrescem.

Další výhodou je, že Alfresco je ucelený systém, který se o veškerou správu souborů (kontrola pravomocí přístupu, ...) stará sám, a integračnímu portálu tak stačí využívat přímo jeho služby bez nutnosti implementace další logiky, jako by tomu bylo v případě využití např. přímého přístupu k CESNET úložišti. Díky tomu mohou uživatelé navíc přistupovat k souborům na Alfrescu prostřednictvím jiných klientů, aniž by hrozilo uvedení dat do nekonzistentního stavu.

Alfresco tedy bude použito jako nadstavba nad úložištěm CESNETu, kdy Alfresco bude mít kontrolu nad správou souborů.

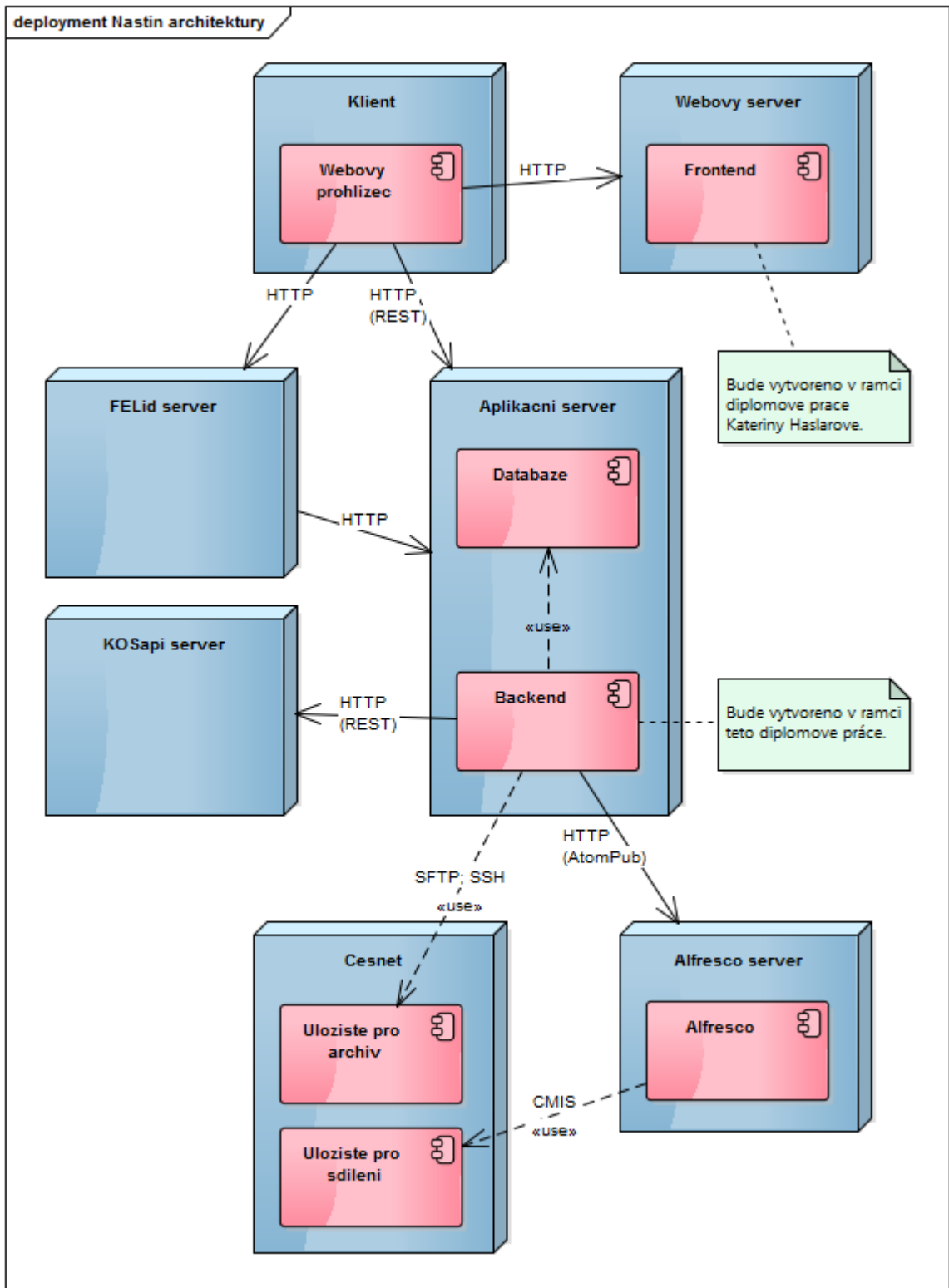
Výhod Alfresca však nebylo možné využít v případě archivace a zálohování, jelikož by to znamenalo ztrátu možnosti kontroly výběru úložných médií prostřednictvím SSH. V případě archivace a zálohování tedy musí probíhat komunikace s CESNET servery přímo prostřednictvím SFTP a SSH, bez nadstavby v podobě Alfresca.

2.1.3 Služby a systémy FEL ČVUT

Pro prvotní integraci byly vybrány jakožto nejdůležitější dvě služby Fakulty elektrotechnické: **FELid** a **KOSapi**.

FELid umožňuje bezpečné a centralizované přihlášení a je již považováno za standard pro všechny systémy Fakulty elektrotechnické, integrační portál by tedy měl umožňovat tento způsob přihlášení.

KOSapi zprostředkovává přístup k informacím o uživatelích a je tedy nezbytné například pro získání seznamu předmětů, do kterých je student zapsán, či které vyučuje profesor.



Obr. 2.1: Základní nástin architektury integračního portálu

2.2 Popis požadavků na integrační portál

Zde jsou představeny všechny plánované funkční i obecné požadavky na integrační portál, včetně těch, které nebudou v rámci této diplomové práce dokončeny. Funkční požadavky jsou následně dále rozvíjeny v kapitole 2.3 Případy užití (use case), kde jsou uvedeni aktéři v rámci integračního portálu, diagramy případů užití, jejich popisy, a konkrétní scénáře případů užití.

2.2.1 Funkční požadavky

Následuje seznam všech funkčních požadavků na systém. Tyto požadavky představují akce, které musí integrační portál uživatelům umožnit provádět. Pro lepší přehlednost jsou rozděleny do tří základních kategorií, odpovídajícím modulům, podle funkcionality. Jedná se o moduly pro **uživatelské účty**, **organizační jednotky**, **domovské složky a sdílení**, a **archiv a zálohování**.

2.2.1.1 Funkční požadavky na uživatelské účty

- F1 Systém umožní akademickým pracovníkům a studentům přihlásit se prostřednictvím globálního autentizačního a autorizačního systému FELid.
- F2 Systém umožní externistům, kterým správce externistů vytvořil v integračním portálu uživatelský účet, přihlásit se přímo prostřednictvím integračního portálu.
- F3 Systém umožní správcům externistů vytvářet uživatelské účty pro externisty.
- F4 Systém umožní správcům externistů mazat uživatelské účty externistů.
- F5 Systém umožní hlavnímu správci vytvářet uživatelské role s konkrétními oprávněními.
- F6 Systém umožní hlavnímu správci mazat uživatelské role.
- F7 Systém umožní hlavnímu správci přiřazovat uživatelům uživatelské role.
- F8 Systém umožní hlavnímu správci odebírat uživatelům uživatelské role.
- F9 Systém umožní hlavnímu správci zobrazení všech uživatelských rolí.
- F10 Systém umožní hlavnímu správci zobrazení všech uživatelských účtů.

2.2.1.2 Funkční požadavky na organizační jednotky

- F11 Systém umožní správcům organizační jednotky přidávat uživatele do organizační jednotky.
- F12 Systém umožní správcům organizační jednotky odebrat uživatele z organizační jednotky.
- F13 Systém umožní správcům organizační jednotky zobrazení obsazeného, dostupného a celkového volného místa v rámci kvóty organizační jednotky.
- F14 Systém umožní správcům organizační jednotky zobrazení obsazeného místa v rámci kvóty organizační jednotky jednotlivými uživateli.
- F15 Systém umožní hlavnímu správci zobrazení obsazeného, dostupného a celkového volného místa v rámci kvóty organizační jednotky.
- F16 Systém umožní hlavnímu správci zobrazení obsazeného místa v rámci kvóty organizační jednotky jednotlivými uživateli.
- F17 Systém umožní hlavnímu správci zobrazení obsazeného a celkového volného místa v rámci datového úložiště.
- F18 Systém umožní hlavnímu správci přiřadit zvolenému uživateli práva správce organizační jednotky.
- F19 Systém umožní hlavnímu správci vytvářet organizační jednotky.
- F20 Systém umožní hlavnímu správci mazat organizační jednotky.
- F21 Systém umožní hlavnímu správci nastavovat kvóty organizačním jednotkám (přidělovat jim prostor v datovém úložišti).

2.2.1.3 Funkční požadavky na domovskou složku a sdílení

- F22 Systém umožní přihlášeným uživatelům vytvářet složky a podsložky v jejich domovské složce.
- F23 Systém umožní přihlášeným uživatelům nahrávat soubory do domovské složky a také do podsložek domovské složky.
- F24 Systém umožní přihlášeným uživatelům mazat složky, podsložky a soubory v jejich domovské složce.
- F25 Systém umožní přihlášeným uživatelům přesouvat a kopírovat složky, podsložky a soubory v jejich domovské složce.

KAPITOLA 2 ANALÝZA A NÁVRH ŘEŠENÍ

- F26 Systém umožní přihlášeným uživatelům sdílení vybraných složek, podsložek a souborů v jejich domovské složce s konkrétními uživateli.
- F27 Systém umožní přihlášeným uživatelům prohlížení složek, podsložek a souborů v jejich domovské složce.
- F28 Systém umožní přihlášeným uživatelům prohlížení složek, podsložek a souborů, které s přihlášeným uživatelem sdílejí ostatní uživatelé.
- F29 Systém umožní přihlášeným uživatelům přesouvání složek a souborů z jejich domovské složky do archivu.

2.2.1.4 Funkční požadavky na archivaci a zálohování

- F30 Systém umožní přihlášeným uživatelům vytváření složek a podsložek v archivu.
- F31 Systém umožní přihlášeným uživatelům nahrávání souborů do složek a podsložek archivu.
- F32 Systém umožní přihlášeným uživatelům mazat složky, podsložky a soubory v archivu.
- F33 Systém umožní přihlášeným uživatelům přesouvat a kopírovat složky, podsložky a soubory v archivu.
- F34 Systém umožní přihlášeným uživatelům prohlížení složek, podsložek a souborů v archivu.
- F35 Systém umožní přihlášeným uživatelům přesouvání složek a souborů z archivu do jejich domovské složky.
- F36 Systém umožní přihlášeným uživatelům přepnout soubory v archivu z „online“ do „offline“ stavu a zpět. Více informací o těchto stavech a jejich významu je uvedeno v kapitole 2.4.3 CESNET.

2.2.2 Obecné požadavky

Obecné požadavky, nebo též nefunkční požadavky, nepopisují konkrétní aktivity či akce, které musí být uživateli umožněny. Namísto toho představují obecné vlastnosti, které musí systém splňovat, aby byl celkově použitelný. Pro integrační portál byly stanoveny následující obecné požadavky.

- N1 Systém bude naprogramován v programovacím jazyce Java.
- N2 Systém bude nasazen na aplikační server a bude dostupný prostřednictvím internetu.

- N3 Systém bude poskytovat **RESTful** [18] aplikační rozhraní (API) pro přístup ke službám. Pro více informací o RESTful webových službách a o REST viz. kapitola 3.2.1 REST.
- N4 Systém bude provádět autentizaci a autorizaci uživatele prostřednictvím protokolu OAuth 2.0 [8].
- N5 Systém bude využívat úložiště CESNET.

2.3 Případy užití (use case)

Případy užití dále rozvádějí funkční požadavky ze seznamu v kapitole 2.2.1 Funkční požadavky. V jednotlivých podkapitolách jsou představeni aktéři v rámci integračního portálu, a dále jednotlivé případy užití podle modulů.

Případy užití jsou modelovány z pohledu uživatele a zachycují tedy i část uživatelského rozhraní, jako např. spojení „systém zobrazí formulář“. Uživatelské rozhraní je však vytvářeno v rámci diplomové práce Kateřiny Hašlarové.

2.3.1 Aktéři

Aktéři představují jednotlivá oprávnění uživatelů. V rámci integračního portálu existuje několik různých oprávnění, které se následně mohou přiřazovat k uživatelským rolím a ty následně konkrétním uživatelům. Systém oprávnění tvoří hierarchickou strukturu, která odpovídá hierarchické struktuře aktérů. Ta je vyobrazena na obrázku 2.2: Diagram hierarchie aktérů.

Konkrétní uživatel může mít přiřazeno více uživatelských rolí, a v důsledku tedy i více uživatelských oprávnění. Může tedy vystupovat jako několik různých aktérů z poslední (spodní) úrovně hierarchie aktérů (**správce organizační jednotky**, **správce externistů**, **hlavní správce**). Vždy se však musí jednat buď o **akademika**, nebo o **externistu**.

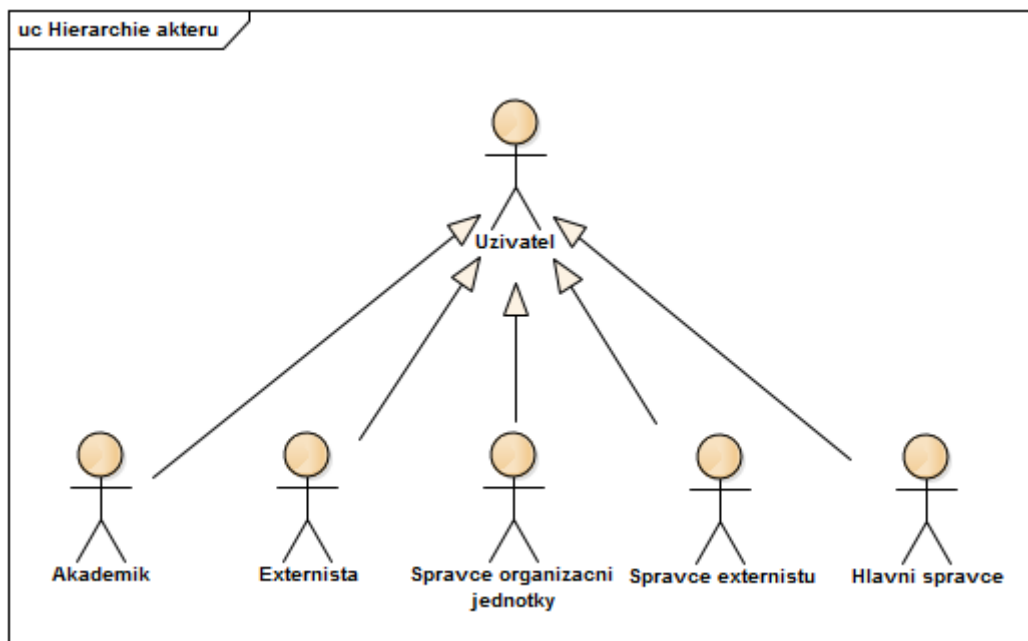
Akademik a **externista** jsou oba **uživatelé**, a oba mohou využívat datové úložiště. Jediný rozdíl mezi těmito dvěma aktéry je tedy ve způsobu přihlašování a dále není možné získat údaje o externistech prostřednictvím KOSapi.

Hlavní správce má na starosti celkovou správu integračního portálu. V rámci organizačních jednotek může jednotkám přiřazovat dostupný prostor v datovém úložišti prostřednictvím nastavení kvót, vytvářet nové jednotky, a celkově monitorovat stav datového úložiště. Dále může vytvářet a přiřazovat uživatelské role konkrétním uživatelům.

Správce externistů se stará o uživatelské účty externistů. Konkrétně může vytvářet nové, či mazat a upravovat stávající účty pro externisty.

KAPITOLA 2 ANALÝZA A NÁVRH ŘEŠENÍ

Správce organizační jednotky má na starosti správu přiřazené organizační jednotky. Toto oprávnění tedy nikdy není globální, ale vždy se vztahuje ke konkrétní jednotce. Takovýto správce si může zobrazovat statistiky o využití úložného prostoru v rámci jednotky, využití místa konkrétním uživatelem, a dále může přidávat a odebírat uživatele z a do organizační jednotky.



Obr. 2.2: Diagram hierarchie aktérů

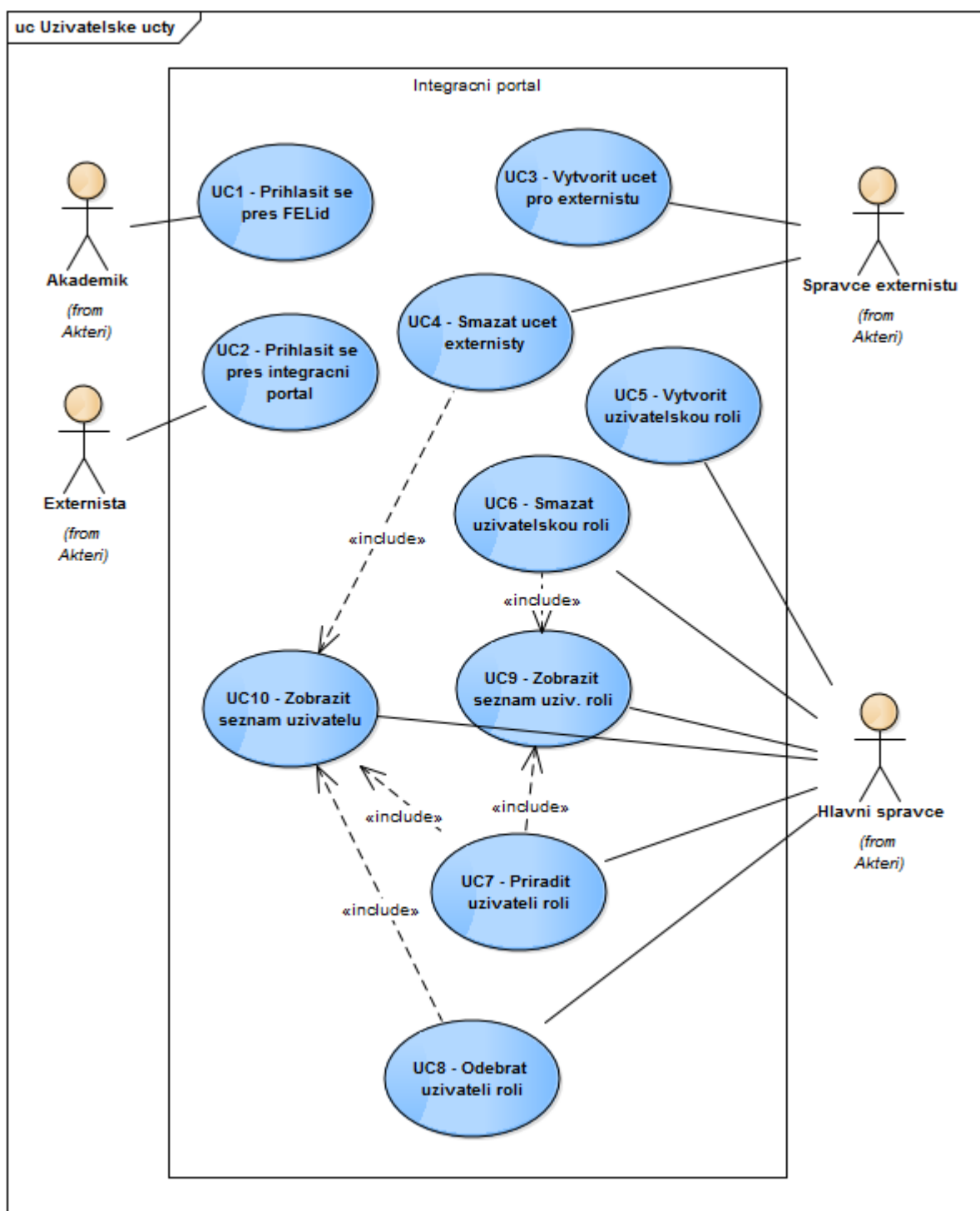
2.3.2 Případy užití uživatelských účtů

V této kapitole jsou rozvinuty funkční požadavky z kapitoly 2.2.1.1 Funkční požadavky na uživatelské účty, konkrétně tedy požadavky F1 až F10. Jednotlivé případy užití, které souvisejí s uživatelskými účty, jsou vyobrazeny na obrázku 2.3: Diagram případů užití pro uživatelské účty. Zde vystupují následující čtyři aktéři: **akademik**, **externista**, **správce externistů** a **hlavní správce**.

Akademikové a **externisté** se mohou pouze přihlašovat, akademikové přes systém FELid, naproti tomu externisté se přihlašují přímo prostřednictvím integračního portálu.

Správce externistů může vytvářet a mazat uživatelské účty pro externisty, nemůže jim ovšem přiřazovat ani odebírat uživatelské role.

Hlavní správce vytváří uživatelské role, které zastřešují jednotlivá oprávnění v rámci integračního portálu, a může je přiřazovat jednotlivým uživatelům.



Obr. 2.3: Diagram případů užití pro uživatelské účty

2.3.2.1 Mapování na funkční požadavky

Pro kontrolu splnění všech postihovaných funkčních požadavků slouží tabulka 2.1: Splnění funkčních požadavků případy užití pro uživ. účty. Ta zachycuje vztahy mezi jednotlivými funkčními požadavky z kapitoly 2.2.1.1 Funkční požadavky na uživatelské účty a konkrétními případy užití, které jsou vyobrazeny na obrázku 2.3: Diagram případů užití pro uživatelské účty a dále rozvíjeny ve scénářích případů užití v následujících podkapitolách.

Případy užití	Požadavky									
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
UC1	↑									
UC2		↑								
UC3			↑							
UC4				↑						
UC5					↑					
UC6						↑				
UC7							↑			
UC8								↑		
UC9									↑	
CU10										↑

Tabulka 2.1: Splnění funkčních požadavků případy užití pro uživ. účty

Jak je z tabulky patrné, funkční požadavky **F1** až **F10** jsou splněny odpovídajícími případy užití **UC1** až **UC10**.

Pokud některý případ užití využívá funkcionalitu jiného případu, mluví se o tzv. **includování případů užití**. To znamená, že se funkcionalita includovaného případu začlení přímo do scénáře toho případu užití, který jej includeje.

Zde jsou includovány případy užití **UC9 – Zobrazit seznam uživ. rolí** a **UC10 – Zobrazit seznam uživatelů**.

2.3.2.2 Případ užití UC1 – přihlášení přes FELid

Tento případ užití zachycuje jeden ze dvou možných způsobů přihlášení se do systému. V tomto případě se jedná o přihlášení prostřednictvím FELid. Takto se mohou přihlašovat pouze **akademikové** (akademičtí pracovníci a studenti).

Hlavní scénář:

1. Případ užití začíná, když se akademik rozhodne přihlásit se do integračního portálu.
2. Integrační portál předá požadavek FELid.
3. Akademik vyplní své přihlašovací údaje a potvrdí zadání.
4. FELid předá požadavek zpět na integrační portál.
5. Integrační portál akademika přihlásí a zobrazí mu zprávu o úspěšném přihlášení.

2.3.2.3 Případ užití UC2 – přihlášení přes integrační portál

Tento případ užití zachycuje druhý ze dvou možných způsobů přihlášení se do systému, a to přímo prostřednictvím integračního portálu. Tato metoda je určena pro **externisty**, kteří nemají účet v databázi Fakulty elektrotechnické ČVUT.

Hlavní scénář:

1. Případ užití začíná, když se externista rozhodne přihlásit se do integračního portálu.
2. Integrační portál zobrazí formulář pro přihlášení, který umožňuje zadat přihlašovací jméno a heslo.
3. Externista vyplní své přihlašovací jméno a heslo, a potvrdí zadání.
4. Integrační portál externistu přihlásí a zobrazí mu zprávu o úspěšném přihlášení.

2.3.2.4 Případ užití UC3 – vytvoření účtu pro externistu

Tento případ užití zachycuje proces vytváření uživatelského účtu pro externistu, tedy pro uživatele, který nemá účet v databázi Fakulty elektrotechnické ČVUT. Vytvářet uživatelské účty pro externisty v rámci integračního portálu může pouze **správce externistů**.

KAPITOLA 2 ANALÝZA A NÁVRH ŘEŠENÍ

Hlavní scénář:

1. Příklad užití začíná, když se správce externistů rozhodne přidat nového externistu.
2. Integrovaný portál zobrazí formulář umožňující zadat externistovo uživatelské jméno, heslo a identifikační číslo organizační jednotky, do které má externista patřit.
3. Správce externistů vyplní povinné údaje uživatelské jméno, heslo a identifikační číslo organizační jednotky, a potvrdí zadání.
4. Integrovaný portál zkontroluje, zda již neexistuje uživatel se stejným jménem, a zda identifikační číslo organizační jednotky existuje. Integrovaný portál vytvoří nový účet pro externistu a zobrazí aktualizovaný seznam uživatelů a zprávu o úspěšném vytvoření.

2.3.2.5 Příklad užití UC4 – smazání účtu externisty

Tento případ užití zachycuje proces smazání existujícího uživatelského účtu pro externistu. Mazat uživatelské účty externistů v rámci integrovaného portálu může opět pouze **správce externistů**.

Hlavní scénář:

1. Příklad užití začíná, když se správce externistů rozhodne smazat existující účet některého externisty.
2. Include (UC10 – Zobrazit seznam uživatelů).
3. Správce externistů vybere požadovaného uživatele.
4. Integrovaný portál zobrazí detail vybraného uživatele.
5. Správce externistů na detailu uživatele klikne na tlačítko pro smazání uživatele.
6. Integrovaný portál smaže vybraný uživatelský účet externisty a zobrazí aktualizovaný seznam uživatelů a zprávu o úspěšném smazání.

2.3.2.6 Příklad užití UC5 – vytvoření uživatelské role

Tento případ užití zachycuje proces vytvoření uživatelské role, což může dělat pouze **hlavní správce**. Uživatelské role se přiřazují konkrétním uživatelům a obsahují seznam oprávnění v rámci systému.

Role může obsahovat pouze globální oprávnění, jako jsou například oprávnění pro **správce externistů** a pro **hlavního správce**. **Správce organizačních jednotek** není globální rolí, jelikož se vždy vztahuje ke konkrétní jednotce. To znamená, že nemůže být přiřazena k uživatelským rolím.

Hlavní scénář:

1. Příklad užití začíná, když se hlavní správce rozhodne vytvořit novou uživatelskou roli.
2. Integrovaný portál zobrazí formulář, který umožní zadat jméno role a její popis. Dále bude formulář obsahovat seznam oprávnění v rámci systému.
3. Hlavní správce vyplní povinné údaje jméno a popis role, dále vybere ze seznamu jedno či více oprávnění, a potvrdí zadání.
4. Integrovaný portál zkontroluje, zda role s daným jménem již neexistuje. Integrovaný portál vytvoří novou uživatelskou roli a zobrazí aktualizovaný seznam rolí a zprávu o úspěšném vytvoření.

2.3.2.7 Příklad užití UC6 – smazání uživatelské role

Tento případ užití zachycuje proces smazání existující uživatelské role. Role může mazat pouze **hlavní správce**.

Hlavní scénář:

1. Příklad užití začíná, když se hlavní správce rozhodne smazat existující uživatelskou roli.
2. Include (UC9 – Zobrazit seznam uživ. rolí).
3. Hlavní správce vybere požadovanou uživatelskou roli.
4. Integrovaný portál zobrazí detail uživatelské role.
5. Hlavní správce na detailu uživatelské role klikne na tlačítko pro smazání role.
6. Integrovaný portál smaže vybranou uživatelskou roli a zobrazí aktualizovaný seznam rolí a zprávu o úspěšném smazání.

2.3.2.8 Příklad užití UC7 – přiřazení role uživateli

Tento případ užití zachycuje proces přiřazení uživatelské role ke konkrétnímu uživateli. Role může přiřazovat pouze **hlavní správce**.

Hlavní scénář:

1. Příklad užití začíná, když se hlavní správce rozhodne přiřadit uživatelskou roli k uživateli.
2. Include (UC10 – Zobrazit seznam uživatelů).
3. Hlavní správce vybere požadovaného uživatele.

KAPITOLA 2 ANALÝZA A NÁVRH ŘEŠENÍ

4. Integrovaný portál zobrazí detail uživatele.
5. Hlavní správce na detailu uživatele klikne na tlačítko pro přidání uživatelské role.
6. Include (UC9 – Zobrazit seznam uživ. Rolí).
7. Hlavní správce vybere požadovanou uživatelskou roli.
8. Integrovaný portál přiřadí vybranou uživatelskou roli k vybranému uživateli a zobrazí detail uživatele s aktualizovaným seznamem rolí a zprávu o úspěšném přiřazení role.

2.3.2.9 Příklad užití UC8 – odebrání role uživateli

Tento případ užití zachycuje proces odebrání uživatelské role konkrétnímu uživateli. Role může odebírat pouze **hlavní správce**.

Hlavní scénář:

1. Příklad užití začíná, když se hlavní správce rozhodne uživateli odebrat uživatelskou roli.
2. Include (UC10 – Zobrazit seznam uživatelů).
3. Hlavní správce vybere požadovaného uživatele.
4. Integrovaný portál zobrazí detail uživatele.
5. Hlavní správce na detailu uživatele vybere ze seznamu rolí uživatele tu roli, kterou chce uživateli odebrat.
6. Integrovaný portál odebere vybranému uživateli vybranou uživatelskou roli a zobrazí detail uživatele s aktualizovaným seznamem rolí a zprávu o úspěšném odebrání role.

2.3.2.10 Příklad užití UC9 – zobrazení seznamu uživ. rolí

Tento případ užití zachycuje proces zobrazení seznamu všech uživatelských rolí v integrovaném portálu. Tuto funkcionalitu využívají i ostatní případy užití, konkrétně **UC6 – Smazat uživatelskou roli** a **UC7 – Přiřadit uživateli roli**.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne zobrazit si seznam uživatelských rolí.
2. Integrovaný portál zobrazí seznam všech existujících uživatelských rolí.

2.3.2.11 Příklad užití UC10 – zobrazení seznamu uživatelů

Tento případ užití zachycuje proces zobrazení seznamu všech uživatelů v integračním portálu. Tuto funkcionalitu využívají i ostatní případy užití, konkrétně UC4 – **Smazat účet externisty**, UC7 – **Přiřadit uživateli roli** a UC8 – **Odebrat uživateli roli**.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne zobrazit si seznam uživatelů.
2. Integrační portál zobrazí seznam všech existujících uživatelských účtů, tedy externistů i akademiků. Účty externistů budou v seznamu odlišeny od účtů akademiků.

2.3.3 Případy užití organizačních jednotek

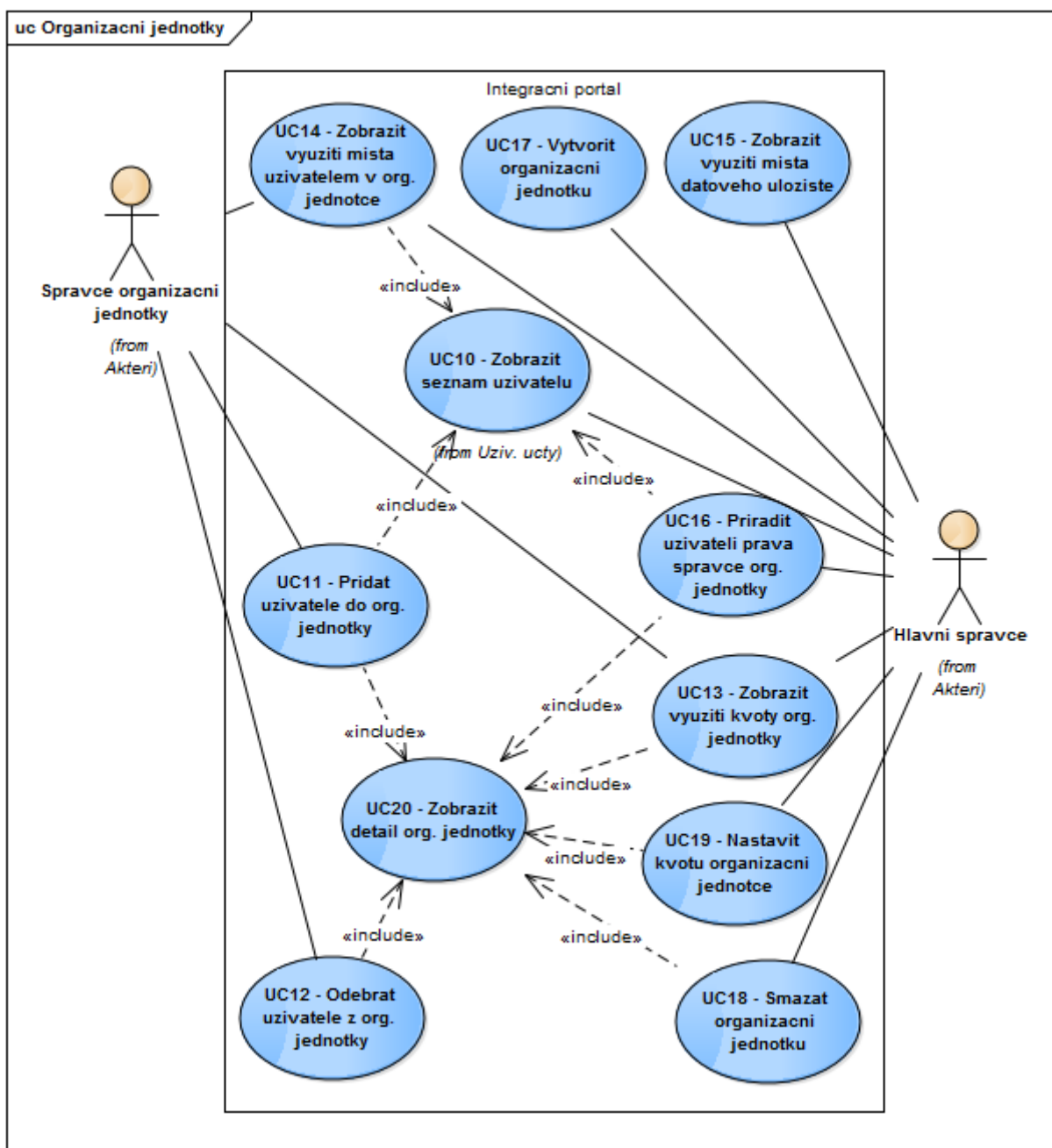
V této kapitole jsou rozvinuty funkční požadavky z kapitoly 2.2.1.2 Funkční požadavky na organizační jednotky, konkrétně tedy požadavky F11 až F21. Jednotlivé případy užití, které souvisejí s organizačními jednotkami, jsou vyobrazeny na obrázku 2.4: Diagram případů užití pro org. jednotky.

V modulu organizačních jednotek tedy vystupují pouze dva aktéři: **správce organizační jednotky** a **hlavní správce**.

Správce organizační jednotky může přidávat a odebírat uživatele do a z organizační jednotky, a dále může monitorovat využití místa v rámci kvóty organizační jednotky, pro kterou je určen jako správce, a využití této místa v této organizační jednotce jednotlivými uživateli.

Hlavní správce může vytvářet a mazat organizační jednotky, nastavovat jim kvóty (přidělené místo v úložišti). Také přiřazuje uživatelům práva správce organizační jednotky, které se vždy vztahuje ke konkrétní organizační jednotce. Dále může monitorovat celkový stav využití místa v datovém úložišti, využití místa v rámci kvót organizačních jednotek, a využití místa jednotlivými uživateli.

Kromě nových případů užití zde také figuruje případ užití UC10 – **Zobrazit seznam uživatelů** z kapitoly 2.3.2 Případy užití uživatelských účtů. Ten je využíván (includován) hned několika případy užití v této kapitole.



Obr. 2.4: Diagram případů užití pro org. jednotky

2.3.3.1 Mapování na funkční požadavky

Pro kontrolu splnění všech postihovaných funkčních požadavků slouží tabulka 2.2: Splnění funkčních požadavků případy užití pro org. jednotky. Ta zachycuje vztahy mezi jednotlivými funkčními požadavky z kapitoly 2.2.1.2 Funkční požadavky na organizační jednotky a konkrétními případy užití, které jsou vyobrazeny na obrázku 2.4: Diagram případů užití pro org. jednotky a dále rozvíjeny ve scénářích případů užití v následujících podkapitolách.

Případy užití	Požadavky										
	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21
UC11	↑										
UC12		↑									
UC13			↑		↑						
UC14				↑		↑					
UC15							↑				
UC16								↑			
UC17									↑		
UC18										↑	
UC19											↑

Tabulka 2.2: Splnění funkčních požadavků případy užití pro org. jednotky

Jak je z tabulky patrné, funkční požadavky **F11** až **F21** jsou splněny případy užití **UC11** až **UC19**. Případ **UC13 – Zobrazit využití kvóty org. jednotky** plní požadavky F13 a F15, jelikož se jedná o požadavky na stejnou funkcionalitu, která má ovšem být umožněna dvěma různými aktéry (správci organizační jednotky a hlavnímu správci). Ze stejného důvodu plní **UC14 – Zobrazit využití místa uživatelem v org. jednotce** požadavky F14 a F16.

2.3.3.2 Případ užití UC11 – přidání uživatele do org. jednotky

Tento případ užití zachycuje proces přidání uživatele do konkrétní organizační jednotky. Přidávat uživatele může pouze **správce organizační jednotky**, a to pouze do těch jednotek, pro které má nastaveno oprávnění správce.

Hlavní scénář:

1. Případ užití začíná, když se správce organizační jednotky rozhodne přidat uživatele do jednotky.
2. Include (UC20 – Zobrazit detail org. jednotky).
3. Správce org. jednotky na detailu jednotky klikne na tlačítko pro přidání uživatele.
4. Include (UC10 – Zobrazit seznam uživatelů).
5. Správce org. jednotky vybere požadovaného uživatele.
6. Integrovaný portál přidá vybraného uživatele do vybrané organizační jednotky a zobrazí detail jednotky s aktualizovaným seznamem členů a zprávu o úspěšném přidání.

2.3.3.3 Případ užití UC12 – odebrání uživatele z org. jednotky

Tento případ užití zachycuje proces odebrání uživatele z organizační jednotky. Odebírat uživatele může pouze **správce organizační jednotky**, a to pouze z těch jednotek, pro které má nastaveno oprávnění správce.

Hlavní scénář:

1. Případ užití začíná, když se správce organizační jednotky rozhodne odebrat uživatele z jednotky.
2. Include (UC20 – Zobrazit detail org. jednotky).
3. Správce organizační jednotky vybere v seznamu členů na detailu vybrané jednotky toho uživatele, kterého si přeje odebrat.
4. Integrovaný portál odebere vybraného uživatele z vybrané jednotky a zobrazí detail jednotky s aktualizovaným seznamem členů a zprávu o úspěšném odebrání.

2.3.3.4 Případ užití UC13 – zobrazení využití kvóty org. jednotky

Tento případ užití zachycuje proces pro zobrazení obsazeného, dostupného a celkového volného místa v rámci kvóty organizační jednotky. Tyto informace si může zobrazit pouze **správce organizační jednotky** u jednotek, pro které má oprávnění správce, a **hlavní správce**.

Hlavní scénář:

1. Případ užití začíná, když se správce rozhodne zobrazit si využití kvóty organizační jednotky.
2. Include (UC20 – Zobrazit detail org. jednotky).
3. Správce na detailu organizační jednotky klikne na tlačítko pro zobrazení statistik využití kvóty.
4. Integrovaný portál zobrazí obsazené, dostupné a celkové volné místo v rámci kvóty jako koláčový graf.

2.3.3.5 Případ užití UC14 – zobrazení využití místa uživatelem v org. jednotce

Tento případ užití zachycuje proces pro zobrazení využitého místa uživatelem v organizační jednotce. Tyto informace si může zobrazit pouze **správce organizační jednotky** u uživatelů v jednotce, pro kterou má oprávnění správce, a **hlavní správce**.

Hlavní scénář:

1. Případ užití začíná, když se správce rozhodne zobrazit si využití místa některým uživatelem v rámci organizační jednotky.
2. Include (UC10 – Zobrazit seznam uživatelů).
3. Správce vybere požadovaného uživatele.
4. Integrovaný portál zobrazí detail uživatele.
5. Správce na detailu uživatele klikne na tlačítko pro zobrazení statistik využití úložného prostoru.
6. Integrovaný portál zobrazí uživatelem obsazené místo v rámci všech organizačních jednotek, ve kterých je členem, a pro které má správce oprávnění.

2.3.3.6 Případ užití UC15 – zobrazení využití místa dat. úložiště

Tento případ užití zachycuje proces pro zobrazení využitého místa proces pro zobrazení obsazeného, dostupného a celkového volného místa v rámci datového úložiště. V základu jsou datová úložiště dvě: jedno pro sdílení souborů, a jedno pro archivaci a zálohování. Tyto informace si může zobrazit pouze **hlavní správce**.

Hlavní scénář:

1. Případ užití začíná, když se hlavní správce rozhodne zobrazit si statistiku využití místa datového úložiště.
2. Integrovaný portál zobrazí seznam datových úložišť.
3. Hlavní správce klikne na tlačítko pro zobrazení statistiky u datového úložiště, pro které si chce prohlížet využití místa.
4. Integrovaný portál zobrazí obsazené, dostupné a celkové volné místo v rámci datového úložiště jako koláčový graf.

2.3.3.7 Případ užití UC16 – přiřazení práv správce org. jednotky

Tento případ užití zachycuje proces pro přiřazení práv správce organizační jednotky uživateli. Toto oprávnění se vždy vztahuje ke konkrétní organizační jednotce, a může jej přiřazovat pouze **hlavní správce**.

KAPITOLA 2 ANALÝZA A NÁVRH ŘEŠENÍ

Hlavní scénář:

1. Příklad užití začíná, když se hlavní správce rozhodne přiřadit uživateli práva správce organizační jednotky.
2. Include (UC20 – Zobrazit detail org. jednotky).
3. Hlavní správce na detailu organizační jednotky klikne na tlačítko pro přidání správce této jednotky.
4. Include (UC10 – Zobrazit seznam uživatelů).
5. Hlavní správce vybere požadovaného uživatele.
6. Integrovaný portál přiřadí vybranému uživateli práva správce vybrané organizační jednotky a zobrazí detail jednotky s aktualizovaným seznamem správců a zprávu o úspěšném přiřazení správce.

2.3.3.8 Příklad užití UC17 – vytvoření org. jednotky

Tento případ užití zachycuje proces vytvoření nové organizační jednotky. Vytvářet organizační jednotky může pouze **hlavní správce**.

Hlavní scénář:

1. Příklad užití začíná, když se hlavní správce rozhodne vytvořit novou organizační jednotku.
2. Integrovaný portál zobrazí formulář umožňující zadat identifikační číslo organizační jednotky a její jméno.
3. Hlavní správce vyplní povinné údaje identifikační číslo a jméno jednotky, a potvrdí zadání.
4. Integrovaný portál zkontroluje, zda organizační jednotka s daným identifikačním číslem nebo jménem již neexistuje. Integrovaný portál vytvoří novou jednotku a zobrazí aktualizovaný seznam jednotek a zprávu o úspěšném vytvoření.

2.3.3.9 Příklad užití UC18 – smazání org. jednotky

Tento případ užití zachycuje proces smazání existující organizační jednotky. Mazat organizační jednotky může pouze **hlavní správce**.

Hlavní scénář:

1. Příklad užití začíná, když se hlavní správce rozhodne smazat organizační jednotku.
2. Include (UC20 – Zobrazit detail org. jednotky).
3. Hlavní správce na detailu organizační jednotky klikne na tlačítko pro smazání jednotky.
4. Integrovaný portál smaže vybranou jednotku a zobrazí aktualizovaný seznam organizačních jednotek a zprávu o úspěšném smazání.

2.3.3.10 Příklad užití UC19 – nastavení kvóty org. jednotky

Tento případ užití zachycuje proces nastavení kvóty, tedy přiřazeného místa v datovém úložišti, pro organizační jednotku. Nastavovat kvóty může pouze **hlavní správce**.

Hlavní scénář:

1. Příklad užití začíná, když se hlavní správce rozhodne nastavit kvótu organizační jednotce.
2. Include (UC20 – Zobrazit detail org. jednotky).
3. Hlavní správce na detailu organizační jednotky klikne na tlačítko pro nastavení kvóty.
4. Integrovaný portál zobrazí formulář umožňující zadat kvótu pro organizační jednotku na jednotlivých datových úložištích.
5. Hlavní správce vyplní kvótu a potvrdí zadání.
6. Integrovaný portál nastaví novou hodnotu kvóty a zobrazí detail organizační jednotky a zprávu o úspěšném nastavení.

2.3.3.11 Příklad užití UC20 – vybrání org. jednotky

Tento případ užití zachycuje proces vybrání uživatelského účtu ze seznamu. Tuto funkcionalitu využívají ostatní případy užití, konkrétně

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne vybrat organizační jednotku.
2. Integrovaný portál zobrazí seznam všech organizačních jednotek.
3. Uživatel kliknutím vybere požadovanou jednotku.
4. Integrovaný portál zobrazí detail vybrané organizační jednotky.

2.3.4 Případy užití domovské složky a sdílení

V této kapitole jsou rozvinuty funkční požadavky z kapitoly 2.2.1.3 Funkční požadavky na domovskou složku a sdílení, konkrétně tedy požadavky F22 až F29.

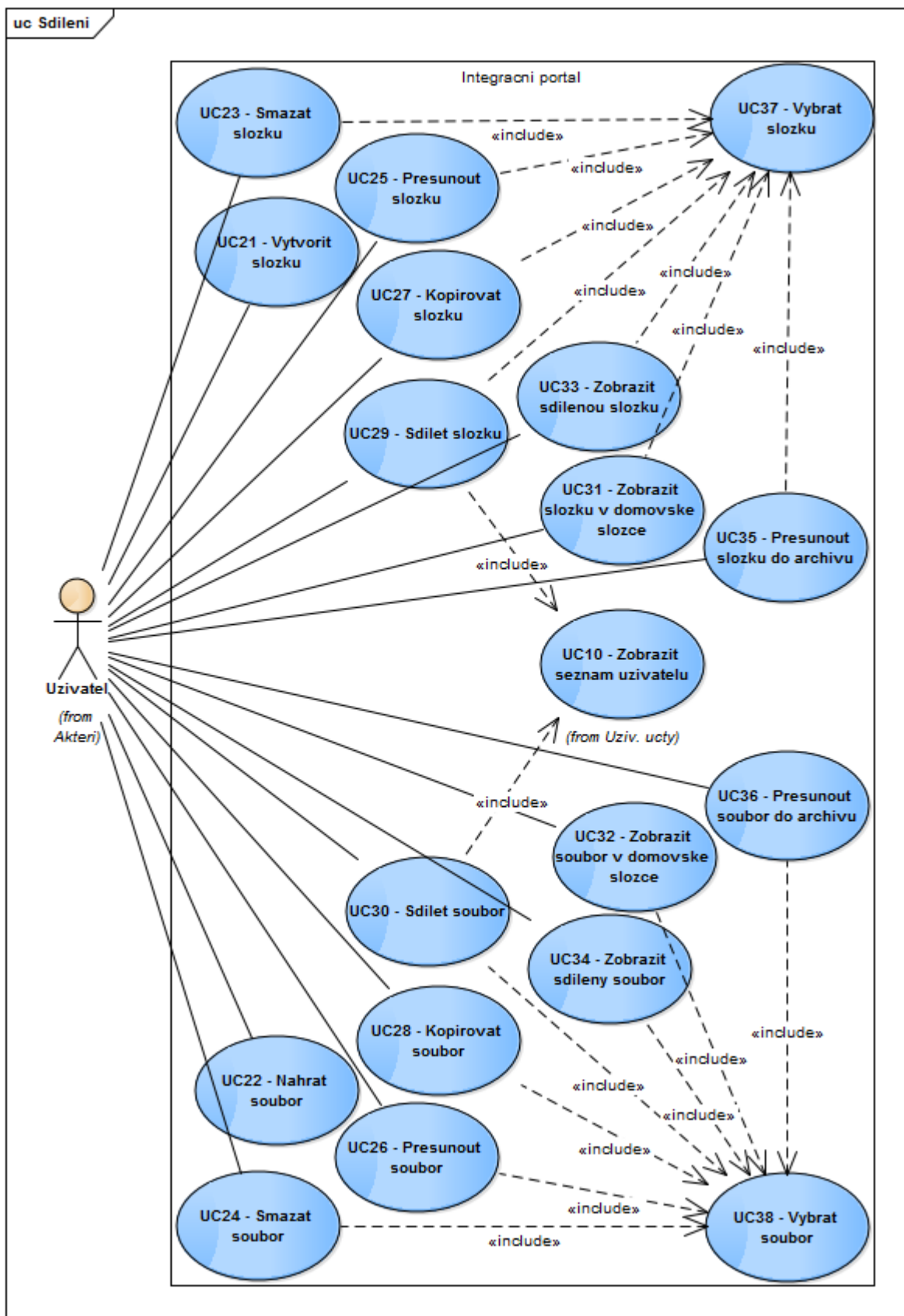
Jednotlivé případy užití, které souvisejí s domovskou složkou a sdílením, jsou vyobrazeny na obrázku 2.5: Diagram případů užití pro domovskou složku a sdílení. V tomto modulu tedy vystupuje pouze jediný aktér, a to obecný **uživatel**. Ten má k dispozici velké množství relativně jednoduchých funkcí pro práci se soubory a složkami.

Případy užití předpokládají, že na jejich začátku je uživatel na obrazovce s domovskou složkou a vidí seznam složek a souborů v ní umístěných. V souvislosti s domovskou složkou může uživatel provádět operace jako je vytváření, nahrávání, mazání, přesouvání, kopírování, prohlížení a sdílení souborů a složek. Pokud s uživatelem někdo sdílí soubor či složku, může si je uživatel pouze prohlížet.

Jelikož jsou si některé případy užití v tomto modulu značně podobné, budou v podkapitolách se scénáři zachyceny pouze některé případy užití. Například případy užití **UC25 – Přesunout složku** a **UC27 – Kopírovat složku** jsou sice dva samostatné případy s rozdílnou funkcionalitou, ovšem jejich popis je téměř totožný a bude tedy popsán pouze scénář prvního z nich.

Dále nejsou uváděny scénáře případů užití pro případy **UC37 – Vybrat složku** a **UC38 – Vybrat soubor**, jelikož se jedná o případy, které jsou totožné s odpovídajícími akcemi ve kterémkoliv správci souborů, a jsou tedy natolik triviální a obecně známé všem uživatelům počítačů, že uvádění jejich detailního scénáře nemá význam.

Kromě nových případů užití zde také figuruje případ užití **UC10 – Zobrazit seznam uživatelů** z kapitoly 2.3.2 Případy užití uživatelských účtů. Ten je využíván (includován) hned několika případy užití v této kapitole.



Obr. 2.5: Diagram případů užití pro domovskou složku a sdílení

2.3.4.1 Mapování na funkční požadavky

Pro kontrolu splnění všech postihovaných funkčních požadavků slouží tabulka 2.3: Splnění funkčních požadavků případy užití pro sdílení. Ta zachycuje vztahy mezi jednotlivými funkčními požadavky z kapitoly 2.2.1.3 Funkční požadavky na domovskou složku a sdílení a konkrétními případy užití, které jsou vyobrazeny na obrázku 2.5: Diagram případů užití pro domovskou složku a sdílení a dále rozvíjeny ve scénářích případů užití v následujících podkapitolách.

Případy užití	Požadavky							
	F22	F23	F24	F25	F26	F27	F28	F29
UC21	↑							
UC22		↑						
UC23			↑					
UC24			↑					
UC25				↑				
UC26				↑				
UC27				↑				
UC28				↑				
UC29					↑			
UC30					↑			
UC31						↑		
UC32						↑		
UC33							↑	
UC34							↑	
UC35								↑
UC36								↑

Tabulka 2.3: Splnění funkčních požadavků případy užití pro sdílení

Jak je z tabulky patrné, funkční požadavky **F22** až **F29** jsou splněny případy užití **UC21** až **UC35**. Některé požadavky jsou splněny hned několika případy užití. To je dáno tím, že zatímco funkční požadavek může být formulován obecněji aby zahrnoval složky i soubory, případně kopírování i přesouvání, případy užití již musejí být pro konkrétní akce nad konkrétním typem objektu.

2.3.4.2 Příklad užití UC21 – vytvoření složky

Tento případ užití zachycuje proces vytvoření složky v domovské složce uživatele, případně v jejích podsložkách.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne vytvořit novou složku.
2. Integrovaný portál zobrazí formulář umožňující zadat jméno složky.
3. Uživatel vyplní jméno složky a potvrdí zadání.
4. Integrovaný portál zkontroluje, jestli již neexistuje složka se stejným jménem na aktuální úrovni hierarchické struktury složek. Integrovaný portál vytvoří složku s daným jménem, zobrazí aktualizovaný seznam složek a souborů, a zobrazí zprávu o úspěšném vytvoření.

2.3.4.3 Příklad užití UC22 – nahrání souboru

Tento případ užití zachycuje proces nahrání (uploadu) souboru z počítače uživatele do jeho domovské složky v integrovaném portálu.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne nahrát soubor.
2. Integrovaný portál zobrazí formulář umožňující vybrat lokální soubor pro nahrání.
3. Uživatel klikne na tlačítko pro vybrání souboru, vybere požadovaný soubor na svém počítači, a následně klikne na tlačítko pro nahrání souboru.
4. Integrovaný portál uloží nahrávaný soubor do aktuální úrovně hierarchické struktury složek v domovské složce uživatele, zobrazí aktualizovaný seznam složek a souborů, a zobrazí zprávu o úspěšném nahrání.

2.3.4.4 Příklad užití UC23 – smazání složky

Tento případ užití zachycuje proces smazání složky. Příklad užití **UC24 – Smazat soubor** je téměř totožný a liší se pouze ve druhém kroku scénáře, kde zahrnuje **UC38 – Vybrat soubor**. Z tohoto důvodu není uváděn jeho scénář.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne smazat složku.
2. Include (UC37 – Vybrat složku).

KAPITOLA 2 ANALÝZA A NÁVRH ŘEŠENÍ

3. Uživatel klikne na tlačítko pro smazání vybrané složky.
4. Integrovaný portál smaže vybranou složku, zobrazí aktualizovaný seznam složek a souborů, a zobrazí zprávu o úspěšném smazání.

2.3.4.5 Příklad užití UC25 – přesunutí složky

Tento případ užití zachycuje proces přesunutí složky do jiné složky. Případy užití **UC26 – Přesunout soubor**, **UC27 – Kopírovat složku** a **UC28 – Kopírovat soubor** jsou opět téměř totožné a liší se pouze ve výběru souboru namísto složky, případně v kliknutí na tlačítko pro kopírování namísto přesunutí. U těchto tří případů tedy není uváděn jejich scénář.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne přesunout složku.
2. Include (UC37 – Vybrat složku) pro vybrání přesouvané složky.
3. Uživatel klikne na tlačítko pro přesunutí vybrané složky.
4. Include (UC37 – Vybrat složku) pro vybrání složky, do které se má přesunout.
5. Integrovaný portál přesune složku vybranou v druhém kroku do složky vybrané ve čtvrtém kroku, zobrazí aktualizovaný seznam složek a souborů, a zobrazí zprávu o úspěšném přesunutí.

2.3.4.6 Příklad užití UC29 – sdílení složky

Tento případ užití zachycuje proces sdílení složky v domovské složce uživatele s jiným uživatelem. Příklad užití **UC30 – Sdílet soubor** je opět téměř totožný a není tedy uváděn jeho scénář.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne sdílet složku.
2. Include (UC37 – Vybrat složku).
3. Uživatel klikne na tlačítko pro sdílení vybrané složky s dalšími uživateli.
4. Include (UC10 – Zobrazit seznam uživatelů).
5. Uživatel vybere požadovaného uživatele ze seznamu.
6. Integrovaný portál nastaví sdílení vybrané složky s vybraným uživatelem a zobrazí zprávu o úspěšném nastavení sdílení.

2.3.4.7 Příklad užití UC32 – zobrazení souboru v domovské složce

Tento případ užití zachycuje proces zobrazení souboru v domovské složce uživatele. Příklad užití **UC31 – Zobrazit složku v domovské složce** je částečně podobný, zobrazuje ovšem podsložky a soubory ve vybrané složce, opět tedy není uváděn jeho scénář.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne zobrazit si soubor ze své domovské složky.
2. Include (UC38 – Vybrat soubor).
3. Uživatel klikne na tlačítko pro zobrazení / stažení souboru.
4. Integrovaný portál vybraný soubor uživateli buď přímo zobrazí, pokud je možné jej zobrazit v prohlížeči (obrázek, textový soubor, ...), nebo jej uživateli nabídne ke stažení.

2.3.4.8 Příklad užití UC33 – zobrazení sdílené složky

Tento případ užití zachycuje proces zobrazení obsahu složky, kterou s uživatelem sdílí některý jiný uživatel. Příklad užití **UC34 – Zobrazit sdílený soubor** je téměř totožný, jen místo obsahu složky zobrazí přímo vybraný soubor jako je popsáno v kapitole 2.3.4.7 Příklad užití UC32 – zobrazení souboru v domovské složce, a není tedy uváděn jeho scénář.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne zobrazit si obsah složky, která je s ním sdílena.
2. Uživatel klikne na odkaz pro zobrazení složky s obsahem, který s ním sdílejí ostatní uživatelé.
3. Integrovaný portál zobrazí seznam všech složek a souborů, které s uživatelem sdílejí ostatní uživatelé.
4. Include (UC37 – Vybrat složku).
5. Integrovaný portál zobrazí všechny podsložky a soubory ve vybrané složce.

2.3.4.9 Příklad užití UC35 – přesunout složku do archivu

Tento případ užití zachycuje proces přesunutí složky do archivu. Příklad užití **UC36 – Přesunout soubor do archivu** je téměř totožný a není tedy uváděn jeho scénář.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne přesunout složku do archivu.
2. Include (UC37 – Vybrat složku).
3. Uživatel klikne na tlačítko pro přesunutí složky do archivu.
4. Integrační portál přesune vybranou složku do archivu, zobrazí aktualizovaný seznam složek a souborů, a zobrazí zprávu o úspěšném archivaci.

2.3.5 Případy užití archivace a zálohování

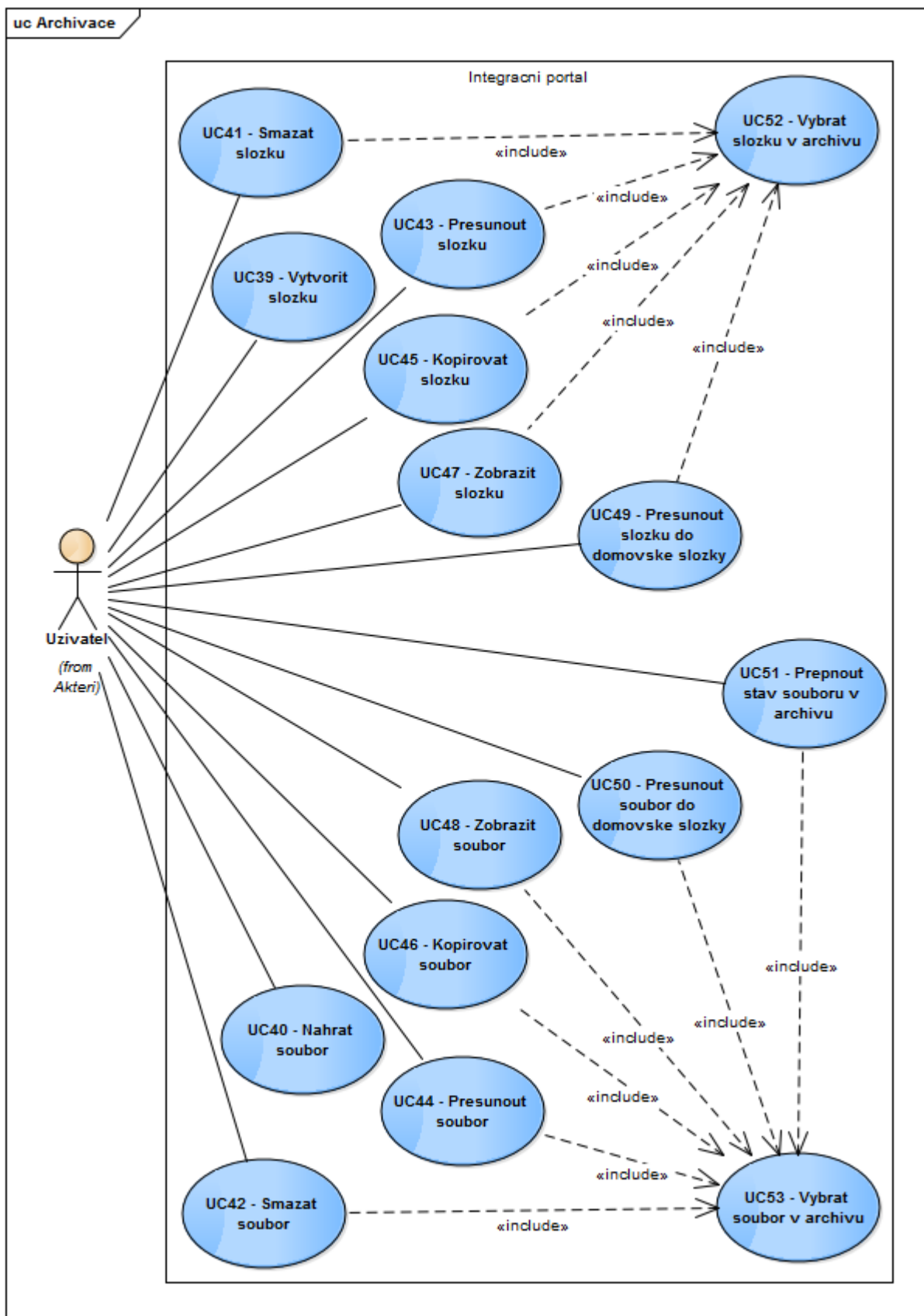
V této kapitole jsou rozvinuty funkční požadavky z kapitoly 2.2.1.4 Funkční požadavky na archivaci a zálohování, konkrétně tedy požadavky F30 až F36.

Jednotlivé případy užití, které souvisejí s archivací a zálohováním, jsou vyobrazeny na obrázku 2.6: Diagram případů užití pro archiv a zálohování. V tomto modulu tedy vystupuje pouze jediný aktér, a to obecný **uživatel**.

Případy užití předpokládají, že na jejich začátku je uživatel na obrazovce s archivem a vidí seznam složek a souborů v něm umístěných. V archivu může uživatel provádět operace vytváření, nahrávání, mazání, přesouvání, kopírování a prohlížení souborů a složek. Navíc může u souborů měnit jejich stav z „online“ na „offline“ a zpět, což umožňuje kontrolu nad druhem média, na kterém je soubor uložen: rychlé nízkokapacitní online médium, nebo pomalé offline médium s velkou kapacitou.

Jelikož je velká část případů užití totožná s případy užití z kapitoly 2.3.4 Případy užití domovské složky a sdílení, nejsou pro tyto případy uváděny jejich scénáře. Konkrétně se jedná o vytváření, nahrávání, mazání, přesouvání, kopírování a prohlížení souborů a složek. Ty se od stejných případů užití v kapitole o domovské složce a sdílení liší pouze tím, že jsou prováděny na obrazovce archivu.

Jediný případ užití, kterým se archiv liší od domovské složky, je tedy změna stavu souboru. Scénář pro tento případ je uveden dále v této kapitole.



Obr. 2.6: Diagram případů užití pro archiv a zálohování

KAPITOLA 2 ANALÝZA A NÁVRH ŘEŠENÍ

2.3.5.1 Mapování na funkční požadavky

Pro kontrolu splnění všech postihovaných funkčních požadavků slouží tabulka 2.4: Splnění funkčních požadavků případy užití pro archiv. Ta zachycuje vztahy mezi jednotlivými funkčními požadavky z kapitoly 2.2.1.4 Funkční požadavky na archivaci a zálohování a konkrétními případy užití, které jsou vyobrazeny na obrázku 2.6: Diagram případů užití pro archiv a zálohování.

Případy užití	Požadavky						
	F30	F31	F32	F33	F34	F35	F36
UC39	↑						
UC40		↑					
UC41			↑				
UC42			↑				
UC43				↑			
UC44				↑			
UC45				↑			
UC46				↑			
UC47					↑		
UC48					↑		
UC49						↑	
UC50						↑	
UC51							↑

Tabulka 2.4: Splnění funkčních požadavků případy užití pro archiv

Jak je z tabulky patrné, funkční požadavky **F30** až **F36** jsou splněny případy užití **UC39** až **UC51**. Požadavky, které obecněji zahrnují složky i soubory, případně kopírování i přesouvání, jsou opět splněny více případy užití.

2.3.5.2 Příklad užití UC51 – přepnutí stavu souboru v archivu

Tento případ užití zachycuje proces přepnutí stavu souboru v archivu z „online“ na „offline“ a naopak. Jedná se o jediný případ užití, který je specifický pro archiv.

Hlavní scénář:

1. Příklad užití začíná, když se uživatel rozhodne přepnout stav souboru.
2. Include (UC53 – Vybrat soubor v archivu).
3. Uživatel klikne na tlačítko, které reprezentuje aktuální stav souboru, a které slouží pro jeho změnu.
4. Integrační portál změní stav souboru, zobrazí aktualizované tlačítko reprezentující stav souboru, a dále zobrazí zprávu o úspěšné změně.

Tlačítko ve třetím kroku slouží jako přepínač stavu. Pokud je soubor v „online“ stavu, pak toto tlačítko ukazuje tento stav a kliknutí na něj soubor přepne do „offline“ stavu. Přepnutí stavu však může trvat i několik minut, a tlačítko bude během této doby neaktivní.

2.4 Integrované služby

V této kapitole jsou popsány všechny služby, které bude integrační portál využívat, a to včetně těch, které budou do systému integrovány až v navazující práci. Jedná se tedy jak o služby systémů Fakulty elektrotechnické na ČVUT, tak o služby datových úložišť.

2.4.1 FELid

Jedná se o globální autentizační a autorizační systém pro aplikace určené pro akademické pracovníky a studenty Fakulty elektrotechnické na ČVUT.

Tento systém podporuje tzv. **single sign-on**, tedy jednorázové přihlášení uživatelů. Díky tomu se uživatel přihlásí pouze jednou, a to při prvním přístupu k některé z aplikací, které využívají FELid. Následně se při jakýchkoliv dalších přístupech k libovolným aplikacím využívajícím FELid již přihlašovat nemusí, je totiž přihlášen automaticky. [14]

Uživatel navíc své přihlašovací údaje nevyplňuje v samotných aplikacích, nýbrž na speciální centrální přihlašovací stránce. Aplikacím následně nejsou odeslány přihlašovací údaje, ale pouze informace o úspěšném přihlášení uživatele. Přihlašování přes FELid je tedy nejen příjemnější (odpadá nutnost přihlašovat se do každé aplikace zvlášť), ale také bezpečnější.

2.4.2 KOSapi

KOSapi je systém, který umožňuje aplikacím Fakulty elektrotechnické přístup k vybraným částem dat v databázi KOS. Mezi dostupná data patří informace o studentech, rozvrzích předmětů, obsah bílé knihy, apod. [15] Přístup k těmto datům je zprostředkován prostřednictvím RESTful webových služeb [18]. Pro více informací o RESTful webových službách a o REST viz. kapitola 3.2.1 REST.

Z KOSapi bude integrační portál mimo jiné získávat seznam organizačních jednotek [35] na Fakultě elektrotechnické, jež odpovídají jednotlivým katedrám. Každý uživatel integračního portálu bude patřit do některé z těchto organizačních jednotek.

Organizační jednotky budou mít přiděleny konkrétní kvóty pro dostupný úložný prostor v rámci integračního portálu. Tyto kvóty se budou vztahovat na soubory všech uživatelů v dané organizační jednotce.

2.4.3 CESNET

Sdružení CESNET založily vysoké školy a Akademie věd České republiky v roce 1996. Jeho hlavním cílem je výzkum a vývoj informačních a komunikačních technologií, budování a rozvoj e-infrastruktury CESNET určené pro výzkum a vzdělávání. CESNET nabízí řadu služeb jak pro připojené instituce i jejich uživatele. [1]

Integrační portál bude zajišťovat přístup k datovému úložišti, které bude umístěno na serverech CESNETu. Ten umožňuje nahrávání a stahování uložených dat prostřednictvím **SFTP** (Secure File Transfer Protocol [20]) protokolu, a také umožňuje připojení přes **SSH** (Secure Shell [19]) pro správu souborů prostřednictvím příkazové řádky (bash). Kromě místa pro ukládání dat poskytuje CESNET navíc možnost buď automatické, nebo manuální archivace uložených souborů na magnetické pásky. K tomu používá tzv. **hierarchický model ukládání dat (HSM – Hierarchical Storage Management)**.

2.4.3.1 Hierarchický model ukládání dat

Hierarchický model ukládání dat (**HSM – Hierarchical Storage Management** [22]), umožňuje udržovat často používané soubory na rychlých „online“ médiích, která mají menší kapacitu. Naproti tomu zřídka používaná data se mohou odkládat pro archivaci na „offline“ média, která mají daleko větší kapacitu, avšak jsou znatelně pomalejší. [21]

V případě CESNETu se jako online média používají rychlé SATA disky, zatímco pro offline archivaci jsou použity buď magnetické pásky, nebo vypínatelné disky (**MAID** – Massive Array of Idle Disks [23]).

Proces **migrace**, tedy přesun dat z online na offline média a zpět, je v základu plně zautomatizován. Dlouho nepoužívané soubory jsou tedy s ubývajícím místem na online médiích přesouvány na offline média. Naopak soubory na offline médiích jsou přesunuty zpět na online média ve chvíli, kdy se s nimi někdo pokusí pracovat.

Uživatel tedy vůbec nemusí vědět o tom, že jsou použity dva druhy úložišť. Zároveň je však možné tento proces řídit ručně, neboť migrace větších souborů může představovat zpomalení práce. Ručně je možné soubory migrovat prostřednictvím příkazů příkazové řádky (je tedy nutné být k serveru připojen prostřednictvím **SSH**).

2.4.3.1.1 Příkazy HSM

Pro odeslání (zmigrování) souboru z online média na offline médium je určen speciální příkaz **dmput -r [soubor]**. Nepovinný přepínač „-r“ zde značí, že se má soubor po úspěšném přesunutí na offline médium smazat z online média.

Pro přesun souboru z offline média zpět na online médium slouží příkaz **dmget [soubor]**. V takovémto případě soubor vždy zůstává uložen na offline médiu pro případ, že by se soubor v budoucnu opět migroval na offline médium. Zároveň má offline médium dostatek dostupného úložného prostoru a není tedy důvod z něj soubor odstraňovat.

Pro vypsání informací (metadat) o souboru je určen příkaz **dmis -l [soubor]**. Jedná se o rozšířenou verzi unixového příkazu `ls -l`, která do tabulky s výpisem metadat o souboru přidává před jeho název další sloupec s reprezentací stavu archivace. Jedná se o třípísmennou zkratku, která je vždy uvedena v závorkách. Seznam všech zkratk pro stav archivace je uveden v kapitole 2.4.3.1.2 Seznam stavů archivace. Výstup příkazu může vypadat například následovně:

```
-rw----- 1 user storage 304 2012-06-22 14:17 (DUL) soubor.txt
```

HSM také umožňuje vyhledávat soubory podle konkrétního stavu archivace. K tomu slouží příkaz **dmfind [adresář] -state [stav]**. Příkaz prohledá adresář, který byl uveden jako první parametr, a vypíše seznam všech souborů, jejich stav archivace odpovídá parametru uvedenému za přepínačem „state“. Tento parametr musí být jednou z třípísmenných zkratk, které jsou uvedeny v kapitole 2.4.3.1.2 Seznam stavů archivace.

2.4.3.1.2 Seznam stavů archivace

HSM definuje několik standardních možných stavů archivace, ve kterých se soubor může nacházet. Každý takovýto stav je reprezentován třípísmennou zkratkou, a to jednou z následujících [9]:

- REG: soubor není spravován systémem DMF (je pouze na online médiu),
- MIG: soubor je právě migrován na offline médium,
- ARC: soubor je právě archivován,
- DUL: soubor je v duálním stavu (na online i offline médiu),
- OFL: soubor je pouze na offline médiu,
- UNM: soubor je právě migrován zpět na online médium,
- NMG: nemigrovatelný soubor,
- PAR: soubor částečně zmigrován,
- N/A: u souboru se nepodařilo zjistit DMF stav,
- INV: soubor je v konfliktním/nevalidním stavu.

V základu je soubor ihned po vytvoření uložen pouze na online médiu a je tedy ve stavu **REG**. Během migrování na offline médium se může nacházet ve stavu **MIG**, ze kterého následně přejde do stavu **OFL**, pokud byl použit přepínač „-r“, nebo do stavu **DUL**.

2.4.4 Alfresco

Alfresco je open source **ECM** systém (Enterprise Content Management [24]), tedy systém pro vytváření, správu, zabezpečení, prohlížení a distribuci dat. Jedná se o ucelené řešení správy dokumentů, které přímo poskytuje většinu potřebné funkcionality.

Jednou z nejdůležitějších funkcí Alfresca je zabezpečení kontroly přístupu k souborům. K tomu využívá **ACL** (Access Control List). ACL představuje definovaný seznam uživatelů a jejich pravomocí pro provádění konkrétních operací s daným souborem či složkou [11]. V Alfrescu má každý soubor i složka přidělen svůj vlastní ACL seznam, a zároveň dědí ACL seznam nadřazené složky.

Alfresco také umožňuje jednoduché **verzování souborů**. Soubor má po vytvoření nastaveno číslo verze na 1 a při každé změně obsahu se číslo verze inkrementálně navýší. Uživatel si následně může zobrazit kteroukoli verzi souboru.

Alfresco přímo podporuje **CMIS** (Content Management Interoperability Services [3]). Díky tomu je možné se k Alfresco repozitáři připojit prostřednictvím standardizovaných knihoven bez nutnosti implementace samotné komunikace. Jedním z příkladů takovýchto knihoven je například OpenCMIS, o kterém pojednává kapitola 3.2.5 OpenCMIS.

Pro přímější přístup k Alfresco repozitáři mohou uživatelé využít vestavěnou webovou aplikaci pro prohlížení a správu obsahu Alfresco Share [43]. Alfresco také umožňuje snadnou integraci s nástroji z balíku Microsoft Office.

Jak je z předchozího výčtu patrné, Alfresco umožňuje přistupovat k uloženému obsahu prostřednictvím mnoha různých klientů. Bylo by tedy vhodné, aby integrační portál uživatele neomezoval, a umožňoval (alespoň v budoucnu) využít pro práci se soubory libovolné jiné programy. Zároveň je však nutné, aby data uložená na Alfrescu měla určitou strukturu.

Oba zmíněné požadavky je možné splnit tím, že se uživatel nejprve musí přihlásit prostřednictvím integračního portálu. Při tomto prvním přihlášení integrační portál vytvoří pro uživatele potřebnou hierarchickou strukturu složek. V rámci takto vytvořené hierarchické struktury může uživatel následně pracovat se soubory a složkami prostřednictvím integračního portálu, či může přímo upravovat jejich obsah např. prostřednictvím programu Microsoft Word. Pro pokročilejší operace, jako je sdílení, přesouvání do archivu apod. však uživatel bude muset použít integrační portál.

Další výhodou Alfresca je, že ČVUT již má zprovozněný Alfresco repozitář, a to přinejmenším na Fakultě elektrotechnické [10]. Pro potřeby integračního portálu by tedy bylo možné během vývoje využít tento již existující Alfresco repozitář bez nutnosti vytvářet nový. Pro výsledný systém integračního portálu však bude vytvořen nový repozitář, který bude využívat úložiště CESNETu.

Kapitola 3 Implementace

V této kapitole je popsán výsledek implementace integračního portálu v rámci této diplomové práce. V jednotlivých podkapitolách následuje seznam všech funkční požadavků, které byly implementovány, dále seznam nejpodstatnějších technologií a knihoven použitých při implementaci, a nakonec popis jednotlivých komponent integračního portálu.

3.1 Implementované požadavky

Jak již bylo uvedeno v kapitole 2.2 Popis požadavků na integrační portál, kvůli rozsahu práce byly implementovány pouze podstatné požadavky související s administrací, domovskou složkou, sdílením a archivací.

V rámci modulu uživatelských účtů z kapitoly 2.2.1.1 Funkční požadavky na uživatelské účty byly implementovány požadavky:

- F2 – přihlašování externistů,
- F3 – vytváření uživatelských účtů pro externisty,
- F5 – vytváření uživatelských rolí,
- F7 – přiřazování uživatelských rolí uživatelům,
- F8 – odebrání uživatelských rolí uživatelům,
- F9 – zobrazení všech uživatelských rolí,
- F10 – zobrazení všech uživatelských účtů.

KAPITOLA 3 IMPLEMENTACE

V rámci modulu organizačních jednotek z kapitoly 2.2.1.2 Funkční požadavky na organizační jednotky musely být organizační jednotky pouze nasimulovány ukázkovými daty, jelikož nebyla implementována integrace s KOSapi. Byly implementovány požadavky:

- F11 – přidávání uživatelů do organizační jednotky,
- F12 – odebírání uživatelů z organizační jednotky,
- F18 – přiřazování práv správce organizační jednotky.

V rámci modulu domovské složky a sdílení z kapitoly 2.2.1.3 Funkční požadavky na domovskou složku a sdílení byly implementovány požadavky:

- F22 – vytváření složek a podsložek v domovské složce,
- F23 – nahrávání souborů do domovské složky,
- F24 – mazání složek a souborů v domovské složce,
- F26 – sdílení souborů s uživateli – implementováno tedy jen částečně, jelikož chybí podpora pro sdílení složek,
- F27 – prohlížení složek a souborů v domovské složce,
- F28 – prohlížení složek a souborů sdílených ostatními uživateli.

V rámci modulu archivace a zálohování z kapitoly 2.2.1.4 Funkční požadavky na archivaci a zálohování byly implementovány požadavky:

- F30 – vytváření složek a podsložek v archivu,
- F31 – nahrávání souborů do archivu,
- F32 – mazání složek a souborů v archivu,
- F34 – prohlížení složek a souborů v archivu,
- F36 – přepnutí stavu souboru v archivu (online/offline).

3.2 Použité technologie, knihovny a API

V této kapitole je seznam těch nejpodstatnějších technologií, knihoven a aplikačních rozhraní, které byly při implementaci integračního portálu použity.

3.2.1 REST

REST, tedy Representational State Transfer, je **architektonický styl** pro komunikaci, který není orientován na volání procedur, ale na data jako taková a na přenos jejich reprezentací. **RESTful** aplikační rozhraní je takové rozhraní, které dodržuje všechny zásady a principy, které definuje REST. [18]

Integrační portál poskytuje RESTful aplikační rozhraní, ke kterému se mohou připojovat jednotlivé klientské aplikace a využívat jeho služeb. Konkrétně toto rozhraní využívá grafický klient vytvořený v rámci diplomové práce Kateřiny Hašlarové.

3.2.2 Maven

Apache Maven je nástroj pro správu, řízení a automatizaci kompilací aplikací [7] a souvisejících akcí, jako je například spouštění testů. Zároveň usnadňuje správu dependencí projektu (knihoven, které projekt používá). Díky tomu není nutné každou knihovnu ručně stahovat a kopírovat do projektu, nýbrž stačí její uvedení v seznamu dependencí, a Maven se již postará o její stažení a začlenění do projektu.

Integrační portál využívá maven pro kompilování, a také pro nasazení (deploy) na **Tomcat** server (open source webový server pro Java aplikace [36]) prostřednictvím pluginu zvaného **Apache Tomcat Maven Plugin** [37]. Zkompilování a nasazení integračního portálu na spuštěný Tomcat server je tedy velmi jednoduché a stačí na něj jediný příkaz:

```
mvn clean tomcat:deploy
```

3.2.3 Spring

Spring je open-source aplikační framework (sada knihoven, podpůrných nástrojů apod.) pro vývoj Java aplikací [4]. Celý framework je rozdělen do mnoha modulů podle funkcionality, a díky tomu tedy může programátor do svého projektu přidat pouze ty moduly, které bude skutečně využívat.

Spring umožňuje konfiguraci prostřednictvím xml konfiguračních souborů, anotací, nebo kombinaci obou. Integrační portál využívá konfigurace pomocí anotací pro nastavení, která mají zůstat neměnná (např. řízení transakce nad jednotlivými metodami). Nastavení, která by správce portálu mohl chtít změnit, jsou uložena v xml souborech. Většinou by však správci mělo stačit změnit hodnoty v souboru `application.properties`, kde jsou uloženy URL a přihlašovací údaje pro databázový server, CESNET server a Alfresco repozitář, apod.

3.2.3.1 Spring Core

Spring Core je základní a hlavní modul, který musí jako jediný být do projektu vždy zahrnut, aby bylo možné používat jakékoliv jiné Spring moduly. Jeho hlavní poskytovanou funkcionalitou je **Inversion of Control**.

Inversion of Control je návrhový vzor, který umožňuje uvolnit vztahy mezi komponentami projektu, a to prostřednictvím **vkládání závislostí** (dependency injection [38]). Pokud třída využívá jiné třídy, za normálních okolností by si udržovala přímou referenci. Inversion of Control místo toho třídě dynamicky dodá odpovídající objekt [6].

3.2.3.2 Spring Web a Spring WebMVC

Jedná se o moduly **Spring Web** a **Spring WebMVC**, které usnadňují psaní webových aplikací a zprostředkovávají funkce, které jsou v takovýchto aplikacích často používané. Obsahují implementaci MVC návrhového vzoru (Model-View-Controller [39]), usnadňují nahrávání souborů a další. Také se starají o automatickou inicializaci všech kontejnerů potřebných pro správnou funkci kontrolérů.

3.2.3.3 Spring Data Access

Spring usnadňuje přístup a práci s databázemi prostřednictvím modulů **Spring JDBC**, **Spring ORM** a **Spring TX**.

Spring JDBC přidává vrstvu abstrakce mezi aplikací a databází. Ta se stará o otevírání, udržování a zavírání spojení s databází. Dovede připravit a odeslat požadavek na databázi (tzv. statement) a následně zpracovat odpověď a postarat se o případné zpracování chyb.

Spring ORM představuje integrační vrstvu pro různé populární API pro objektově-relační mapování (**ORM** – konverze dat z relační databáze na objekty [40]). Hlavními podporovanými ORM API jsou **JPA**, **JDO** a **Hibernate**.

Spring TX přidává podporu pro deklarativní řízení transakcí. Díky tomu je možné jednoduše definovat transakční chování jednotlivých metod pomocí anotací.

3.2.3.4 Spring Security

Jedná se o sbírku modulů, starajících se o bezpečnou autentizaci a autorizaci uživatelů. Automaticky chrání před různými známými typy útoků jako je například **útok fixací na sezení** (session fixation [41]), **CSRF** (cross-site request forgery [42]) a jiné.

Vše je integrováno s dalšími Spring moduly, je tedy možné přidat kontrolu autentizace či autorizace na jednotlivá URL v rámci aplikace, případně reprezentace URL ve tvaru regulárního výrazu, či přímo přidat kontrolu na konkrétní metody tříd.

Spring Security podporuje celou řadu způsobů pro autentizaci uživatele. Od přihlášení prostřednictvím přihlašovacího formuláře až po pokročilé způsoby jako je OAuth verze 1 i 2. Více informací o protokolu OAuth je v kapitole 3.2.4 OAuth.

3.2.4 OAuth

OAuth je protokol pro bezpečnou autentizaci a autorizaci uživatelů, přihlašujících se prostřednictvím webových, mobilních a desktopových aplikací přes HTTP [8].

Klientská aplikace nejprve umožní uživateli přihlášení proti autentizačnímu serveru, který následně klientské aplikaci vygeneruje tzv. **access token**. Jedná se o náhodný sled alfanumerických znaků, který může klientská aplikace používat pro přístup k serverovým službám.

OAuth umožňuje několik způsobů přihlášení pro získání access tokenu, jelikož však integrační portál zastává roli jak autentizačního serveru, tak roli cílové serveru se službami, byl jako způsob přihlášení vybrán tzv. Username-Password Flow způsob, kdy se přímo odesílají přihlašovací údaje uživatele v HTTP požadavku [28].

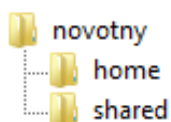
3.2.5 OpenCMIS

OpenCMIS je open source kolekce Java knihoven a nástrojů, které poskytují jednoduchého klienta pro přístup k repozitářům, které podporují CMIS standard.

V rámci integračního portálu je OpenCMIS použito pro přístup k **Alfresco** repozitářům. Jak již bylo zmíněno v kapitole 2.4.4 Alfresco, je pro kompatibilitu s externími klienty nutné, aby integrační portál vytvořil pro uživatele hierarchickou strukturu složek, se kterými může uživatel následně pracovat.

3.2.5.1 Hierarchická struktura složek v Alfrescu

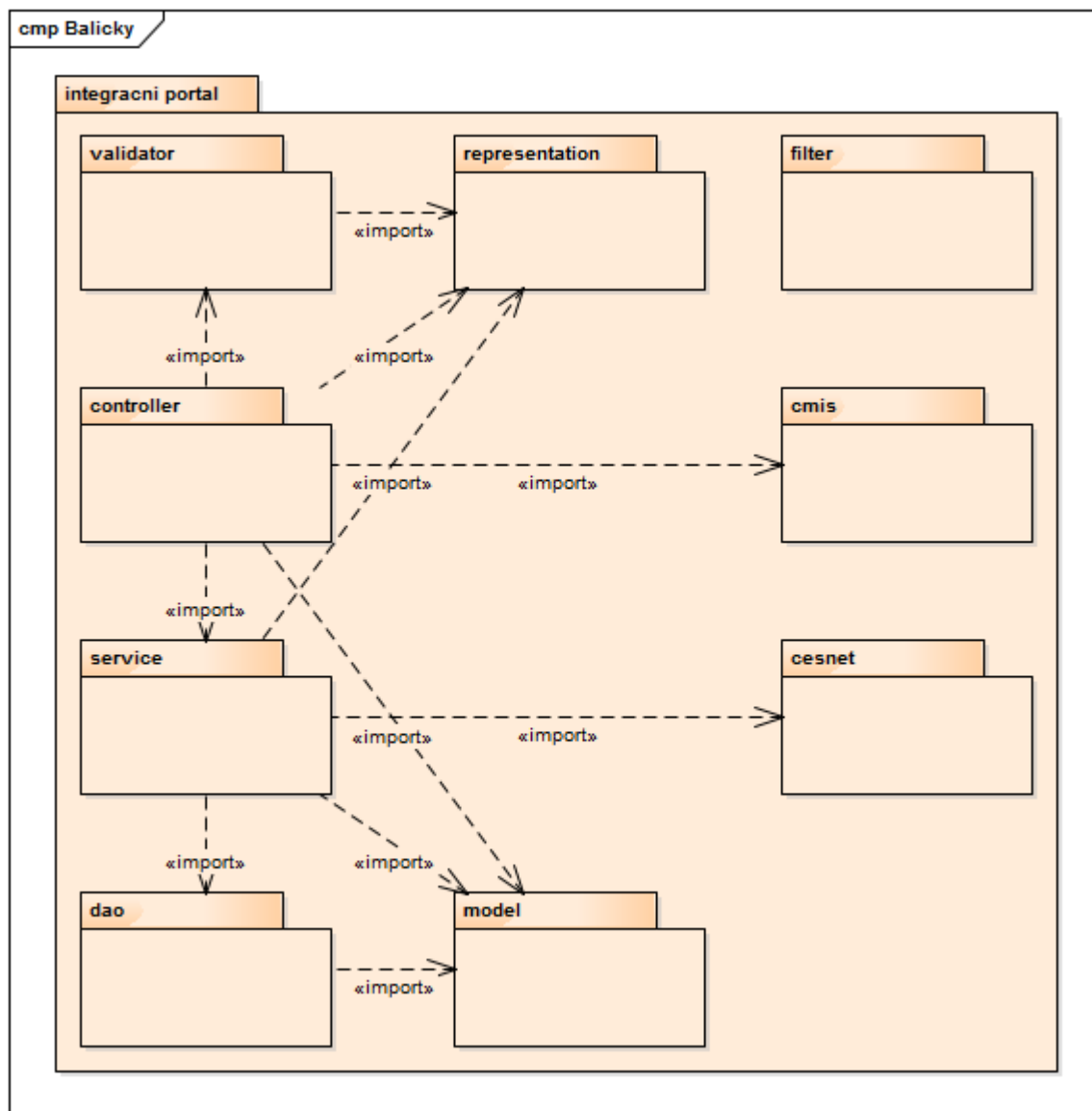
Předně systém vytvoří uživatelskou složku, pojmenovanou podle uživatelského jména. Následně vytvoří podsložky „home“ a „shared“, jako je vyobrazeno na obr. 3.1: Hierarchie uživatelské složky v Alfrescu. Zároveň integrační portál nastaví uživateli pro tyto složky odpovídající oprávnění tak, aby do složky „home“ mohl zapisovat a mohl ji používat jako svou domovskou složku. Naproti tomu ve složce „shared“ bude moci pouze číst, neboť bude obsahovat soubory, které s uživatelem sdílejí ostatní uživatelé.



Obr. 3.1: Hierarchie uživatelské složky v Alfrescu.

3.3 Balíčky integračního portálu

Zdrojový kód integračního portálu je rozdělen do řady balíčků (package) podle jejich účelu. V této kapitole jsou popsány jejich významy a vzájemná provázanost. Závislosti mezi těmito balíčky zachycuje obrázek 3.2: Diagram balíčků integračního portálu.



Obr. 3.2: Diagram balíčků integračního portálu

3.3.1 Balíček controller

Balíček **controller** obsahuje všechny **kontroléry**, které přijímají HTTP požadavky od klientských aplikací. Příchozí požadavky Spring předává podle jejich URL a parametrů odpovídajícím metodám kontrolérů.

U příchozího požadavku kontrolér nejprve převede jeho parametry na objekty reprezentací dat z balíčku **representation**. O tento převod se stará automaticky Spring, kdy stačí, aby byl v hlavičce metody kontroléru uveden jako parametr objekt třídy reprezentace s `@RequestBody`. Hlavička metody může tedy vypadat například takto:

```
... createUser(@RequestBody UserDetailsRepresentation user...)
```

Následně kontrolér tyto reprezentace předá ke zkontrolování odpovídajícímu validátoru z balíčku **validator**, a případně navrátí chybovou hlášku, pokud reprezentace neobsahuje platná data. Po validaci kontrolér vyvolá odpovídající metodu některé ze služeb z balíčku **service** a převede případnou návratovou hodnotu takovéto metody na objekt reprezentace dat, který následně odešle jako odpověď zpět klientské aplikaci.

3.3.2 Balíček validator

Tento balíček obsahuje **validátory**, které používají kontroléry pro kontrolu správnosti dat v parametrech HTTP požadavků. Ve většině případů není samostatný validátor potřebný, neboť Spring automaticky provádí základní kontrolu těchto dat, jako například kontrolu vyplnění jednotlivých proměnných v objektu reprezentace apod.

Hlavním používaným validátor v tomto balíčku je validátor pro reprezentaci uživatelského účtu. Tato reprezentace totiž obsahuje i seznam uživatelských rolí uživatele jako seznam jmen rolí. Je tedy nutné zkontrolovat, zda tento seznam neobsahuje neexistující jména uživatelských rolí.

3.3.3 Balíček representation

Balíček **representation** obsahuje třídy, na jejichž objekty jsou převáděny parametry HTTP požadavků, a které jsou následně navraceny v odpovědích uživatelským aplikacím. Spring se automaticky pokouší provádět konverzi parametrů HTTP požadavků na odpovídající členské proměnné objektů těchto tříd.

Nejprve je nalezena odpovídající proměnná podle jejího jména, které musí odpovídat jménu parametru v požadavku. Pokud žádná takováto proměnná neexistuje, konverze selže a kontroléru je navracena chyba.

KAPITOLA 3 IMPLEMENTACE

Pokud byla odpovídající proměnná nalezena, Spring se pokusí parametr převést na datový typ této proměnné. Pro běžné datové typy je konverze automatická, pokud je však typem třída, pak musí obsahovat bezparametrický konstruktor a setter metody pro všechny členské proměnné.

3.3.4 Balíček service

V balíčku **service** jsou třídy, které obsahují veškerou rozhodovací logiku a většinu byznys logiky. Zajišťují vytváření, upravování a mazání **entit**, tedy objektů balíčku **model**, které jsou ukládány do databáze prostřednictvím **data access objektů**, tedy objektů z balíčku **dao**.

Dále zprostředkovávají samotnou integraci s dalšími službami, konkrétně využívají služby z balíčků **cesnet** a **cmis**.

3.3.5 Balíček dao

DAO, tedy **Data Access Object** představuje objekty, které zajišťují přístup k databázi [44]. Balíček **dao** tedy obsahuje třídy, jejichž metody využívají jednotlivé objekty z balíčku **service** pro získávání a ukládání objektů v databázi.

3.3.6 Balíček model

Tento balíček obsahuje všechny třídy, jejichž objekty jsou ukládány do databáze. Tyto objekty jsou po získání v kontroléru převáděny na jejich reprezentace, které jsou následně vráceny jako odpověď klientským aplikacím.

Konkrétně se jedná o třídy pro uživatelské účty, role, organizační jednotky, a metadata souborů a složek uložených v archivu.

3.3.7 Balíček cmis

Tento balíček obsahuje třídy, které jsou použity pro integraci s repozitářem **Alfresca** prostřednictvím **OpenCMIS** knihovny. Třída **CmisSessionFactory** se stará o přihlášení a vytvoření sezení (session) pro konkrétního uživatele s Alfresco repozitářem prostřednictvím **OpenCMIS**. Třída **AlfrescoService** zprostředkovává metody pro práci s repozitářem a využívá **CmisSessionFactory**.

Pro každého uživatele integračního portálu je nutné vytvořit odpovídající účet v repozitáři **Alfresca**, avšak **OpenCMIS** neumožňuje práci s uživatelskými účty. Pro vytvoření účtu tedy integrační portál využívá **RESTful API Alfresca**, které tuto funkcionalitu zpřístupňuje.

Hlavním problémem při implementaci spojení s Alfresco repozitářem se stala realizace sdílení. Zatímco webový klient Alfresco Share, který je vestavěný v Alfrescu, umožňoval bezproblémové sdílení, ta samá operace není přes CMIS ani REST rozhraní Alfresca možná. Pro sdílení tedy musel být vymyšlen alternativní postup.

Jelikož Alfresco umožňuje, aby soubor byl umístěn ve více složkách najednou, přistoupilo s k řešení sdílení prostřednictvím přidání sdíleného souboru do jednotlivých „shared“ složek uživatelů, se kterými má být sdílen.

3.3.8 Balíček cesnet

Tento balíček obsahuje veškerou logiku pro přímo práci s CESNET úložištěm. Obsahuje vlastní implementaci pro správu SSH sezení (session) s CESNET serverem a umožňuje přístup k uloženým souborům prostřednictvím SFTP a zároveň odesílání libovolných příkazů bash příkazové řádky přímo přes SSH.

3.3.9 Balíček filter

Tento balíček obsahuje všechny používané servletové filtry. Filtry dynamicky zachycují HTTP požadavky a odpovědi serveru, a mohou nad nimi provádět operace [45].

Integrační portál využívá filtrů pro nastavení hlaviček odpovědi, pomocí kterých je klientské aplikaci navrácen seznam povolených HTTP metod (GET, POST, PUT, OPTIONS, DELETE) a seznam povolených hlaviček pro HTTP požadavky.

Kapitola 4 Popis API integračního portálu

V této kapitole je popsáno aplikační rozhraní (API), které bylo vytvořeno v rámci této diplomové práce. Popsány jsou dostupné metody pro jednotlivé URI (Uniform Resource Identifier – identifikátory zdroje [26]), parametry těchto metod, a možné návratové hodnoty ve formě HTTP statusů [34].

Celý popis je pro lepší přehlednost rozdělen do kapitol podle jednotlivých modulů funkcionality. Konkrétně se jedná o kapitoly (potažmo moduly) 4.1 Přihlášení, 4.2 Administrace uživatelů, 4.4 Domovská složka uživatele, 4.5 Složka se soubory sdílenými s uživatelem, 4.6 Archiv a ?? Metody poskytované pro práci se soubory a složkami v archivu jsou totožné jako v kapitole 4.4 Domovská složka uživatele, liší se vždy jen začátek URL, tedy místo /rest/v0.1/home všechna URL pro archiv začínají /rest/v0.1/archive..

Interaktivní forma tohoto popisu je také dostupná na Apiary (online služba pro generování dokumentace RESTových API [29]) na adrese <http://docs.cesnetintegracniportal.apiary.io/>

4.1 Přihlášení

Přihlášení probíhá prostřednictvím OAuth 2.0 Username-Password Flow [28]. Uživatel odešle požadavek na přihlášení, který obsahuje jeho přihlašovací údaje, a je mu vrácen access token, pomocí kterého se autorizuje v rámci integračního portálu.

Klientská aplikace by komunikaci s integračním portálem měla vždy začínat přihlášením. Na server by tedy měla jako první odeslat požadavek na autentizaci prostřednictvím OAuth 2.0 [27]. Tento požadavek v sobě musí obsahovat Authorization hlavičku [30], pomocí které může integrační portál rozpoznat, že se jedná o autorizovanou klientskou aplikaci.

Povolené klientské aplikace se definují v konfiguračním souboru integračního portálu „client-details.xml“, kde je definováno „client-id“ a „secret“ (pro zjednodušení je možné si je představit jako jméno a heslo).

KAPITOLA 4 POPIS API INTEGRAČNÍHO PORTÁLU

Tyto údaje se odešlou na server jako Basic Authentication [31] ve formě textového řetězce „client-id:secret“ zakódovaného prostřednictvím base64 kódování [32]. Pro testovací aplikace vypadá Authorization hlavička například takto:

```
Authorization: Basic ODU5NWM4Mjg0YTUyNDc1ZTUxNGQ2Njd1NDMxM2U4NmE6MjI2ZDI0NjE3ZTY1NTRkNzFhNjg2MTRjMzQ0MzZkNjc=
```

Požadavek pro přihlášení se odesílá jako HTTP **GET** požadavek na URL **/oauth/token**. Potřebný formát požadavku je uveden v tabulce 4.1: Požadavek pro přihlášení GET **/oauth/token**, možné druhy odpovědi serveru na tento požadavek jsou uvedeny v tabulce 4.3: Odpověď na požadavek GET **/oauth/token**.

Při úspěšném přihlášení server vrátí HTTP status 200 (OK) a v těle odpovědi je uveden **access token** (viz. Dále), typ tokenu (vždy bearer), **refresh token**, který je možné použít pro opětovné vygenerování access tokenu bez nutnosti zasílání přihlašovacích údajů, a čas platnosti tokenu ve vteřinách.. V případě, že přihlašovací údaje neodpovídají žádnému existujícímu uživatelskému účtu v integračním portálu, server navrátí HTTP status 400 (Bad Request).

Po úspěšné autentizaci uživatele tedy získá klientská aplikace z odpovědi serveru tzv. **access token**. Všechny požadavky na integrační portál (kromě požadavku na autentizaci) musí obsahovat Authorization hlavičku s access tokenem. Například access tokenu 278d50d1-2b7b-4ebf-98ce-5d25e26cf743 odpovídá hlavička:

```
Authorization: Bearer 278d50d1-2b7b-4ebf-98ce-5d25e26cf743
```

Následně již klientská aplikace může využívat služeb integračního portálu, a to do té doby, než vyprší časová platnost takto uděleného access tokenu.

Pro opětovné vygenerování access tokenu prostřednictvím **refresh tokenu** je možné využít požadavek uvedený v tabulce 4.2 Požadavek pro vygenerování access tokenu prostřednictvím refresh tokenu. Jak je patrné, požadavek je velmi podobný požadavku pro přihlášení, pouze parametr grant_type je nastaven na refresh_token, a parametry username a password jsou nahrazeny parametrem refresh_token.

URI	/oauth/token	
Metoda	GET	
Hlavičky	Authentication	Basic [login údaje aplikace]
Parametry	grant_type	password
	username	[přihlašovací jméno uživatele]
	password	[heslo uživatele]

Tabulka 4.1: Požadavek pro přihlášení GET /oauth/token

URI	/oauth/token	
Metoda	GET	
Hlavičky	Authentication	Basic [login údaje aplikace]
Parametry	grant_type	refresh_token
	refresh_token	[refresh token]

Tabulka 4.2: Požadavek pro vygenerování access tokenu prostřednictvím refresh tokenu

HTTP status	Tělo odpovědi
200 (OK)	<pre>{ "access_token": "[access token]", "token_type": "bearer", "refresh_token": "[refresh token]", "expires_in": [platnost tokenu] }</pre>
400 (Bad Request)	<pre>{ "error": "invalid_grant", "error_description": "Bad credentials" }</pre>

Tabulka 4.3: Odpověď na požadavek GET /oauth/token

4.2 Administrace uživatelů

Pro administraci uživatelů integrační portál zpřístupňuje metody pro správu uživatelských účtů a uživatelských rolí. Konkrétně jsou k dispozici následující metody:

- **GET** `/rest/v0.1/user` – vrátí seznam všech uživatelských účtů,
- **POST** `/rest/v0.1/user` – vytvoří nového uživatele,
- **GET** `/rest/v0.1/user/{id}` – vrátí uživatele s daným id,
- **PUT** `/rest/v0.1/user/{id}` – upraví údaje uživatele s daným id,
- **GET** `/rest/v0.1/user/current` – vrátí aktuálně přihlášeného uživatele,
- **GET** `/rest/v0.1/role` – vrátí seznam všech uživatelských rolí,
- **POST** `/rest/v0.1/role` – vytvoří novou uživatelskou roli,
- **GET** `/rest/v0.1/role/{id}` – vrátí uživatelskou roli s daným id,
- **PUT** `/rest/v0.1/role/{id}` – upraví údaje uživatelské role s daným id,
- **GET** `/rest/v0.1/permission` – vrátí seznam oprávnění, která mohou být přiřazena k uživatelským rolím.

Metody pro vytvoření a editaci uživatelské účtu, a dále metody pro vytvoření a editaci role vyžadují, aby v těle HTTP požadavku byl platný JSON objekt s odpovídajícími údaji. Pro uživatelský účet může obsahovat proměnné `username`, `password`, `unitId`, seznam přímých oprávnění `directPermissions`, který jako jediný může obsahovat oprávnění správce org. jednotek, a seznam uživatelských rolí `userRoles`. Může tedy vypadat například následovně:

```
{
  "username": "jan",
  "password": "heslo",
  "unitId": 13136,
  "directPermissions": [ "units" ],
  "userRoles": [ "externists_admin" ]
}
```

Pro uživatelskou roli se jedná o proměnné `name`, `description` a seznam oprávnění `permissions`:

```
{
  "name": "externists_admin",
  "description": "Spravce externistu.",
  "permissions": [ "externists" ]
}
```

4.3 Organizační skupina

Jelikož v rámci této diplomové práce nebyla implementována integrace s KOSapi, jedná se pouze o simulovaná (mockovaná) data organizačních jednotek. Pro administraci organizačních skupin integrační portál zpřístupňuje následující seznam metod:

- **GET** `/rest/v0.1/unit` – vrátí seznam všech organizačních jednotek,
- **GET** `/rest/v0.1/unit/{id}` – vrátí organizační jednotku s daným id,
- **PUT** `/rest/v0.1/unit/{id}` – upraví organizační jednotku s daným id.

Upravovat je v tuto chvíli možné pouze seznam administrátorů, a v těle HTTP PUT požadavku tedy může být seznam uživatelských jmen administrátorů jako parametr `admins`.

4.4 Domovská složka uživatele

Domovská složka uživatele slouží pro ukládání a práci s jeho vlastními soubory. K tomu slouží následující metody:

- **GET** `/rest/v0.1/home` – vrátí seznam složek a souborů v domovské složce,
- **POST** `/rest/v0.1/home` – pokud požadavek obsahuje hlavičku `Content-Type: application/json`, pak vytvoří novou složku v domovské složce, jinak nahraje do domovské složky soubor z parametru `file`,
- **GET** `/rest/v0.1/home/file/{id}` – vrátí metadata souboru s daným id,
- **PUT** `/rest/v0.1/home/file/{id}` – upraví metadata souboru s daným id,
- **DELETE** `/rest/v0.1/home/file/{id}` – smaže soubor s daným id,
- **GET** `/rest/v0.1/home/file/{id}/content` – vrátí obsah souboru s daným id,
- **PUT** `/rest/v0.1/home/file/{id}/content` – přepíše obsah souboru s daným id souborem z parametru `file`,
- **GET** `/rest/v0.1/home/folder/{id}` – vrátí metadata složky s daným id a seznam jejích podsložek a souborů,
- **PUT** `/rest/v0.1/home/folder/{id}` – upraví metadata složky s daným id,
- **DELETE** `/rest/v0.1/home/folder/{id}` – smaže složku s daným id,
- **POST** `/rest/v0.1/home/folder/{id}` – viz. **POST** `/rest/v0.1/home`.

KAPITOLA 4 POPIS API INTEGRAČNÍHO PORTÁLU

Jedinými metadaty složky je její jméno, které může být nastaveno prostřednictvím proměnné `name`. Metadatum souboru sestává z jeho jména `filename` a seznamu jmen uživatelských účtů, se kterými má být soubor sdílen `sharedWith`. Prostřednictvím tohoto seznamu je tedy možné sdílet soubory s ostatními uživateli.

4.5 Složka se soubory sdílenými s uživatelem

Tato složka obsahuje všechny soubory, které s přihlášeným uživatelem sdílejí ostatní uživatelé, a do budoucna bude obsahovat také sdílené složky. Integrovaný portál umožňuje sdílené soubory pouze prohlížet, a pro práci se sdílenou složkou tedy poskytuje následující metody:

- **GET** `/rest/v0.1/shared` – vrátí seznam souborů, které s uživatelem někdo sdílí,
- **GET** `/rest/v0.1/shared/file/{id}` – vrátí metadatum sdíleného souboru s daným `id`,
- **GET** `/rest/v0.1/shared/file/{id}/content` – vrátí obsah sdíleného souboru s daným `id`.

4.6 Archiv

Metody poskytované pro práci se soubory a složkami v archivu jsou totožné jako v kapitole 4.4 Domovská složka uživatele, liší se vždy jen začátek URL, tedy místo `/rest/v0.1/home` všechna URL pro archiv začínají `/rest/v0.1/archive`.

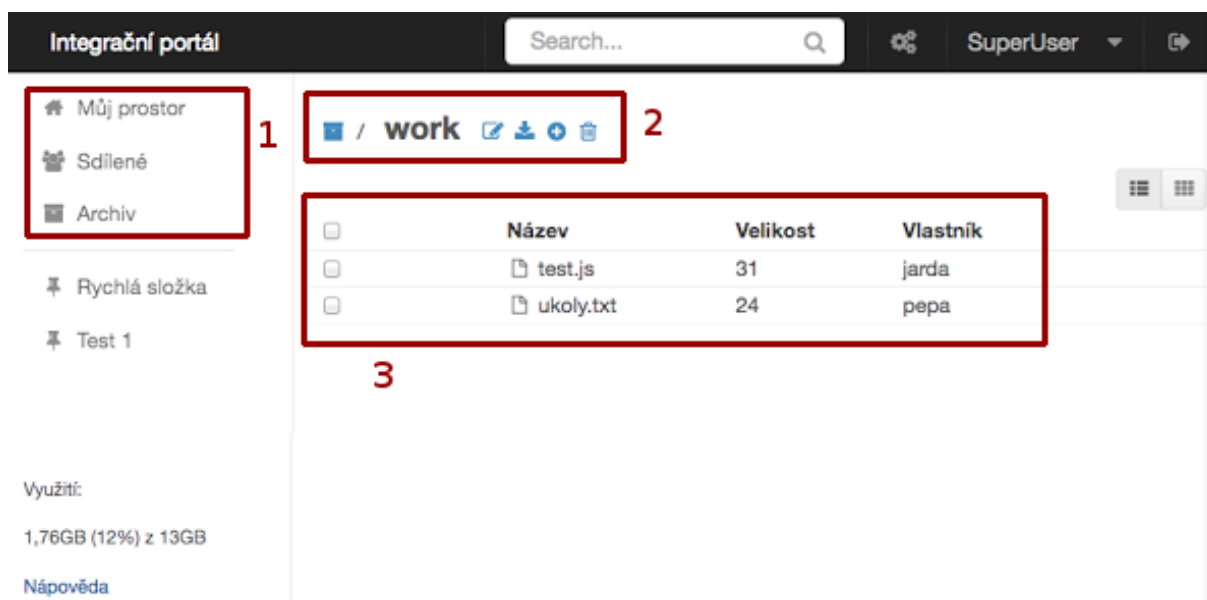
Jediná metoda, která v rámci archivu obsahuje dodatečné proměnné navíc, je metoda **PUT** `/rest/v0.1/archive/file/{id}`. Tato metoda pro editaci metadat souboru může kromě proměnných společných s domovskou složkou obsahovat navíc proměnnou `state`, která může nabývat hodnot `REG` nebo `OFL`. Tímto způsobem je možné soubor přepínat buď do „online“ (`REG`) nebo „offline“ (`OFL`) stavu.

Kapitola 5 Ověření výsledku

V této kapitole jsou uvedeny výsledky testování funkčnosti vytvořeného integračního portálu a dále propojení s grafickým klientem, který vznikl v rámci diplomové práce Kateřiny Hašlarové. Jelikož však v době psaní této práce nebyl grafický klient ještě zcela dokončen, nemohla být jeho prostřednictvím otestována veškerá funkcionalita integračního portálu. Některé funkce tedy musely být otestovány prostřednictvím jednoduché klientské REST aplikace.

5.1 Grafický klient

Grafický klient představuje hlavní uživatelské rozhraní integračního portálu, které budou uživatelé využívat pro práci se systémem. Jelikož je jeho detailní popis uveden v diplomové práci Kateřiny Hašlarové, je zde uveden pouze popis hlavních prvků rozhraní.



Obr. 5.1: Grafické uživatelské rozhraní

KAPITOLA 5 OVĚŘENÍ VÝSLEDKU

Jak je patrné z obrázku 5.1: Grafické uživatelské rozhraní, uživatelské rozhraní sestává ze tří hlavních prvků, které byly v obrázku označeny odpovídajícím číslem.

Číslem 1 je označeno menu pro přechod do jednotlivých hlavních složek. První složkou je domovská složka uživatele, na obrázku označená jako „Můj prostor“. Druhá složka, označená jako „Sdílené“, obsahuje všechny složky a soubory, které s uživatelem sdílejí ostatní uživatelé. Poslední složka „Archiv“ slouží pro přístup k archivovaným souborům.

Číslem 2 je označena drobečková navigace nebo-li breadcrumb, tedy seznam nadřazených složek právě aktuální složky. Dále jsou zde tlačítka akcí, které je možné v aktuální složce provádět.

Číslem 3 je označen samotný obsah aktuální složky. Zde je seznam všech podsložek a souborů, které jsou přímými potomky složky.

5.2 Jednoduchý REST klient

Jelikož grafický klient v době psaní této práce ještě nepodporoval všechnu funkcionalitu vytvořeného integračního portálu, nemohl být použit pro komplexní testování všech funkcí, a musel být tedy použit jiný nástroj. Existuje sice hned několik aplikací pro testování RESTful webových služeb, žádná z nich však nepodporuje OAuth 2.0 Username-Password Flow. Pracovat jejich prostřednictvím s integračním portálem je tedy nadměrně složité, jelikož uživatel musí sám sestavovat HTTP požadavky pro získání access tokenu.

Z tohoto důvodu byl vytvořena jednoduchá REST klientská aplikace, která byla napsána s použitím html a javascriptu. Na obrázku 5.2: Jednoduchý REST klient je vyobrazeno rozhraní této klientské aplikace, které je rozděleno do několika pojmenovaných oddílů.

Oddíl **Server URL** umožňuje vybrat buď lokální localhost URL, nebo zadání libovolné konkrétní URL. Jako výchozí je zadána URL testovacího serveru **147.32.80.219**.

Login oddíl umožňuje přihlášení uživatele. Aplikace se již postará o sestavení a odeslání správného přihlašovacího HTTP požadavku a zpracování odpovědi, všechny ostatní požadavky jsou následně odesílány s hlavičkou s odpovídajícím access tokenem.

Last Response oddíl zobrazuje zformátovanou odpověď serveru na poslední HTTP požadavek. Většinou se jedná o JSON objekt.

Request oddíl slouží pro zadání libovolného požadavku, který má být odeslán na server. Umožňuje zadání URL v rámci serveru, a případně tělo požadavku v poli „Data“. Obsahuje také čtyři tlačítka pro odeslání odpovídajícího typu požadavku (GET, POST, PUT, DELETE).

Oddíly **Upload to home** a **Upload to archive** slouží pro nahrání souboru do domovské složky a do archivu.

Server URL

URL

Login (logged as admin)

admin

Last Response

```
[200] OK
{
  "access_token": "be967c54-c0aa-4164-8d49-f3680671a4b9",
  "token_type": "bearer",
  "refresh_token": "d8e71762-23ba-41c2-a55f-5116fc0ca8fd",
  "expires_in": 3599
}
```

Request

http://147.32.80.219:8080/integracni-portal/rest/v0.1/

Data:

Upload to home

Folder id to upload to:

Soubor nevybrán

Upload to archive

Folder id to upload to:

Soubor nevybrán

Obr. 5.2: Jednoduchý REST klient

REST klient je součástí projektu integračního portálu a je k dispozici na URL <http://147.32.80.219:8080/integracni-portal/simple-rest-client.html> a na příloženém CD.

5.3 Testování

V této kapitole je uveden postup a výsledky testování nejdůležitější funkčnosti aplikačního rozhraní, které poskytuje integrační portál. Celé testování probíhalo proti testovacímu serveru, URL jednotlivých HTTP požadavků tedy začínalo **http://147.32.80.219:8080/integracni-portal/**. Jako nejdůležitější je zde uváděno testování funkčnosti uživatelských účtů externistů, a dále testování integrace se službami Alfresca a přímého spojení s CESNET úložištěm, tedy domovská složka, sdílení a archiv.

5.3.1 Uživatelské účty

V rámci testování uživatelských účtů jsem nejprve vytvořil nový účet pro externistu. Prostřednictvím REST klientské aplikace jsem se tedy přihlásil jako výchozí administrátor (na testovacím serveru se jedná o uživatele „admin“ s heslem „admin“). Následně jsem odeslal **POST** požadavek na URL `.../rest/v0.1/user`, který měl ve svém těle uveden následující JSON objekt:

```
{
  "username": "jan",
  "password": "heslo",
  "unitId": 1
}
```

Integrační portál na tento požadavek odpověděl zprávou o úspěšném vytvoření, která měla HTTP status 200 (OK) a v těle obsahovala JSON objekt reprezentující nově vytvořeného uživatele:

```
{
  "id": 79,
  "username": "jan",
  "unitId": 1,
  "roles": [],
  "permissions": {
    "externists": false,
    "password": false,
    "units": false
  },
  "directPermissions": []
}
```

Z odpovědi je tedy patrné, že uživatelský účet nemá žádné role ani přímá oprávnění. Následně jsem se přihlásil jako tento nový uživatel vyplněním přihlašovacího jména „jan“ a hesla „heslo“. Přihlášení proběhlo úspěšně a tento účet byl použit i pro následující testování.

5.3.2 Domovská složka

Pro testování domovské složky jsem se nejprve přihlásil jako uživatel „jan“, jehož uživatelský účet byl vytvořen v předchozí kapitole testování uživatelských účtů. Odeslání **GET** požadavku na `.../rest/v0.1/home` ukázalo, že domovská složka tohoto uživatele je prázdná:

```
{
  "id": "f4b26f6f-9ec6-4acf-bbe1-6dc73fc7ebeb",
  "name": "home",
  "breadcrumbs": [],
  "folders": [],
  "files": [],
  "createdOn": "2014-12-31T13:01+0100",
  "changedOn": "2014-12-31T13:01+0100"
}
```

Vytvořil jsem tedy složku „prace“ pomocí požadavku **POST** na `.../rest/v0.1/home` s tělem obsahujícím metadata pro tuto složku:

```
{"name": "prace"}
```

Integrační portál odpověděl zprávou o úspěšném vytvoření a uvedl URL na kterém je možné tuto složku najít. Do složky jsem následně nahrál textový soubor `todo.txt` za použití oddílu „Upload to home“ v REST klientovi. Následně jsem si nechal složku zobrazit požadavkem **GET** na `.../rest/v0.1/home/folder/3cf4e5a2-35ca-449d-b683-5da284ab7cfe`. Odpověď integračního portálu ověřila, že byl soubor úspěšně nahrán:

```
{
  "id": "3cf4e5a2-35ca-449d-b683-5da284ab7cfe",
  "name": "prace",
  "breadcrumbs": [],
  "folders": [],
  "files": [
    {
      "uuid": "1ff076ae-7f8d-4d07-bb03-71aef2141f91",
      "filename": "todo.txt",
      "mimetype": "text/plain",
      "filesize": 86,
      "createdOn": "2014-12-31T13:09+0100",
      "changedOn": "2014-12-31T13:09+0100"
    }
  ],
  "createdOn": "2014-12-31T13:05+0100",
  "changedOn": "2014-12-31T13:09+0100"
}
```

KAPITOLA 5 OVĚŘENÍ VÝSLEDKU

Obsah souboru jsem také zkontroloval prostřednictvím odeslání **GET** požadavku na URL `.../rest/v0.1/home/file/1ff076ae-7f8d-4d07-bb03-71aef2141f91/content`, na který integrační portál odpověděl zprávou s obsahem souboru.

5.3.3 Sdílení

Pro účely sdílení jsem nejprve vytvořil další uživatelský účet „karel“, opět s heslem „heslo“. Z uživatelského účtu „jan“ jsem uživateli „karel“ nasdílel soubor `todo.txt`, který byl nahrán v rámci předchozí kapitoly testování domovské složky. Toho jsem dosáhl odesláním požadavku **PUT** na `.../rest/v0.1/home/file/1ff076ae-7f8d-4d07-bb03-71aef2141f91` v jehož těle byl uveden seznam uživatelů, se kterými má být soubor sdílen:

```
{"sharedWith": ["karel"]}
```

Následně jsem se přihlásil jako uživatel „karel“ a odeslal požadavek **GET** na `.../rest/v0.1/shared` pro zobrazení obsahu sdílené složky. V seznamu sdílených souborů, který vrátil integrační portál jako odpověď, skutečně byl soubor `todo.txt`:

```
{
  "id": "c507f1dd-ddab-4e69-b9c5-9ec11a95cce0",
  "name": "shared",
  "breadcrumbs": [],
  "folders": [],
  "files": [
    {
      "uuid": "1ff076ae-7f8d-4d07-bb03-71aef2141f91",
      "filename": "todo.txt",
      ...
    }
  ],
  ...
}
```

5.3.4 Archiv

První část testování archivu probíhala totožně, jako testování domovské složky v kapitole 5.3.2 Domovská složka a není zde proto uváděna. V druhé části byla testována funkcionality specifická pro archiv, tedy přepínání souboru mezi stavy „online“ a „offline“.

Po nahrání souboru jsem nejprve prostřednictvím HTTP požadavku **GET** na URL `.../rest/v0.1/archive/file/ff8081814aa05500014aa06e42d40000` v jeho metadatech zkontroloval, že je soubor ve stavu „online“, reprezentovaným textovým řetězcem „REG“:

```
{
  "uuid": "ff8081814aa05500014aa06e42d40000",
  "filename": "todo.txt",
  '...',
  "state": "REG"
}
```

Následně jsem se pokusil přepnout stav souboru pomocí požadavku **PUT** na stejné URL, který v těle obsahoval:

```
{"state": "OFL"}
```

Opětovná kontrola metadat souboru ukázala, že soubor se nacházel ve stavu „MIG“, který značí, že je soubor právě přesouván (migrován) na offline médium.

```
{
  "uuid": "ff8081814aa05500014aa06e42d40000",
  "filename": "todo.txt",
  '...',
  "state": "MIG"
}
```

V tomto stavu soubor setrval po dobu třiceti minut. Po této době byl již soubor kompletně přesunut na offline médium a smazán z online média, a byl tedy ve stavu „OFL“. Přesunutí souboru zpět na online médium již proběhlo v rámci několika málo vteřin, po jejichž uběhnutí byl soubor ve stavu „DUL“, tedy uložen jak na online, tak offline médium.

5.3.4.1 Čas potřebný pro migraci souborů

Jak vyplynulo z předchozího testování, soubory nejsou přesunuty na offline médium okamžitě. Proběhlo tedy další testování, v jehož rámci byl měřen čas potřebný pro migraci souborů různých velikostí. Pro testování byly použity čtyři soubory o velikosti 1, 100, 200 a 1000 MB, a každý z nich byl desetkrát přepnut do offline a následně online stavu. Mezi jednotlivými testy byl každému souboru pozměněn obsah, aby se zajistilo, že je opětovně nahráván na offline médium.

Průměry výsledných hodnot jsou uvedeny v tabulce 5.1: Čas potřebný pro přepnutí stavu souborů podle jejich velikosti a zaneseny do grafu 5.3: Graf času potřebného pro přepnutí souboru do offline stavu.

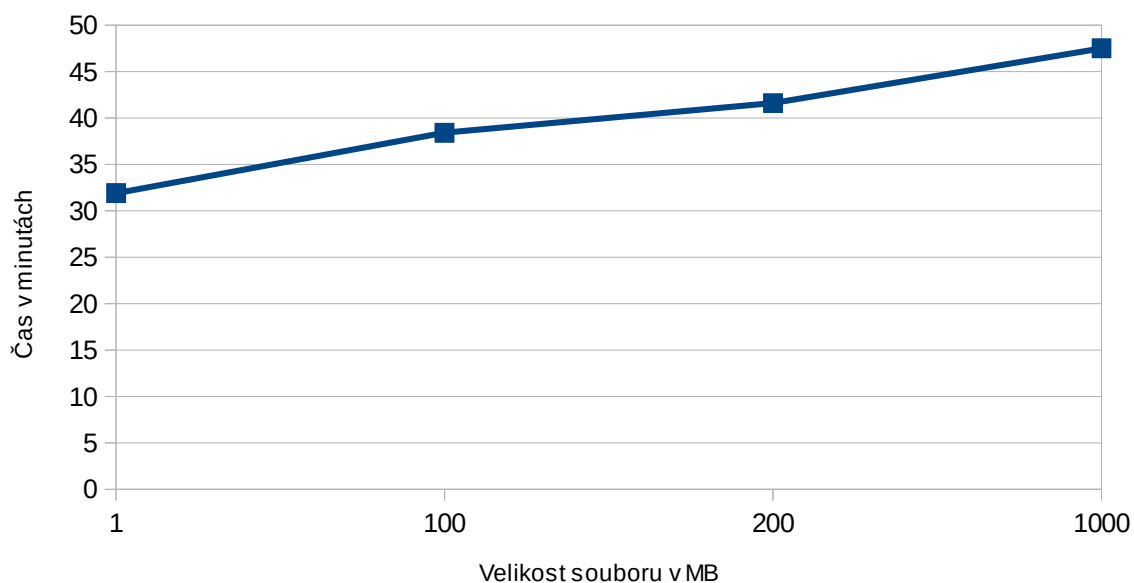
Jak je tedy patrné z naměřených dat, výsledný čas potřebný pro migraci na offline média sestává z konstantní časové složky, která se pohybuje okolo třiceti minut, a proměnné složky, která se odvíjí od velikosti souboru.

KAPITOLA 5 OVĚŘENÍ VÝSLEDKU

Naměřené hodnoty jsou však pouze orientační, a přestože platí ve většině případů, občas dochází ke značným výkyvům. Během opakovaných testování se stejnými soubory byl například pro přepnutí 100MB souboru nejnižší naměřený čas 19 minut, zatímco nejvyšší byl přes 5 hodin.

Velikost (MB)	Čas potřebný pro přepnutí do stavu	
	offline	online
1	00:31:58	00:00:02
100	00:38:28	00:00:03
200	00:41:40	00:00:04
1000	00:47:35	00:00:07

Tabulka 5.1: Čas potřebný pro přepnutí stavu souborů podle jejich velikosti



Obr. 5.3: Graf času potřebného pro přepnutí souboru do offline stavu

5.4 Shrnutí

V rámci testování funkčnosti vytvořeného integračního portálu byla otestována funkcionalita portálu, tedy přihlašování uživatelů a administrace uživatelských účtů, a dále integrace s externími službami, sestávající z práce se soubory v domovské složce uživatele, sdílení a prohlížení sdílených souborů, a práce se soubory v archivu.

Výsledky testů ukazují, že integrační portál je ve funkčním stavu, je schopen bez problémů komunikovat s Alfresco repozitářem i CESNET úložištěm a je použitelný pro administraci a práci se soubory.

Navíc byla testována a měřena doba potřebná pro provedení operace přepnutí stavu souboru v archivu. Těchto naměřených hodnot může v budoucnu využít uživatelské rozhraní, aby u souboru mohlo zobrazit odpovídající odhad času, který bude potřebný pro provedení této operace.

Kapitola 6 Závěr

V rámci této diplomové práce byl sestaven seznam funkčních a obecných požadavků na integrační portál, který bude integrovat služby datových úložišť CESNETu se službami systémů Fakulty elektrotechnické na ČVUT.

Jednotlivé funkční požadavky byly rozvedeny do případů užití integračního portálu tak, aby portál mohl zprostředkovávat služby pro sdílení a archivaci dat, a administraci uživatelských účtů a organizačních jednotek. Zároveň byli identifikováni jednotliví aktéři v integračním portálu, tedy konkrétní role, které budou uživatelé zastávat.

V rámci rešerše byly analyzovány a testovány jednotlivé technologie a knihovny, které byly potřebné pro realizaci integračního portálu.

Dále byl úspěšně vytvořen samotný základ systému pro integrační portál a implementovány nejpodstatnější požadavky na sdílení a archivaci souborů a na administraci uživatelských účtů a pravomocí. Integrační portál také obsahuje simulovaná (mockovaná) data pro organizační jednotky, a je tedy částečně připraven na práci s nimi v rámci budoucí integrace služeb KOSapi.

6.1 Budoucí práce

V rámci této práce byly zpracovány požadavky na systém a implementován jeho základ. Již od samotného začátku se však počítalo s tím, že kvůli rozsahu projektu nebude celý systém dokončen v této diplomové práci. Dokončená část integračního portálu tedy bude předána dalším studentům v rámci jejich bakalářských či diplomových prací.

Nejdůležitější částí projektu pro budoucí práce je integrace služeb Fakulty elektrotechnické na ČVUT, hlavně FELid a KOSapi. Jelikož jsou tyto dva systémy často využívány v dalších systémech Fakulty elektrotechnické, je možné částečně převzít již hotové implementace z jiných aplikací a upravit je pro potřeby integračního portálu.

KAPITOLA 6 ZÁVĚR

Jako další je potřeba implementovat zbývající funkční požadavky, které nebyly implementovány v rámci této práce. Mezi ty nejdůležitější patří administrace organizačních jednotek, která byla zatím v systému pouze nastíněna. Dalšími podstatnými chybějícími funkcemi je sdílení celých složek s uživateli a přesouvání souborů z domovské složky uživatele do archivu.

6.2 Shrnutí

Cílem projektu bylo vytvoření systému pro integraci služeb systémů Fakulty elektrotechnické na ČVUT a datových úložišť pro nahrávání, sdílení a archivaci souborů. Výsledkem práce je analýza technologií nutných pro realizaci systému integračního portálu, seznam požadavků na funkčnost, implementace základu systému a ověření jeho funkčnosti.

Jedná se o dobrý a robustní základ integračního portálu pro budoucí práci.

Použité zdroje

- [1] web: cesnet.cz. CESNET – O nás.
<http://www.cesnet.cz/sdruzeni/> stav z 15.12.2014.
- [2] web: cesnet.cz. Domovské adresáře a migrační politiky na HSM.
<https://du.cesnet.cz/cs/navody/home-migrace-plzen/start> stav z 15.12.2014.
- [3] Příspěvatelé Wikipedie. CMIS [online]. [cit.15.12.2014]. Dostupné z:
http://en.wikipedia.org/wiki/Content_Management_Interoperability_Services
- [4] Příspěvatelé Wikipedie. Spring Framework [online]. [cit.15.12.2014]. Dostupné z:
http://cs.wikipedia.org/wiki/Spring_Framework
- [5] web: spring.io. Spring Framework.
<http://projects.spring.io/spring-framework/> stav z 15.12.2014.
- [6] web: signaly.cz. Inversion of Control
<https://sqdw.signaly.cz/0804/inversion-of-control-dependency> stav z 15.12.2014.
- [7] Příspěvatelé Wikipedie. Apache Maven [online]. [cit.15.12.2014]. Dostupné z:
http://cs.wikipedia.org/wiki/Apache_Maven
- [8] web: oauth.net. OAuth Community Site.
<http://oauth.net/> stav z 15.12.2014.
- [9] web: cesnet.cz. Práce na serveru přes ssh.
<https://du.cesnet.cz/wiki/doku.php/cs/navody/ssh/start> stav z 22.9.2014.
- [10] web: fel.cvut.cz. Alfresco na FEL ČVUT.
<http://www.fel.cvut.cz/user-info/alfresco/> stav z 16.12.2014.
- [11] Příspěvatelé Wikipedie. Access Control List [online]. [cit.16.12.2014]. Dostupné z:
http://cs.wikipedia.org/wiki/Access_control_list
- [12] Příspěvatelé Wikipedie. OpenCMIS [online]. [cit.16.12.2014]. Dostupné z:
<http://en.wikipedia.org/wiki/OpenCMIS>
- [13] web: chemistry.apache.org. OpenCMIS.
<http://chemistry.apache.org/java/opencmis.html> stav z 16.12.2014.
- [14] web: wiki.fel.cvut.cz. O systému FELid.
<https://wiki.fel.cvut.cz/net/felid/about> stav z 18.12.2014

POUŽITÉ ZDROJE

- [15] web: kosapi.fit.cvut.cz. Projekt KOSapi.
<https://kosapi.fit.cvut.cz/projects/kosapi/wiki> stav z 18.12.2014
- [16] web: google.com. Disk Google.
<https://www.google.com/drive/> stav z 18.12.2014
- [17] web: onedrive.live.com. Microsoft OneDrive.
<https://onedrive.live.com/about/cs-cz/> stav z 18.12.2014
- [18] Příspěvatelé Wikipedie. REST [online]. [cit.19.12.2014]. Dostupné z:
http://en.wikipedia.org/wiki/Representational_state_transfer
- [19] Příspěvatelé Wikipedie. Secure Shell [online]. [cit.19.12.2014]. Dostupné z:
http://cs.wikipedia.org/wiki/Secure_Shell
- [20] Příspěvatelé Wikipedie. SFTP [online]. [cit.19.12.2014]. Dostupné z:
http://cs.wikipedia.org/wiki/SSH_file_transfer_protocol
- [21] web: cesnet.cz. Hierarchical Storage Management.
<https://du.cesnet.cz/cs/navody/hsm> stav z 19.12.2014
- [22] web: techtarget.com. HSM (Hierarchical Storage Management)
<http://searchstorage.techtarget.com/definition/HSM> stav z 19.12.2014
- [23] web: techtarget.com. MAID (Massive Arrays of Idle Disks)
<http://searchstorage.techtarget.com/definition/MAID> stav z 19.12.2014
- [24] Příspěvatelé Wikipedie. ECM [online]. [cit.19.12.2014]. Dostupné z:
http://en.wikipedia.org/wiki/Enterprise_content_management
- [25] Příspěvatelé Wikipedie. Alfresco [online]. [cit.19.12.2014]. Dostupné z:
[http://en.wikipedia.org/wiki/Alfresco_\(software\)](http://en.wikipedia.org/wiki/Alfresco_(software))
- [26] Příspěvatelé Wikipedie. URI [online]. [cit.19.12.2014]. Dostupné z:
http://cs.wikipedia.org/wiki/Uniform_Resource_Identifier
- [27] web: aaronparecki.com. OAuth2 Simplified.
<https://aaronparecki.com/articles/2012/07/29/1/oauth2-simplified> stav z 19.12.2014
- [28] web: help.salesforce.com. OAuth 2.0 Username-Password Flow
https://help.salesforce.com/HTViewHelpDoc?id=remoteaccess_oauth_username_password_flow.htm&language=en_US stav z 19.12.2014
- [29] web: apiary.io. Apiary.
<http://apiary.io/> stav z 19.12.2014
- [30] web: w3.org. HTTP/1.1: Header field definitions.
<http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html> stav z 20.12. 2014

- [31] web: [httpwatch.com. HTTP Authentication](http://www.httpwatch.com/httpgallery/authentication/)
<http://www.httpwatch.com/httpgallery/authentication/> stav z 20.12.2014
- [32] Příspěvatelé Wikipedie. Base64 [online]. [cit.20.12.2014]. Dostupné z:
<http://cs.wikipedia.org/wiki/Base64>
- [33] Příspěvatelé Wikipedie. JSON [online]. [cit.20.12.2014]. Dostupné z:
http://cs.wikipedia.org/wiki/JavaScript_Object_Notation
- [34] Příspěvatelé Wikipedie. List of HTTP status codes [online]. [cit.20.12.2014]. Dostupné z:
http://en.wikipedia.org/wiki/List_of_HTTP_status_codes
- [35] web: [fel.cvut.cz. Seznam organizačních jednotek FEL ČVUT.](http://www.fel.cvut.cz/rozvoj/organizacioni-jednotky.pdf)
<http://www.fel.cvut.cz/rozvoj/organizacioni-jednotky.pdf> stav z 20.12.2014
- [36] Příspěvatelé Wikipedie. Apache Tomcat [online]. [cit.29.12.2014]. Dostupné z:
http://en.wikipedia.org/wiki/Apache_Tomcat
- [37] web: [tomcat.apache.org. Apache Tomcat Maven Plugin](http://tomcat.apache.org/maven-plugin-2.2/index.html)
<http://tomcat.apache.org/maven-plugin-2.2/index.html> stav z 29.12.2014
- [38] Příspěvatelé Wikipedie. Dependency Injection [online]. [cit.29.12.2014]. Dostupné z:
http://en.wikipedia.org/wiki/Dependency_injection
- [39] Příspěvatelé Wikipedie. Model View Controller [online]. [cit.29.12.2014]. Dostupné z:
<http://en.wikipedia.org/wiki/Model-view-controller>
- [40] Příspěvatelé Wikipedie. Objektově relační mapování [online]. [cit.29.12.2014].
Dostupné z:
http://cs.wikipedia.org/wiki/Objektově_relační_mapování
- [41] Příspěvatelé Wikipedie. Session fixation [online]. [cit.29.12.2014]. Dostupné z:
http://cs.wikipedia.org/wiki/Session_fixation
- [42] Příspěvatelé Wikipedie. Cross-site request forgery [online]. [cit.29.12.2014]. Dostupné z:
http://cs.wikipedia.org/wiki/Cross-site_request_forgery
- [43] web: [alfresco.com. Alfresco Share](https://wiki.alfresco.com/wiki/Alfresco_Share)
https://wiki.alfresco.com/wiki/Alfresco_Share stav z 29.12.2014
- [44] Příspěvatelé Wikipedie. Data Access Object [online]. [cit.29.12.2014]. Dostupné z:
http://en.wikipedia.org/wiki/Data_access_object
- [45] web: [oracle.com. The Essentials of Filters](http://www.oracle.com/technetwork/java/filters-137243.html)
<http://www.oracle.com/technetwork/java/filters-137243.html> stav z 30.12.2014