

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra počítačů



Bakalářská práce

Backend pro integrační portál

Vlastimil Fengl

Vedoucí práce: Ing. Ondřej Macek, Ph.D.

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Softwarové inženýrství

20. května 2015

Poděkování

Děkuji vedoucímu práce Ing. Ondřeji Mackovi, Ph.D., Jiřímu Blažkovi a Radku Ježdíkovi za perfektní projektový tým a rady při implementaci .

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 21.5.2015

.....

Abstract

The aim of this thesis is to extend the application interface of an existing project, which integrates various data storages and systems from the Faculty of Electrical Engineering at the Czech Technical University. The result of this work is an analysis of technologies used in project, refactoring of the existing code, implementation new functions (for example integration with KOSapi) and verification of its functionality.

Abstrakt

Cílem této práce je rozšířit aplikační rozhraní existujícího projektu zabývající se integrací služeb datových úložišť a systémů Fakulty elektrotechnické na ČVUT. Výsledkem práce je analýza technologií použitých v systému, úprava stávajícího kódu, implementace nových funkcí (například integrace s KOSapi) a ověření funkčnosti těchto funkcí.

Obsah

1	Úvod.....	1
2	Analýza a návrh	2
2.1	Popis architektury	2
2.2	Frontend projektu.....	2
2.3	Backend projektu	2
2.3.1	Archivace a zálohování	3
2.3.2	Sdílení	3
2.3.3	KOSapi služba FEL ČVUT.....	3
2.4	Popis funkčních požadavků	5
2.4.1	Funkční požadavky na organizační jednotky	5
2.4.2	Funkční požadavky na štítky.....	5
2.4.3	Funkční požadavky na oblíbené složky	5
2.4.4	Funkční požadavky pro integraci KOSapi	6
2.4.5	Funkční požadavky pro archiv CESNET	6
2.5	Případy užití	6
2.5.1	Aktéři	6
2.5.2	Případy užití organizačních jednotek.....	7
2.5.2.1	Případ užití - Přidat uživatele do organizační jednotky	9
2.5.2.2	Případ užití - Odebrat uživatele z organizační jednotky	9
2.5.2.3	Případ užití - Vytvořit organizační jednotku.....	9
2.5.2.4	Případ užití - Přidat administrátora do organizační jednotky.....	10
2.5.2.5	Případ užití - Odebrat administrátora z organizační jednotky.....	10
2.5.2.6	Případ užití - Nastavení kvóty organizační jednotky	11
2.5.2.7	Případ užití - Zobrazit seznam uživatelů.....	11
2.5.2.8	Případ užití - Zobrazit seznam uživatelů v organizační jednotce.....	11
2.5.2.9	Případ užití - Zobrazit seznam organizačních jednotek	12
2.5.3	Případy užití štítků	12
2.5.3.1	Případ užití - Přidat štítek.....	14
2.5.3.2	Případ užití - Upravit štítek	14
2.5.3.3	Případ užití - Odebrat štítek	14
2.5.3.4	Případ užití - Přidat štítek ke složce.....	15
2.5.3.5	Případ užití - Odebrat štítek u složky.....	15
2.5.3.6	Případ užití - Přidat štítek k souboru.....	15

2.5.3.7	Případ užití - Odebrat štítek u souboru	16
2.5.3.8	Případ užití - Zobrazit seznam štítků	16
2.5.3.9	Případ užití - Zobrazit seznam složek	16
2.5.3.10	Případ užití - Zobrazit seznam souborů	16
2.5.4	Případy užití oblíbených složek	17
2.5.4.1	Případ užití - Přidat oblíbenou složku	17
2.5.4.2	Případ užití - Odebrat oblíbenou složku	18
2.5.4.3	Případ užití - Zobrazit oblíbené složky	18
2.5.4.4	Případ užití - Zobrazit seznam složek	18
2.5.5	Případy užití KOSapi	19
2.5.5.1	Případ užití - Zobrazit seznam kateder na FEL	19
2.5.5.1	Případ užití - Zobrazit seznam zaměstnanců katedry	20
2.5.6	Případy užití archivu CESNET	20
2.5.6.1	Případ užití - Přesunout soubor do offline režimu	21
2.5.6.2	Případ užití - Přesunout soubor do online režimu	21
3	Implementace	22
3.1	Implementované požadavky	22
3.2	Použité technologie	23
3.2.1	Spring	23
3.2.1.1	Spring Core	24
3.2.1.2	Spring Web a Spring WebMVC	24
3.2.1.3	Spring Security	24
3.2.2	OAuth	24
3.2.3	REST	26
3.2.4	Maven	26
3.3	Balíčky integračního portálu	26
3.3.1	Balíček model	28
3.3.2	Balíček dao	28
3.3.3	Balíček service	28
3.3.4	Balíček controller	28
3.3.5	Balíček representation	29
3.3.6	Balíček validator	29
3.3.7	Balíček filter	29
3.3.8	Balíček cesnet	29
4	Popis API integračního portálu	30
4.1	Štítky	30

4.2	Organizační jednotky	32
4.3	Oblíbené složky	33
4.4	Archiv CESNET	33
4.5	KOSapi.....	33
5	Testování.....	34
5.1	Unit testování	34
5.1.1	Testování štítků	35
5.1.2	Testování organizačních jednotek.....	35
5.1.3	Testování oblíbených složek.....	36
5.2	Testování implementace KOSapi.....	36
6	Závěr	39
	Použité zdroje.....	40

A	Obsah přiloženého CD	42
----------	-----------------------------	-----------

Seznam ilustrací

<i>Obr. 1: Architektura integračního portálu</i>	<i>4</i>
<i>Obr. 2: Diagram hierarchie aktérů [1]</i>	<i>7</i>
<i>Obr. 3: Diagram případu užití - Administrace organizační jednotky.....</i>	<i>8</i>
<i>Obr. 4: Diagram případu užití - Administrace štítků.....</i>	<i>13</i>
<i>Obr. 5: Diagram případu užití - Administrace oblíbených složek</i>	<i>17</i>
<i>Obr. 6: Diagram případu užití - KOSapi</i>	<i>19</i>
<i>Obr. 7: Diagram případu užití - Archiv CESNET</i>	<i>20</i>
<i>Obr. 8: Průběh autentizace pomocí protokolu OAuth [24]</i>	<i>25</i>
<i>Obr. 9: Diagram balíčků integračního portálu</i>	<i>27</i>
<i>Obr. 10: Webové rozhraní testovacího serveru</i>	<i>37</i>
<i>Obr. 11: Tvar odpovědi na požadavek seznamu kateder</i>	<i>37</i>
<i>Obr. 12: Tvar odpovědi na požadavek seznamu zaměstnanců dané katedry.....</i>	<i>38</i>

Seznam tabulek

<i>Tabulka 1: Význam HTTP metod u REST rozhraní</i>	<i>26</i>
---	-----------

1 Úvod

Doby, kdy se přenos, zálohování či sdílení dat provádělo pomocí datových nosičů (od magnetických pásek, disket, CD a DVD až po USB Flash disky) jsou už skoro pryč. V dnešní době uživatelé preferují snadný přístup k datům a možnost stažení těchto dat odkudkoliv s co nejmenším vynaloženým úsilím. Proto vznikla spousta cloudových služeb, které sdílení a ukládání dat umožňují. Mezi nejznámější cloudové služby patří například Google Drive[3], Microsoft OneDrive[4] nebo Dropbox[5]. Tyto služby ovšem mají řadu omezení jako je velikost celkového úložiště nebo velikost sdílených souborů.

Mezi aktivní uživatele těchto služeb patří i akademičtí pracovníci a studenti ČVUT, kteří je využívají skoro denně a musí pak řešit několik problémů. Mezi hlavní problémy patří především nemožnost archivace velkých dat nebo omezení úložného místa. Z těchto důvodů jsou akademičtí pracovníci nuceni využít velké množství různých datových úložišť.

Problém, který tyto služby přináší pro akademické pracovníky a studenty ČVUT, je, že neumožňují přihlašování přes FELid[6]. Je sice pravda, že Google Drive je propojen se systémem FELid, tedy s autentizačním a autorizačním systémem pro Fakultu elektrotechnickou na ČVUT, ale ostatní systémy ČVUT už nepodporuje. Přitom dalším často potřebným systémem je KOSapi[12], který umožňuje přístup k části dat v databázi ČVUT. Dále, podle podmínek společnosti Google, není Google Drive určen pro dlouhodobé uchování dat.

Proto bylo potřeba vytvořit systém, který bude provázaný se systémy ČVUT a bude integrovat jednotlivé služby pro sdílení a archivaci dat. Vznikl tedy projekt "Integrační portál pro sdílení a zálohování dat". Tento systém vzniká pod grantem CESNET, integrace datových služeb pro vědecko-výukové skupiny[2]. Samotný systém využívá úložné prostory, které pro univerzitu poskytuje CESNET.

Projekt byl v minulosti rozdělen na dvě menší části, backend[1] a frontend[2]. V mé bakalářské práci se budeme zabývat jen backend částí, kterou podrobněji rozebereme v další kapitole. Cílem bakalářské práce je seznámit se s návrhem projektu integrační portál, sestavení požadavků a jejich následná implementace pro funkcionalitu popsanou níže:

- administrace organizačních jednotek
- administrace štítků
- administrace oblíbených složek uživatelů
- přesun z a do archivu CESNET
- integrace KOSapi

2 Analýza a návrh

2.1 Popis architektury

Již v úvodu jsme se dozvěděli, že projekt byl rozdělen na dvě části, na backend a frontend, a že tato bakalářská práce se zabývá jen backendovou částí projektu. V následujících podkapitolách si rozebereme obě části projektu, podrobněji se zaměříme na backendovou část, vypíšeme si funkční požadavky, které mají být implementovány do projektu, a popíšeme chování implementovaných požadavků pomocí případů užití.

2.2 Frontend projektu

Frontend část, neboli klientská část projektu, poskytuje přístup ke službám serverové části pomocí intuitivního grafického rozhraní. Využívá API poskytované serverovou částí a pomocí grafického rozhraní umožňuje uživateli tyto služby využívat. Touto částí se zabývá bakalářská práce Jiřího Blažka[29].

2.3 Backend projektu

Backend část, neboli serverová část projektu, se stará o integraci se systémem ČVUT a datovými úložišti. Tato část je nasazena na serveru a poskytuje aplikační rozhraní, které uživatelé využívají pro přístup k poskytovaným službám. Tato bakalářská práce se zabývá rozšířením právě této části projektu. Celkový přehled nynější architektury projektu zachycuje obr. 1: Architektura integračního portálu.

2.3.1 Archivace a zálohování

Jeden z hlavních požadavků na projekt integrační portál byla schopnost dlouhodobého uchovávání a zálohování dat. Pro archivaci a zálohování souborů jsou využívána datová úložiště CESNETu, které poskytuje v rámci CESNET grantu o integraci datových služeb vědecko-výukové skupiny.

Tato datová úložiště CESNETu poskytují dostatečný úložný prostor pro potřeby integračního portálu. Přístup k těmto úložištím lze realizovat pomocí SFTP (Secure File Transfer Protocol[7]) nebo přes FTPS (File Transfer Protocol over Secure Socket Layer[8]).

Dále je k serverům CESNET umožněn přístup i přes SSH (Secure Shell[9]). Integrační portál pomocí SSH může zjišťovat, kde se nachází jednotlivé soubory, jestli na pomalých, vysokokapacitních úložištích nebo se nacházejí na rychlejších mediích.

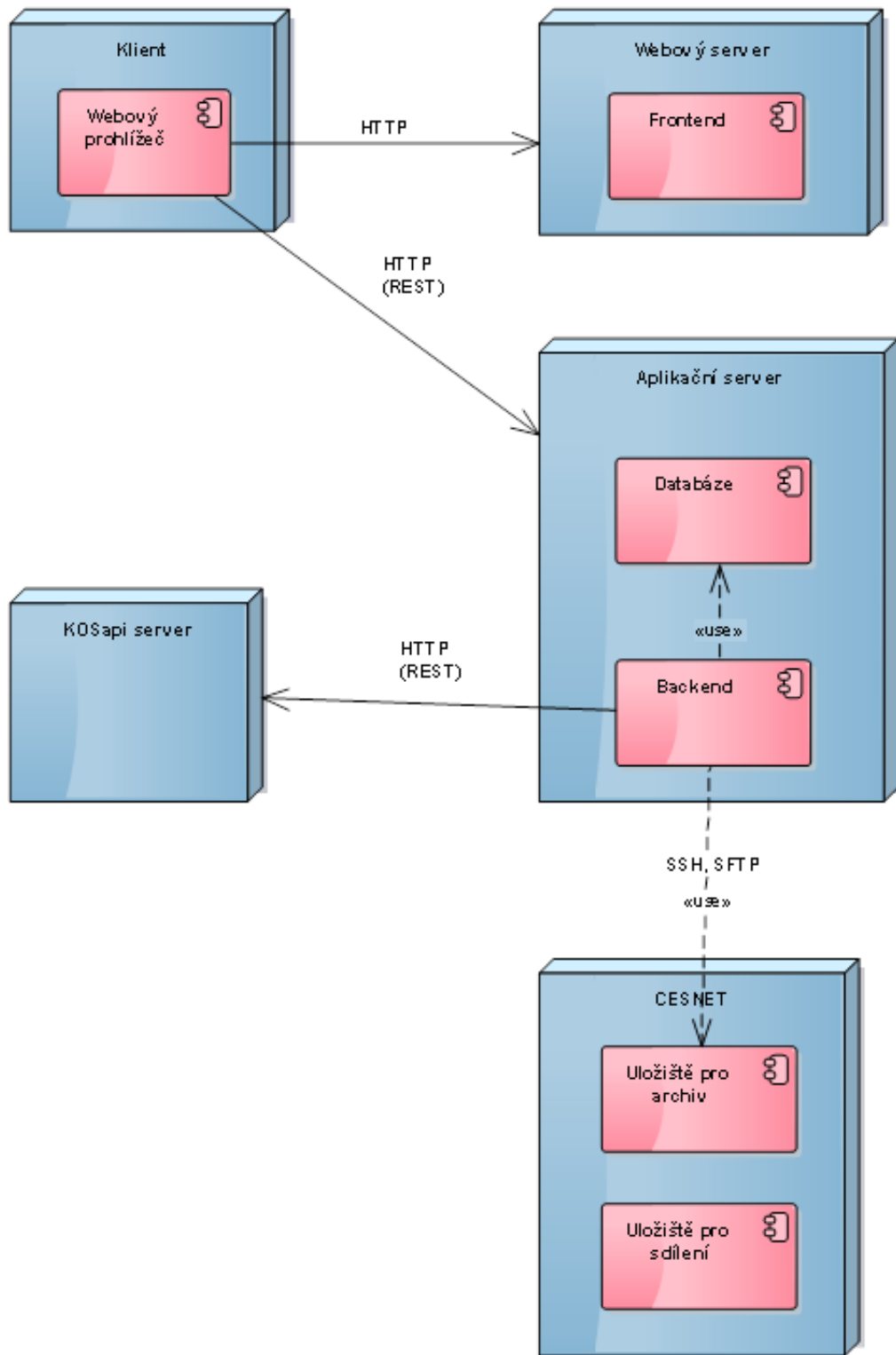
2.3.2 Sdílení

Pro sdílení souboru byl do projektu nasazen systém Alfresco[10], který je open-source ECM systém (Enterprise Content Management - systém pro vytváření, správu, zabezpečení, prohlížení a distribuci dat[11]). Ale v rámci diplomové práce jiného studenta, který rozšíří integrační portál o integraci dalších datových úložišť (Google Drive, Microsoft OneDrive), bylo Alfresco odstraněno z důvodu jednodušší komunikace s těmito úložišti (například Google Drive nabízí pro komunikaci REST API) a snížení nároků hardwaru na server. V tuto chvíli je tedy v projektu upload a download souborů na úložiště CESNET znemožněn.

2.3.3 KOSapi služba FEL ČVUT

V rámci této bakalářské práce bylo jedno z požadovaných funkcionalit integrace KOSapi[12]. Jde o aplikační rozhraní RESTových webových služeb, která umožňují přístup k vybrané části dat v databázi KOS.

Data, která služby poskytují, jsou reprezentována ve formátu Atom[22]. Atom je formát založený na XML[23] a vytvořený pro publikování syndikovaného obsahu.



Obr. 1: Architektura integračního portálu

2.4 Popis funkčních požadavků

V této části bakalářské práce si popíšeme všechny naplánované funkční požadavky. Následně budou funkční požadavky rozvíjeny v kapitole 2.5 Případy užití, kde si uvedeme aktéry integračního portálu, diagramy případu užití a jejich popisy.

Nyní následují seznam požadovaných funkčních požadavků, o které by měl být integrační portál rozšířen. Jde vlastně o služby, které by měl integrační portál uživateli umožnit využívat. Požadavky jsou rozděleny, podle funkcionality, do modulů pro lepší přehlednost.

2.4.1 Funkční požadavky na organizační jednotky

- Systém bude umožňovat uživateli vytvářet, upravovat a mazat organizační jednotky.
- Systém bude umožňovat správci organizační jednotky přidávat uživatele jako dalšího správce organizační jednotky.
- Systém bude umožňovat správci organizační jednotky odebírat jiného správce organizační jednotky.
- Systém bude umožňovat správci organizační jednotky přidávat nebo odebírat uživatele z organizační jednotky.
- Systém bude umožňovat uživateli zobrazit seznam vytvořených organizačních jednotek.
- Systém bude umožňovat uživateli zobrazit seznam uživatelů dané organizační jednotky

2.4.2 Funkční požadavky na štítky

- Systém bude umožňovat uživateli vytvářet, upravovat a mazat štítky.
- Systém bude umožňovat připínat nebo odepínat štítky na soubory.
- Systém bude umožňovat připínat nebo odepínat štítky na složky.

2.4.3 Funkční požadavky na oblíbené složky

- Systém bude umožňovat uživateli přidat složku do oblíbených složek.
- Systém bude umožňovat uživateli odebrat složku z oblíbených složek.
- Systém bude umožňovat uživateli zobrazit seznam jeho oblíbených složek.

2.4.4 Funkční požadavky pro integraci KOSapi

- Systém bude umožňovat uživateli vypsát seznam kateder na Fakultě elektrotechnické ČVUT.
- Systém bude umožňovat uživateli vypsát seznam zaměstnanců dané katedry z Fakulty elektrotechnické na ČVUT.

2.4.5 Funkční požadavky pro archiv CESNET

- Systém bude umožňovat uživateli přesun souboru z online módu do offline mód.
- Systém bude umožňovat uživateli přesun souboru z offline módu do online módu.

2.5 Případy užití

V této kapitole si dále rozvedeme funkční požadavky ze seznamu v kapitole 2.4 Popis funkčních požadavků. V následujících podkapitolách jsou představeny jednotlivé případy užití, podle modulů, a aktéři integračního portálu.

2.5.1 Aktéři

Aktéři představují jednotlivá oprávnění uživatelů. Integrační portál má několik různých oprávnění, které se mohou přiřazovat k uživatelským rolím a ty se pak přiřazují jednotlivým uživatelům. Na obrázku obr. 2: Diagram hierarchie aktérů je vyobrazena hierarchická struktura aktérů, která odpovídá struktuře systému oprávnění.

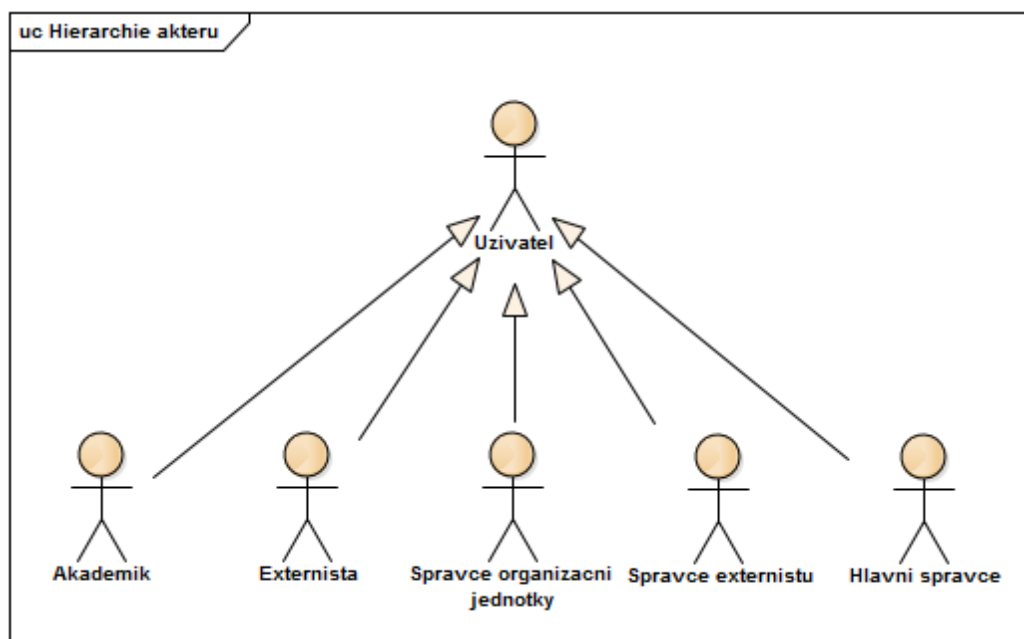
Uživatel systému integračního portálu může mít přiřazeno více uživatelských rolí, a tím i více uživatelských oprávnění. V důsledku to znamená, že může vystupovat jako několik různých aktérů ze spodní úrovně hierarchie aktérů (**správce organizační jednotky, správce externistů, hlavní správce**). Vždy musí mít oprávnění **akademik** nebo **externista**.

Akademik a **externista** jsou oba uživatelé systému a oba mohou plně využívat služeb integračního portálu. Jediný rozdíl mezi těmito aktéry je ve způsobu přihlašování do systému.

Správce organizační jednotky spravuje přiřazené organizační jednotky. Toto oprávnění není globálního typu, ale vždy se vztahuje na konkrétní organizační jednotku. Správce může například přidávat nebo odebírat uživatele z organizační jednotky nebo zobrazovat využití úložného prostoru v rámci organizační jednotky.

Hlavní správce se stará o správu celého integračního portálu. Může například přiřazovat dostupný prostor v datovém úložišti organizačním jednotkám prostřednictvím kvót, vytvářet nové organizační jednotky nebo vytvářet a přiřazovat uživatelské role jednotlivým uživatelům.

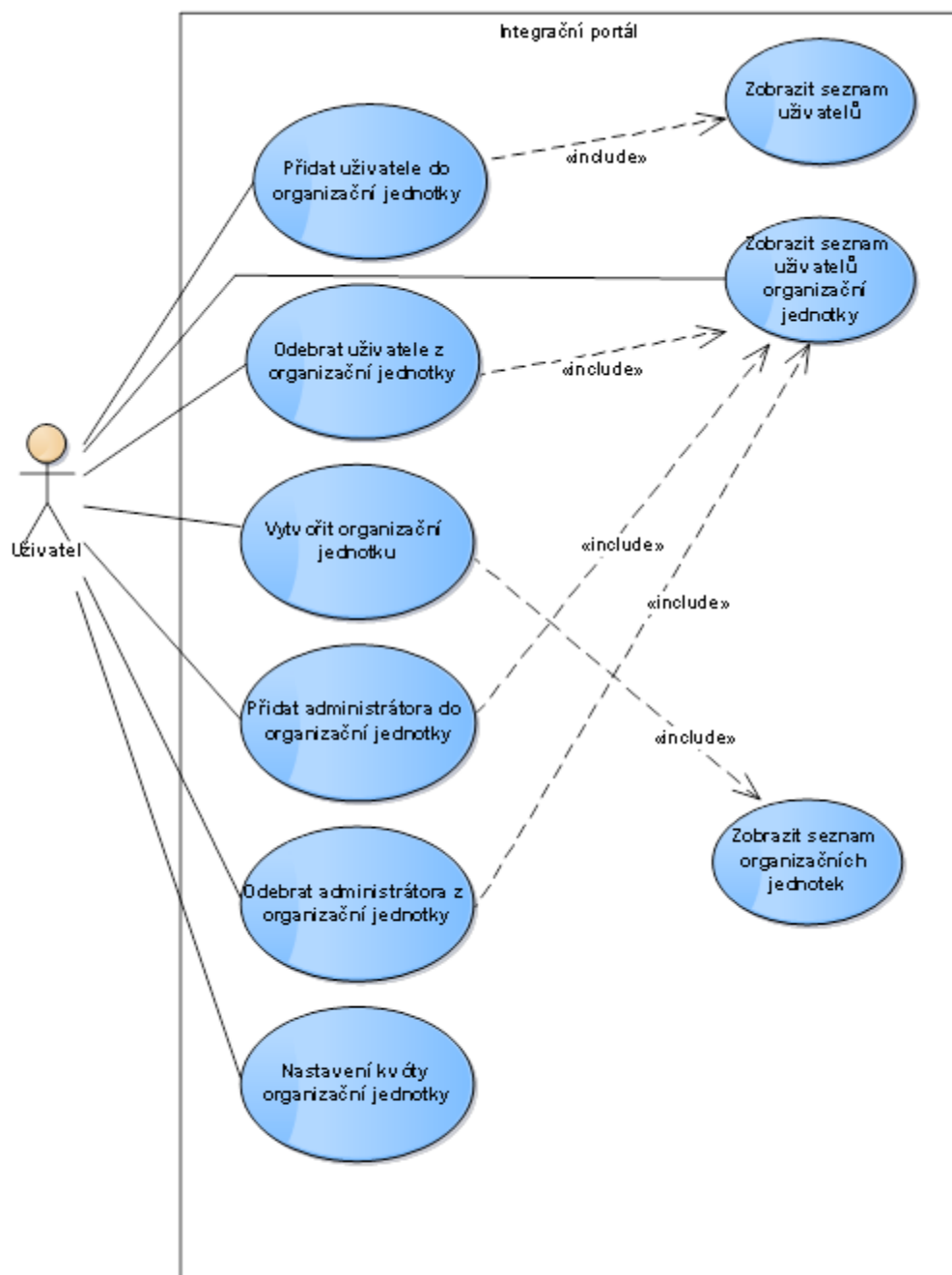
Správce externistů má na starosti uživatelské účty externistů. Může vytvářet nové nebo upravovat či mazat již existující účty pro externisty.



Obr. 2: Diagram hierarchie aktérů [1]

2.5.2 Případy užití organizačních jednotek

V této kapitole jsou detailněji popsány funkční požadavky z kapitoly 2.4.1 Funkční požadavky na organizační jednotky. Na obrázku Obr. 3: Diagram případu užití - Administrace organizační jednotky jsou vyobrazeny jednotlivé případy užití. V tomto modulu vystupuje více aktérů (Akademik, Externista, Správce organizační jednotky a Hlavní správce), ale v diagramu jsou všichni vyobrazeni jako obecný aktér **Uživatel**. Každý případ užití má popsáno, který aktér ho může vykonávat, pokud je aktérem **Uživatel** mohou tento případ užití použít všichni.



Obr. 3: Diagram případu užití - Administrace organizační jednotky

2.5.2.1 Příklad užití - Přidat uživatele do organizační jednotky

Tento případ zachycuje proces přidání uživatele do organizační jednotky. Přidávat uživatele může jen správce organizační jednotky.

Hlavní scénář:

1. Příklad užití začíná, když chce správce organizační jednotky přidat uživatele do organizační jednotky.
2. Include(Zobrazit seznam uživatelů).
3. Správce organizační jednotky vybere požadovaného uživatele.
4. Integrovaný portál přidá uživatele do organizační jednotky a zobrazí aktualizovaný seznam uživatelů v organizační jednotce a zprávu o úspěšném přidání.

2.5.2.2 Příklad užití - Odebrat uživatele z organizační jednotky

Tento případ užití zachycuje proces odebrání uživatele z organizační jednotky. Odebrat uživatele může jen správce organizační jednotky.

Hlavní scénář:

1. Příklad užití začíná, když chce správce organizační jednotky odebrat uživatele z organizační jednotky.
2. Include(Zobrazit seznam uživatelů organizační jednotky)
3. Správce organizační jednotky vybere požadovaného uživatele.
4. Integrovaný portál odebere uživatele z organizační jednotky a zobrazí aktualizovaný seznam uživatelů v organizační jednotce a zprávu o úspěšném odebrání.

2.5.2.3 Příklad užití - Vytvořit organizační jednotku

Tento případ užití zachycuje proces vytváření nové organizační jednotky. Vytvoření organizační jednotky může zatím každý.

Hlavní scénář:

1. Příklad užití začíná, když chce uživatel vytvořit novou organizační jednotku.
2. Integrovaný portál zobrazí formulář, který umožní zadat potřebné údaje a vybrat jméno ze seznamu Include(Zobrazit seznam organizačních jednotek).v rámci Fakulty elektrotechnické.
3. Uživatel vyplní potřebná data a vybere název organizační jednotky ze seznamu a potvrdí zadání.
4. Integrovaný portál zkontroluje, zda organizační jednotka z daným názvem neexistuje. Integrovaný portál vytvoří novou organizační jednotku a zobrazí detail nové organizační jednotky a zprávu o úspěšném vytvoření.

2.5.2.4 Příklad užití - Přidat administrátora do organizační jednotky

Tento případ užití zachycuje proces přidání role administrátora uživateli v organizační jednotce. Přidat administrátora může správce organizační jednotky.

Hlavní scénář:

1. Příklad užití začíná, když chce správce organizační jednotky přidat administrátora organizační jednotky.
2. Include(Zobrazit seznam uživatelů organizační jednotky).
3. Správce organizační jednotky vybere požadovaného uživatele.
4. Integrovaný portál zobrazí detail uživatele.
5. Správce organizační jednotky v detailu klikne na tlačítko pro přidání uživatele mezi administrátory organizační jednotky.
6. Integrovaný portál přidá vybraného uživatele mezi administrátory organizační jednotky a zobrazí zprávu o úspěšném přidání

2.5.2.5 Příklad užití - Odebrat administrátora z organizační jednotky

Tento případ užití zachycuje proces odebrání role administrátora uživateli v organizační jednotce. Odebrat administrátora může správce organizační jednotky.

Hlavní scénář:

1. Příklad užití začíná, když chce správce organizační jednotky odebrat administrátora organizační jednotky.
2. Include(Zobrazit seznam uživatelů organizační jednotky).
3. Správce vybere požadovaného administrátora.
4. Integrovaný portál zobrazí detail administrátora.
5. Správce organizační jednotky v detailu klikne na tlačítko pro odebrání role administrátora v organizační jednotce.
6. Integrovaný portál odebere vybraného uživatele ze seznamu administrátorů organizační jednotky a zobrazí zprávu o úspěšném odebrání.

2.5.2.6 Příklad užití - Nastavení kvóty organizační jednotky

Tento případ užití zachycuje proces nastavení kvóty datového úložiště pro organizační jednotku. Nastavit kvótu organizační jednotky může správce organizační jednotky.

Hlavní scénář:

1. Příklad užití začíná, když chce hlavní správce nastavit kvótu pro organizační jednotku.
2. Integrovaný portál zobrazí formulář, který umožní zadat požadovanou kvótu.
3. Hlavní správce vyplní kvótu a potvrdí zadání.
4. Integrovaný portál nastaví požadovanou kvótu organizační jednotce a zobrazí zprávu o úspěšném nastavení.

2.5.2.7 Příklad užití - Zobrazit seznam uživatelů

Tento případ užití zachycuje proces zobrazení seznamu uživatelů integrovaného portálu.

Hlavní scénář:

1. Příklad užití začíná, když chce uživatel zobrazit seznam uživatelů integrovaného portálu.
2. Integrovaný portál odešle seznam všech uživatelů integrovaného portálu.

2.5.2.8 Příklad užití - Zobrazit seznam uživatelů v organizační jednotce

Tento případ užití zachycuje proces zobrazení seznamu uživatelů v organizační jednotce.

Hlavní scénář:

1. Příklad užití začíná, když chce uživatel zobrazit seznam uživatelů v organizační jednotce.
2. Integrovaný portál vrátí seznam uživatelů v organizační jednotce. Administrátoři organizační jednotky budou odlišeni.

2.5.2.9 Příklad užití - Zobrazit seznam organizačních jednotek

Tento případ užití zachycuje proces zobrazení seznamu organizačních jednotek na Fakultě elektrotechnické ČVUT.

Hlavní scénář:

1. Příklad užití začíná, když chce uživatel zobrazit seznam organizačních jednotek.
2. Integrovaný portál vrátí seznam organizačních jednotek.

2.5.3 Případy užití štítků

V této kapitole jsou detailněji popsány funkční požadavky z kapitoly 2.4.2 Funkční požadavky na štítky. Na obrázku Obr. 4: Diagram případu užití - Administrace štítků jsou vyobrazeny jednotlivé případy užití. V tomto modulu vystupuje pouze jediný aktér, obecný aktér Uživatel.



Obr. 4: Diagram případu užití - Administrace štítků

2.5.3.1 Příklad užití - Přidat štítek

Tento případ užití zachycuje proces přidání štítku. Štítky může vytvářet každá role.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce přidat nový štítek.
2. Integrovaný portál zobrazí formulář, který umožní zadat název a vybrat barvu štítku.
3. Uživatel vyplní povinné údaje a klikne na tlačítko pro vytvoření štítku.
4. Integrovaný portál vytvoří nový štítek a zobrazí zprávu o úspěšnosti vytvoření.

2.5.3.2 Příklad užití - Upravit štítek

Tento případ užití zachycuje proces úpravy dat štítku. Úpravu štítku může provádět jen autor štítku.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce upravit štítek.
2. Include(Zobrazit seznam štítků)
3. Uživatel vybere požadovaný štítek.
4. Integrovaný portál zobrazí detail vybraného štítku.
5. Uživatel v detailu klikne na tlačítko pro úpravu.
6. Integrovaný portál zobrazí formulář, který umožní změnu názvu nebo barvy štítku.
7. Uživatel upraví údaje a potvrdí úpravu.
8. Integrovaný portál upraví detaily štítku a zobrazí aktualizovaný seznam štítků a zprávu o úspěšnosti úpravy.

2.5.3.3 Příklad užití - Odebrat štítek

Tento případ užití zachycuje proces odebrání štítku. Odebrat štítek může jen autor štítku.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce smazat štítek.
2. Include(Zobrazit seznam štítků)
3. Uživatel vybere štítek.
4. Integrovaný portál zobrazí detail vybraného štítku.
5. Uživatel v detailu klikne na tlačítko pro smazání štítku.
6. Integrovaný portál smaže štítek a zobrazí zprávu o úspěšném smazání.

2.5.3.4 Příklad užití - Přidat štítek ke složce

Tento případ užití zachycuje proces přidávání štítku na složku.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce připnout štítek na složku.
2. Include(Zobrazit seznam složek)
3. Uživatel vybere složku.
4. Integrovaný portál zobrazí detail vybrané složky.
5. Uživatel klikne na tlačítko pro připnutí štítku.
6. Include(Zobrazit seznam štítků)
7. Uživatel vybere štítek a potvrdí výběr.
8. Integrovaný portál přiřadí složce štítek a zobrazí zprávu o úspěšném přidání

2.5.3.5 Příklad užití - Odebrat štítek u složky

Tento případ užití zachycuje proces odebrání štítku připnutý na složce.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce odebrat připnutý štítek na složce.
2. Include(Zobrazit seznam složek)
3. Uživatel vybere složku.
4. Integrovaný portál zobrazí detail vybrané složky.
5. Uživatel klikne na tlačítko pro odebrání štítku
6. Integrovaný portál odebere štítek a zobrazí zprávu o úspěšném odebrání.

2.5.3.6 Příklad užití - Přidat štítek k souboru

Tento případ užití zachycuje proces přidávání štítku k souboru.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce připnout štítek k souboru.
2. Include(Zobrazit seznam souborů)
3. Uživatel vybere soubor.
4. Integrovaný portál zobrazí detail vybraného souboru.
5. Uživatel klikne na tlačítko pro připnutí štítku.
6. Include(Zobrazit seznam štítků)
7. Uživatel vybere štítek a potvrdí výběr.
8. Integrovaný portál přiřadí souboru štítek a zobrazí zprávu o úspěšném přidání

2.5.3.7 Příklad užití - Odebrat štítek u souboru

Tento případ užití zachycuje proces odebrání štítku u souboru.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce odebrat štítek připnutý na souboru.
2. Include(Zobrazit seznam souborů)
3. Uživatel vybere soubor.
4. Integrovaný portál zobrazí detail vybrané složky.
5. Uživatel klikne na tlačítko pro odebrání štítku.
6. Integrovaný portál odebere štítek a zobrazí zprávu o úspěšném přidání

2.5.3.8 Příklad užití - Zobrazit seznam štítků

Tento případ užití zachycuje proces zobrazení seznamu štítků.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce zobrazit seznam štítků.
2. Integrovaný portál zobrazí seznam všech štítků uživatele.

2.5.3.9 Příklad užití - Zobrazit seznam složek

Tento případ užití zachycuje proces zobrazení seznamu složek.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce zobrazit seznam složek.
2. Integrovaný portál zobrazí seznam všech složek v rámci organizační jednotky.

2.5.3.10 Příklad užití - Zobrazit seznam souborů

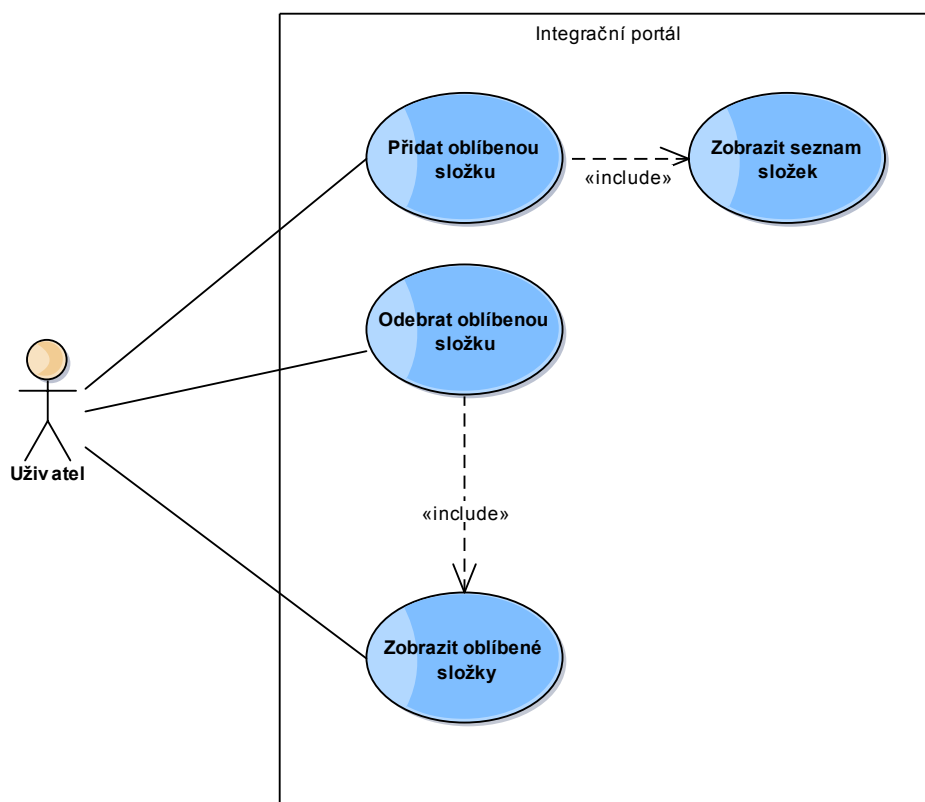
Tento případ užití zachycuje proces zobrazení seznamu souborů.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce zobrazit seznam souborů.
2. Integrovaný portál zobrazí seznam všech souborů v rámci organizační jednotky.

2.5.4 Případy užití oblíbených složek

V této kapitole jsou detailněji popsány funkční požadavky z kapitoly 2.4.3 Funkční požadavky na oblíbené složky. Na obrázku Obr. 5: Diagram případu užití - Administrace oblíbených složek jsou vyobrazeny jednotlivé případy užití. V tomto modulu vystupuje jediný aktér, obecný aktér Uživatel.



Obr. 5: Diagram případu užití - Administrace oblíbených složek

2.5.4.1 Příklad užití - Přidat oblíbenou složku

Tento případ užití zachycuje proces přidávání složky mezi oblíbené složky.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce přidat složku mezi oblíbené.
2. Include(Zobrazit seznam složek)
3. Uživatel vybere složku.
4. Integrační portál zobrazí detail složky.
5. Uživatel v detailu klikne na tlačítko pro přidání mezi oblíbené.
6. Integrační portál přidá složku mezi oblíbené složky a zobrazí zprávu o úspěšnosti přidání.

2.5.4.2 Příklad užití - Odebrat oblíbenou složku

Tento případ užití zachycuje proces odebrání složky ze seznamu oblíbených složek.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce přidat složku mezi oblíbené.
2. Include(Zobrazit oblíbené složky)
3. Uživatel vybere složku.
4. Integrovaný portál zobrazí detail vybrané složky.
5. Uživatel v detailu klikne na tlačítko pro odebrání z oblíbených složek.
6. Integrovaný portál odebere složku ze seznamu oblíbených složek a zobrazí zprávu o úspěšnosti odebrání.

2.5.4.3 Příklad užití - Zobrazit oblíbené složky

Tento případ užití zachycuje proces zobrazení seznamu oblíbených složek uživatele.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce zobrazit seznam oblíbených složek.
2. Integrovaný portál zobrazí seznam oblíbených složek uživatele.

2.5.4.4 Příklad užití - Zobrazit seznam složek

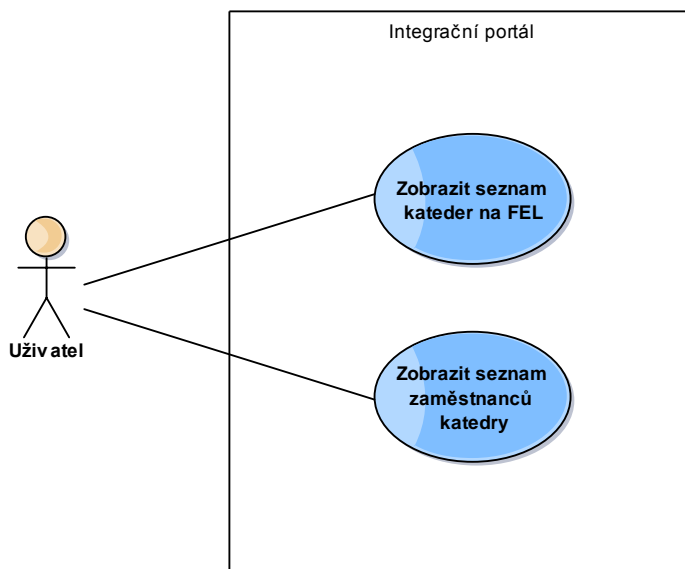
Tento případ užití zachycuje proces zobrazení seznamu složek v rámci organizační jednotky.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce zobrazit seznam složek.
2. Integrovaný portál zobrazí seznam složek organizační jednotky.

2.5.5 Případy užití KOSapi

V této kapitole jsou detailněji popsány funkční požadavky z kapitoly 2.4.4 Funkční požadavky pro integraci KOSapi. Na obrázku Obr. 6: Diagram případu užití - KOSapi jsou vyobrazeny jednotlivé případy užití. V tomto modulu vystupuje jediný aktér, obecný aktér Uživatel.



Obr. 6: Diagram případu užití - KOSapi

2.5.5.1 Příklad užití - Zobrazit seznam kateder na FEL

Tento případ užití zachycuje proces zobrazení seznamu kateder z Fakulty elektrotechnické na ČVUT.

Hlavní scénář:

1. Příklad užití začíná, když uživatel chce zobrazit seznam kateder z Fakulty elektrotechnické..
2. Integrační portál zobrazí seznam kateder na Fakultě elektrotechnické.

2.5.5.2 Příklad užití - Zobrazit seznam zaměstnanců katedry

Tento případ užití zachycuje proces zobrazení seznamu zaměstnanců určené katedry z Fakulty elektrotechnické na ČVUT.

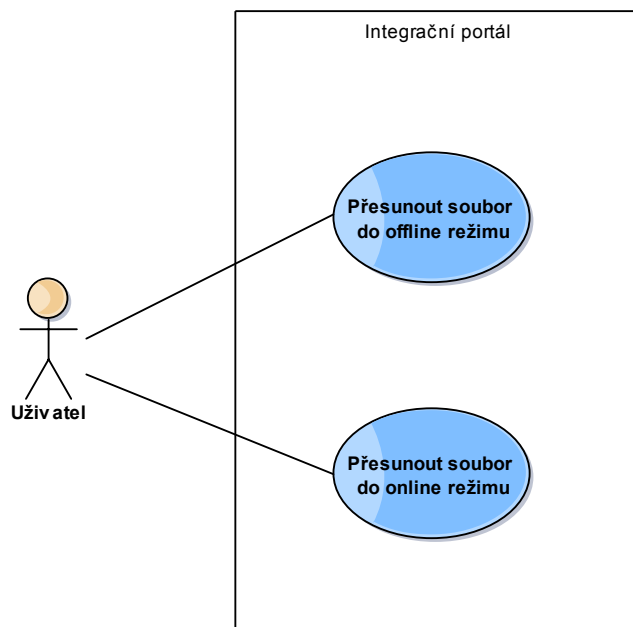
Hlavní scénář:

1. Příklad užití začíná, když uživatel chce zobrazit seznam zaměstnanců z určené katedry.
2. Integrovaný portál zobrazí seznam zaměstnanců pracujících na dané katedře z Fakulty elektrotechnické.

2.5.6 Případy užití archivu CESNET

V této kapitole jsou detailněji popsány funkční požadavky z kapitoly 2.4.5 Funkční požadavky pro archiv CESNET. Na obrázku Obr. 7: Diagram případu užití - Archiv CESNET jsou vyobrazeny jednotlivé případy užití.

Případy užití předpokládají, že je uživatel na jejich začátku na obrazovce se seznamem složek a souborů v organizační jednotce.



Obr. 7: Diagram případu užití - Archiv CESNET

2.5.6.1 Případ užití - Přesunout soubor do offline režimu

Tento případ užití zachycuje proces žádosti o přesun souboru do offline režim, neboli do archivu na pomalý velkokapacitní úložiště.

Hlavní scénář:

1. Případ užití začíná, když se uživatel rozhodne přesunout soubor do offline režimu.
2. Integrovaný portál zobrazí detail souboru.
3. Uživatel klikne na tlačítko pro přesun souboru do offline režimu.
4. Integrovaný portál provede migraci souboru na pomalé velkokapacitní medium, aktualizuje detail souboru a zobrazí zprávu o úspěšném přesunu.

2.5.6.2 Případ užití - Přesunout soubor do online režimu

Tento případ užití zachycuje proces žádosti o přesun souboru do online režim, neboli na datové úložiště s rychlými médii.

Hlavní scénář:

1. Případ užití začíná, když se uživatel rozhodne přesunout soubor do online režimu.
2. Integrovaný portál zobrazí detail souboru.
3. Uživatel klikne na tlačítko pro přesun souboru do online režimu.
4. Integrovaný portál provede migraci souboru na rychlejší medium, aktualizuje detail souboru a zobrazí zprávu o úspěšném přesunu.

3 Implementace

Backend projektu je navržen jako RESTful server, což přináší řadu výhod jako například možnost psát tenký klienty[26], který využívají poskytované rozhraní tohoto serveru.

Zde budeme popisovat výsledek implementace integračního portálu vzhledem k zadání této bakalářské práce. Níže si vypíšeme všechny požadavky, které byly do projektu v rámci bakalářské práce naimplementovány, dále si vypíšeme seznam nejdůležitějších technologií a knihoven použité v projektu a popíšeme si jak vypadá struktura zdrojového kódu.

3.1 Implementované požadavky

Již v kapitole 2.4 Popis funkčních požadavků jsme si uvedli všechny funkční požadavky, které mají být naimplementovány do projektu. Podrobně si zde popíšeme naimplementované požadavky.

Modul "Administrace organizačních jednotek" byl z minulé diplomové práce jen nasimulován daty, protože nebyla naimplementována integrace s KOSapi. Součástí této bakalářské práce byla implementace integrace s KOSapi, takže v rámci organizačních jednotek byly naimplementovány tyto požadavky:

- Přidávání uživatele do organizační jednotky
- Odebírání uživatele z organizační jednotky
- Vytvoření nové organizační jednotky
- Přidávání administrátora do organizační jednotky
- Odebrání administrátora z organizační jednotky
- Nastavení kvóty organizační jednotky

V rámci modulu "Administrace štítků" byly v této práci naimplementovány tyto požadavky:

- Přidávání nového štítku
- Upravování existujícího štítku
- Odebírání existujícího štítku
- Přiřazení štítku ke složce
- Odebírání štítku složky

V rámci modulu "Administrace oblíbených složek uživatelů" byly naimplementovány tyto požadavky:

- Přidávání oblíbených složek uživateli
- Odebírání oblíbených složek uživatele
- Zobrazení oblíbených složek uživatele

V rámci modulu "Přesun z a do archivu CESNET" již byly naimplementovány následující požadavky a tak došlo spíše k refactoringu tohoto modulu. Naimplementované požadavky modulu:

- Přesun souboru z offline režimu do online režimu a zpět

Při realizaci modulu "Integrace s KOSapi" se vyskytly menší problémy s autentizací pro přístup k datům (byl používám špatný grant type), ale po přečtení dokumentace o zabezpečení[25] byl problém vyřešen. V rámci této bakalářské práce byly naimplementovány následující požadavky:

- Výpis organizačních jednotek (kateder) na Fakultě elektrotechnické ČVUT
- Výpis zaměstnanců dané organizační jednotky (katedry)

3.2 Použité technologie

Zde se dočteme o nejpodstatnějších technologiích a knihoven, které jsou používány při implementaci integračního portálu.

3.2.1 Spring

Jde o open-source aplikační framework[13], softwarovou strukturu sloužící jako podpora při programování a vývoji softwaru a obsahující sadu knihoven, pro vývoj J2EE aplikací (součást platformy Java pro vývoj podnikových aplikací[14]). Celý framework je tvořen mnoha moduly, které jsou rozdělené podle funkcionality, takže programátor může ve svém projektu používat jen potřebné moduly.

Výhodou Spring frameworku je, že umožňuje konfiguraci pomocí XML souboru, anotací, nebo obojí. V projektu je použito jak konfigurace pomocí XML souboru, tak i pomocí anotací. Kde bylo potřeba nastavení, která mají zůstat neměnná (např. řízení transakcí nad jednotlivými metodami), bylo použito nastavení pomocí anotací. Nastavení, která by mohl správce portálu potřebovat změnit (např. URL a přihlašovací údaje k databázovému serveru), jsou uložena v XML souboru.

3.2.1.1 Spring Core

Spring core je nejdůležitější modul ve Spring frameworku, který musí být vždy zahrnut v projektu, aby bylo možné využívat další Spring moduly. Jeho hlavní funkcionalitou jsou Inversion of Control a Dependency Injection.

Inversion of Control[15] je návrhový vzor, který umožňuje přesunutí zodpovědnosti za provázání objektů z aplikace na framework. Objekty jsou pak získávány prostřednictvím Dependency Injection[16] (vsazování závislostí). To znamená, že třída využívající jiné třídy by si obvykle držela jejich reference, ale Inversion of Control této třídě dynamicky dodá odpovídající objekty.

3.2.1.2 Spring Web a Spring WebMVC

Toto jsou moduly, které zprostředkovávají funkce, které jsou používané pro webové aplikace. Usnadňují psaní webových aplikací a starají se o inicializaci všech kontejnerů pro správnou funkci kontrolérů.

3.2.1.3 Spring Security

Spring security je sbírka modulů starajících se o autentizaci a autorizaci uživatelů. Umožňuje přidat kontrolu nebo autorizaci na jednotlivá URL v rámci aplikace či přidat kontrolu na konkrétní metody tříd.

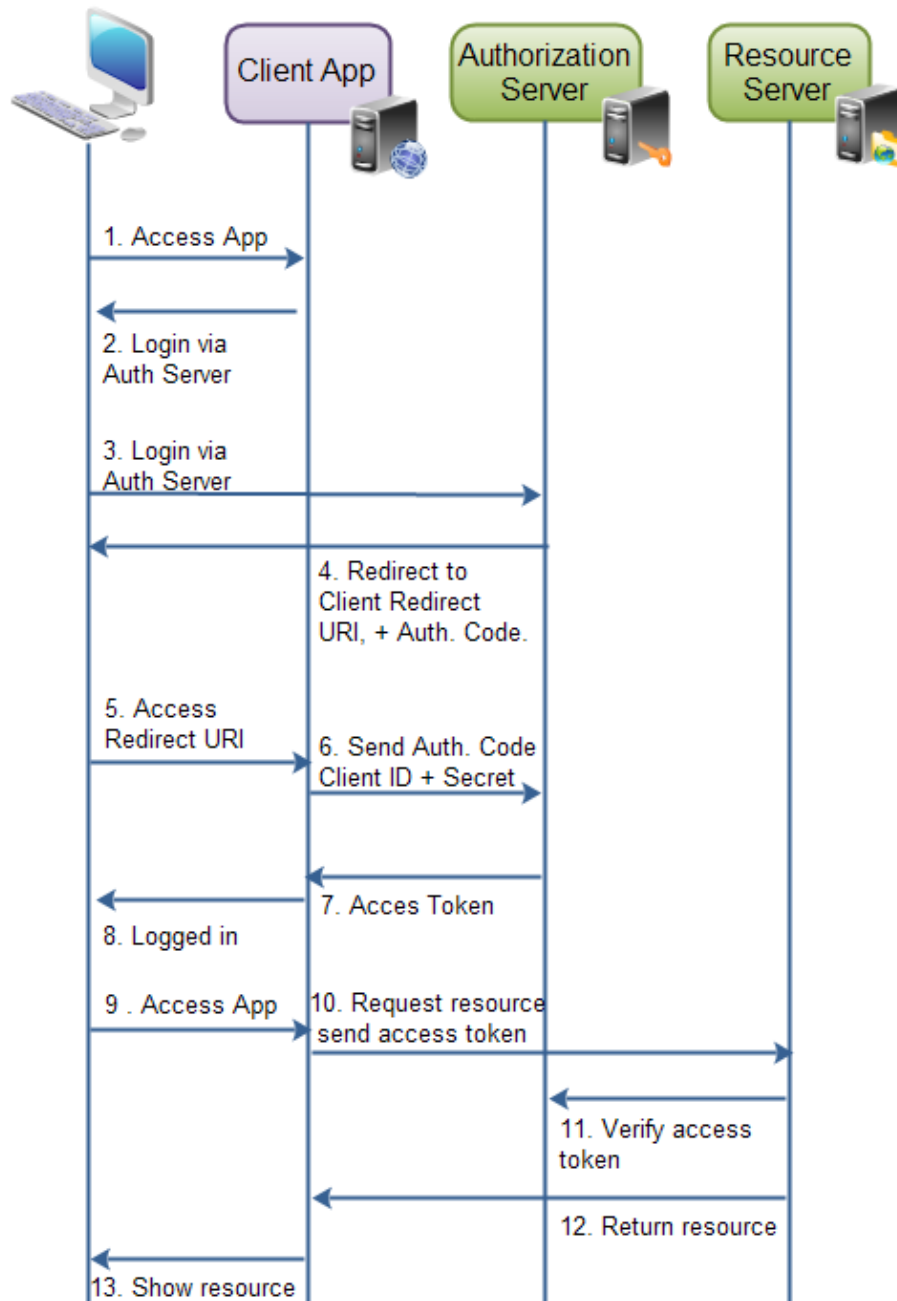
Tento modul podporuje celou řadu technologií pro autentizaci uživatele. Od klasické jednoduché autentizace pomocí HTTP hlavičky až po sofistikovanější způsoby jako je například OAuth, o kterém si povíme víc v následující kapitole.

3.2.2 OAuth

Jedná se o protokol pro autentizaci a autorizaci uživatelů, kteří se přihlašují prostřednictvím desktopových, webových a mobilních aplikací přes HTTP[17].

Klientská aplikace nejprve přihlásí uživatele proti autentizačnímu serveru, který vrátí aplikaci vygenerovaný tzv. access token. Access token je náhodný sled číslic a písmen, který je používán pro přístup k serverovým službám. Průběh autentizace je znázorněn na obr. 7: Průběh autentizace pomocí protokolu OAuth.

OAuth samozřejmě umožňuje i několik způsobů pro získání access tokenu. Integrovaný portál je nyní v roli jak autentizačního serveru tak i cílového serveru se službami a je použit způsob přihlášení Username-Password Flow[18].



Obr. 8: Průběh autentizace pomocí protokolu OAuth [24]

3.2.3 REST

REpresentational State Transfer[19] je architektura rozhraní, které není orientován procedurálně, ale datově (tzn. že určuje jak se přistupuje k datům). Všechny data mají vlastní URI identifikátor a **REST** definuje čtyři základní metody pro přístup k nim, viz tabulka 1.

Integrační portál poskytuje právě toto **REST**ové rozhraní a je využíváno grafickým klientem vytvořeným v rámci diplomové práce Kateřiny Hašlarové.

HTTP metoda	Význam metody
GET	Získání dat ze serveru
PUT	Používá se pro změnu existujících dat na serveru
POST	Odesílání dat nebo formulářů na server
DELETE	Smazání určitých dat na serveru

Tabulka 1: Význam HTTP metod u REST rozhraní

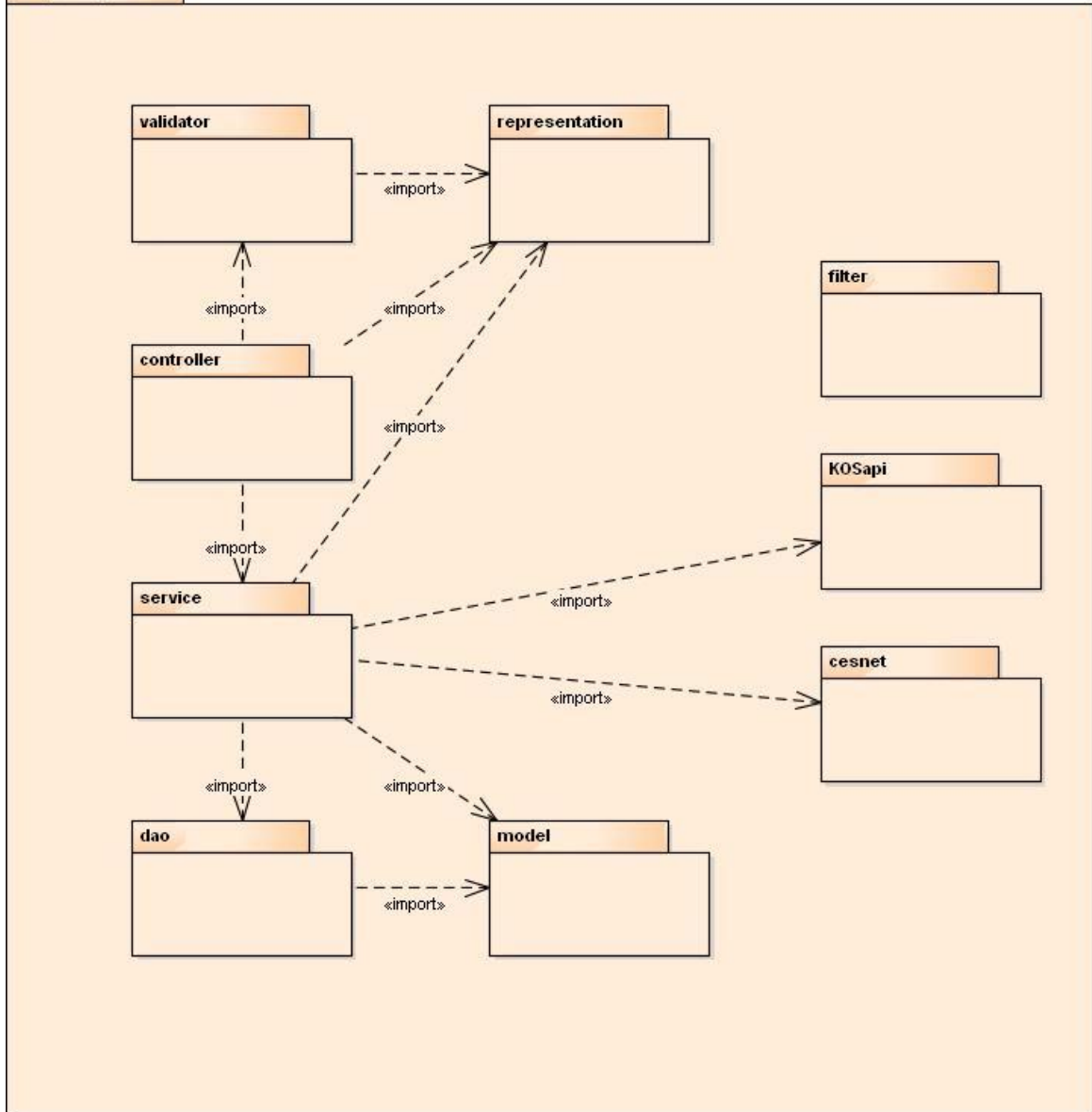
3.2.4 Maven

Apache maven je nástroj pro správu, řízení a automatizaci kompilování aplikací[20]. Usnadňuje správu jednotlivých knihoven použitých v projektu. To znamená, že už není třeba každou knihovnu ručně stahovat a vkládat do projektu, ale stačí danou knihovnu uvést do seznamu závislostí (dependencies) a Maven si pak dané knihovny do projektu stáhne a integruje. Náš projekt samotný využívá maven pro kompilování pro nasazení na server.

3.3 Balíčky integračního portálu

Zdrojový kód projektu je rozdělen do několika balíčků podle účelu daného kódu. Zde si popíšeme významy jednotlivých balíčků a vzájemnou provázanost mezi nimi. Závislost mezi balíčky je zachycena na obr. 9: Diagram balíčků integračního portálu.

Integrační portál



Obr. 9: Diagram balíčků integračního portálu

3.3.1 Balíček model

V tomto balíčku nalezneme třídy, jejichž objekty jsou ukládány do databáze. Tyto objekty se v kontroléru následně převádějí na jejich reprezentace, které se pak vrací jako odpověď klientským aplikacím.

Nalezneme zde třídy jako uživatelské údaje, role, organizační jednotky, štítky, metadata souborů a složek uložených na datových úložištích.

3.3.2 Balíček dao

Tento balíček obsahuje třídy, pomocí jejich metod se získávají data z databáze či se data ukládají do databáze. Název balíčku vychází ze slova Data Access Object, což má představovat objekty, které zajišťují přístup k databázím.

3.3.3 Balíček service

Balíček service představují třídy, které mají veškerou rozhodovací logiku a byznys logiku v integračním portálu. Pomocí nich se vytváří, upravují nebo mažou jednotlivé objekty z balíčku model. Tyto třídy využívají pro komunikaci s databází objekty z balíčku dao a mohou zprostředkovávat integraci s dalšími službami, v tomto případě využívají službu cesnet.

3.3.4 Balíček controller

Balíček controller obsahuje třídy (tzv. kontroléry), které přijímají HTTP požadavky od klientských aplikací a provádějí odpovídající metody k požadavkům. Spring odpovídající metody vybírá podle jejich URL a parametrů, na které byly HTTP požadavky poslány.

Třídy tohoto balíčku využívají objekty z balíčku representation (při převedení parametrů požadavků na objekty), validator (využívány kontroléry pro validaci správného převodu parametrů na objekt) a service (používány kontroléry pro vykonání požadované logiky aplikace).

3.3.5 Balíček representation

V balíčku representation se vyskytují třídy, na jejichž objekty jsou převáděny parametry HTTP požadavků. Tuto konverzi se Spring pokouší automaticky provádět na odpovídající proměnné objektu těchto tříd.

Spring nejprve nalezne odpovídající proměnnou podle jejího jména, které odpovídá jménu parametru v požadavku. Pokud neexistuje, je vrácena kontroléru chyba.

Pro běžné datové typy probíhá konverze automaticky. Jedná-li se typem o třídu, pak musí tato třída obsahovat bezparametrický konstruktor a setter metody pro vlastní proměnné.

Objekty vytvořené ze tříd tohoto balíčku jsou využívány převážně třídami z balíčku controller, když potřebují parametry HTTP požadavků převést do objektů a tyto objekty jsou pak využívány pro komunikaci s třídami v balíčku service.

3.3.6 Balíček validator

Balíček validator obsahuje třídy, které jsou používány kontroléry pro validaci dat v parametrech HTTP požadavků. Validátor ve většině případů není potřeba, neboť Spring provádí základní kontrolu dat automaticky.

V projektu je však validátor použit jen pro reprezentaci třídy uživatelské účty, kde tato reprezentace obsahuje seznam uživatelských rolí uživatele jako seznam jmen těchto rolí. Zde je nutné zkontrolovat, jestli tento seznam neobsahuje jména rolí, které neexistují.

3.3.7 Balíček filter

Balíček filter obsahuje třídu implementující rozhraní Filter. Filtry[21] v J2EE dynamicky zachycují HTTP požadavky nebo odpovědi serveru a mohou na nimi provádět operace.

V projektu je filtr využit k tomu, aby klientské aplikace věděly povolené HTTP metody (GET, POST, PUT, OPTIONS, DELETE) a povolené HTTP hlavičky pro požadavky.

3.3.8 Balíček cesnet

V tomto balíčku jsou obsaženy třídy, které slouží pro komunikaci s CESNET úložištěm. Obsahuje implementaci SSH a SFTP protokoly. SSH protokol je využit pro komunikaci s CESNET serverem a SFTP protokol umožňuje přístup k uloženým souborům na úložišti.

Třídy z balíčku cesnet jsou využity třídami z balíčku service, které mají na starosti správu souborů a složek integračního portálu.

4 Popis API integračního portálu

V této kapitole si popíšeme aplikační rozhraní integračního portálu, které bylo vytvořeno v rámci této bakalářské práce. Budeme popisovat dostupné metody pro jednotlivé URI[30], parametry těchto metod a možné návratové hodnoty, které jsou ve formě HTTP statutů[31].

Pro lepší přehlednost popisů si rozdělíme popisy na kapitoly podle jednotlivých modulů funkcionality. Půjde tedy o kapitoly 5.1 Štítky, 5.2 Organizační jednotky, 5.3 Oblíbené složky, 5.4 archiv CESNET a 5.5 KOSapi.

Na Apiary (online služba pro dokumentace RESTových API[32]) je interaktivní forma tohoto popisu, které je dostupné ze stránky <http://docs.cesnetintegracniportal.apiary.io/>.

4.1 Štítky

Štítky slouží pro označování souborů a složek podle jejich důležitosti. Slouží k tomu následující metody:

- **GET** `/rest/v0.2/label` - vrátí seznam štítků přihlášeného uživatele
- **POST** `/rest/v0.2/label` - vytvoří nový štítek
- **PUT** `/rest/v0.2/label/{id}` - upraví údaje štítku s daným id
- **DELETE** `/rest/v0.2/label/{id}` - smaže štítek s daným id
- **POST** `/rest/v0.2/space/{spaceId}/folder/{folderId}/addLabel` - přidá štítek na složkou s daným id (folderId) v úložišti s daným id (spaceId). Požadavek odesílá id štítku, který chceme použít, jako JSON objekt.
- **POST** `/rest/v0.2/space/{spaceId}/folder/{folderId}/removeLabel` - odebere štítek ze složky s daným id (folderId) v úložišti s daným id(spaceId). Požadavek odesílá id odebíraného štítku jako JSON objekt.
- **POST** `/rest/v0.2/space/{spaceId}/file/{fileId}/addLabel` - - přidá štítek na soubor s daným id (fileId) v úložišti s daným id (spaceId). Požadavek odesílá id štítku, který chceme použít, jako JSON objekt.
- **POST** `/rest/v0.2/space/{spaceId}/file/{fileId}/removeLabel` - odebere štítek ze souboru s daným id (fileId) v úložišti s daným id(spaceId). Požadavek odesílá id odebíraného štítku jako JSON objekt.

Metody pro vytvoření a upravení štítku a metody pro přidání nebo odebrání štítku ze souboru či složky požadují, aby v těle HTTP požadavku byl platný JSON objekt. Pro metody pro vytvoření nebo úpravy štítku musí JSON objekt obsahovat proměnné *text* a *color*. Objekt může vypadat asi takhle:

```
{  
  text: "work",  
  color: "#ff0000"  
}
```

Pro metody přidání a odebrání štítku ze souboru nebo složky musí tělo požadavku obsahovat *id* štítku. Tělo požadavku vypadá asi takhle:

```
{  
  "id": 1  
}
```

4.2 Organizační jednotky

Zde budou vypsány metody umožňující administraci organizačních jednotky. Seznam použitelných metod je zde:

- **GET** `/rest/v0.2/unit` - vypíše seznam organizačních jednotek existujících v integračním portálu.
- **GET** `/rest/v0.2/unit/{id}` - vrátí detail organizační jednotky s danou id
- **POST** `/rest/v0.2/unit/{id}/nameChange` - změni název organizační jednotky s danou id
- **POST** `/rest/v0.2/unit/{id}/quotaChange` - změni velikost úložného prostor, který může organizační jednotka používat.
- **POST** `/rest/v0.2/unit/{id}/adminsAssignment` - přidá administrátory do organizační jednotky s danou id.

I zde některé metody potřebují mít JSON objekt v těle HTTP požadavku. U metody pro změnu názvu musí tělo obsahovat parametr *name*. Tělo může vypadat asi takto:

```
{  
  "name": "Katedra počítačů"  
}
```

Metoda pro změnu velikosti úložného prostoru naopak musí mít parametr *quota*:

```
{  
  "quota": 100000  
}
```

A metoda pro přidání administrátorů do jednotky má pole parametrů *admins*:

```
{  
  "admins": [1, 2]  
}
```

4.3 Oblíbené složky

Jsou to jen složky, ke kterým chce mít uživatel rychlý přístup. Integrovaný portál by měl poskytovat:

- **POST** `/rest/v0.2/space/{spaceId}/folder/{folderId}/addFavorite` - přidá složku s danou id (folderId) v úložišti s daným id (spaceId) uživateli mezi oblíbené složky
- **POST** `/rest/v0.2/space/{spaceId}/folder/{folderId}/removeFavorite` - odebere složku s danou id (folderId) v úložišti s daným id (spaceId) uživateli z oblíbených složek.

4.4 Archiv CESNET

Zde budou vypsány jen metody umožňující přesun souboru či celé složky do online nebo offline módu. V rámci práce byla implementace psána jen pro datové úložiště CESNET. Pro archivaci souboru z jiných úložišť se bude muset implementace v budoucnu upravit. Integrovaný portál by měl poskytovat následující metody k archivaci:

- **POST** `/rest/v0.2/space/{spaceId}/file/{fileId}/online` - přesune soubor s daným id(fileId) z úložiště (spaceId) do online módu
- **POST** `/rest/v0.2/space/{spaceId}/file/{fileId}/offline` - - přesune soubor s daným id(fileId) z úložiště (spaceId) do offline módu
- **POST** `/rest/v0.2/space/{spaceId}/folder/{folderId}/online` - přesune všechny soubory ve složce s daným id (folderId) v úložišti (spaceId) do online módu.
- **POST** `/rest/v0.2/space/{spaceId}/folder/{folderId}/offline` - přesune všechny soubory ve složce s daným id (folderId) v úložišti (spaceId) do offline módu

4.5 KOSapi

Integrovaný portál zatím využívá rozhraní KOSapi, aby získal z databáze existující katedry a zaměstnance v jednotlivých katedrách na Fakultě elektrotechnické ČVUT. Zde jsou vypsány metody, které by zatím integrovaný portál měl poskytovat:

- **GET** `/rest/v0.2/kos/departments` - vrátí seznam jmen a jejich unikátních kódů všech kateder na Fakultě elektrotechnické.
- **GET** `/rest/v0.2/kos/{departmentCode}/teachers` - vrátí seznam všech zaměstnanců na katedře, která má daný unikátní kód(departmentCode).

5 Testování

Testování je nedílnou součástí vývoje softwaru. Používá se pro ověření funkčnosti jednotlivých modulů, tříd či metod. Existuje spousta způsobů jak kód otestovat, ale v našem projektu jsme využili jen jeden způsob. Je používáno takzvané unit testování[27].

Většina času byla věnována jen implementaci a psaní práce a na psaní testů nezbylo moc času. Vzhledem k tomu, že testy se psaly jen v přestávkách od psaní kódu nebo bakalářské práce a ještě se muselo nastudovat psaní testů pro webové aplikace, nezbylo moc času na řešení chyb v kódu.

5.1 Unit testování

Jde o automatizované testování a ověřování správné funkčnosti implementace dílčích částí (jednotek) zdrojového kódu. Jednotka zdrojového kódu, z pohledu objektově orientovaného programování, se považuje třída nebo konkrétní metoda.

V integračním portálu je tato metoda testování využívána. Testuje se s ní většina implementace, i ta která je v rámci bakalářské práce uskutečněna. V rámci testování je používáno takzvané mockování objektů[28]. Mock objekty jsou objekty, které napodobují chování skutečných objektů (dají se přirovnat ke crashtestovému panáku, který má skoro stejné vlastnosti jako skutečný člověk a je použit místo skutečného člověka na otestování chování auta při různých typech simulovaných nehod). Výhodou těchto mock objektů je, že se jim dá nadefinovat různé odpovědi na různé podněty od testovaných objektů. Používají se třeba pro nasimulování databáze, ze které se získávají data, nebo nasimulování tříd, jejichž metody testovaná třída využívá.

V následujících podkapitolách nalezneme moduly testované pomocí unit testů.

5.1.1 Testování štítků

V této kapitole se dozvíme, co všechno se testovalo v rámci štítků a zda testy prošly bez chyby. Následuje seznam testovaných funkcí v rámci štítků:

- Vytvoření nového štítku - prošlo bez chyby
- Vyhledání štítku podle id - prošlo bez chyby
- Vyhledání štítku podle jména - prošlo bez chyby
- Vrácení seznamu všech štítků - prošlo bez chyby
- Vrácení seznamu štítků patřící uživateli - prošlo bez chyby
- Upravení názvu a barvy štítku - prošlo bez chyby
- Smazání existujícího štítku - prošlo bez chyby
- Připnutí štítku na složku - prošlo bez chyby
- Odepnutí štítku ze složky - prošlo bez chyby
- Připnutí štítku na soubor - prošlo bez chyby
- Odepnutí štítku ze souboru - prošlo bez chyby

5.1.2 Testování organizačních jednotek

Testování organizačních jednotek nebylo úspěšné, protože při vytváření jednotky mi test vyhodil chybu 'Pro sloupec "MEMBERS_USER_ID" není hodnota NULL povolena'. Tudíž jsem nemohl dále testovat, když nešlo vytvořit organizační jednotku. Snažil jsem se chybu opravit, ale nebyl na to čas. Zde je seznam funkcí, které jsem chtěl otestovat:

- Vytvoření nové organizační jednotky
- Upravení názvu organizační jednotky
- Změna kvóty organizační jednotky
- Vrácení seznamu všech organizačních jednotek
- Vyhledání organizační jednotky podle id
- Vyhledání organizační jednotky podle jména
- Přidání administrátora do organizační jednotky
- Odebrání administrátora z organizační jednotky
- Přidání člena do organizační jednotky
- Odebrání člena z organizační jednotky

5.1.3 Testování oblíbených složek

Zde nalezneme seznam testovaných funkcí, který modul oblíbené složky umožňuje.

- Přidat složku mezi oblíbené u uživatele - prošlo bez chyby
- Odebrat složku z oblíbených u uživatele - prošlo bez chyby

5.2 Testování implementace KOSapi

V rámci bakalářské práce byla implementace KOSapi pro získání seznamu kateder z Fakulty elektrotechnické na ČVUT nebo pro získání zaměstnanců dané katedry. Veškerá data se získávají z databáze KOSu (to znamená, že je potřeba autorizovaný přístup).

Vzhledem k tomu, že mě nenapadl způsob jak danou service třídu, která se stará o autorizaci přístupu a o poskytnutí požadovaných údajů z databáze KOS, správně testovat pomocí unit testů, vytvořil jsem testovací server, který danou service třídu implementuje a zobrazuje výsledky použití jednotlivých metod této service třídy.

Jde tedy o otestování metod, které service třída poskytuje. Díky tomuto testu jsem se ujistil, že funguje autorizace aplikace pro přístup k databázi KOS a že fungují metody zajišťující získání seznamu kateder nebo zaměstnanců dané katedry. Dále je popsán postup použitý k otestování těchto metod.

Na obrázku Obr. 10: Webové rozhraní testovacího serveru můžete vidět jednoduché rozhraní. Po kliknutí na odkaz "Vypis kateder" se odešle požadavek na backend testovacího serveru, jehož součástí je service třída, kterou chceme otestovat, který vrátí odpověď na daný požadavek. V našem případě přijde seznam kateder, které jsou na Fakultě elektrotechnické na ČVUT, viz obrázek Obr. 11: Tvar odpovědi na požadavek seznamu kateder. Metoda starající se o poskytnutí seznamu kateder z databáze KOS funguje správně a metoda, která se stará o přístup k databázi, také funguje správně. Kdyby jedna z těchto metod nefungovala korektně, vyskočil by error ze strany testovacího serveru nebo by jsme dostali HTTP error kód ze strany autorizačního serveru na ČVUT.

Pro otestování funkčnosti metody zajišťující výpis zaměstnanců dané katedry použijeme druhý odkaz "Vypis ucitelu v katedre". Nejprve musíme zadat parametr kod katedry (aby service třída hledala zaměstnance v námi určené katedře) a offset (udává od kolikátého zaměstnance má začínat výpis, maximální počet zaměstnanců na stránku je totiž 1000). Po zadání parametru (18102 a 0) a kliknutí na odkaz "Vypis ucitelu v katedre" se odešle požadavek se zadanými parametry a server vrátí odpověď, viz obrázek Obr. 12: Tvar odpovědi na požadavek seznamu zaměstnanců dané katedry.

Testování service třídy proběhlo úspěšně. Všechny metody fungují tak jak mají. Vždy jsme dostali požadovaná data a ve formátu, ve kterém přijít měli.

Test server implementace KOSapi

[Vypis kateder](#)

zadejte kod katedry	offset	Vypis ucitelu v katedre
---------------------	--------	---

Obr. 10: Webové rozhraní testovacího serveru

```
<html>
  <body>
    <atom:feed ...>
      <atom:entry>
        <atom:title>katedra softwarového inženýrství</atom:title>
        <atom:content>
          <code>18102</code>
        </atom:content>
      </atom:entry>
      <atom:entry>...</atom:entry>
      <atom:entry>...</atom:entry>
      <atom:entry>...</atom:entry>
      <atom:entry>...</atom:entry>
      ...
      <atom:entry>...</atom:entry>
    </atom:feed>
  </body>
</html>
```

Obr. 11: Tvar odpovědi na požadavek seznamu kateder

```

<html>
  <body>
    <atom:feed ...>
      <atom:entry>...</atom:entry>
      ...
      <atom:entry>...</atom:entry>
      <atom:entry>
        <atom:title>Ing. Adam Šenk</atom:title>
        <atom:content>
          <username>senkadam</username>
          <email>senkadam@fit.cvut.cz</email>
        </atom:content>
      </atom:entry>
      <atom:entry>...</atom:entry>
      <atom:entry>...</atom:entry>
      <atom:entry>...</atom:entry>
      <atom:entry>...</atom:entry>
      ...
      <atom:entry>...</atom:entry>
    </atom:feed>
  </body>
</html>

```

Obr. 12: Tvar odpovědi na požadavek seznamu zaměstnanců dané katedry

6 Závěr

V rámci této bakalářské práce bylo rozšířeno nebo upraveno API projektu integrační portál o podporu KOSapi, možnost používání štítků, označení oblíbených složek a manipulaci dat na datovém úložišti.

Jednotlivé požadavky byly popsány textově i graficky pomocí diagramu případu užití. Dále byla popsána nynější struktura zdrojového kódu a současná architektura integračního portálu.

Všechny požadavky se otestovat automatickými testy nepovedlo, protože testy se psaly jako poslední a nezbylo na ně moc času. Do časové tísně jsem se dostal, kvůli implementování KOSapi a přesunu souboru z a do archivu CESNET.

U implementace KOSapi bylo náročné naimplementovat autorizaci aplikace proti autorizačnímu serveru vyvíjenému na Fakultě informačních technologií, protože dokumentace tohoto serveru není až tak přehledná a většina publikací o OAuth je v angličtině, tak porozumění problematice chvíli trvalo. Proto bylo naimplementováno jen výpis seznamu kateder Fakulty elektrotechnické a seznam zaměstnanců dané katedry, pro zbytek implementace nezbyl čas.

U implementace přesunu souboru z a do archivu CESNET se zas protáhlo učení používání pro mě nové knihovny, která poskytuje možnost použití SSH protokolu.

Cílem práce bylo seznámení se z projektem integrační portál a diplomovými pracemi studentů Petra Strnada a Kateřiny Hašlarové a rozšíření stávající API o další funkcionality. Výsledkem práce je lehká úprava zdrojového kódu, rozšíření rozhraní integračního portálu a ověření funkčnosti těchto implementací.

Použité zdroje

- [1] *Strnad, P.* Integrační portál pro sdílení a zálohování dat. 2014 ČVUT, diplomová práce
- [2] *Hašlarová, K.* Front-end pro portál pro sdílení souborů. 2014 ČVUT, diplomová práce
- [3] *Google Drive* [online]. [cit. 17.5.2015]. Dostupné z https://www.google.com/intl/cs_CZ/drive/
- [4] *Microsoft OneDrive* [online]. [cit. 17.5.2015]. Dostupné z <https://onedrive.live.com/about/cs-cz/>
- [5] *Dropbox* [online]. [cit. 17.5.2015]. Dostupné z <https://www.dropbox.com/>
- [6] *Systém FELid* [online]. [cit. 17.5.2015]. Dostupné z <https://wiki.fel.cvut.cz/net/felid/about>
- [7] *SFTP protokol* [online]. [cit. 17.5.2015]. Dostupné z http://cs.wikipedia.org/wiki/SSH_file_transfer_protocol
- [8] *SFTP protokol* [online]. [cit. 17.5.2015]. Dostupné z <http://en.wikipedia.org/wiki/FTPS>
- [9] *SSH protokol* [online]. [cit. 17.5.2015]. Dostupné z http://cs.wikipedia.org/wiki/Secure_Shell
- [10] *Alfresco* [online]. [cit. 17.5.2015]. Dostupné z [http://en.wikipedia.org/wiki/Alfresco_\(software\)](http://en.wikipedia.org/wiki/Alfresco_(software))
- [11] *ECM* [online]. [cit. 17.5.2015]. Dostupné z http://en.wikipedia.org/wiki/Enterprise_content_management
- [12] *Projekt KOSapi* [online]. [cit. 17.5.2015]. Dostupné z <https://kosapi.fit.cvut.cz/projects/kosapi/wiki>
- [13] *Spring Framework* [online]. [cit. 17.5.2015]. Dostupné z http://cs.wikipedia.org/wiki/Spring_Framework
- [14] *J2EE* [online]. [cit. 17.5.2015]. Dostupné z http://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition
- [15] *Inversion of Control* [online]. [cit. 17.5.2015]. Dostupné z <https://sqdw.signaly.cz/0804/inversion-of-control-dependency>
- [16] *Dependency Injection* [online]. [cit. 17.5.2015]. Dostupné z http://en.wikipedia.org/wiki/Dependency_injection
- [17] *Hypertext Transfer protokol* [online]. [cit. 17.5.2015]. Dostupné z http://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol
- [18] *OAuth 2.0 Username-Password Flow* [online]. [cit. 17.5.2015]. Dostupné z https://help.salesforce.com/HTViewHelpDoc?id=remoteaccess_oauth_username_password_flow.htm&language=en_US

- [19] *Representational State Transfer* [online]. [cit. 17.5.2015]. Dostupné z <http://en.wikipedia.org/wiki/Representational_state_transfer>
- [20] *Apache Maven Project* [online]. [cit. 17.5.2015]. Dostupné z <<https://maven.apache.org/what-is-maven.html>>
- [21] *The Essentials of Filters* [online]. [cit. 17.5.2015]. Dostupné z <<http://www.oracle.com/technetwork/java/filters-137243.html>>
- [22] *Atom (standard)* [online]. [cit. 17.5.2015]. Dostupné z <[http://en.wikipedia.org/wiki/Atom_\(standard\)](http://en.wikipedia.org/wiki/Atom_(standard))>
- [23] *Extensible Markup Language* [online]. [cit. 17.5.2015]. Dostupné z <http://cs.wikipedia.org/wiki/Extensible_Markup_Language>
- [24] *OAuth 2.0 Authorization* [online]. [cit. 17.5.2015]. Dostupné z <<http://tutorials.jenkov.com/oauth2/authorization.html>>
- [25] *Dokumentace OAuth 2 na ČVUT* [online]. [cit. 17.5.2015]. Dostupné z <<https://rozvoj.fit.cvut.cz/Main/oauth2>>
- [26] *Tenký klient* [online]. [cit. 17.5.2015]. Dostupné z <http://cs.wikipedia.org/wiki/Tenk%C3%BD_klient>
- [27] *Unit testing* [online]. [cit. 17.5.2015]. Dostupné z <http://cs.wikipedia.org/wiki/Unit_testing>
- [28] *Mock Object* [online]. [cit. 17.5.2015]. Dostupné z <http://en.wikipedia.org/wiki/Mock_object>
- [29] *Blažek, J.* Frontend pro integrační portál. 2015 ČVUT, bakalářská práce
- [30] *Uniform Resource Identifier* [online]. [cit. 20.5.2015]. Dostupné z <http://cs.wikipedia.org/wiki/Uniform_Resource_Identifier>
- [31] *List of HTTP status codes* [online]. [cit. 20.5.2015]. Dostupné z <http://en.wikipedia.org/wiki/List_of_HTTP_status_codes>
- [32] *Apiary* [online]. [cit. 20.5.2015]. Dostupné z <<https://apiary.io/>>

Příloha A

Obsah příloženého CD

Příložené CD obsahuje složky:

- **text:** obsahuje text práce ve formátu PDF a naskenované zadání práce
- **source:** obsahuje ZIP soubor se zdrojovým kódem aplikace