

Sem vložte zadání Vaší práce.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA . . . SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Návrh internetového systému pro podporu komunitního vzdělávání

Ing. Hana Lukešová

Vedoucí práce: Ing Pavel Náplava

12. února 2015

Poděkování

Ráda bych poděkovala především svému vedoucímu práce Ing. Pavlu Náplavovi za řešení zajímavé problematiky, užitečné rady a pravidelné konzultace. Dále bych ráda poděkovala své rodině za podporu během psaní této práce a celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 12. února 2015

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2015 Hana Lukešová. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Lukešová, Hana. *Návrh internetového systému pro podporu komunitního vzdělávání*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Bakalářská práce se věnuje analýze a návrhu internetového informačního systému pro podporu komunitního vzdělávání. Pro tyto účely využívá agregace cloudových služeb do systému. V první části se nachází teoretický úvod do problematiky vzdělávání, komunit a konkrétních cloudových řešení. Druhá část navazuje analýzou a návrhem systému, který realizuje propojení těchto služeb.

Klíčová slova komunita, cloud, vzdělávání, agregace

Abstract

Bachelor thesis focuses on concept of internet education system for various communities. System uses aggregation of existing cloud services for these purposes. First part contains theoretical base to education, communities and exact cloud services. Second part follows with analysis and concept of the system bringing these services together.

Keywords community, cloud, education, aggregation

Obsah

| | |
|--|-----------|
| Úvod | 1 |
| Struktura práce | 1 |
| 1 Vzdělávání | 3 |
| 1.1 Úloha internetu ve vzdělávání | 4 |
| 2 Internetové služby | 7 |
| 2.1 Cloudová řešení | 7 |
| 2.2 Vybrané cloudové služby pro účely vzdělávání | 10 |
| 3 YouTube | 13 |
| 3.1 Popis YouTube API | 13 |
| 4 Scribd | 17 |
| 4.1 Popis Scribd API | 17 |
| 5 Imgur | 21 |
| 5.1 Popis Imgur API | 21 |
| 6 SoundCloud | 23 |
| 6.1 Popis SoundCloud API | 23 |
| 7 Dropbox | 25 |
| 7.1 Popis Dropbox API | 25 |
| Shrnutí teoretické části | 27 |
| 8 Analýza | 29 |
| 8.1 Evidence uživatelů | 30 |
| 8.2 Evidence informací | 32 |
| 8.3 Model požadavků | 32 |

| | | |
|----------|--|-----------|
| 8.4 | Model případů užití | 35 |
| 8.5 | Activity diagramy | 42 |
| 8.6 | Doménový model | 47 |
| 9 | Návrh | 53 |
| 9.1 | Model architektury | 53 |
| 9.2 | Model komunikace | 53 |
| 9.3 | Datový model | 56 |
| 9.4 | Možnosti praktického využití | 61 |
| | Shrnutí praktické části | 63 |
| | Závěr | 65 |
| | Literatura | 67 |
| A | Seznam použitých zkratk | 71 |
| B | Obsah přiloženého CD | 73 |

Seznam obrázků

| | | |
|------|--|----|
| 1.1 | Pilíře pro koncept vzdělávání. | 4 |
| 2.1 | Schéma obecného principu cloudových řešení. [16] | 8 |
| 2.2 | Výhody a nevýhody cloudových řešení. [21] | 9 |
| 7.1 | Prvotní návrh internetového informačního systému pro podporu komunitního vzdělávání. | 28 |
| 8.1 | Hierarchie uživatelů v systému. | 30 |
| 8.2 | Životní cyklus uživatele. | 31 |
| 8.3 | Model požadavků. | 32 |
| 8.4 | Funkční požadavky. | 33 |
| 8.5 | Nefunkční požadavky. | 34 |
| 8.6 | Model případů užití. | 35 |
| 8.7 | UC1 Správa uživatelů. | 36 |
| 8.8 | UC2 Vkládání a editace informací. | 37 |
| 8.9 | UC3 Vyhledávání a propojování informací. | 39 |
| 8.10 | UC4 Kontrola aktuálnosti dat. | 40 |
| 8.11 | UC5 Zasílání informačních zpráv. | 41 |
| 8.12 | Účastníci. | 42 |
| 8.13 | Act1 Registrace uživatele do databáze. | 43 |
| 8.14 | Act2 Zasílání novinek. | 44 |
| 8.15 | Act3 Vkládání dat. | 44 |
| 8.16 | Act4 Editace a mazání dat. | 45 |
| 8.17 | Act5 Kontrola aktuálnosti dat. | 46 |
| 8.18 | Act6 Životní cyklus uživatele. | 47 |
| 8.19 | Doménový model – evidence uživatelů | 48 |
| 8.20 | Doménový model – evidence dat | 48 |
| 9.1 | Model architektury | 54 |
| 9.2 | Model komunikace – Vkládání a editace dat | 55 |

| | | |
|-----|---|----|
| 9.3 | Model komunikace – registrace uživatele | 56 |
| 9.4 | Model komunikace – vyhledání a zobrazení příspěvků. | 57 |
| 9.5 | Datový model – evidence uživatelů. | 58 |
| 9.6 | Datový model – evidence dat. | 60 |

Seznam tabulek

| | | |
|------|---|----|
| 8.1 | Atributy entity Uživatel. | 49 |
| 8.2 | Atributy entity Přihlašovací údaje. | 49 |
| 8.3 | Atributy entity Příspěvatel. | 49 |
| 8.4 | Atributy entity Správce kategorie. | 50 |
| 8.5 | Atributy entity Administrátor. | 50 |
| 8.6 | Atributy entity Kategorie. | 50 |
| 8.7 | Atributy entity Příspěvek. | 51 |
| 8.8 | Atributy entity Editace. | 51 |
| 8.9 | Atributy entity Služba. | 51 |
| 8.10 | Atributy entity Typ příspěvku. | 52 |
| 9.1 | Atributy tabulky Uzivatel. | 57 |
| 9.2 | Atributy tabulky PrihlasovaciUdaje. | 58 |
| 9.3 | Atributy tabulky SkupinaUzivatele. | 59 |
| 9.4 | Atributy tabulky Kategorie. | 59 |
| 9.5 | Atributy tabulky Prispivek. | 59 |
| 9.6 | Atributy tabulky Editace. | 60 |
| 9.7 | Atributy tabulky Sluzba. | 61 |

Úvod

Bakalářská práce se zabývá problematikou internetového vzdělávání a s ním spojeným využíváním cloudových služeb.

Cloudové služby jsou trendem současnosti i budoucnosti vzhledem k jejich snadnému použití a malým nárokům především na hardwarové zdroje z hlediska uživatele. Tyto služby mohou být využity také k rozšíření možností dalšího vzdělávání pro zájmové skupiny (komunity). Komunitou se v tomto případě myslí jakákoliv skupina se zájmem o obohacení svých znalostí. S rozšířením internetu do domácností je možné přenést i tuto sféru do elektronické oblasti.

Cílem práce je navrhnout informační systém, který bude v sobě agregovat cloudové služby využitelné pro oblast vzdělávání.

Vzhledem k charakteru práce je většina zdrojů v elektronické podobě. Knihy a další literární podklady jsou v této oblasti méně používané.

Struktura práce

První kapitola se zabývá obecnou problematikou vzdělávání, kde se zabývá konceptem vzdělávacích informačních systémů, charakteristikou komunit a možnostmi využití těchto principů v informačních systémech.

Ve druhé kapitole se zaměřuji na poskytování internetových služeb, především pak cloudových řešení. Podrobněji popisují ta řešení, která se dají využít z hlediska vzdělávání.

Ve třetí až sedmé kapitole se podrobněji věnuji vybraným cloudovým řešením. Jedná se o obecné popisy služeb a specifikace jejich veřejných API rozhraní.

Následující dvě kapitoly obsahují samotnou analýzu a návrh informačního systému pro podporu komunitního vzdělávání. Jedná se o agregaci cloudových

ÚVOD

služeb a možnosti jejich využití v daném kontextu. Analytická část je zaměřena na obecnou funkčnost systému a specifikaci požadavků, případů užití a procesů. Zároveň je zde uveden návrh databáze včetně doménového modelu s popisem jednotlivých entit. Návrhová část se zabývá podrobněji samotnou implementací databáze a propojení s cloudovými službami z hlediska technologického řešení.

Vzdělávání

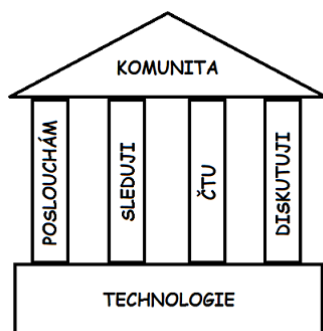
Se stále se rozvíjejícími lidskými znalostmi je na vzdělávání kladen čím dál větší důraz. Ke každé činnosti v současném světě je potřeba mít určité znalosti [5]. Školy jsou odborně zaměřovány na úzké oblasti s tím, že ostatních oborů se dotýkají pouze okrajově, především z hlediska souvislostí s daným tématem. Člověk tedy získává ucelené a podrobné znalosti, ale pouze v úzkém spektru toho, co je známo. Školy samozřejmě umožňují i studium nesouvisejících předmětů v rámci volitelných, ale to je umožněno pouze po dobu, kdy je dotyčný studentem.

Pokud se chce člověk dozvídat více i po opuštění takových institucí, je třeba hledat v jiných zdrojích. Typické místo je především knihovna, kde se informace nacházejí v podobě odborných tisků, které je možné pročitat a doplňovat tak své dosavadní znalosti [2]. Při větším rozsahu požadovaného vzdělání ale může být tato činnost dosti náročná a ne vždy dostatečná. Ne každá kniha nebo jiný písemný materiál může obsahovat přesně to, co člověk hledá a bez možnosti textového vyhledávání je to běh na dlouhou trať.

Vzdělávání má také své zákonitosti a dá se popsat v rámci čtyř pilířů, na nichž je založen. Na obrázku 1.1 je znázorněn koncept sdílení znalostí. Na tomto principu jsou také založeny vzdělávací informační systémy. Pro příklad lze uvést některé systémy, které těchto pilířů využívají:

- edux [3]
- moodle [4]

Navrhovaný systém je zaměřen na podporu komunitního vzdělávání. Komunitou je v tomto případě myšlena jakákoliv zájmová skupina, která sdílí touhu po šíření znalostí. Nejedná se pouze o vytváření jednotlivých datových záznamů, ale zároveň jde i o jejich propojení. Toho lze dosáhnout například schůzkami v rámci sdílení znalostí, kdy jednotliví účastníci přispějí svou částí.



Obrázek 1.1: Pilíře pro koncept vzdělávání.

1.1 Úloha internetu ve vzdělávání

S příchodem tzv. moderních technologií a především internetu se ale otevřela i další možnost, jak se nadále průběžně vzdělávat, pokud o to má člověk zájem [1]. Existuje množství publikací dostupných v elektronické podobě ve formátu pdf. To umožňuje textové vyhledávání a další manipulaci s textem, která urychluje zjištění přesně toho, co člověk potřebuje.

V rámci vzdělávání se využívá také jiných zdrojů, jako jsou audiozáznamy, videozáznamy a další podpůrné služby. I v těchto případech lze najít ekvivalenty v elektronické podobě podobně, jako jsem to již zmínila u textových záznamů.

Internetové vyhledávače navíc zajišťují téměř okamžitý přístup ke všem informacím, které jsou na dané téma k dispozici a to nejen v rodném jazyce, ale také i v jiných jazycích. Záleží, jaké parametry si při zadávání člověk zvolí. Vyhledávání funguje na základě zadaných klíčových slov a výsledky jsou řazeny podle relevance [6].

V současné době se také začíná rozšiřovat možnost internetových kurzů pořádaných především univerzitami. Ty umožňují uživateli v rámci webových stránek další vzdělávání formou on-line přednášek přímo z prohlížeče svého PC, tabletu či jiného zařízení [7]. Informace jsou podávány nejrůznější formou od prostého textu po videa, kde vyučující vysvětluje danou látku. Vzhledem k těmto možnostem je snadnější získat přesné informace, které člověk hledá i v daném kontextu. Jsou to zdroje na úrovni výuky na školách většinou zajišťované vyučujícími na univerzitách, které dané stránky provozují. Příkladem internetových stránek, na nichž lze najít velké množství tematicky rozdělených přednášek, jsou:

- Coursera [8]

- Udacity [9]
- FutureLearn [10]
- edX [11]
- Khanova škola [12]

Kromě zaštitění univerzitní půdou se také rozšiřují studentské kluby, kde si mohou zájemci vyměňovat zdroje informací a další potřebné detaily. Sdružení lidí se stejnými zájmy může rozšířit možnosti propojování znalostí a zrychluje přístup k nim. Tohoto principu má navrhovaný systém využívat, protože takový přístup funguje i u již výše zmíněných komunit. V navrhovaném systému se tedy nejedná o duplikaci již existujících kurzů, zaměřuje se na jinou oblast vzdělávání, než je univerzitní výuka.

Internetové služby

Sféra podpůrných služeb spojených se vzděláním se s masovým rozšířením internetu přesunula také do elektronické podoby [13]. V té se jedná především o poskytování například e-mailových služeb, komunikačních protokolů, ale také externího úložiště nebo zobrazování uložených informací. Jedná se jak o komerčně využitelné služby (placené), tak i o služby poskytované široké veřejnosti zdarma. Ve většině případů ovšem i takové získávají příspěvky pomocí dobrovolného daru.

Toto rozdělení ovšem není úplně navzájem se vylučující. Některé služby používají oba způsoby zároveň, kdy pro platící uživatele nabízí větší kapacitu, či další podpůrné služby. Neplatící tak mají omezené možnosti používání. Záleží jen na konkrétním způsobu využití, zda tyto kapacity postačují, nebo je třeba rozšířit možnosti za cenu příspěvku provozovateli.

Služby na internetu se neustále rozšiřují a mnoho firem již nabízí řešení přímo na míru podle přání zákazníka. Takové služby na míru jsou placené. Příkladem mohou být již výše zmiňované univerzitní projekty typu Coursera, ale i firemní portály na soukromou správu účetnictví a podobně.

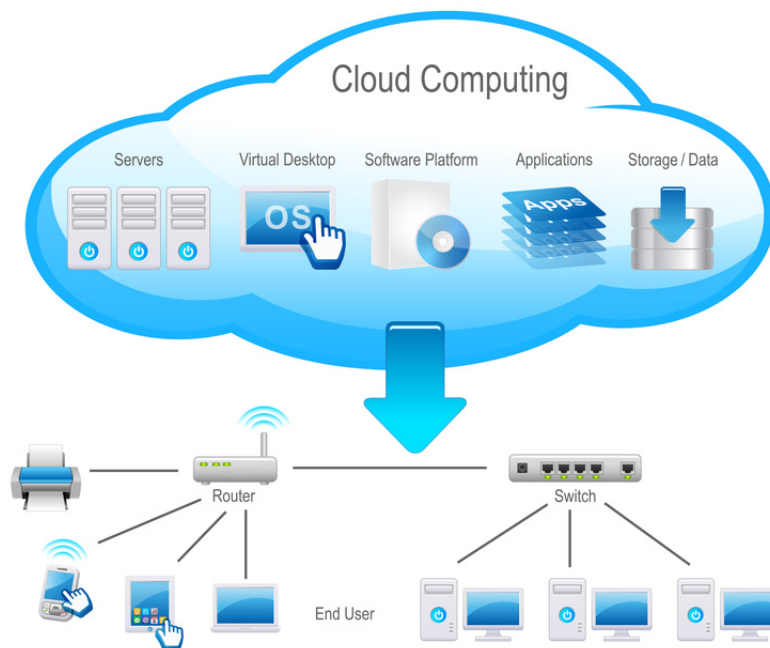
Jelikož je navrhovaný systém zaměřený na podporu vzdělávání komunity, budu se nadále zaměřovat pouze na služby poskytované zdarma. Důvodem je minimalizace nákladů na provoz a tudíž možnost rozšíření mezi libovolnou komunitu pokud možno nezávisle na množství použitých zdrojů.

2.1 Cloudová řešení

Mezi poskytované služby prostřednictvím internetu patří také cloudová řešení [15]. Jedná se o takové řešení, při němž poskytuje firma část kapacity svého serveru pro externí úložiště dat a jejich případnou strukturalizaci. Lze si například objednat vytvoření a správu databáze informací. Výhodou těchto řešení je, že samotný zákazník nemusí řešit vlastní server a jeho kapacitní možnosti, ale může se spolehnout na externí zdroje. Zároveň také má přístup k takto uloženým informacím v podstatě odkudkoliv, stačí pouhé připojení na internet

2. INTERNETOVÉ SLUŽBY

a ověření zpravidla pomocí přihlášení pod svůj uživatelský účet. Mezi nevýhody patří případné problémy ohledně vlastnictví dat zde uložených a dalšího nakládání s nimi, jakož i otázky jejich bezpečnosti a případného zneužití.



Obrázek 2.1: Schéma obecného principu cloudových řešení. [16]

Na obrázku 2.1 je znázorněn obecný princip fungování cloudových řešení. Provozovatel poskytuje zázemí provozované služby (servery, softwarovou platformu, aplikace, databázi, apod.) a uživatel jich pouze využívá prostřednictvím vnější komunikace. Jinými slovy to, co je uzavřeno v cloudu (oblaku) není podstatné pro použití služby, protože je tato struktura spravována poskytovatelem [14].

Každé cloudové řešení má pět specifických vlastností:

1. On-demand Self Service - poskytování prostředků cloudu na požádání, umožňuje rozšiřitelnost potřebné infrastruktury
2. Resource Pooling - poskytování služeb více uživatelům, sdílení základní infrastruktury
3. Broad Network Access - dostupnost cloudových prostředků z široké škály zařízení (tablet, PC, smartphone) a z různých míst
4. Rapid Elasticity - možnost automaticky zažádat o další úložné místo na cloudu

5. Measured Services - přizpůsobení poskytování služby na základě jejího využívání a z toho plynoucích modelů a cenových sazeb

Takové využití přes internet je praktické a nenáročné pro zákazníka, vše řeší provozovatel. Jak již jsem zmínila výše, i zde platí, že jsou cloudy přístupné zdarma a cloudy placené. Může být i cloudové řešení vyrobené na míru přesně podle potřeb specifikovaných zákazníkem. Z hlediska této bakalářské práce je ovšem primární využití cloudů přístupných zdarma. Umožní to tak malé náklady na provoz, a tudíž může být aplikace volně nekomerčně šiřitelná a přístupná pro širokou veřejnost (včetně již zmíněných komunit).



Obrázek 2.2: Výhody a nevýhody cloudových řešení. [21]

Na obrázku 2.2 lze vidět přehledně výhody a nevýhody cloudových řešení. Z hlediska využití navrhovaného systému komunitou mezi hlavní výhody patří cena (jelikož se zaměřujeme na služby zdarma) a rozšiřitelnost úložiště bez nutnosti dovybavení hardwarových kapacit. Kromě těchto zjevných výhod ovšem nabízí tato řešení další přínosy. Vzhledem k tomu, že jsou tyto služby poměrně náročné nejen na provoz, ale i na programování, není nutné vytvářet znovu to, co již bylo uděláno. Z tohoto důvodu je pro komunitu (ne univerzitu či jiný větší či komerčně zaměřený subjekt) praktičtější agregace již existujících služeb místo vytváření vlastních.

2.2 Vybrané cloudové služby pro účely vzdělávání

Ke vzdělávacím účelům jsou vhodné především některé služby, u nichž se dají cloudová řešení využít. Mezi takové hlavní zdroje jsou řazeny videozáznamy, audiozáznamy, dokumenty, obrázky a samozřejmě také obecná úložiště, pokud se informace nedá zařadit mezi výše uvedené. Tento typ informací bývá náročný nejen na kapacitu úložiště, ale i na potřebný výkon. Využitím cloudu je eliminována nutnost navyšování těchto zdrojů, jelikož o to se stará provozatel služby.

Z hlediska samotného využívání dostupného úložiště lze vzít cloudovou službu v takové podobě, v jaké je primárně poskytována (zpravidla to bývá dostupnost přes webové rozhraní), nebo lze využít veřejného API rozhraní [17]. To umožňuje implementovat zobrazení takto uložených informací přímo v originální podobě do systému. Jedná se o to, že není nutné stahovat přímo video ze serveru, ale může se rovnou přehrávat pomocí API rozhraní v rámci aplikace v okně v podobě streamovaného záznamu. To opět snižuje nároky na systém, nevyžaduje totiž stahování velkého množství dat, pokud člověk chce zjistit informace, ale umožní rychlé prohlížení a prohledávání uložených znalostí bez nutnosti kapacitních nároků vlastního přístupového zařízení.

Pro účely využívání systému komunitou, kde se klade důraz na nízké nároky na cenu a hardware je možnost veřejného API rozhraní poměrně zásadní. S ohledem na tento fakt také byly vybrány následující služby.

Videozáznamy se dají ukládat pomocí nejrůznějších dostupných aplikací. Mezi těmi nejznámějšími a hlavními zmíním Vimeo a YouTube. Obě tyto služby jsou známé a poskytují externí úložiště.

Pro audiozáznamy existuje velké množství služeb, kam lze tyto informace uložit, ovšem ne všechny poskytují veřejné API rozhraní pro komunikaci a případné streamované přehrávání. Zvukové záznamy bez videodoprovodu lze uložit též na YouTube, přičemž služba typu Soundcloud využívá specificky pouze zvukové stopy, které pak přehrává.

Dokumenty jsou často hlavní součástí předávání znalostí a informací. Nejedná se pouze o elektronické verze knih, které nemusí mít veřejnou licenci, jedná se také o nejrůznější poznámky či články, o něž se lidé chtějí podělit. K zobrazování takových textů lze použít například Google docs, ovšem známější je v této oblasti služba Scribd, která se na ně specializuje.

Pro ukládání a zobrazování obrázků je asi nejrozšířenější pole služeb. Vizualní informace jsou jednou z nejčastějších činností, které široké spektrum uživatelů internetu využívá. Většina těchto řešení má přístup k nahrávání obrázků přes webovou stránku. Mezi ty, které umožňují nahrání i přes veřejné API rozhraní, patří například Imgur. K zobrazování není třeba dalšího přehrávacího zařízení jako je tomu u dokumentů, audiozáznamů a videozáznamů, lze je jednoduše vložit do programu pomocí jednoduchého odkazu na adresu webového uložení a pomocí tagů obrázků zobrazit. Není třeba jakékoliv manipulace jako u výše zmíněných, proto je potřeba API rozhraní pouze na nahrávání

a získávání adresy uložených dat.

V neposlední řadě je třeba zajistit nějaké obecné úložiště pro cokoliv, co nespadá do výše uvedených kategorií, nebo je v nějakém nepodporovaném formátu. U těchto dat ovšem není možné zobrazení pomocí veřejného API rozhraní z prostého důvodu, že není znám charakter všech dat, která jsou takto uložena. API tedy v tomto případě slouží hlavně pro nahrávání, aby se nemusela tato činnost řešit separátně přes danou službu, ale bylo možno ji integrovat přímo do aplikace. Mezi taková obecná externí úložiště patří například uloz.to. Pro účely programu jsem ovšem zvolila Dropbox, který má zázemí známé firmy provozující takové úložiště dlouhodobě.

Vzhledem k plánovanému využívání navrhovaného systému komunitou je kladen velký důraz na redukci finančních a kapacitních nároků. Z hlediska toho bude výsledný systém agregovat výše uvedené služby a nebude je znovu vytvářet. Zároveň se částečně vyřeší problémy kapacitní, jelikož informace budou uloženy na externím cloudovém serveru. To s sebou nese ovšem komplikace ve formě možnosti ztráty dat. S takovou možností je nutné při návrhu a implementaci takového systému počítat. Kromě možnosti konfliktu ohledně autorských práv může dojít ke ztrátě také problémy na straně externího úložiště, které uživatel nemůže ovlivnit.

YouTube

YouTube je služba poskytující ukládání, přehrávání a vyhledávání videozáznamů. Přehrávání probíhá přes API rozhraní, které zobrazuje video ve streamované podobě bez nutnosti stažení celého záznamu. Umožňuje také manipulaci s přehráváním, jako je přepínání v časové ose záznamu, pozastavení, zastavení. Kromě toho je také možno k videu připojit titulky, či vybrat kvalitu nahrávky [18].

Službu provozuje firma Google, která zajišťuje běh serveru a jeho správu. Pro registraci a přihlášení není nutno poskytovat finanční příspěvek, je dostupná také zdarma. Pro tyto účely má každý účet vyhrazeno místo, na něž může své záznamy ukládat.

Ukládání na server probíhá po přihlášení na uživatele, pod kterým budou videa zavedena do databáze. Za samotnou uloženou nahrávku, zodpovídá majitel účtu.

Pro účely použití videozáznamů v externím programu, YouTube nabízí veřejné API rozhraní, které má dvě části. Data API, které zajišťuje nahrávání záznamů na server a samotné přehrávací API rozhraní, které umožňuje zobrazení videa včetně manipulačních tlačítek přímo v grafickém uživatelském rozhraní programu. K použití je nutno program přes jednoduchou sérii úkonů zaregistrovat na stránkách YouTube.

3.1 Popis YouTube API

Jak již bylo zmíněno, se serverem program komunikuje přes dvě API rozhraní, Data API a přehrávač API.

3.1.1 Data API

Předtím, než je možno toto API používat, je nutno registrovat aplikaci. Ta probíhá pomocí získání API klíče pod účtem, se kterým bude služba spojo-

3. YOUTUBE

vána. Poté je zvolena služba Data API, která umožňuje především následující operace:

- list – vrátí (GET) seznam obsahující jeden či více záznamů
- insert – vytváří (POST) nový zdroj
- update – upravuje (PUT) existující zdroj podle příkazů v žádosti o změnu
- delete – odstraní (DELETE) specifikovaný zdroj

Data API obsahuje ještě další operace, které mohou být specifické pouze pro konkrétní zdroj, ale pro účely navrhované aplikace jsou tyto čtyři základní a dostatečné. Pro další možné použití se lze obrátit na stránky YouTube Data API vysvětlující všechny operace, které lze provádět včetně příkladů použití a odpovědi serveru.

Vzhledem k tomu, že Data API využívá veškerých možností, které YouTube nabízí, nemusí být tyto operace asociovány pouze s konkrétním videozáznamem. Příkazy se používají ve spojení s konkrétním zdrojem, na který jsou aplikovány následujícím způsobem: „zdroj“.„operace“ (například `video.insert`). Mezi hlavní zdroje, se kterými lze pomocí výše zmíněných operací manipulovat patří:

- channel – obsahuje informace o jednom YouTube kanálu
- playlist – reprezentuje YouTube playlist (obsahuje seznam videí, která uživatel spolu asocioval a ta se dají shlédnout buď v sekvenci za sebou, nebo jednotlivě)
- search result – obsahuje informace o videu, kanálu nebo playlistu, který odpovídá vyhledávacím parametrům specifikovaným v API žádosti
- video – reprezentuje jedno YouTube video
- videoCategory – identifikuje kategorii, se kterou je možno nahrané video spojovat

Aby aplikace příliš nepřetěžovala servery (vzhledem k tomu, že YouTube podporuje velké množství zákazníků a umožňuje jim použití těchto služeb), každá žádost má svojí hodnotu, kterou si server přepočítává a při velkém zatížení omezí přístup.

Na každý požadavek, který je zaslán pomocí rozhraní na YouTube server, je zpět obdržena patřičná odpověď, která obsahuje nejdůležitější informace o právě provedené akci. Při nahrání videa se především jedná o ID, pod kterým je dané video uloženo v databázi na serveru. Konkrétní odpovědi serveru s případnými ukázkami aplikačního kódu jsou obsaženy na stránkách YouTube Data API [19].

3.1.2 Přehrávač API

Pro samotné zobrazování videa přímo v programu v YouTube přehrávači je využito veřejné API pro přehrávání. Na jeho načtení slouží jednoduchý požadavek pomocí URL `http://www.youtube.com/v/VIDEO_ID?version=3`, kde VIDEO_ID je unikátní 11-ti znakový identifikátor, pod kterým se daný záznam nachází na serveru. Přehrávač umožňuje veškerou manipulaci se zobrazovaným videem a popis jednotlivých funkcí, které k tomu používá, je opět uveden na stránkách YouTube API přehrávače [20].

Scribd

Scribd je služba, která umožňuje nahrávání a zobrazování dokumentů v originální digitální podobě. Tím je myšleno, že daná informace si zachovává svůj formát a styl textu, ve kterém se dá listovat podobně jako v kterémkoliv editačním programu. Díky veřejnému API rozhraní je umožněno toto zobrazení integrovat přímo do programu, který jej využívá. Na grafickém uživatelském rozhraní se tedy objeví přímo zobrazení dokumentu, které se načte ze serveru, aniž by uživatel musel cokoli ze serveru stahovat [22].

Služba byla založena trojicí Adler, Friedman, Bernstam a sídlo má v San Franciscu ve státě California. Kromě nahrávání vlastních dokumentů také umožňuje šíření oficiálních dokumentů a časopisů, ovšem tyto služby jsou zpravidla placené.

K uložení dokumentu na server Scribd vyžaduje autentizaci pomocí uživatele, pod jehož vlastnictví bude daná informace náležet. Vzhledem k tomuto systému za dodržování licenčních podmínek zodpovídá majitel registrovaného účtu.

Pro nahrávání na server slouží Platform API, které umožňuje veškerou komunikaci se serverem přímo z aplikačního rozhraní. K zobrazování informací přímo v rámci grafického uživatelského rozhraní aplikace využívá javascript API, které vloží dokumenty přímo v podobě, v jaké jsou uloženy na serveru bez nutnosti stahování. Podobně jako v případě YouTube, Scribd vyžaduje registraci aplikace, která bude API rozhraní používat. Server má velice benevolentní podmínky, co se týče nahrávaných souborů, protože umožňuje neomezeně uploadovat veřejné dokumenty a u soukromých má limit 50000 na jeden API účet.

4.1 Popis Scribd API

Podobně jako YouTube, i Scribd využívá dvou API rozhraní, kde jedno je využíváno pro komunikaci se serverem a ukládání a editaci dokumentů a druhé, které zajišťuje zobrazení v grafickém uživatelském rozhraní.

4.1.1 Platform API

Rozhraní, které je využíváno pro manipulaci s dokumenty v rámci komunikace mezi aplikací a serverem, je platform API, které zajišťuje veškerou komunikaci se serverem a jeho odezvy. Všechny požadavky jsou posílány přes `http://api.scribd.com/api`.

Pro každý požadavek je nutno uvést především dvě věci, `api_key` a `method`. `Api_key` je jednoznačný identifikátor, který je přiřazený k účtu, pod kterým je rozhraní používáno. `Method` obsahuje informace o metodě, kterou momentálně bude příkaz používat.

Scribd umožňuje kromě manipulace s jednotlivými dokumenty také vytvářet kolekce, ve kterých se sdružují dokumenty se stejným tématem. API rozhraní podporuje manipulaci i s dalšími prvky, ovšem ty nejsou z hlediska navrhované aplikace podstatné.

Pro manipulaci s dokumenty umožňuje API rozhraní následující metody, které jsou vždy ve formátu docs. „metoda“:

- `upload` – umožňuje nahrání (POST) dokumentu na server
- `getList` – vrátí (GET) seznam dokumentů daného uživatele
- `delete` – smaže (DELETE) existující dokument ze serveru (při úspěchu vrátí OK zprávu)
- `getSettings` – vrátí (GET) meta-data existujícího dokumentu
- `changeSettings` – upraví (UPDATE) meta-data dokumentu podle vyplněných polí, vyplňují se pouze ta, která jsou ke změně určena
- `browse` – vrátí (GET) seznam dokumentů, které odpovídají hledanému kritériu

Pro manipulaci s kolekcemi dokumentů umožňuje API rozhraní následující metody, které jsou vždy ve formátu collections. „metoda“:

- `addDoc` – přidá dokument do existující kolekce
- `create` – vytvoří novou kolekci
- `update` – metoda umožňuje upravit název, popis nebo typ kolekce
- `delete` – metoda smaže existující kolekci
- `getList` – vrátí seznam kolekcí daného uživatele
- `removeDoc` – odstraní dokument z existující kolekce
- `listDocs` – vrátí seznam dokumentů v dané kolekci

Každá metoda, kterou zašle API rozhraní požadavek na server, má své specifické parametry. Z hlediska využití pomocí aplikace uvedu pouze povinné parametry k jednotlivým metodám. Odpověď serveru je řešena pomocí XML, ve kterém jsou uvedeny veškeré parametry spojené s provedením požadavku. Mezi nejdůležitější patří `doc_id` (unikátní identifikátor uloženého dokumentu) a `access_key` (klíč umožňující připnutí dokumentu na externí stránku). Kromě těchto také umožňuje rozhraní použití dalších nepovinných parametrů, které dále specifikují požadavek. O těchto se lze dočíst na velmi podrobné dokumentaci, kterou obsahují stránky Scribd platform API včetně příkladů použití v kódu a vzorových odpovědi serveru (spolu s vysvětlením jednotlivých vrácených parametrů) [23].

- `docs.upload` – `file` (cesta k dokumentu, který má být nahrán), `api_key` (jednoznačný identifikátor uživatele)
- `docs.getList` – `api_key` (jednoznačný identifikátor uživatele)
- `docs.delete` – `doc_id` (jednoznačný identifikátor uloženého dokumentu), `api_key` (jednoznačný identifikátor uživatele)
- `docs.getSettings` – `doc_id` (jednoznačný identifikátor uloženého dokumentu), `api_key` (jednoznačný identifikátor uživatele)
- `docs.changeSettings` – `doc_id` (jednoznačný identifikátor uloženého dokumentu), `api_key` (jednoznačný identifikátor uživatele), parametr, který si přejeme změnit
- `docs.browse` – `api_key` (jednoznačný identifikátor uživatele)

V případě kolekcí je použití obdobné, nejdůležitějšími parametry je `api_key` (jednoznačný identifikátor uživatele), `doc_id` (jednoznačný identifikátor uloženého dokumentu), `collection_id` (jednoznačný identifikátor existující kolekce).

4.1.2 Javascript API

Pro připnutí dokumentů přímo do grafického uživatelského rozhraní aplikace slouží javascript API. Podobně jako u YouTube je vloženo zobrazovací zařízení, které načte informace uložené na serveru bez nutnosti stahování konkrétního dokumentu. V zobrazovaných informacích lze listovat úplně stejně, jako by byly uloženy na lokálním disku a otevřeny, rychlost je ovšem odvislá od komunikace se serverem a rychlosti připojení.

Využití tohoto rozhraní není náročné, stačí pouze postupovat pomocí jednoduchých kroků. Include `scribd_api.js`, vytvoření dokumentu, odeslání verze API rozhraní a zavolání objektové `write` metody. Pro názornost zde uvedu základní funkce pro komunikaci s vloženým API rozhraním.

Nejčastěji využívanými konstruktory jsou:

4. SCRIBD

- `scribd.Document.getDoc` (*document_id*, *access_key*) – tento konstruktor zobrazí dokument podle `document_id` (jednoznačný identifikátor dokumentu uloženého na serveru) a `access_key` (identifikátor, který umožňuje zobrazení v javascript API)

Nejčastěji využívané metody:

- `write` (*element_id:Number*) – vloží dokument pomocí `iframe`, doporučuje se v použití s `DIV` tagem
- `seamless` (*element_id:Number*) – alternativní metoda k `write`, která vkládá dokument bez `iframe`, tedy tak, že text vypadá jako součást stránky

Kromě výše uvedeného umožňuje rozhraní komunikaci ještě podle dalších metod a parametrů, které jsou podrobně vysvětleny na stránkách Scribd v sekci javascript API [24].

Imgur

Imgur je služba, která zajišťuje ukládání a zobrazování obrázků, pro které využívá vlastních serverových kapacit. Vzhledem k tomu, že se jedná pouze o grafické zobrazení, nejsou možné jakékoliv manipulace s obsahem, jako je tomu u videa či dokumentu [25].

Služba byla vytvořena Schaafem, který ji provozuje v Athenach ve státě Ohio. Firma je sponzorována pomocí prodeje Pro účtů, dobrovolných příspěvků a inzerce.

Vzhledem k tomu, že k zobrazování obrázků není třeba speciálního zobrazovacího zařízení, jako tomu bylo u výše uvedených služeb, veškerá komunikace probíhá pouze přes jediné univerzální API rozhraní. Samotné zobrazování je nezávislé na této komunikaci, data jsou jednoduše přehrána ze serveru podobně, jako je tomu u webových stránek.

V samotném používání této služby jsou uvedeny limity, aby se serverová kapacita rovnoměrně rozdělila mezi jednotlivé uživatele. Jedná se o maximální počet 1250 nahrání za den a 12500 požadavků za den. Tyto limity jsou stanoveny pro jednotlivé aplikace, ne uživatele. Stejně jako v předchozích případech je pro použití služby nutno nejprve aplikaci zaregistrovat podle pokynů uvedených na stránkách poskytovatele.

5.1 Popis Imgur API

Imgur provozuje pouze jedno API rozhraní, protože pro zobrazení není třeba speciálních nástrojů. Ke komunikaci je využito jednoduchých požadavků v podobě `https://api.imgur.com/3/metoda`, přičemž v položce metoda je uvedený text podle daného požadavku.

Mezi základní příkazy patří:

- image – vrátí (GET) informace o obrázku uloženém na serveru, metoda = image/{id}

5. IMGUR

- image upload – nahraje (POST) nový obrázek na server, metoda = image nebo upload
 - parametrem je cesta k obrázku, který má být nahrán do velikosti 10MB
- image deletion – smaže (DELETE) obrázek ze serveru, metoda = image/{id}
- update image information – upraví (POST nebo PUT) informace o obrázku (název nebo popis), metoda = image/{id}

V základní odpovědi serveru, pokud nevrací informace o obrázku, jsou uvedeny tyto informace:

- data – jedná se o návratovou hodnotu provedení akce, pokud se jedná o nahrání obrázku, obsahuje ID, pod kterým je uložen na server
- success – jednoduchá boolean odpověď serveru, zda byl požadavek úspěšný
- status – http status kód

Pokud se jedná o zjišťování informací o obrázku, v odpovědi jsou uvedeny veškeré informace, které jsou o něm na serveru uloženy. Mezi ně patří ID (jednoznačný identifikátor uloženého obrázku), název, popis, datum uložení a podobně. Veškeré tyto parametry včetně dalších možných využití API rozhraní jsou podrobněji popsány přímo na stránkách imgur v sekci API [26].

SoundCloud

Soundcloud je jednou z poměrně malého množství služeb, která se specificky zaměřuje na ukládání a přehrávání čistě audiozáznamů. Ostatní služby typu YouTube toto umožňují také, ale primárně jsou určeny pro kombinaci zvuku s videem. Vzhledem ke svému zaměření umožňuje nahrávat, ale i přehrávat zvukové záznamy přímo z externího úložiště [27].

Služba byla založena dvojicí Ljung a Wahlforss původně ve Stockholmu, ovšem nyní sídlí firma v Berlíně.

Nahrávání a následné přehrávání umožňuje http API rozhraní, které zajišťuje veškerou komunikaci se serverem.

Vzhledem k poskytovaným službám jsou jednotlivé audiozáznamy, které se nahrávají, omezeny svojí velikostí, nesmí být větší než 500MB. K používání API rozhraní je navíc jako v předešlých případech nutno aplikaci nejprve zaregistrovat.

6.1 Popis SoundCloud API

Soundcloud používá jednoduchého http API rozhraní, které umožňuje pomocí jednoduchých scriptů nahrávání, přehrávání a vyhledávání zvuků. Pro přehrávání vloží do aplikace přes příkaz widget, který zobrazí nahrávaný zvuk, ve kterém jsou možnosti jako spuštění, pozastavení, přetáčení a podobně.

Pro komunikaci s API rozhráním je využíván přístup pomocí `https://api.soundcloud.com`. Za každým požadavkem, který je prováděn, navíc musí být specifikován uživatel, pod kterým je akce zaštitěna. To je zajištěno parametrem `client_id`, což je jednoznačný identifikátor registrovaného uživatele. Příklad takového dotazu včetně identifikace je například `https://api.soundcloud.com/tracks?client_id=YOUR_CLIENT_ID`, kde `YOUR_CLIENT_ID` je identifikátor uživatelského účtu.

Jednotlivé požadavky se posílají přes `https://api.soundcloud.com/POZADAVEK`, kde za `POZADAVEK` se dosadí přímo metoda, kterou chceme použít. Mezi nejdůležitější požadavky, které lze tímto přístupem používat, patří následující:

6. SOUNDCLOUD

- upload – nahrání (POST) zvukového záznamu na server, požadavek = /tracks
- remove – odstranění (DELETE) zvukového záznamu ze serveru, požadavek = /tracks/{id}
- update – modifikuje (PUT) informace o zvukovém záznamu, požadavek = /tracks/{id}
- play – získání (GET) zvukového záznamu ze serveru, požadavek = /tracks/{id}

Podobně jako u výše uvedených služeb i Soundcloud umožňuje slučování jednotlivých zvuků do skupin a následnou manipulaci s nimi. Bližší informace o tom jsou uvedeny na stránkách Soundcloudu v API dokumentaci [28].

Pro zobrazení widgetu přímo v aplikaci, který umožní streamované přehrávání zvuků, je možno použít požadavek /oembed, který pomocí URL adresy nahraného zvuku jej spustí do widgetu.

Dropbox

Dropbox je služba, která nemá specifické zaměření na uložené informace. Nezáleží na tom, jaký formát či charakter mají nahrávaná data, vše je přístupno z jednoho bodu. Vzhledem k těmto skutečnostem se tedy nejedná o upload a následné zobrazení v grafickém uživatelském rozhraní, ale veškeré informace jsou buď nahrávány, nebo získávány pomocí klasického stažení [29].

Služba je provozována společností Dropbox Inc. založenou dvojicí Houston a Ferdowsie.

Pro účet, který je veden zdarma, jsou omezení na velikost úložiště a maximální velikost souborů, které jsou ukládány. Z hlediska kapacity, která je dostupná, jsou to 2GB dat a maximální velikost 10MB na soubor. Ostatní služby jsou placené.

Dropbox má výhodu nejen přístupu přes veřejné API rozhraní, ale umožňuje přístup i přes mobilní aplikace. Pro komunikaci v rámci navrhovaného programu je vhodné využití Core API, které zajišťuje veškerou komunikaci se serverem. K využití této služby je nutno aplikaci nejprve pod API rozhraní zaregistrovat.

7.1 Popis Dropbox API

Jak již bylo uvedeno, Core API zajišťuje komunikaci se serverem. Z hlediska použití v rámci navrhované aplikace jsou níže popsány především metody, které umožňují správu souborů nahrávaných na server.

Mezi jednotlivé požadavky včetně jejich cesty a popisu parametrů patří:

- /files (GET)
 - stáhne metodou GET soubor, který je uložen na serveru
 - `https://api-content.dropbox.com/1/files/auto/<path>`
 - path – cesta, ze které se má soubor stáhnout

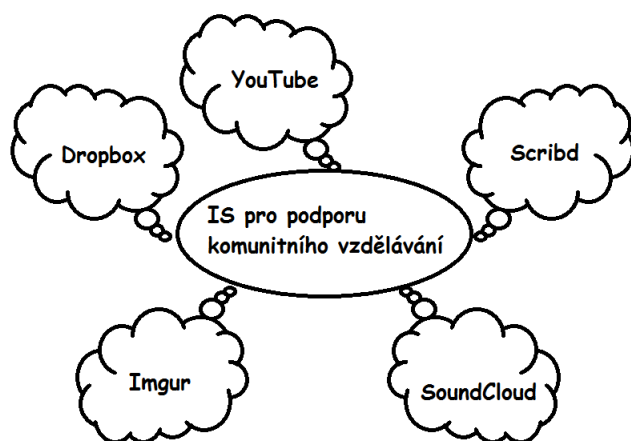
7. DROPBOX

- /files_put
 - nahraje soubor pomocí PUT nebo POST metody
 - `https://api-content.dropbox.com/1/files_put/auto/<path>?param=val`
 - path – cesta, ze které se má soubor nahrát, param=val – URL kódování parametrů
- /fileops/copy
 - zkopíruje soubor metodou POST do nového umístění
 - `https://api.dropbox.com/1/fileops/copy`
 - parametry – root (kořenový adresář, ze kterého se berou hodnoty odkud kam), from_path (odkud se soubor má zkopírovat), to_path (kam se má soubor zkopírovat)
- /fileops/delete
 - smaže pomocí metody DELETE soubor
 - `https://api.dropbox.com/1/fileops/delete`
 - parametry – root (kořenový adresář, pomocí kterého je specifikována cesta), path (cesta k souboru, který má být smazán)
- /fileops/move
 - přesune pomocí metody POST soubor do nového umístění
 - `https://api.dropbox.com/1/fileops/move`
 - parametry – root (kořenový adresář, ze kterého se berou hodnoty odkud kam), from_path (odkud se soubor má zkopírovat), to_path (kam se má soubor zkopírovat)

Kromě výše uvedeného umožňuje rozhraní ještě další manipulaci včetně tvorby adresářů. Tyto a další informace jsou podrobněji popsány na stránkách dropbox v sekci Core API [30]. Jsou zde uvedeny i příklady zdrojových kódů v různých programovacích jazycích, ne pouze webový způsob.

Shrnutí teoretické části

Vzdělávání je forma získávání informací, která se opírá o čtyři základní pilíře. Jsou jimi sledování, poslouchání, čtení a komunikace. S jejich pomocí je možno vytvářet informační systém pro podporu vzdělávání. Vzhledem k zaměření na komunitu je lepším řešením agregace již existujících služeb místo jejich vytváření. Eliminují se tím částečně finanční a kapacitní požadavky. Z dostupných cloudů poskytujících potřebné služby a možnost externího úložiště jsem z nich vybrala pět pro účely tohoto návrhu. Kritéria, která splňují všechny z nich, jsou poskytnutí služeb zdarma a existence veřejného API rozhraní. API je důležité především pro snadnou použitelnost systému bez nutnosti technických znalostí a nutnosti využívání dalších webových služeb. YouTube a Scribd navíc již obsahují určité výukové materiály, které se dají použít bez nutnosti nahrávání svých vlastních zdrojů. Scribd jsem zvolila navíc proto, že zobrazuje veškeré dokumenty v jejich původní formě a podporuje velkou řadu formátů. Co se týče záznamů zvukových a obrázků, hlavním kritériem byla možnost veřejného API rozhraní, ačkoliv mnoho výukových materiálů neobsahují. Ty bude nutno nejprve na server před použitím nahrát. Dropbox byl zvolen z hlediska jeho dostupnosti a rozšířenosti, je v současnosti jedním z nejpoužívanějších všeobecných úložišť. Prvotní návrh vyplývající z těchto faktů je uveden na obrázku 7.1.



Obrázek 7.1: Prvotní návrh internetového informačního systému pro podporu komunitního vzdělávání.

Analýza

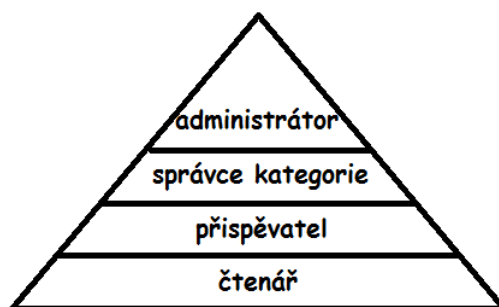
Aby bylo možno k navrhovanému systému přistupovat odkudkoliv a nebyly kladeny velké požadavky na hardware a cenu, je koncipován jako online aplikace, při jejímž spuštění je nutný přístup k internetu. Nejedná se o pouhé shromažďování a vytváření datových záznamů, ale také o jejich vzájemné propojení. Jak již bylo uvedeno v teoretické části, navrhovaný systém je určen pro komunitní vzdělávání. S tím mimo jiné souvisí specifická hierarchie, kterou jsem do návrhu systému promítla.

Jednotliví uživatelé systému jsou rozděleni do následujících skupin (hierarchie viz obrázek 8.1):

- administrátor
- správce kategorie
- přispěvatel
- čtenář

Každá z těchto skupin má svoji specifickou roli v systému. Administrátor je nejvyšší správce celého systému. Vzniká nainstalováním databáze, kdy je automaticky majiteli přidělen status administrátora. Ten má dohled nad celým systémem bez omezení, může blokovat uživatele, schvalovat jejich vytvoření, mazat záznamy z databáze a podobně. Zároveň zajišťuje provoz jediné části systému, která není řešena pomocí cloudových služeb. Tou je samotná databáze uživatelů a příspěvků. Vzhledem k její koncepci není náročná na provoz a lze ji proto umístit buď na lokální PC, nebo na pronajatý server. Konkrétní volba je již na samotném administrátorovi.

Správce kategorie funguje jako další dozor nad fungováním systému, ovšem jeho práva nejsou tak vysoká, jako u administrátora. Stará se pouze o přidělenou kategorii, na kterou ještě musí splňovat podmínky především počtem příspěvků. Toto kritérium je tam z důvodu, že přispěvatelů do systému může být mnoho a ten, který má dohled, by měl mít o dané části přehled. Jelikož



Obrázek 8.1: Hierarchie uživatelů v systému.

administrátor nemůže prověřovat každého kandidáta zvlášť, je zde stanovena tato jednoduchá podmínka pro usnadnění správy.

Přispěvatelem se stává uživatel tehdy, vloží-li příspěvek do systému. Tato kategorie zajišťuje, že větší manipulace v rámci systému budou prováděny zkušenějšími uživateli.

Při první registraci (kromě administrátora při instalaci) je automaticky každý uživatel zařazen do kategorie čtenář, kterému je umožněno prohlížet obsah systému.

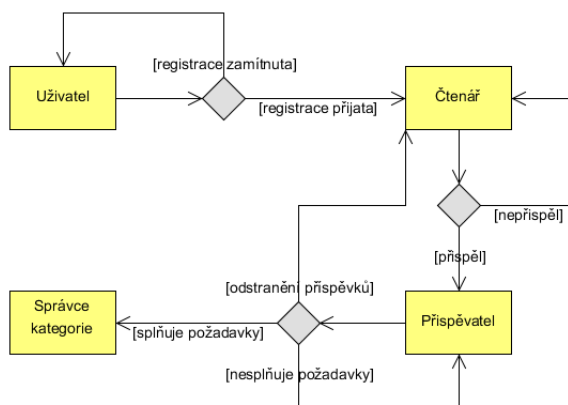
Jak již bylo uvedeno výše, navrhovaný systém tedy bude přes internet komunikovat přes API rozhraní s cloudovými službami a začlení tak obsah zde uložených dat v grafickém uživatelském rozhraní. Samotná aplikace (přes kterou bude navrhovaný systém přístupný) není součástí tohoto návrhu, budu ji považovat tedy za black box. Navržená je databáze osob a dat, která bude umístěna na serveru, jehož prostřednictvím bude systém provozován a spravován. Jednotlivé případy užití simulují komunikaci s databází i s vnějšími cloudovými zdroji přes rozhraní aplikace.

8.1 Evidence uživatelů

Při první registraci do systému je uživatel automaticky označen jako čtenář. Má přístup ke čtení veškerých informací, které jsou ostatními uživateli vkládány.

Poté, co poprvé vyplní formulář na přidání jakékoliv informace, je automaticky přesunut do skupiny přispěvatelů, kteří navíc mají již právo propojovat také informace mezi sebou. Každá vložená informace či odkaz na související data jsou označena pak jménem vlastníka, který je vytvořil.

Lepší dohled nad fungováním celého systému je zajištěn pomocí skupiny správce kategorie, který má dohled nad příslušnou částí systému. Podmínkou pro získání takového statusu je dostatečný počet vložených příspěvků a po-



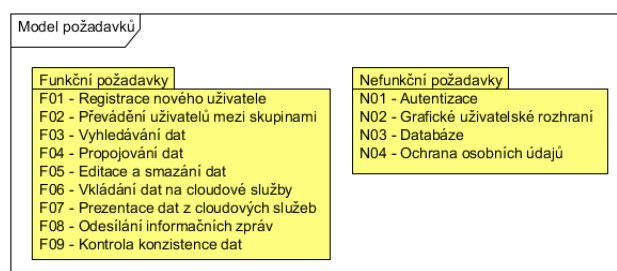
Obrázek 8.2: Životní cyklus uživatele.

slední slovo má administrátor celého systému, který jej do této skupiny musí přesunout.

Dohled nad celým fungováním má administrátor, který má přístup ke všem informacím. Administrátor má také přidělena ID cloudových služeb, aby se daly informace vkládat na veřejně přístupná místa.

Každá registrace má podmínku unikátního e-mailu (v rámci databáze uživatelů), který je pro ochranu systému nutné potvrdit a schválit administrátorem, aby se zamezilo útokům a zneužívání systému. Pod takovým útokem je myšleno například zahlcení systému nesmyslnými příspěvky. Pokud navíc někdo nebude dodržovat pravidla používání systému, má administrátor možnost takového uživatele zablokovat.

V databázi jsou uživatelé rozřazováni do skupin. Každá skupina má jinou funkci a práva. 8.2 Počátečním aktem je vytvoření registrace pomocí registračního formuláře. Pokud je přijata, je automaticky uživatel zařazen mezi skupinu čtenářů. V opačném případě nezískává přístup do systému. Přispěvatelem se stává ten, kdo začne vkládat informace do databáze. Ten poté může i navzájem mezi sebou propojovat informace, které se v databázi již nacházejí. Ve chvíli, kdy by byly smazány z jakéhokoliv důvodu veškeré příspěvky od daného uživatele, přestává být přispěvatelem a vrací se do skupiny čtenářů. Pokud splní požadavky na počet příspěvků či odkazů v dané sekci, může požádat o pozici správce kategorie. Tuto skupinu schvaluje administrátor na základě ověření korektnosti požadavku. Správce kategorie má následně povinnost spravovat příslušnou kategorii a dohlížet na její správné fungování.



Obrázek 8.3: Model požadavků.

8.2 Evidence informací

Hlavní částí systému jsou informace, které jednotliví uživatelé do systému vloží. Ty jsou poté vlastníkem přiřazeny do sekce, kterou si vybere či založí.

Součástí přidání dat je také zapsání klíčových slov, která následně slouží pro rychlé vyhledávání. Kromě toho také mohou i ostatní uživatelé, kteří si data prohlédnou, přiřadit odkaz na jiné informace, které s daným tématem nějak souvisí. Tímto je umožněno komplexnější sdílení dat v souvislostech.

V databázi jsou uloženy pouze informace o jednotlivých kategoriích a odkazy na ID na příslušné cloudové službě, která je zde využita.

Každý takový příspěvek do systému je poté označen jménem uživatele, který informaci vložil (vlastník) a datum, kdy to bylo provedeno. Editovat data může pouze jejich vlastník a do databáze se uloží datum změny. K takovým úpravám je nutno připojit komentář, který krok popisuje: důvod a způsob změny. Umožňuje to pak lepší přehlednost a možnou kontrolu administrátorů. Podobně funguje i mazání příspěvků.

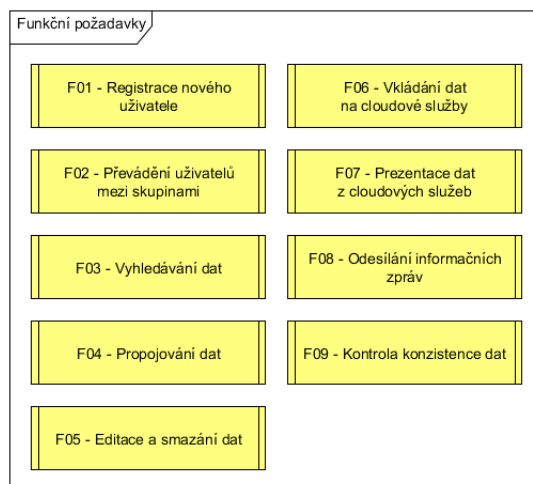
Propojování informací pomocí odkazujících souvislostí je doprovázeno zapsáním komentáře, proč byla daná akce provedena a také jménem uživatele, který je za ní zodpovědný.

8.3 Model požadavků

V této kapitole jsou uvedeny požadavky na systém, které musí nutně splňovat, aby odpovídal zde uvedenému návrhu. Jsou rozděleny na funkční a nefunkční.

8.3.1 Funkční požadavky

Kapitola obsahuje popis jednotlivých funkčních požadavků na navrhovaný systém. 8.4



Obrázek 8.4: Funkční požadavky.

8.3.1.1 F01: Registrace nového uživatele

System bude umožňovat volnou registraci uživatele, který má zájem se účastnit a přispívat do systému. Záznam je vytvořen manuálně vyplněním dotazníku o osobních údajích.

8.3.1.2 F02: Převádění uživatelů mezi skupinami

System bude umožňovat převádění uživatelů mezi skupinami administrátorem systému na základě splnění požadavků pro vstup do skupiny.

8.3.1.3 F03: Vyhledávání dat

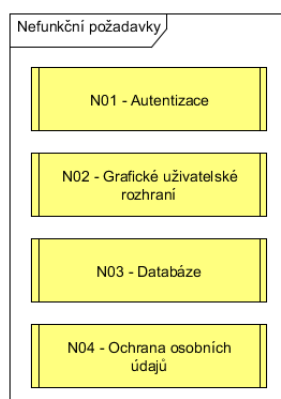
System bude umožňovat vyhledávání dat na základě klíčových slov uvedených u jednotlivých záznamů. Ta jsou přidána vlastníkem při vkládání příspěvku do databáze.

8.3.1.4 F04: Propojování dat

System bude umožňovat odkazování na další data v databázi. Takové propojení je vytvořeno uživateli programu na základě souvislostí daných témat.

8.3.1.5 F05: Editace a smazání dat

System bude umožňovat editaci dat vlastníkem. Ten musí při tomto úkonu uvést důvod a způsob změny.



Obrázek 8.5: Nefunkční požadavky.

8.3.1.6 F06: Vkládání dat na cloudové služby

Systém bude umožňovat vkládání informací za využití cloudových služeb, které budou přístupné zdarma. Program sám provede přes veřejné datové rozhraní upload na vzdálený server a uloží do databáze ID příspěvku.

8.3.1.7 F07: Prezentace dat z cloudových služeb

Systém bude umožňovat zobrazování informací, které byly uživateli vloženy za využití cloudových služeb. Pro tyto účely program zajistí načtení informací ze vzdáleného serveru za použití ID vloženého příspěvku.

8.3.1.8 F08: Odesílání informačních zpráv

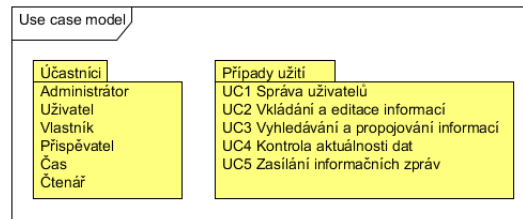
Systém bude umožňovat uživatelům označit kategorii, která je zajímavá. Program poté bude zajišťovat automatické odesílání e-mailů s novinkami.

8.3.1.9 F09: Kontrola konzistence dat

Systém bude umožňovat výpis změn v databázi a příspěvků splňující časové kritérium pro možnosti kontroly konzistence dat.

8.3.2 Nefunkční požadavky

Kapitola obsahuje popis jednotlivých nefunkčních požadavků na navrhovaný systém. 8.5



Obrázek 8.6: Model případů užití.

8.3.2.1 N01: Autentizace

System bude při přihlášení požadovat heslo o délce alespoň 6 znaků.

8.3.2.2 N02: Grafické uživatelské rozhraní

System bude mít grafické uživatelské rozhraní, které bude propojené s grafickým uživatelským rozhraním cloudových služeb.

8.3.2.3 N03: Databáze

System bude odpovídat na databázové dotazy maximálně do deseti vteřin.

8.3.2.4 N04: Ochrana osobních údajů

System bude zajišťovat, že s osobními údaji, která uživatel uvede, bude nakládáno v souladu se zákonem č. 101/2000 Sb. (zákon o ochraně osobních údajů).

8.4 Model případů užití

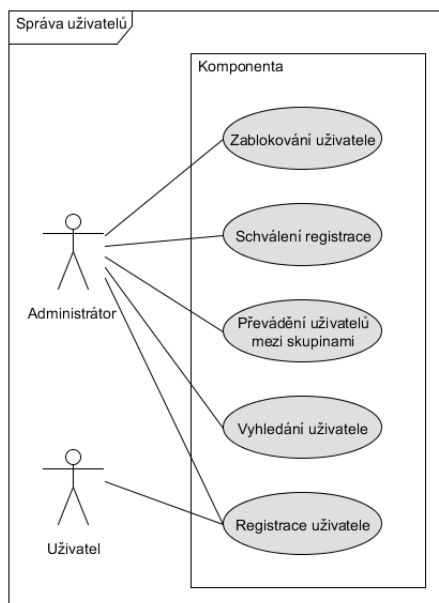
V této kapitole jsou uvedeny případy užití navrhovaného systému. Především se jedná o správu databáze, vkládání, vyhledávání a propojování informací, ale i o kontrolu aktuálnosti dat. 8.6

8.4.1 UC1 Správa uživatelů

Případ užití ukazuje přístup k databázi uživatelů a manipulaci s ní. 8.7

8.4.1.1 Zablokování uživatele

Případ užití začíná požadavkem administrátora na zablokování uživatele z databáze. To může být na základě žádosti ze strany samotného uživatele, či jako



Obrázek 8.7: UC1 Správa uživatelů.

následek porušení pravidel užívání aplikace. Přístupový záznam je následně v databázi označen jako blokový, není ovšem smazán.

8.4.1.2 Schválení registrace

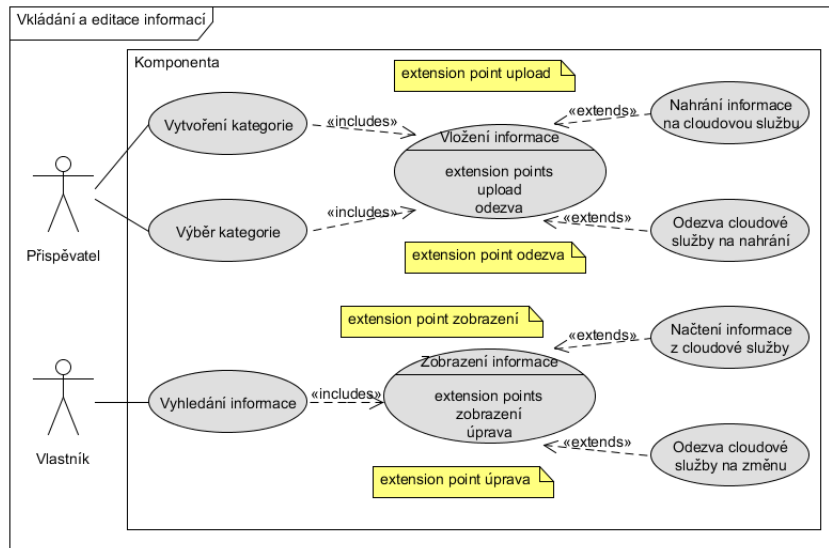
Případ užití začíná, když uživatel vyplní formulář o registraci a pošle žádost o její přijetí. Administrátor poté odešle přijetí či zamítnutí požadavku. Při přijetí je umožněn přístup uživatele do systému, při zamítnutí je mu přístup odepřen a administrátor může záznam nechat v databázi jako nežádoucí, nebo jej smazat.

8.4.1.3 Převádění uživatelů mezi skupinami

Případ užití umožňuje administrátorovi převést záznamy o uživatelích mezi jednotlivými skupinami. To je prováděno na základě splnění požadavků na přesun do příslušné kategorie.

8.4.1.4 Vyhledání uživatele

Případ užití umožňuje administrátorovi vyhledat libovolného uživatele na základě zadaných kritérií.



Obrázek 8.8: UC2 Vkládání a editace informací.

8.4.1.5 Registrace uživatele

Případ užití začíná kliknutím na registraci a následném vyplnění formuláře. Je provedena kontrola e-mailové adresy, pokud již v databázi existuje, je registrace automaticky zamítnuta. V opačném případě je odeslán požadavek na schválení registrace a umožnění přístupu do systému.

8.4.2 UC2 Vkládání a editace informací

Případ užití slouží k popisu vkládání a editace informací v databázi s využitím cloudových služeb. 8.8

8.4.2.1 Vytvoření kategorie

1. Systém umožňuje vytvořit kategorii, pokud již není v databázi obsažena.
2. Zahrnout (Vložení informace).

8.4.2.2 Výběr kategorie

1. Případ užití začíná vyhledáním příslušné kategorie, kam chce uživatel informaci vložit.
2. Zahrnout (Vložení informace).

8.4.2.3 Vložení informace

1. Do vybrané kategorie v databázi je uložen záznam o vkládané informaci, následuje komunikace s cloudovou službou.

extension point: upload

extension point: odezva

8.4.2.4 Nahrání informace na cloudovou službu

Systém pomocí veřejného API rozhraní kontaktuje cloudovou službu a odešle požadavek na nahrání informace na server.

8.4.2.5 Odezva cloudové služby na nahrání

Systém obdrží odezvu z cloudové služby v podobě ID nahrané informace, které je přidáno do záznamu v databázi.

8.4.2.6 Vyhledání informace

1. Případ užití začíná vyhledáním informace, kterou na server uživatel vložil a chce jí změnit.
2. Zahrnout (Zobrazení informace).

8.4.2.7 Zobrazení informace

1. Systém zobrazí vyhledanou informaci.

extension point: zobrazení

extension point: úprava

8.4.2.8 Načtení informace z cloudové služby

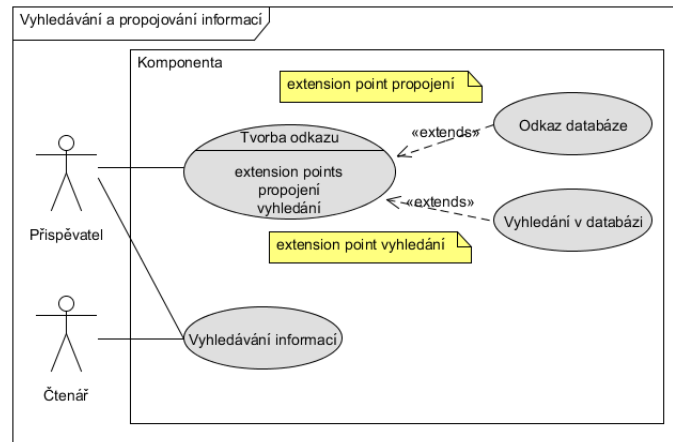
Systém pomocí veřejného API rozhraní kontaktuje cloudovou službu a nahraje uložená data pod příslušným ID.

8.4.2.9 Odezva cloudové služby na změnu

Systém umožní uživateli editovat či odstranit záznam, který uložil na cloudovou službu. Při smazání je odstraněn záznam z databáze i z cloudové služby. Při editaci je odstraněna původní informace a nahrazena novou informací.

8.4.3 UC3 Vyhledávání a propojování informací

Případ užití slouží k vytváření odkazů mezi jednotlivými informacemi v databázi a jejich následné vyhledávání. 8.9



Obrázek 8.9: UC3 Vyhledávání a propojování informací.

8.4.3.1 Tvorba odkazu

1. Případ užití začíná vyplněním formuláře pro odkazování mezi jednotlivými daty. Systém umožňuje propojovanou informaci vyhledat.

extension point: propojení
extension point: vyhledání

8.4.3.2 Odkaz databáze

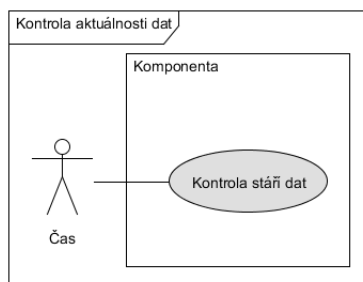
Systém umožňuje doplnit do záznamu o informaci vytvoření odkazu, doplní automaticky datum, autora a odkazované informace mezi sebou propojí v databázi.

8.4.3.3 Vyhledání v databázi

Pro snadnější propojování informací systém umožňuje vyhledat v databázi konkrétní data, která se mezi sebou mají odkazovat.

8.4.3.4 Vyhledávání informací

Systém umožňuje všem uživatelům vyhledávat informace na základě klíčových slov, která uvádí vlastník při nahrání, či data vložení/editace. Dále také je možné najít příspěvky podle autorů. Vlastníkovi je navíc umožněn výpis všech příspěvků, které do databáze vložil.



Obrázek 8.10: UC4 Kontrola aktuálnosti dat.

8.4.4 UC4 Kontrola aktuálnosti dat

Případ užití popisuje pravidelnou kontrolu aktuálnosti dat v databázi. 8.10

8.4.4.1 Kontrola stáří dat

1. Případ užití probíhá v daných časových intervalech.
2. Pro každý příspěvek v databázi:
 - a) Systém provede kontrolu data vytvoření/editace.
 - b) Pro příspěvky starší, než je stanovená doba, odešle upozornění administrátorovi a vlastníkovi příspěvku.

8.4.5 UC5 Zasílání informačních zpráv

Případ užití ukazuje zasílání novinek uživatelům podle jejich preferencí. 8.11

8.4.5.1 Zasílání informací

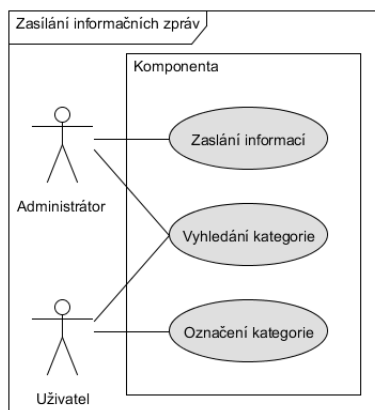
Administrátor na základě vyhledané kategorie zašle seznam nově přidaných či upravených příspěvků všem uživatelům, kteří si kategorii označili.

8.4.5.2 Vyhledání kategorie

Systém umožňuje vyhledat kategorii a vypsat seznam uživatelů, kteří si ji označili.

8.4.5.3 Označení kategorie

Uživatel si může označit kategorii, o které chce získávat informační novinky, jako jsou nové příspěvky či aktualizace dat.



Obrázek 8.11: UC5 Zasílání informačních zpráv.

8.4.6 Účastníci

Popis jednotlivých účastníků případů užití. 8.12

8.4.6.1 Administrátor

Osoba s plným přístupem do systému, zodpovědná především za správu databáze. Schvaluje registrace a stará se o běh programu a korektnost dat a uživatelů.

8.4.6.2 Uživatel

Souhrnné označení pro všechny registrované uživatele, kteří mají přístup do systému (čtenář, příspěvateľ, správce kategorie, administrátor).

8.4.6.3 Vlastník

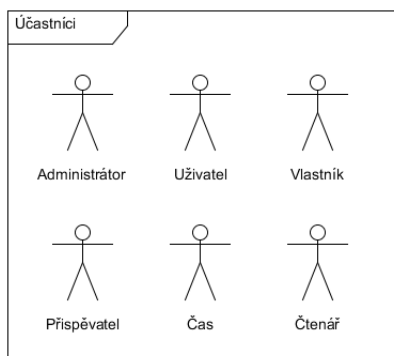
Uživatel, který vložil data do systému. Tímto aktem se stal jejich vlastníkem a má právo s nimi dále nakládat (editovat, mazat).

8.4.6.4 Příspěvateľ

Uživatel, který má práva odkazovat mezi jednotlivými daty. Stává se jím tehdy, když vloží příspěvek do jakékoliv kategorie.

8.4.6.5 Čas

Účastník, jenž zajišťuje automatickou kontrolu některých funkcí systému. Jsou prováděny periodicky v daných časových intervalech.



Obrázek 8.12: Účastníci.

8.4.6.6 Čtenář

Uživatel, který má práva pouze ke čtení a vyhledávání dat, která se nacházejí v databázi. Prozatím nevložil žádný záznam do databáze.

8.5 Activity diagramy

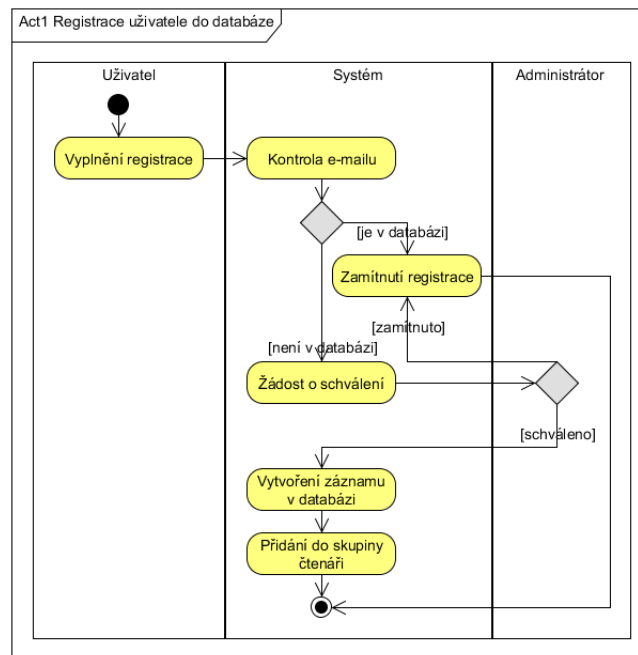
Kapitola obsahuje vysvětlení a popis jednotlivých procesů. Ty bude výsledný systém podporovat.

8.5.1 Act1 Registrace uživatele do databáze

Activity diagram je zobrazen na obrázku 8.13. Uživatel si vytvoří registraci pomocí vyplnění formuláře v programu a odešle jej. Systém automaticky zkontroluje e-mail, zda se již nenachází v databázi. Pokud jej nalezne, registrace je zamítnuta, v opačném případě je odeslána žádost o schválení administrátorovi. Ten může buď odmítnout a registraci tím přerušit (zároveň má možnost přidat záznam do seznamu nežádoucích jako ochranu proti útokům), nebo přijmout. V druhém případě je vytvořen záznam o registrovaném uživateli v databázi a automaticky je přiřazen do základní skupiny, kterou jsou čtenáři.

8.5.2 Act2 Zasílání novinek

Activity diagram je zobrazen na obrázku 8.14. Uživatel si vyhledá kategorii, která jej zajímá. Poté má možnost si ji označit. Tímto aktem se přihlásí k zasílání novinek k danému tématu. Systém automaticky pravidelně každý měsíc vyhledá v každé kategorii záznamy podle data nahrání či editace. Vzhledem k měsíčnímu intervalu jsou vyfiltrována data, která jsou přidána či změněna tentýž měsíc. Poté je vytvořen jejich seznam a ten je pod názvem kategorie



Obrázek 8.13: Act1 Registrace uživatele do databáze.

odeslán administrátorovi. Ten si nechá vyhledat ke každé kategorii uživatele, kteří ji mají označenou a těm zašle zprávu s příslušným seznamem novinek.

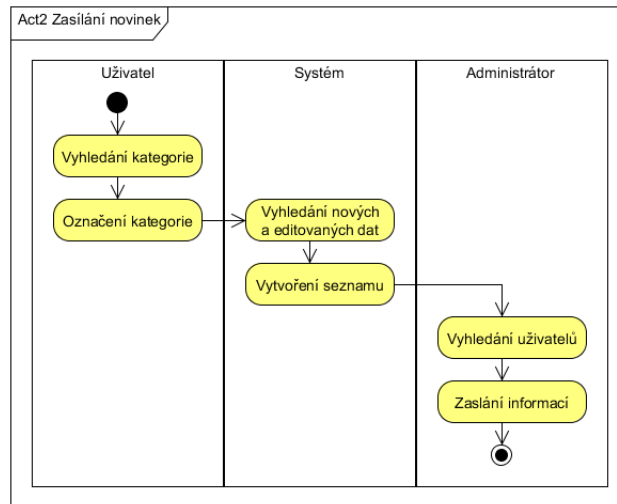
8.5.3 Act3 Vkládání dat

Activity diagram je zobrazen na obrázku 8.15. Uživatel, který si přeje přispět do databáze nějakou informací, musí nejprve vyplnit formulář, kde se mimo jiné nachází název kategorie, kam chce daný příspěvek zařadit. Ta je následně vyhledána systémem v databázi. V případě, že není nalezena, je automaticky vytvořena. Následně jsou informace z formuláře zaevidovány do příslušného záznamu v databázi. Jelikož systém využívá k ukládání dat externí servery cloudových služeb, kontaktuje poté systém přes veřejné API rozhraní příslušnou službu a na tu nahraje data, která uživatel vložil. Zpět odešle odezvu, ve které je uvedena informace o uložení na vzdálený server společně s ID, pod kterým je tam uveden. Toto ID je přidáno do databázového záznamu.

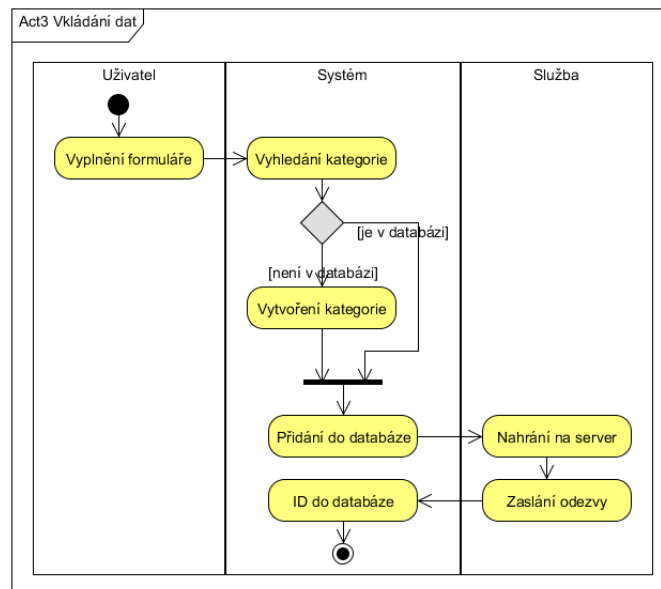
8.5.4 Act4 Editace a mazání dat

Activity diagram je zobrazen na obrázku 8.16. Pro možnost editace či mazání je nutno mít práva buď administrátorská, nebo být vlastníkem. V tomto

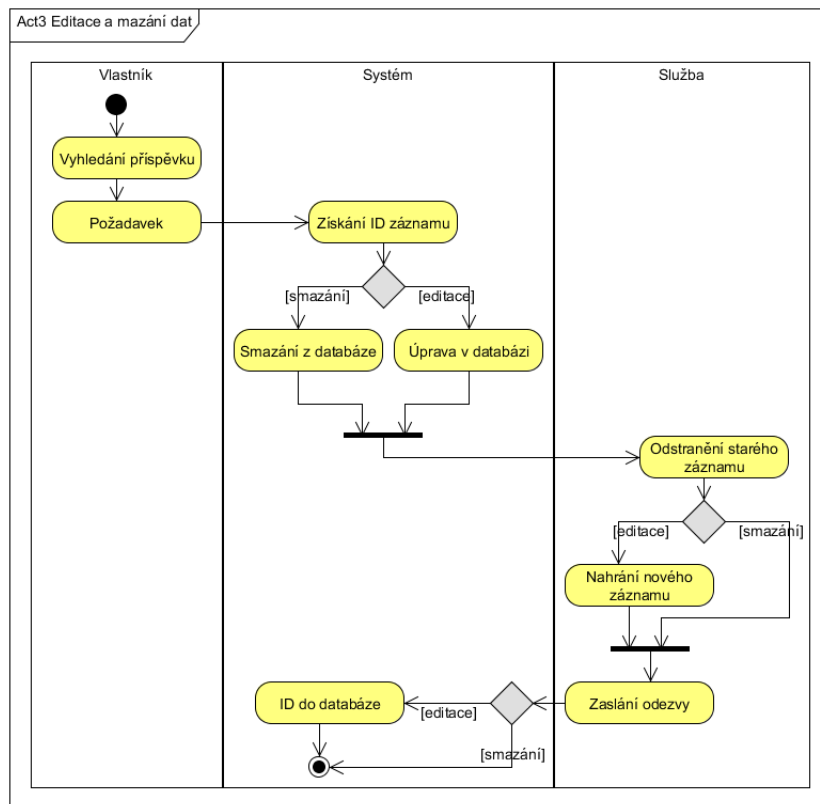
8. ANALÝZA



Obrázek 8.14: Act2 Zasilání novinek.

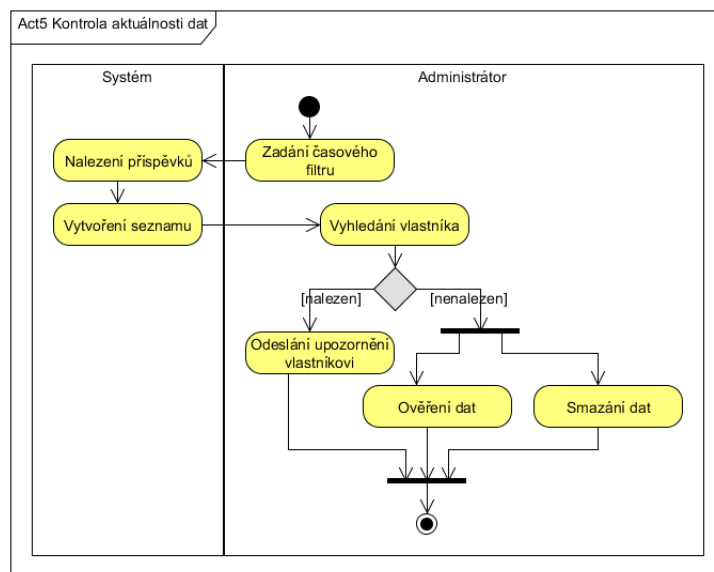


Obrázek 8.15: Act3 Vkládání dat.



Obrázek 8.16: Act4 Editace a mazání dat.

případě je popsána editace a mazání dat samotným vlastníkem. Nejprve si vyhledá vlastní příspěvek, kde poté vyplní formulář. Pokud se jedná o smazání, nejsou dodány další informace, při editaci je potřeba vložit nová data, která mají ta stará nahradit. Systém si automaticky vyhledá ID záznamu, který má být upraven. Při mazání je záznam odstraněn z databáze, u editace jsou přidány informace o změnách (datum, důvod, ...). Vzhledem k použití externích úložišť přes cloudové služby je poté kontaktována příslušná instance a záznam, který zde byl uložen, je označen ke smazání. Pokud se jedná o editaci, je poté provedeno nové nahrání dalšího záznamu na externí server. V obou případech úprav je odeslána zpráva z cloudu o požadavku a v případě editace obsahuje ID nového záznamu. Toto ID je následně přidáno do změn v databázi v příslušné entitě.



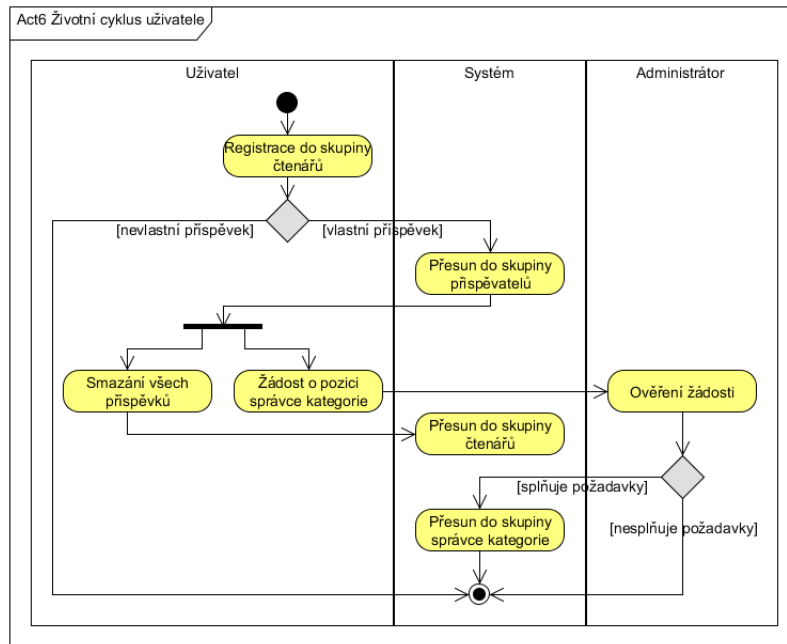
Obrázek 8.17: Act5 Kontrola aktuálnosti dat.

8.5.5 Act5 Kontrola aktuálnosti dat

Activity diagram je zobrazen na obrázku 8.17. Vzhledem ke vzdělávací povaze navrhovaného systému je potřeba kontrolovat, zda data nejsou zastaralá a udržovat je aktuální, případně je označit za „starší“. Administrátor má možnost zadat časový filtr, na základě kterého poté systém vyhledá všechny odpovídající příspěvky. Jedná se o data, která mají datum nahrání či editace (primárně se kontroluje datum editace, pokud není uvedeno, bere se datum nahrání) starší, než je zadané časové období. Následně systém vytvoří seznam, který administrátor zpracuje. Vyhledá vlastníka, zda je stále přítomen v systému. Pokud ano, odešle mu upozornění, že je příspěvek zastaralý. V opačném případě má dvě možnosti. Buď data z databáze vymaže s tím, že je již nemá kdo rozvinout, nebo je může ověřit a ponechat v systému nezměněné.

8.5.6 Act6 Životní cyklus uživatele

Activity diagram je zobrazen na obrázku 8.18. Při registraci je každý nový uživatel automaticky zařazen mezi skupinu čtenářů. Ti mají možnost si prohlížet příspěvky v systému, ale nemají práva propojovat informace. Pokud je chtějí získat, musí vložit do systému data a pak jsou zařazeni mezi přispěvatele. Z této pozice mohou dále postupovat, případně se vrátit zpět do skupiny čtenářů. Ve chvíli, kdy jsou smazány veškeré vlastněné příspěvky, ztrácí automaticky uživatel post přispěvatele. Pokud se chce začít více zapojovat do ko-



Obrázek 8.18: Act6 Životní cyklus uživatele.

munitního úsilí, může zaslat žádost administrátorovi o přiřazení do skupiny správce kategorie. Administrátor poté ověří, zda je oprávněná a je potřeba. Podmínkou pro vstup do správce kategorie je dostatečná účast na jejím dění v roli přispěvatele (dostatečný počet vložených dat či propojení). Pokud to splňuje, je přeřazen do jiné skupiny, jinak zůstává nadále ve své původní.

8.6 Doménový model

V kapitole jsou uvedeny jednotlivé entity navrhované databáze a popis jejich atributů. 8.19 8.20

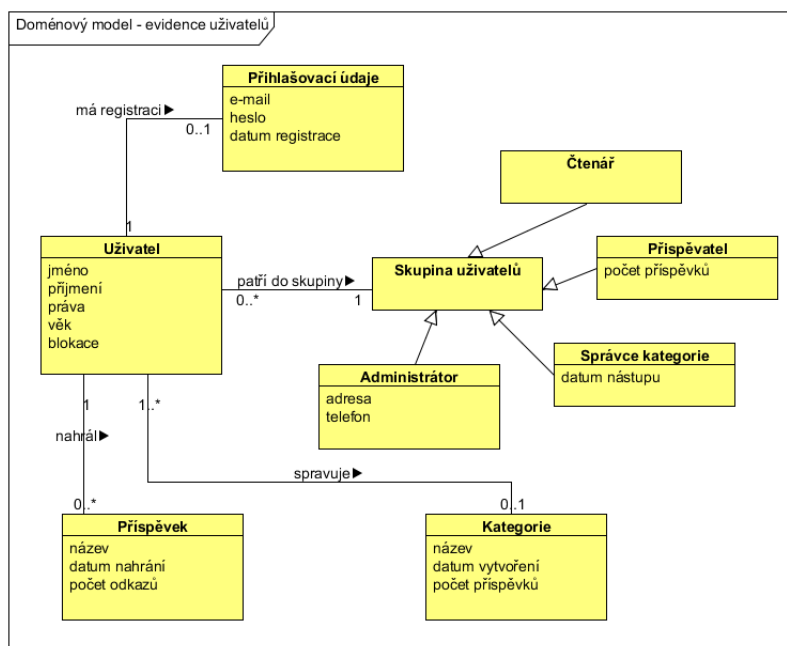
8.6.1 Uživatel

Entita obsahující základní údaje o uživateli v databázi. Jedná se o osoby, které se zaregistrují do systému. 8.1

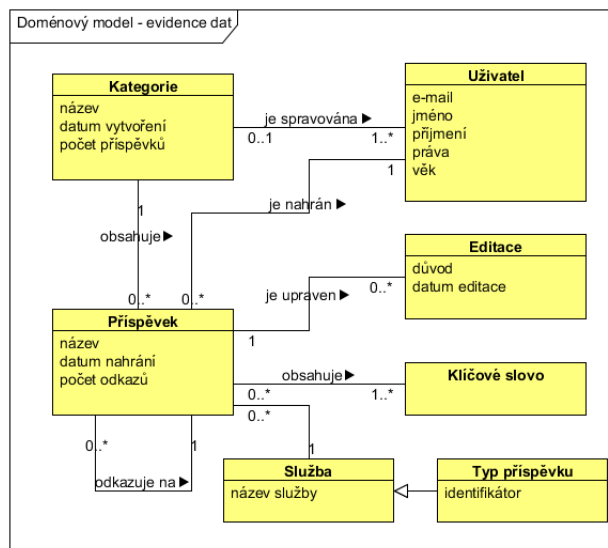
8.6.2 Přihlašovací údaje

Entita obsahující data potřebná k přihlašování do systému. 8.2

8. ANALÝZA



Obrázek 8.19: Doménový model – evidence uživatelů



Obrázek 8.20: Doménový model – evidence dat

Tabulka 8.1: Atributy entity Uživatel.

| Atributy | Poznámky |
|----------|---|
| jméno | Křestní jméno uživatele. |
| příjmení | Příjmení uživatele. |
| práva | Specifikace práv, která v systému uživatel má. |
| věk | Aktuální věk. |
| blokace | Informace o tom, zda je uživatel blokový, či nikoliv. |

Tabulka 8.2: Atributy entity Přihlašovací údaje.

| Atributy | Poznámky |
|------------------|--|
| e-mail | Aktuální e-mail uživatele, pod kterým se přihlašuje. |
| heslo | Bezpečné heslo. |
| datum registrace | Datum, kdy byla provedena registrace do systému. |

8.6.3 Skupina uživatelů

Skupina, do které je uživatel zařazen.

8.6.4 Čtenář

Uživatel, který ještě nepřidal žádná data do systému, může pouze volně prohlížet a číst příspěvky.

8.6.5 Příspěvatel

Uživatel, který vložil alespoň jeden příspěvek do systému. Může odkazovat jednotlivé příspěvky mezi sebou. 8.3

Tabulka 8.3: Atributy entity Příspěvatel.

| Atributy | Poznámky |
|-----------------|---|
| počet příspěvků | Počet vlastních příspěvků (vlozil je do systému). |

8.6.6 Správce kategorie

Uživatel, který má na starosti správu kategorie, ve které dostatečně přispíval.
8.4

Tabulka 8.4: Atributy entity Správce kategorie.

| Atributy | Poznámky |
|---------------|---|
| datum nástupu | Datum, kdy byl přeřazen do skupiny správce kategorie. |

8.6.7 Administrátor

Uživatel, který má celkový dohled nad systémem. Jedná se o zodpovědnou osobu za chod celého systému. 8.5

Tabulka 8.5: Atributy entity Administrátor.

| Atributy | Poznámky |
|----------|--|
| adresa | Aktuální adresa. |
| telefon | Aktuální telefon pro případy problémů se systémem. |

8.6.8 Kategorie

Entita obsahující základní údaje o kategoriích, do kterých jsou rozřazovány příspěvky v databázi. 8.6

Tabulka 8.6: Atributy entity Kategorie.

| Atributy | Poznámky |
|-----------------|-----------------------------------|
| název | Název kategorie. |
| datum vytvoření | Datum vytvoření kategorie. |
| počet příspěvků | Počet příspěvků v dané kategorii. |

8.6.9 Příspěvek

Entita obsahující základní údaje o příspěvcích vkládaných do databáze. 8.7

Tabulka 8.7: Atributy entity Příspěvek.

| Atributy | Poznámky |
|---------------|-------------------------------------|
| název | Název příspěvku. |
| datum nahrání | Datum, kdy byl do databáze vložen. |
| počet odkazů | Počet příspěvků, na které odkazuje. |

8.6.10 Editace

Entita obsahující základní údaje o editaci příspěvku. 8.8

Tabulka 8.8: Atributy entity Editace.

| Atributy | Poznámky |
|---------------|-----------------------------------|
| důvod | Důvod, proč byl příspěvek změněn. |
| datum editace | Datum provedení editace. |

8.6.11 Klíčové slovo

Entita obsahuje seznam klíčových slov, která slouží k přehledu a vyhledávání mezi příspěvky.

8.6.12 Služba

Obsahuje informace o použité službě. 8.9

Tabulka 8.9: Atributy entity Služba.

| Atributy | Poznámky |
|--------------|--|
| název služby | Název cloudové služby, kterou příspěvek využívá. |

8.6.13 Typ příspěvku

Obsahuje informace o konkrétním příspěvku pro účely kontaktování cloudové služby přes její API rozhraní.

Tabulka 8.10: Atributy entity Typ příspěvku.

| Atributy | Poznámky |
|---------------|--|
| identifikátor | ID nebo název, pod kterým je příspěvek uložen. |

Návrh

9.1 Model architektury

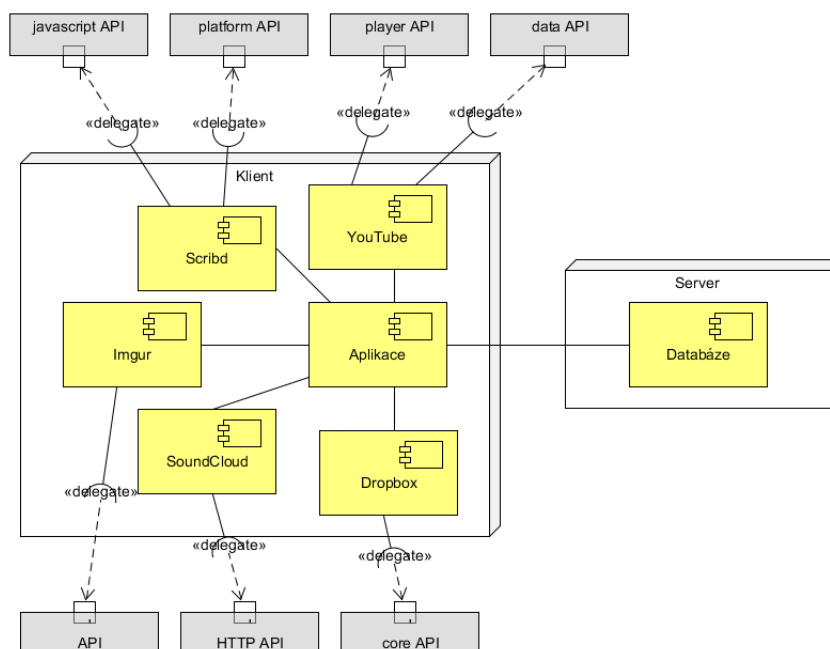
Kapitola obsahuje podrobnější návrh systému. Systém využívá ve velké míře externích serverů pomocí cloudových služeb, aby se redukovaly požadavky na hardware. Přesto je třeba využít serveru pro spravování základních informací pro běh systému. Z tohoto důvodu jsem navrhla dvouvrstvou architekturu.

Server bude spravovat hlavní činnosti klientských aplikací a bude obsahovat databázi osob a příspěvků. Hlavní objem dat bude ovšem spravován externími servery pomocí cloudových služeb. Veškerá data budou nahrávána na tyto služby pomocí veřejných API rozhraní. Zobrazování dat v klientské části systému bude rovněž provozováno přes veřejná API rozhraní. Viz obrázek 9.1.

Jednotlivé části aplikace, které jsou využívány, jsou pouze skripty pro komunikaci a nejsou součástí návrhu. Pro účely této práce je budu považovat za black box s jasně danými vstupními a výstupními parametry. Samotná aplikační logika podobně není součástí návrhu.

9.2 Model komunikace

V této kapitole jsou popsány základní modely komunikace. Vzhledem k tomu, že návrh je především zaměřen na propojení cloudových služeb se samotnou aplikací, je celý návrh veden v abstraktnější rovině. Není konkrétně navržena aplikační logika, databáze, ani jednotlivé skripty. V komunikaci jsou tedy využity tzv. diagramy SSD (System Sequence Diagram), které zobrazují především komunikaci s rozhraními dalších systémů či vnějšími účastníky. Samotná aplikace je pro tyto účely brána pouze jako black box a veškeré její metody jsou popisovány pomocí pseudokódu. Popíši zde tři modelové případy komunikace včetně diagramů, aby bylo zřejmé, jak komunikuje systém s cloudovými službami a serverem.

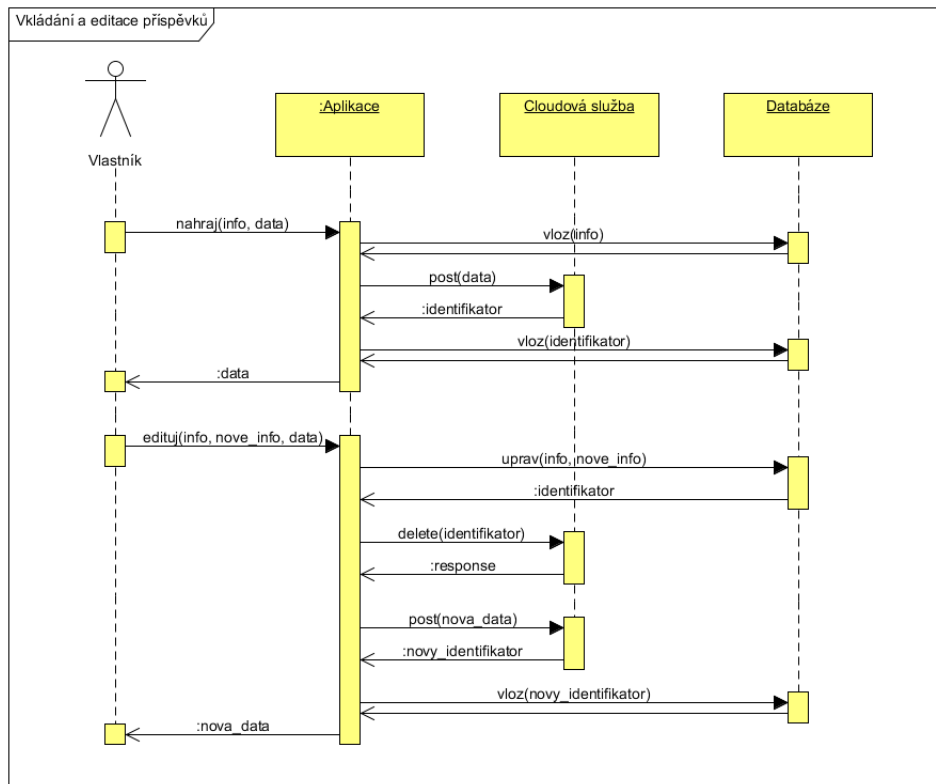


Obrázek 9.1: Model architektury

Vzhledem k tomu, že je využito více cloudových služeb a každá má své unikátní API rozhraní, veškeré metody komunikace jsou zjednodušeny do univerzálního jazyka. Například pro nahrání na externí server je použito v diagramech metody post, ačkoliv podle popisu rozhraní se může metoda jmenovat odlišně. Je to z důvodu, aby zde nebyly popisovány jednotlivé služby, ale ukázal se obecný model komunikace, který je pro všechny shodný.

9.2.1 Vkládání a editace příspěvků

Na diagramu 9.2 je ukázáno, jak komunikují mezi sebou aplikace, databáze a cloudová služba při vkládání a editaci příspěvků. Uživatel (vlastník příspěvku) odešle požadavek na nahrání dat, aplikace uloží informace o přidávaných datech do databáze a odešle zprávu cloudové službě na přidání samotných dat (video, dokument, ...) na externí server. V odpovědi je uveden mimo jiné identifikátor, pomocí kterého lze data na serveru nalézt. Ten je následně přidán k záznamu v databázi. Uživateli se zobrazí v aplikaci nahraná data. Při následné editaci jsou odeslány aplikaci informace o editovaných datech, nové informace a nová data, která mají ta původní nahradit. Při změně dat v databázi je zároveň nazpět vyhledán identifikátor, pod kterým je uložen příspěvek na cloudové službě. Pomocí jej pak aplikace odešle požadavek na smazání z



Obrázek 9.2: Model komunikace – Vkládání a editace dat

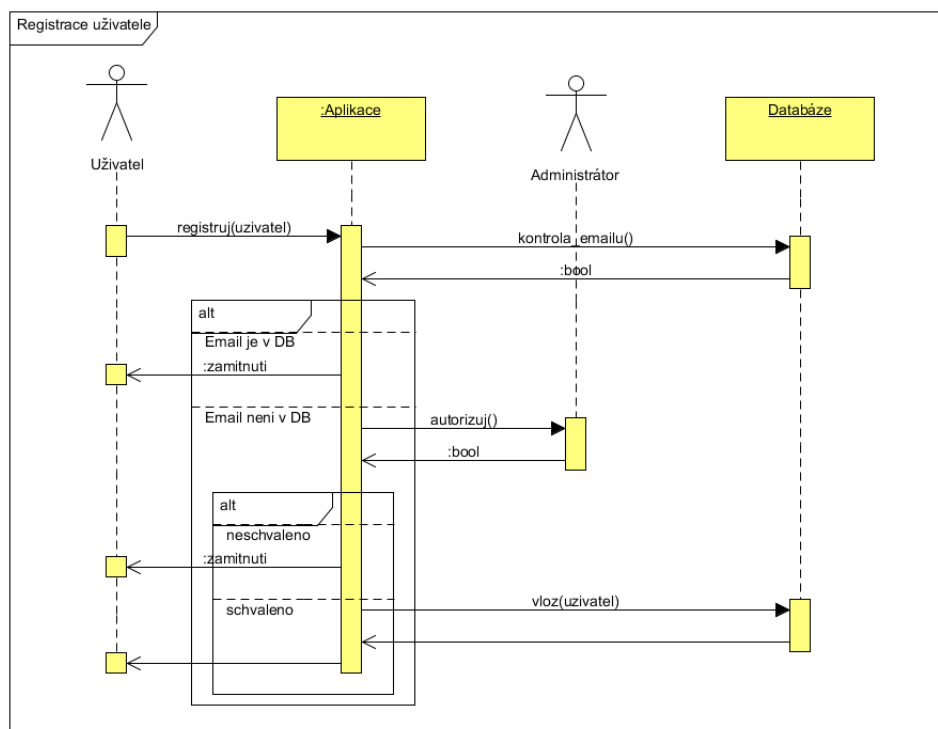
externího serveru. Následně jsou uložena nová data na cloud a zpět je získán nový identifikátor, který je následně přidán v databázi k informacím. Uživateli jsou zobrazena editovaná data.

9.2.2 Registrace uživatele

Na diagramu 9.3 je ukázáno, jak probíhá komunikace mezi aplikací a databází při registraci uživatele. Poté, co odešle uživatel požadavek aplikaci, je v databázi zkontrolováno, zda již e-mail není přítomen. Pokud je nalezen, odešle se zamítnutí. V opačném případě je odeslán požadavek administrátorovi, aby registraci schválil. Při neschválení je opět vráceno zamítnutí, jinak jsou údaje o uživateli zaneseny do databáze.

9.2.3 Vyhledání a zobrazení příspěvků

Na diagramu 9.4 je ukázáno, jak probíhá komunikace mezi aplikací, databází a cloudovou službou při vyhledávání a zobrazování výsledků. Uživatel vyhle-



Obrázek 9.3: Model komunikace – registrace uživatele

dává pomocí klíčových slov, která jsou následně poslána na databázi, ve které vyhledají výsledky. Ty jsou aplikací zobrazeny uživateli v podobě jednoduchého výpisu. Jakmile si vybere konkrétní, který ho zajímá, aplikace zjistí z databáze identifikátor a službu, ve které je příspěvek uložen. Poté kontaktuje přes API rozhraní cloudové úložiště, aby načetlo postupně data zde uložená, která následně zobrazuje v grafickém uživatelském rozhraní.

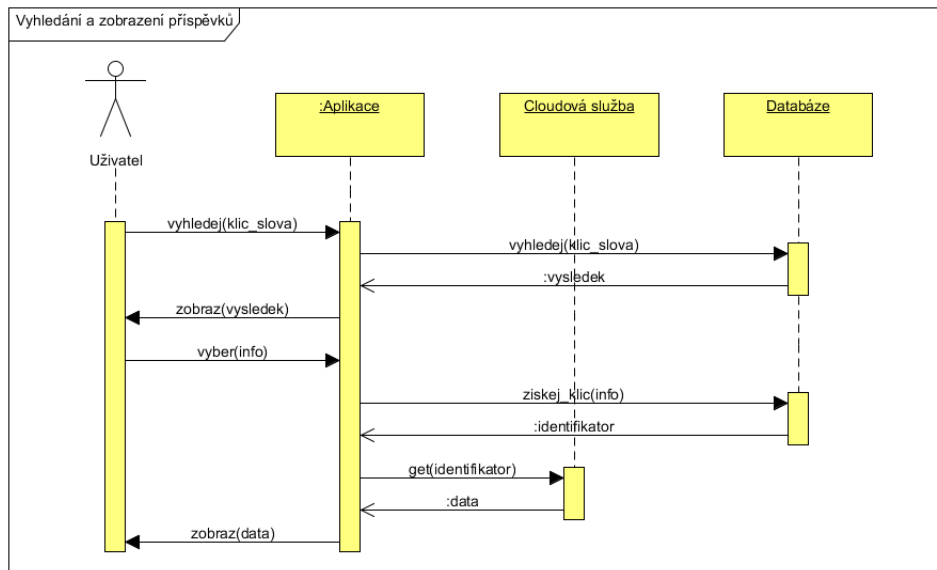
9.3 Datový model

9.3.1 Evidence uživatelů

Viz obrázek ??.

9.3.1.1 Tabulka „Uživatel“

Třída reprezentuje informace o uživateli. 9.1



Obrázek 9.4: Model komunikace – vyhledání a zobrazení příspěvků.

Tabulka 9.1: Atributy tabulky Uživatel.

| Atributy | Datový typ | Poznámky |
|------------|------------|--------------------------------------|
| IDuzivatel | integer | Jednoznačný identifikátor uživatele. |
| jmeno | varchar | Křestní jméno uživatele. |
| prijmeni | varchar | Příjmení uživatele. |
| prava | varchar | Seznam práv uživatele. |
| vek | integer | Současný věk uživatele. |
| blokace | boolean | Indikátor blokace uživatele. |
| IDskupina | integer | ID skupiny, do které uživatel patří. |

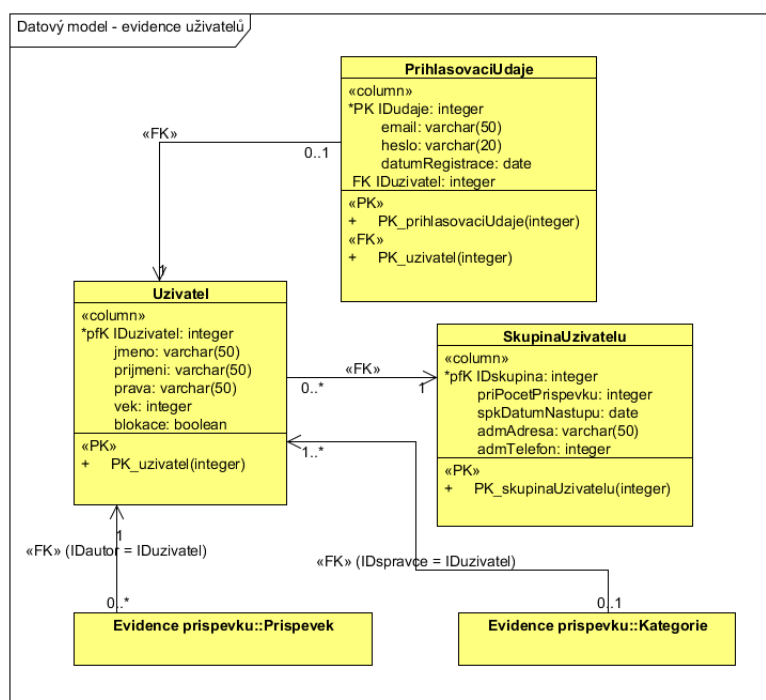
9.3.1.2 Tabulka „PrihlasovaciUdaje“

Třída reprezentuje přihlašovací údaje uživatele, které vyplňuje při registraci.
9.2

9.3.1.3 Tabulka „SkupinaUzivatelu“

Třída reprezentuje záznamy o skupinách, do kterých jsou přiřazováni jednotliví uživatelé. Příspěvatel, správce kategorie a administrátor mají přidány další údaje. 9.3

9. NÁVRH



Obrázek 9.5: Datový model – evidence uživatelů.

Tabulka 9.2: Atributy tabulky PrihlasovacíUdaje.

| Atributy | Datový typ | Poznámky |
|-----------------|------------|---------------------------------------|
| IDudaje | integer | Jednoznačný identifikátor údaje. |
| email | varchar | Aktuální email uživatele. |
| heslo | varchar | Heslo pro přihlášení. |
| datumRegistrace | date | Datum, kdy byla provedena registrace. |
| IDuzivatel | integer | ID uživatele, ke které údaje patří. |

9.3.2 Evidence příspěvků

Viz obrázek ??.

9.3.2.1 Tabulka „Kategorie“

Třída reprezentuje informace o kategorii, ve které jsou umístěny příspěvky.

9.4

Tabulka 9.3: Atributy tabulky SkupinaUzivatelu.

| Atributy | Datový typ | Poznámky |
|-------------------|------------|--|
| IDskupina | integer | Jednoznačný identifikátor skupiny. |
| priPocetPrispevku | integer | Příspěvatel – počet jím vlastněných příspěvků. |
| spkDatumNastupu | date | Správce kategorie – datum nástupu na pozici. |
| admAdresa | varchar | Administrátor – současné sídlo. |
| admTelefon | integer | Administrátor – aktuální telefon. |

Tabulka 9.4: Atributy tabulky Kategorie.

| Atributy | Datový typ | Poznámky |
|----------------|------------|---|
| IDkategorie | integer | Jednoznačný identifikátor kategorie. |
| nazev | varchar | Název kategorie. |
| datumVytvoreni | date | Datum, kdy byla kategorie vytvořena. |
| pocetPrispevku | integer | Počet příspěvků v dané kategorii. |
| IDSpravce | integer | ID uživatele, který je správcem této kategorie. |

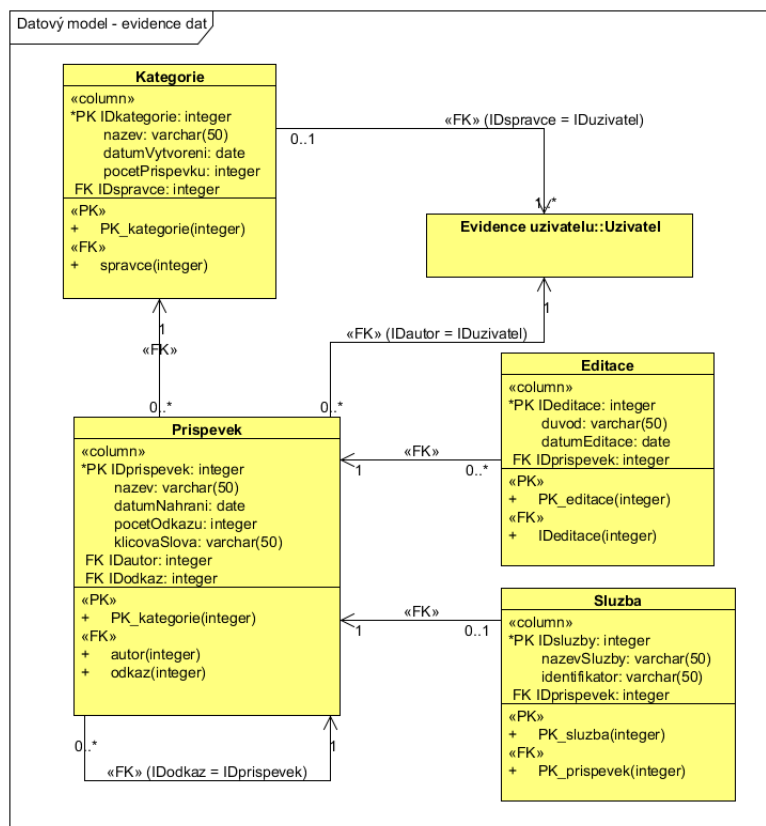
9.3.2.2 Tabulka „Prispevek“

Třída reprezentuje informace o jednotlivých příspěvcích, které jsou do systému vkládány. 9.5

Tabulka 9.5: Atributy tabulky Prispevek.

| Atributy | Datový typ | Poznámky |
|--------------|------------|--|
| IDprispevek | integer | Jednoznačný identifikátor příspěvku. |
| nazev | varchar | Název příspěvku. |
| datumNahrani | date | Datum, kdy byl příspěvek nahrán do databáze. |
| pocetOdkazu | integer | Počet příspěvků, na které tento odkazuje. |
| klicovaSlova | varchar | Seznam klíčových slov, podle kterých lze vyhledávat. |
| IDautor | integer | ID autora, který vložil příspěvek. |
| IDodkaz | integer | ID příspěvku, na který odkazuje. |

9. NÁVRH



Obrázek 9.6: Datový model – evidence dat.

9.3.2.3 Tabulka „Editace“

Třída reprezentuje informace o editaci příspěvku, pokud byla nějaká provedena. 9.6

Tabulka 9.6: Atributy tabulky Editace.

| Atributy | Datový typ | Poznámky |
|--------------|------------|---|
| IDeditace | integer | Jednoznačný identifikátor editace. |
| duvod | varchar | Důvod, proč byla původní informace změněna. |
| datumEditace | date | Datum, kdy byla editace provedena. |
| IDpríspevek | integer | ID příspěvku, ke kterému editace patří. |

9.3.2.4 Tabulka „Sluzba“

Třída reprezentuje informace o službě, pomocí které je daná informace uložena. 9.7

Tabulka 9.7: Atributy tabulky Sluzba.

| Atributy | Datový typ | Poznámky |
|---------------|------------|---|
| IDsluzby | integer | Jednoznačný identifikátor cloudové služby. |
| nazevSluzby | varchar | Název cloudové služby. |
| identifikator | varchar | Identifikátor, pod kterým je uložena informace na cloudové službě (číslo ID, název souboru,...). |
| IDprispevek | integer | ID příspěvku, který službu využívá k uložení dat. |

9.4 Možnosti praktického využití

V následující sekci popíši tři příklady praktického využití navrhovaného systému. Veškeré příklady zde uvedené demonstrují výhody, jež přináší používání systému. Nejedná se o nové postupy, vše je založeno na již existujících systémech, jedná se hlavně o jejich zefektivnění a zpřístupnění většímu množství lidí (větší komunity).

9.4.1 Studenti před státními závěrečnými zkouškami

Studium před závěrečnými zkouškami bývá náročné a obsáhlé. Sebrání veškerých potřebných informací tedy zabere jednomu člověku velké množství času. Jedním z trendů je scházet se ve skupinkách studentů (komunita dle definice uvedené v první kapitole), kdy každý účastník přinese část informací a společně je pak sdílí s ostatními. Stejným způsobem se dá využít i výše navrhovaný systém s tím, že se nemusí pořádat žádné osobní schůzky, veškeré sebrané informace lze přes systém jednoduše nahrát na cloudovou službu a propojit je s ostatními souvisejícími příspěvky od kolegů. Tím se ušetří další čas a navíc přispěvatelů může být větší množství.

9.4.2 Šachisté

Příkladem mimo studentskou sféru je komunita šachistů. Jedná se o skupinu se stejným zájmovým zaměřením, kteří si chtějí mezi sebou sdělovat nejen informace, jež se dají dohledat volně na internetu, ale také sdílet partie, kterých se sami účastnili, či je pouze sledovali. Takové záznamy pak mohou do výše navrhovaného systému nahrát ať už v podobě videí, či textové šachové notace.

Kromě toho nahrávání obrázků může sloužit například k řešení složitějších šachových situací, nebo řešení kvízových typů úloh. Veškeré tyto činnosti šachisté zpravidla provozují při osobních schůzkách a výše navržený systém jim jen usnadní tuto formu sdílení informací.

9.4.3 Sportovci

Jelikož systém nemá specifikována omezení co se týče komunity, která jej využívá, další příklad uvedu na sportovcích. Kromě očividných znalostí šířících se v takové komunitě jako je zdravá výživa, techniky sportování a podobně systém umožňuje i sdílení vlastních výsledků, porovnávání se světovými údaji, které mohou být uloženy do systému v přehledných tabulkách (Scribd umožňuje i zobrazování spreadsheet typu dokumentů). Navíc informace o zdravé výživě vložené do systému mohou být propojeny s lékařskými informacemi (ať už oficiálními, nebo formou domácího lékaře) a umožnit tak snadnou dohledatelnost souvisejících faktorů.

Shrnutí praktické části

Výše navrhovaný systém (jeho první verze) umožňuje evidenci uživatel a příspěvků. Obsahuje způsob ukládání a zobrazování dat ve formě videozáznamů, audiozáznamů, dokumentů a obrázků. Vše ostatní je řešeno obecným úložištěm. Všechny tyto způsoby uchovávání znalostí jsou ukládány na cloudové služby a pro jejich zobrazení není nutné stažení s výjimkou obecného úložiště. Tímto splňuje systém tři ze čtyř pilířů vzdělávání stanovených v první kapitole (sleduji, poslouchám, čtu). Co se týče komunikace, prozatím je v této verzi umožněna pouze formou vkládání dokumentů. Je to z toho důvodu, že samotná komunikace v rámci komunity je poměrně rozsáhlá kapitola a odvíjí se od samotného charakteru uživatelů. V úvahu přichází například propojení se sociálními sítěmi (Facebook, Twitter), různými fóry případně prosté vkládání komentářů přes textové pole do databáze.

Jak jsem již uvedla výše, jedná se o prvotní verzi systému, která pokrývá pouze základní funkčnost. Prvním krokem pro správné fungování by tedy bylo doplnění stávajícího návrhu ještě o komunikační část umožňující efektivnější komentování shromažďovaných dat.

Popsané možnosti praktického využití jsou jen příklady, o které jsem se opírala v rámci navrhování celého systému. Jedná se však o dostatečně obecný návrh, aby se dal aplikovat i na další podobné komunity.

V návrhu je počítáno s tím, že jediný vlastník bude mít práva administrátora. To by mohl být problém, pokud by více komunit chtělo sdílet stejný systém a mít tak více oddělených samostatně spravovaných částí, které by ovšem obsluhoval jediný server. Vzhledem k obecnosti návrhu ovšem není problém upravit částečně toto pravidlo, samotná struktura systému se tím nijak nenaruší a zásah je minimální.

Závěr

Cílem této práce byl popis konceptu informačních systémů, analýza existujících řešení, které je možno v tomto kontextu využít, a následně návrh systému pro jejich agregaci.

V první kapitole jsem rozebrala podrobně problematiku vzdělávání a koncept souvisejících informačních systémů. Popsala jsem pilíře, s jejichž pomocí lze takové systémy koncipovat. Základními faktory jsou videozáznamy, audiozáznamy, dokumenty a komunikace.

Druhá kapitola se zaměřuje na dostupnost internetových služeb a jejich možnosti využití v rámci komunitního vzdělávání. Z těchto služeb jsem zvolila cloudové služby, které redukuje požadavky na hardwarové zázemí. Zároveň je výběr omezen na služby poskytované zdarma, díky čemuž se snižují i finanční náklady na provoz navrhovaného systému. Následně jsou specifikovány jednotlivé cloudy, které je možno využít v navrhovaném systému především díky jejich veřejnému API rozhraní.

Následujících pět kapitol popisuje jednotlivé vybrané cloudové služby a jejich API rozhraní. Jedná se konkrétně o YouTube (videozáznamy), Scribd (dokumenty), Imgur (obrázky), Soundcloud (audiozáznamy) a Dropbox (obecné úložiště). Ve všech případech splňují požadavky na poskytování služeb zdarma a existenci veřejného API rozhraní. To umožňuje přenos dat mezi cloudem a navrhovaným systémem bez nutnosti stahování. V případě YouTube a Scribd je to i existence relevantních materiálů.

Poslední dvě kapitoly se zabývají samotnou analýzou a návrhem informačního systému pro komunitní vzdělávání. Ten agreguje výše zmíněné cloudové služby. Na serveru, kde systém bude provozován, bude uložena pouze databázi uživatelů a příspěvků a bude jej možno tedy provozovat jak na osobním počítači, tak i na pronajatém serveru. V rámci zaměření na komunitu je zde řešena i hierarchie uživatelů usnadňující správu systému.

Pro úplné splnění požadavků pro informační vzdělávací systém stanove-

ných v úvodní kapitole by systém měl být rozšířen ještě o komunikační prostředky a vytváření komentářů. Jelikož se v tomto případě jedná o poměrně rozsáhlou oblast a záleží také na zaměření konkrétní komunity, která bude systém využívat, je toto zmíněno pouze v dalších krocích. Navržený systém je pouze první verzí obsahující základní funkčnost.

Práce je koncipována abstraktně a analyticky, aby bylo možno ji upravit podle konkrétních požadavků.

Výsledná práce by měla přiblížit možnosti vzdělávání pomocí internetových služeb a především také upozornit na možnosti využití cloudových řešení. Ty se stále více rozšiřují a vzhledem k jejich specifikacím umožňujícím snadné využití úložiště bez technických prostředků jsou trendem i do budoucna. Zároveň je zde přiblížena možnost tyto služby agregovat za účelem jednotného schraňování informací a rozšiřování znalostí v rámci komunity.

Při zpracovávání práce jsem si vyzkoušela průběh návrhu informačního systému. Ačkoliv to není rozsáhlé dílo, přesto bylo náročné všechny myšlenky smysluplně skloubit dohromady a udržet konzistentní data. Každá sebemenší úprava návrhu měla za následek revizi dosud navržených částí a mnohých úprav. Práce na tomto návrhu byla pro mne cennou zkušeností.

Literatura

- [1] *Think „Exciting“: E-Learning and the Big „E“* [online]. Bernard LUSKIN, 2010-, strana naposledy edit. 2010-03-03 [cit. 2015-02-02]. Anglická verze. Dostupný z: <http://www.educause.edu/ero/article/think-exciting-e-learning-and-big-e>
- [2] *Natural Learning in Higher Education* [online]. J. Scott ARMSTRONG, *Encyclopedia of the Sciences of Learning*, 2010-, strana naposledy edit. 1-1-2011 [cit. 2015-02-02]. Anglická verze. Dostupný z: <https://marketing.wharton.upenn.edu/files/?whdmsaction=public:main.file&fileID=8113>
- [3] *Edux* [online]. Edux s.r.o, strana naposledy edit. 2014 [cit. 2015-02-02]. Anglická verze. Dostupný z: <http://www.edux.cz/>
- [4] *Moodle* [online]. Moodle.com, 2012-, naposledy edit. 2014 [cit. 2015-02-03]. Anglická verze. Dostupný z: <https://moodle.org/>
- [5] Lyn TETT. *Community Education, Lifelong Learning And Social Inclusion (Policy and Practice in Education)* 2. vydání Edinburgh: Dunedin Academic Press, 2006. 90s. ISBN 1-903765-56-0.
- [6] *2. díl - SEO - Princip internetových vyhledávačů a Google Page-Rank* [online]. David ČÁPKA, itnetwork.cz, 2015- [cit. 2015-02-03]. Dostupný z: <http://www.itnetwork.cz/seo-optimalizace-webu-pro-vyhledavace-google-pagerank>
- [7] *Jak (ne)zabíjet čas na internetu – online vzdělávání* [online]. Pavel HORALÍK, Websupport.cz, 2013-, strana naposledy edit. 2013-10-28 [cit. 2015-02-05]. Dostupný z: <http://blog.websupport.cz/2013/10/jak-nezabijet-cas-na-internetu-online-vzdelavani/>
- [8] *Coursera* [online]. Coursera Inc., 2015-, strana naposledy edit. 2015 [cit. 2015-02-05]. Anglická verze. Dostupný z: <https://www.coursera.org/>

- [9] *Udacity* [online]. Udacity Inc., 2011-2015, strana naposledy edit. 2015 [cit. 2015-02-05]. Anglická verze. Dostupný z: <https://www.udacity.com/>
- [10] *Future Learn* [online]. FutureLearn, 2015-, strana naposledy edit. 2015 [cit. 2015-02-05]. Anglická verze. Dostupný z: <https://www.futurelearn.com/>
- [11] *edX* [online]. edX Inc., 2015-, strana naposledy edit. 2014-10-22 [cit. 2015-02-05]. Anglická verze. Dostupný z: <https://www.edx.org/>
- [12] *Khanova škola* [online]. Khan Academy Inc., 2015-, strana naposledy edit. 2014-05-08 [cit. 2015-02-05]. Anglická verze. Dostupný z: <https://www.khanacademy.org/>
- [13] Timeline of popular Internet services. In *Wikipedie : otevřená encyklopedie* [online]. St. Petersburg (Florida) : Wikimedia Foundation, 2001-, strana naposledy edit. 2015-01-29 [cit. 2015-02-07]. Anglická verze. Dostupný z: http://en.wikipedia.org/wiki/Timeline_of_popular_Internet_services
- [14] *Prozkoumejte multitenantní podobu cloudových služeb.* [online]. Markéta PASTUCHOVÁ, 2012 [cit. 2014-25-11]. Dostupný z: <http://www.ictmanazer.cz/2012/03/prozkoumejte-multitenantni-podobu-cloudovych-sluzeb/>
- [15] *Cloudové služby: data i počítače v oblacích.* [online]. Jan RYLICH, Ikaros [online]. 2012, ročník 16, číslo 9 [cit. 2015-02-12]. ISSN 1212-5075. Dostupný z: <http://ikaros.cz/node/13965>
- [16] *Cloud computing.* [online]. www.uc-solutions.net, [cit. 2015-02-10]. Anglická verze. Dostupný z: <http://www.uc-solutions.net/index.php/solutions/cloud-computing>
- [17] *ProgrammableWeb: the world's largest API repository.* [online]. ProgrammableWeb.com, 2015 [cit. 2015-01-13]. Anglická verze. Dostupný z: <http://www.programmableweb.com/apis/directory>
- [18] *YouTube.* [online]. Google Inc., 1998- [cit. 2015-02-07]. Anglická verze. Dostupný z: <https://developers.google.com/youtube/?hl=cs>
- [19] *Getting Started with the YouTube Data API.* [online]. Google Inc., 1998- [cit. 2015-02-07]. Anglická verze. Dostupný z: <https://developers.google.com/youtube/v3/getting-started?hl=cs>
- [20] *YouTube ActionScript 3.0 Player API Reference.* [online]. Google Inc., 1998- [cit. 2015-02-07]. Anglická verze. Dostupný z: https://developers.google.com/youtube/flash_api_reference?hl=cs

-
- [21] *Cloud computing*. [online]. Maria SPÍNOLA, B2B TIC Content Marketing & Copywriting, 2015 [cit. 2015-02-09]. Anglická verze. Dostupný z: <http://www.mariaspinola.com/cloud-computing/>
- [22] *Scribd*. [online.] Scribd Inc., 2007- [cit. 2015-02-07]. Anglická verze. Dostupný z: <https://www.scribd.com/developers/>
- [23] *Platform API Reference*. [online.] Scribd Inc., 2007- [cit. 2015-02-07]. Anglická verze. Dostupný z: <https://www.scribd.com/developers/platform/api>
- [24] *Scribd Javascript API*. [online.] Scribd Inc., 2007- [cit. 2015-02-07]. Anglická verze. Dostupný z: https://www.scribd.com/developers/javascript_api
- [25] *Imgur*. [online.] Imgur.com, 2009- [cit. 2015-02-07]. Anglická verze. Dostupný z: <https://api.imgur.com/>
- [26] *Imgur API endpoints*. [online.] Imgur.com, 2009- [cit. 2015-02-07]. Anglická verze. Dostupný z: <https://api.imgur.com/endpoints/image>
- [27] *Soundcloud*. [online.] Soundcloud, 2007- [cit. 2015-02-07]. Anglická verze. Dostupný z: <https://soundcloud.com/>
- [28] *Soundcloud HTTP API Guide*. [online.] Soundcloud, 2007- [cit. 2015-02-07]. Anglická verze. Dostupný z: <https://developers.soundcloud.com/docs/api/guide>
- [29] *Dropbox*. [online.] Dropbox Inc., 2013- [cit. 2015-02-07]. Anglická verze. Dostupný z: <https://www.dropbox.com/developers>
- [30] *Dropbox Core API*. [online.] Dropbox Inc., 2013- [cit. 2015-02-07]. Anglická verze. Dostupný z: <https://www.dropbox.com/developers/core/docs>

Seznam použitých zkratk

GUI Graphical user interface

API Application interface

HTTP Hypertext transfer protocol

HTTPS Hypertext transfer protocol service

SQL Structured query language

SSD System sequence diagram

UC Use case

URL Uniform resource locator

Obsah přiloženého CD

| | | |
|------------------------|-------|---|
| images | | originální verze obrázku v maximálním rozlišení |
| text.pdf | | text práce ve formátu PDF |
| text.tex | | zdrojová forma práce ve formátu LATEX |
| cvut-logo-bw.pdf | | PDF s logem CVUT |
| FITthesis.cls | | šablona práce pro LATEX |
| mybibliographyfile.bib | | soubor s bibliografií pro BibTex |
| readme.txt | | stručný popis obsahu CD |