



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Databáze konferencí a publikací II
Student:	Bc. Stanislav Štípl
Vedoucí:	doc. Ing. Petr Fišer, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2018/19

Pokyny pro vypracování

Cílem práce je pokračovat ve vývoji stávající aplikace pro evidenci konferencí a publikací. Použitá platforma: Nette framework, MySQL.

Aplikace podporuje evidenci konferencí, jejich ro níků a publikací, tyto informace jsou propojené. V rámci diplomové práce proveďte následující:

- odstraní drobných chyb, vylepšení uživatelské p ív tivosti,
- zavedení pojmu "citace publikace",
- zavedení automatického rozpoznávání referencí publikace, alespo ň jakým jednoduchým zp ůsobem,
- zavedení správy "vlastních" (autorských) publikací a správy citací,
- zavedení uživatelských "tag ů" u publikací,
- vylepšení vyhledávání publikací (podle kombinace r ůzných kritérií, vyhledávání podle poznámek, tag ů).

ásti, ve kterých budete d lat úpravy, d ůkladn ů otestujte na funk nost, proveďte uživatelské testování, p ípadn ů zaveďte automatizované testování.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdíková, CSc.
d ůkan

V Praze dne 9. zá ří 2017



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Diplomová práce

Databáze konferencí a publikací II

Bc. Stanislav Štipl

Katedra softwarového inženýrství

Vedoucí práce: doc. Ing. Petr Fišer, Ph.D.

16. února 2018

Poděkování

Děkuji doc. Ing. Petru Fišerovi, Ph.D. za odborné vedení diplomové práce.
Děkuji svým rodičům za podporu při studiu.

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 16. února 2018

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2018 Stanislav Štipl. 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 a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Štipl, Stanislav. *Databáze konferencí a publikací II*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato diplomová práce se zabývá rozšířením aplikace pro správu konferencí a publikací, která je využívána na FIT ČVUT. Práce uvádí detailní popis a analýzu stávající aplikace. Na základě této analýzy je navrženo její rozšíření. Úpravy spočívají ve vylepšení uživatelské přívětivosti, zavedení evidence referencí, citací a jejich rozpoznání, evidence vlastních publikací uživatele, zavedení uživatelských štítků u publikací, zavedení možnosti uvádět více ISBN/ISSN u časopisů, ročníků konferencí a publikací a dalších drobnějších úpravách.

Klíčová slova Správa publikací, správa konferencí, aplikace, reference, citace, štítky, webová aplikace, Nette, MySQL.

Abstract

The thesis presents extension of Web application for publication and conference management software used at the Faculty of Information Technology of Czech Technical University in Prague. The thesis describes and analyses the current application. On the basis of this analysis an extension of the application is designed. The revision improves user interface and implements possibility to add references and citations, their recognition, to keep a record

of user's own publications, user's own tags for publications, possibility to add more ISBN/ISSN to journals, conference years and publications as well as additional minor modifications.

Keywords Publication management software, conference management software, references, citations, tags, Web application, Nette, MySQL.

Obsah

Úvod	1
1 Cíl práce	3
1.1 Odstranění drobných chyb	3
1.2 Vylepšení uživatelské přívětivosti	3
1.3 Zavedení pojmu citace publikace	4
1.4 Zavedení automatického rozpoznávání referencí publikace	4
1.5 Zavedení správy vlastních (autorských) publikací	4
1.6 Zavedení uživatelských tagů u publikací	4
1.7 Vylepšení vyhledávání publikací	4
2 Analýza a návrh	5
2.1 Obecný popis stávající aplikace	5
2.2 Technický popis stávající aplikace	6
2.3 Bezpečnost stávající aplikace	7
2.4 Požadavky na úpravy aplikace	8
2.5 Role v systému	10
2.6 Případy užití	11
2.7 Stručný popis datového modelu původní aplikace	21
2.8 Návrh změn datového modelu	27
3 Realizace	31
3.1 Implementace nových funkcí	31
3.2 Nové CRUD komponenty	31
3.3 Rozpoznávání referencí	32
3.4 Dependency Injection	34
3.5 Zabezpečení	35
3.6 Správa závislostí pomocí Composeru	36
3.7 Aktualizace knihoven	36

4 Testování	37
Závěr	39
Literatura	41
A Seznam použitých zkratk	43
B Obsah přiloženého CD	45

Seznam obrázků

2.1	Role v systému	10
2.2	Případy užití	12
2.3	UML Class diagram: původní stav	22
2.4	UML Class diagram: návrh změn	28
2.5	UML Class diagram: finální stav	30

Seznam tabulek

3.1	Úspěšnost rozpoznávání referencí	33
3.2	Shrnutí úspěšnosti rozpoznávání referencí	34

Úvod

Po celém světě se pořádá velké množství odborných konferencí a vzniká řada vědeckých publikací. Jejich množství každým dnem stoupá a současně s tím se objevuje i požadavek, jak se v takovém množství nových informací co nejlépe orientovat a využívat je. Touto problematikou se zabývají i jednotlivé vysoké školy a vědecké ústavy, které musí ve své činnosti vycházet z již publikovaných údajů.

Skupina pedagogů a studentů z ČVUT pro své potřeby vyvinula aplikaci, která slouží pro správu publikací a konferencí. Tato aplikace byla postupně vyvíjena v rámci několika bakalářských a diplomových prací. Má práce přímo navazuje na bakalářskou práci [1] a diplomovou práci [2]. Na základě požadavků uživatelů aplikace se jí snaží dále rozšířit a opravit případné nedostatky.

Cíl práce

Zadání práce ukládá pokračovat ve vývoji stávající aplikace pro evidenci konferencí a publikací. Konkrétně jsou v zadání zmíněny následující body:

- odstranění drobných chyb, vylepšení uživatelské přívětivosti,
- zavedení pojmu „citace publikace“,
- zavedení automatického rozpoznávání referencí publikace, alespoň nějakým jednoduchým způsobem,
- zavedení správy „vlastních“ (autorských) publikací a správy citací,
- zavedení uživatelských „tagů“ u publikací,
- vylepšení vyhledávání publikací (podle kombinace různých kritérií, vyhledávání podle poznámek, tagů).

Níže se jednotlivým bodům věnuji podrobněji.

1.1 Odstranění drobných chyb

Při používání aplikace uživatelé postupně zjišťovali chyby a nedostatky, které je třeba opravit. Další chyby pak byly způsobeny mými úpravami. Chyby byly evidovány v systému GitLab a postupně jsem je opravoval.

1.2 Vylepšení uživatelské přívětivosti

Uživatelské rozhraní je implementováno ve frameworku Bootstrap. Ten zajišťuje jednotný vzhled různých částí aplikace a možnost responzivního zobrazení. Aplikace často využívá modálních oken. Pro některé funkce jsou použity další JS knihovny na straně klienta. Vylepšení uživatelské přívětivosti v tomto případě spočívá například v umístění již existujících ovládacích prvků na místa, kde budou pro uživatele snadněji dostupné.

1.3 Zavedení pojmu citace publikace

Aplikace eviduje jednotlivé publikace, ale neumí evidovat vztahy mezi nimi. Tedy žádným způsobem nezaznamenává, které další publikace jsou v publikaci citovány, nebo naopak, které ji citují. Ze zadání vyplývá potřeba doplnit možnost vazby mezi publikacemi formou citací.

1.4 Zavedení automatického rozpoznávání referencí publikace

Zavedení automatického rozpoznávání referencí publikace navazuje na předchozí bod. Cílem je, aby nebylo nutné ručně vyhledávat každou citovanou publikaci, ale alespoň u některých publikací došlo k automatickému rozpoznání na základě zkopírovaného seznamu citací. Nutnou podmínkou pro rozpoznání je, že citované publikace musí být v systému již vedeny.

1.5 Zavedení správy vlastních (autorských) publikací

Aplikace dosud nemá podporu pro to, že někdo z jejích uživatelů je zároveň autorem některých publikací. Autoři publikací a uživatelé jsou v aplikaci vedeni zcela odděleně. Cílem tohoto bodu je zavést relaci mezi autory a uživatele a umožnit uživatelům správu vlastních publikací a citací.

1.6 Zavedení uživatelských tagů u publikací

Podobně jako anotace, které již aplikace umí, budou „tagy“ (štítky) moci být zadávány každým uživatelem jako soukromé nebo veřejné. Na rozdíl od anotací však štítky budou nejprve vytvořeny na jednom místě a až následně přiřazeny ke konkrétním publikacím.

1.7 Vylepšení vyhledávání publikací

Aplikace v současné době umí vyhledávat publikace, ale jen pomocí fulltextového vyhledávání v kombinaci s kategoriemi, které jsou spojeny pomocí logického součtu (OR) nebo součinu (AND). Kromě toho je k vyhledávání možné připojit filtr na publikace označené hvězdičkou. Cílem vylepšení je umožnit složitější podmínky vyhledávání (kombinace podmínek) a umožnit vyhledávání podle anotací, štítků atd.

Analýza a návrh

2.1 Obecný popis stávající aplikace

Aplikace slouží pro uchování informací o vědeckých konferencích a publikacích. Byla vyvinuta v rámci několika bakalářských a diplomových prací na FEL a FIT ČVUT.

2.1.1 Správa konferencí

Uživatelé mohou přidávat a editovat konference a jejich ročníky, přiřadit k nim publikace a workshopy. Záznam konference obsahuje její název, zkratku, její první ročník a popis. Správce konferencí může u každého ročníku označit příznak archivace, případně označit celou konferenci jako mrtvou. Každý uživatel konferencí může označovat ročníky konferencí uživatelskou hvězdičkou. Konference lze zařazovat do dvou typů kategorií. ACM kategorie [3] mají stromovou strukturu, zatímco vlastní kategorie stromovou strukturu nemají. Ročníky konferencí lze filtrovat podle klíčových slov, obou typů kategorií, stavu konference a jejího označení uživatelskou hvězdičkou a podle písmen anglické abecedy. Ročníky konferencí lze řadit podle názvu, zkratky, roku, místa a několika dat. Výsledky filtrování jsou stránkované.

2.1.2 Správa publikací

Uživatelé mohou přidávat a editovat publikace. Informace zaznamenávané u publikace se liší podle typu publikace. Typy publikace vycházejí z formátu BibTex [4]. Nejčastějšími typy publikace jsou inproceedings (článek na základě konference) a article (článek v časopise). Nově přidaná publikace musí být schválena administrátorem. Publikace lze filtrovat podle klíčových slov, označení uživatelskou hvězdičkou a podle písmen anglické abecedy. Výsledky filtrování jsou stránkované. Vedle toho lze publikace vyhledávat fulltextově, přičemž výsledek vyhledávání lze filtrovat podle logického součtu nebo logic-

kého součinu vlastních kategorií. Fulltextové vyhledávání prohledává i nahrané dokumenty. Kromě hvězdiček mohou uživatelé k publikacím přidat vlastní anotace. Ty mohou být buď privátní (zobrazí se pouze danému uživateli a administrátorovi) nebo globální (zobrazí se všem uživatelům). Je možné zobrazit citační záznamy publikací ve formátech definovaných administrátorem (např. ČSN ISO 690, IEEE, APA). Publikaci je také možné importovat z formátů BibTex, EndNote a RefWorks a exportovat její údaje do těchto formátů.

2.1.3 Ostatní funkce

Uživatelé se do aplikace přihlašují přes SSO, konkrétně přes fakultní Shibboleth. Z historických důvodů aplikace obsahuje také podporu pro LDAP a pro přihlašování uživatelským jménem a heslem. Ve třetím uvedeném případě je možné obnovit zapomenuté heslo pomocí e-mailu. Tato možnost však není dostupná uživatelům, kteří se přihlašují např. přes Shibboleth. Každý uživatelský účet může využívat právě jeden způsob autentizace. Administrátor může uživatelům přidělovat role, podle kterých se určují jejich oprávnění v systému. Pro uživatele je dostupná změna hesla, přehled jejich anotací a publikací a možnost změny uživatelských nastavení. Administrátor dále může schvalovat zadané publikace a spravovat autory, časopisy, vydavatele, formáty citací kategorie a další hodnoty.

2.2 Technický popis stávající aplikace

Stávající aplikace pro evidenci konferencí a publikací je webovou aplikací.

2.2.1 Aplikace na straně serveru

Aplikace je vytvořena ve frameworku Nette ve verzi 2.2. Uvedená verze frameworku je v době psaní práce již zastaralá. Aplikace požaduje PHP ve verzi 5.3.7 nebo novější (v rámci řady 5). Na serveru je aplikace provozována na PHP ve verzi 5.6.32.

2.2.2 Databáze

Aplikace používá pro uložení dat (s výjimkou souborů, které jsou uloženy na souborovém systému) relační databázi MySQL. Při tvorbě nové aplikace bych zvolil robustnější řešení, například PostgreSQL, pro aplikaci je však MySQL zcela dostačující a není důvod databázi měnit.

Pro rozpoznávání referencí bude třeba fulltextové vyhledávání. MySQL podporuje FULLTEXT indexy pro tabulky s engine MyISAM a od verze 5.6 i pro tabulky InnoDB.

2.2.3 Aplikace na straně klienta

Pro vzhled využívá aplikace front-end framework Twitter Bootstrap. Dále jsou na straně klienta použity JS knihovny jQuery, jQuery UI, jqxWidgets a další. Použitá knihovna jQuery je ve verzi 1.11.1. Tato verze již dávno není podporována. Zastaralé jsou i další knihovny. Jiné knihovny jsou sice do kódu vloženy, není však zřejmé, zda se skutečně využívají.

2.3 Bezpečnost stávající aplikace

Při zkoumání aplikace jsem narazil na několik věcí, které nepovažuji z hlediska zabezpečení za ideální. Tyto záležitosti popisuji níže. Zároveň však je třeba uvést, že se v žádném případě nejedná o vyčerpávající zkoumání zabezpečení aplikace. Tato problematika je velmi rozsáhlá a pravděpodobně by vydala na samostatnou diplomovou práci. Pro ověření bezpečnosti aplikace je možné použít OWASP Application Security Verification Standard Project [5].

2.3.1 Spuštění vlastního kódu na serveru

Aplikace umožňuje zadavateli publikací definovat formáty citací. Kromě toho, že způsob zadání formátů citací není příliš uživatelsky přívětivý (upravuje se přímo kód šablony latte), obsahuje v sobě také bezpečnostní hrozbu. Do latte šablony je totiž možné vložit libovolný PHP kód. Samozřejmostí je zároveň možnost spouštět jakékoliv dotazy v databázi. Protože funkce exec není v instalaci PHP zakázaná, je zároveň možné získat shellový přístup k serveru pod uživatelem webového serveru. Není třeba dodávat, že pod stejným uživatelem běží i další webové aplikace a že k plnému ovládnutí serveru stačí, aby se objevila vhodná zranitelnost v jádře operačního systému.

2.3.2 Uložení hesel

Pro ukládání hesel používá aplikace hashovací funkci SHA-512 s 5000 opakováními. Přestože hashovací funkce SHA-512 nebyla prolomena, pro účely ukládání hesel není vhodná z důvodu příliš velké rychlosti výpočtu, zejména na grafických kartách, a to i při použití 5000 opakování. Bylo by vhodnější použít funkce přímo navržené k účelu hashování hesel, například bcrypt, pbkdf2 nebo argon2. Nicméně většina uživatelů se do aplikace přihlašuje přes SSO, takže toto nepředstavuje zásadní problém.

2.3.3 Zranitelnosti typu XSS

Zranitelnosti typu XSS jsou typem útoku, kdy uživatel vloží do webové stránky škodlivý kód. Protože aplikace (s výjimkou přihlašovacího formuláře) není veřejně dostupná z internetu, neměl by tento typ zranitelnosti být problematický. Aplikace slouží pouze přihlášeným uživatelům, kterým můžeme do značné

míry důvěřovat. Nicméně teoreticky by bylo možné tento typ útoku zneužít ze strany běžného uživatele aplikace proti administrátorovi aplikace. Základní ochranu proti XSS v podobě Context-Aware Escaping poskytuje framework Nette, resp. jeho šablonovací aplikace Latte [6]. Je důležité, aby programátor tuto ochranu nevypínal, pokud si není jistý bezpečností takového postupu v daném konkrétním případě.

2.3.4 Zranitelnosti typu CSRF

Aplikace může obsahovat zranitelnosti typu CSRF. To by teoreticky útočníkovi umožňovalo například smazat publikaci prostřednictvím přihlášené oběti pomocí zaslání odkazu nebo jeho otevření v rámci zcela jiného webu. V minulosti se proti této zranitelnosti používaly jednorázové tokeny, ale dnes tomu lze bránit nastavením HTTP hlaviček.

2.4 Požadavky na úpravy aplikace

2.4.1 Funkční požadavky

Ze zadání vyplývají následující funkční požadavky:

- Evidence referencí a citací publikace
- Automatické rozpoznání referencí publikace
- Správa vlastních publikací
- Evidence štítků u publikací
- Rozšířené vyhledávání publikací

Další funkční požadavkek vyplynul z konzultací:

- Přiřazení více ISBN a ISSN publikaci, ročníku konference a časopisu

Evidence referencí publikace znamená, že u každé publikace bude dostupný seznam publikací, které tato publikace cituje. Tento seznam se u publikace neobjeví automaticky, ale je jej možné zadat dvěma způsoby:

- výběrem z dostupných publikací, kdy uživatel vybere ze seznamu citované publikaci,
- zadáním textového řádku reference, kdy uživatel zadá text reference (typicky zkopírovaný z odpovídající části publikace).

Textové řádky reference se budou zadávat ve formě seznamu. Typicky uživatel zkopíruje všechny reference z dané publikace (každou na samostatný řádek).

Aplikace se bude snažit zadané textové reference rozpoznávat. Bude tak činit dvěma způsoby.

První způsob spočívá v tom, že se aplikace při zadávání textových referencí pomocí parseru pokusí z textového řádku získat název publikace. Pokud se tento název shoduje s některou z publikací v databázi, je velmi pravděpodobné, že se jedná o identickou publikaci. Proto bude aplikace takto nalezené publikace nabízet administrátorovi ke schválení.

Druhý způsob vychází z MySQL FULLTEXT indexu. Aplikace umožní uživateli u textové reference provést výběr publikace ze seznamu tak, že nabízené publikace budou seřazeny podle shody na základě FULLTEXT indexu. Pokud se citovaná publikace v databázi nachází, je velice pravděpodobné, že bude v seznamu na prvním místě. V případě, že publikace v databázi není, bude aplikaci přesto nabízet publikaci s „nejbližším“ podobným názvem. Z tohoto důvodu nebude aplikace takto rozpoznané publikace nabízet administrátorovi ke schválení, ale nechá pouze na zadavateli publikací, zda je bude chtít zkontrolovat jednotlivě.

Evidence citací je jen opačnou stranou stejné relace. U citované publikace se zobrazí seznam publikací, které ji citují. Na této straně relace není možné citace upravovat, jen mazat.

Správa vlastních publikací znamená především spojení mezi autorem publikací a uživatelem. Uživatel pak může zobrazovat vlastní publikace.

Evidence štítků u publikací znamená možnost vytvářet uživatelské štítky a tyto přiřazovat k publikacím. Podobně jako anotace mohou být tyto štítky veřejné nebo privátní.

Rozšířené vyhledávání zohlední nově přidané možnosti pro vyhledávání publikací. Zároveň také umožní vyhledávání podle kategorií, které je v současné době poněkud problematické.

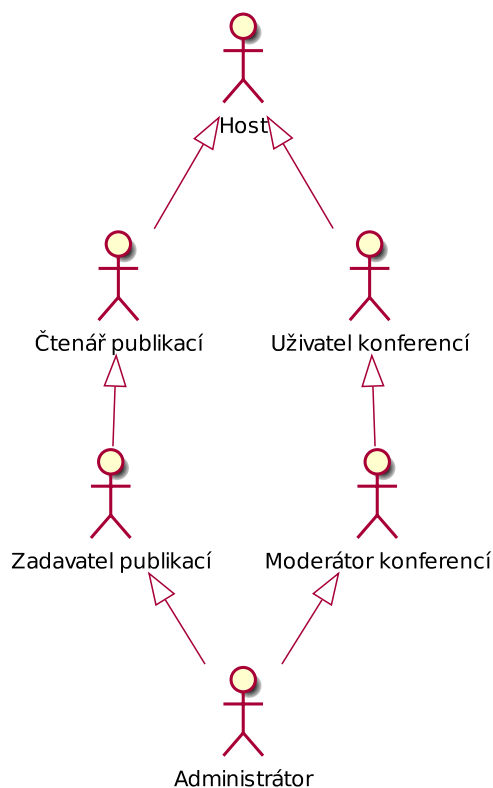
Přiřazení více ISBN a ISSN k publikaci, ročníku konference a časopisu umožní mít u každé publikace libovolný počet ISBN/ISSN.

2.4.2 Nefunkční požadavky

Uvádím všechny nefunkční požadavky, které má aplikace splňovat.

- Webová aplikace
- Uživatelská přívětivost
- Bezpečnost

Aplikace byla od počátku vyvíjena jako webová aplikace a není možné ani vhodné na tom něco měnit. Výhodou je možnost použití na různých zařízeních



Obrázek 2.1: Role v systému

a operačních systémech bez nutnosti něco instalovat na klientské počítače. Další výhodou je snadná správa, kdy případné aktualizace je možné provádět na serveru, tedy na jednom místě.

Uživatelská přívětivost je dána použitým front-end frameworkem, javascriptovými knihovnami a rozmístěním jednotlivých prvků. Na základě konkrétních připomínek uživatelů provádím změny směřující k jejímu vylepšení.

Přestože otázka bezpečnosti není hlavním předmětem této práce, podařilo se mi najít několik míst, kde bylo možné provést úpravy směřující k lepší bezpečnosti aplikace.

2.5 Role v systému

Role vycházejí z původního systému. Není třeba vytvářet nové role ani rušit některou z rolí. Obrázek 2.1 ukazuje role v systému.

2.5.1 Host

Výchozí role. Může se přihlásit do systému, čímž se stane přihlášeným uživatelem s jednou nebo více dalšími rolemi. Dále může požádat o obnovu zapomenutého hesla.

2.5.2 Čtenář publikací

Čtenář publikací může vyhledávat, filtrovat a zobrazovat publikace a pracovat s vlastním uživatelským profilem.

2.5.3 Zadavatel publikací

Zadavatel publikací může provádět vše, co Čtenář publikací. Navíc může spravovat autory publikací, kategorie publikací, formáty citací, skupiny, časopisy a především publikace.

2.5.4 Uživatel konferencí

Uživatel konferencí může zobrazovat konference a svůj uživatelský profil.

2.5.5 Moderátor konferencí

Moderátor konferencí může provádět to, co Uživatel konferencí. Navíc může spravovat kategorie konferencí a ACM kategorie, skupiny, konference a seznam veřejných databází publikací.

2.5.6 Administrátor

Administrátor může vše, co Zadavatel publikací a Moderátor konferencí. Navíc může spravovat uživatele a nastavení systému.

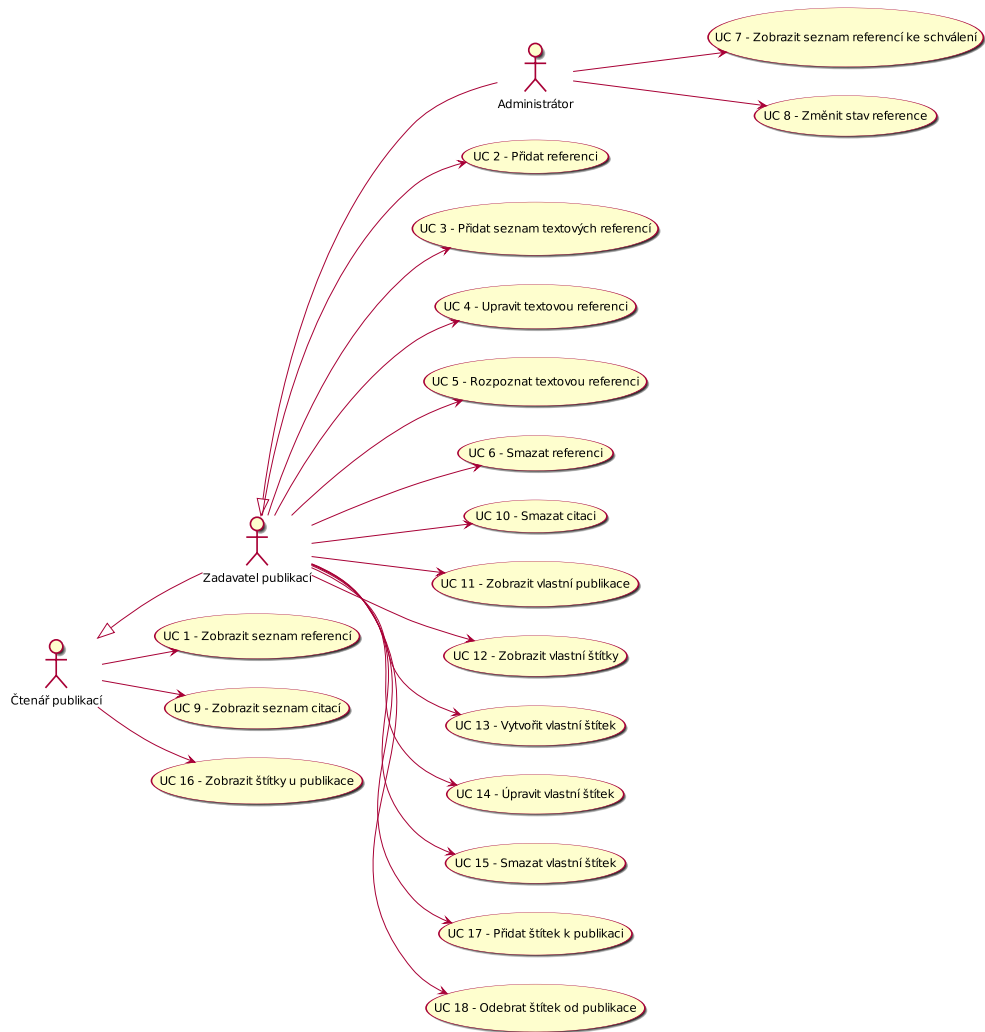
2.6 Případy užití

Níže uvádím případy užití, které jsou v rámci této práce nové nebo dochází k jejich zásadní úpravě. Některé úpravy v rámci práce vyžadují jen drobnou změnu stávajících případů užití, a proto je zde nerozepisují. Jedná se zejména o tyto úpravy:

- odstranění drobných chyb, vylepšení uživatelské přívětivosti,
- zavedení více ISBN a ISSN u ročníků konferencí, časopisů a publikací,
- vylepšení vyhledávání publikací (podle kombinace různých kritérií, vyhledávání podle poznámek, tagů).

Obrázek 2.2 ukazuje případy užití, které budu přidávat do aplikace nebo je významněji upravovat.

2. ANALÝZA A NÁVRH



Obrázek 2.2: Případy užití

2.6.1 UC 1 - Zobrazit seznam referencí

Aktéři Čtenář publikací, systém

Počáteční podmínky Čtenář publikací se nachází v detailu publikace

Hlavní scénář

1. Případ užití začíná, když Čtenář publikací chce zobrazit reference.
2. Čtenář publikací zvolí funkci „zobrazit seznam referencí“.
3. Systém zobrazí seznam referencí publikace.

2.6.2 UC 2 - Přidat referenci

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu referencí u publikace

Hlavní scénář

1. Případ užití začíná, když Zadavatel publikací chce přidat referenci.
2. Zadavatel publikací zvolí funkci „přidat referenci“.
3. Systém nabídne seznam publikací.
4. Zadavatel publikací vybere citovanou publikaci a zvolí „uložit“.
5. Systém uloží referenci k publikaci.

Alternativní scénář

4. Zadavatel publikací zvolí možnost „zrušit“.
5. Systém změny neuloží.

2.6.3 UC 3 - Přidat seznam textových referencí

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu referencí u publikace

Hlavní scénář

1. Případ užití začíná, když Zadavatel publikací chce přidat textové reference k publikaci.

2. Zadavatel publikací zvolí funkci „přidat seznam referencí“.
3. Systém nabídne prostor pro zadání referencí.
4. Zadavatel publikací zadá seznam referencí v textové podobě, každou referencí na jeden řádek a zvolí „uložit“.
5. Systém uloží seznam referencí k publikaci.

Alternativní scénář

4. Zadavatel publikací zvolí možnost „zrušit“.
5. Systém změny neuloží.

2.6.4 UC 4 - Úpravit textovou referenci

Akteři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu referencí u publikace

Hlavní scénář

1. Příklad užití začíná, když Zadavatel publikací chce upravit textovou referenci.
2. Zadavatel publikací zvolí u reference, kterou chce upravit, funkci „upravit referenci“. Tato funkce je dostupná jen u textových referencí, ne u referencí zadaných formou výběru z dostupných publikací.
3. Systém nabídne text reference k úpravě.
4. Zadavatel publikací upraví referenci a vybere volbu „uložit“.
5. Systém uloží změny reference.

Alternativní scénář

4. Zadavatel publikace zvolí možnost „zrušit“.
5. Systém změny neuloží.

2.6.5 UC 5 - Rozpoznat textovou referenci

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu referencí u publikace

Hlavní scénář

1. Případ užití začíná, když Zadavatel publikací chce rozpoznat textovou referenci.
2. Zadavatel publikací zvolí u reference, kterou chce rozpoznat, volbu „rozpoznat referenci“. Tato funkce je dostupná u textově zadaných referencí, ne u referencí zadaných formou výběru.
3. Systém nabídne seznam publikací v pořadí podle pravděpodobnosti (od nejvyšší k nejnižší), že odpovídají dané textové referenci.
4. Zadavatel publikací vybere publikaci a zvolí volbu „uložit“.
5. Systém uloží referenci. Tím se z textové reference stane reference zadaná formou výběru.

Alternativní scénář

4. Zadavatel publikací nevybere žádnou z nabízených publikací a zvolí „zrušit“.
5. Systém změny neuloží.

2.6.6 UC 6 - Smazat referenci

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu referencí u publikace

Hlavní scénář

1. Případ užití začíná, když Zadavatel publikací chce smazat referenci.
2. Zadavatel publikací zvolí u reference, kterou chce smazat, funkci „smazat“. Tato funkce je dostupná jak u textově zadaných referencí, tak u referencí zadaných formou výběru.
3. Systém se zeptá na potvrzení, zda má skutečně provést smazání reference.

2. ANALÝZA A NÁVRH

4. Zadavatel publikací potvrdí smazání reference volbou „potvrdit“.
5. Systém smaže referenci.

Alternativní scénář

4. Zadavatel publikací zvolí „zrušit“.
5. Systém změny neuloží.

2.6.7 UC 7 - Zobrazit seznam referencí ke schválení

Aktéři Administrátor, systém

Počáteční podmínky Administrátor je přihlášený do systému

Hlavní scénář

1. Případ užití začíná, když Administrátor chce schválit rozpoznané reference.
2. Administrátor zvolí funkci „zobrazit seznam referencí ke schválení“.
3. Systém zobrazí seznam referencí ke schválení.

2.6.8 UC 8 - Změnit stav reference

Aktéři Administrátor, systém

Počáteční podmínky Administrátor se nachází v seznamu referencí ke schválení

Hlavní scénář

1. Případ užití začíná, když Administrátor chce změnit stav reference v seznamu referencí ke schválení.
2. Administrátor zvolí funkci „schválit referenci“ u reference, kterou chce schválit.
3. Systém nastaví referenci jako schválenou a vyřadí ji ze seznamu referencí ke schválení.

Alternativní scénář

2. Administrátor zvolí funkci „zrušit referenci“ u reference, kterou chce zrušit.
3. Systém vyřadí referenci ze seznamu referencí ke schválení.

2.6.9 UC 9 - Zobrazit seznam citací

Aktéři Čtenář publikací, systém

Počáteční podmínky Čtenář publikací se nachází v detailu publikace

Hlavní scénář

1. Případ užití začíná, když Čtenář publikací chce zobrazit citace.
2. Čtenář publikací zvolí funkci „zobrazit seznam citací“.
3. Systém zobrazí seznam referencí publikace.

2.6.10 UC 10 - Smazat citaci

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu citací u publikace

Hlavní scénář

1. Případ užití začíná, když Zadavatel publikací chce smazat citaci.
2. Zadavatel publikací zvolí u citace, kterou chce smazat, funkci „smazat“.
3. Systém se zeptá na potvrzení, zda má skutečně provést smazání citace.
4. Zadavatel publikací potvrdí smazání citace volbou „potvrdit“.
5. Systém odstraní citaci. Při odstranění citace se zároveň odstraní příslušná reference.

Alternativní scénář

4. Zadavatel publikací zvolí „zrušit“.
5. Systém změny neuloží.

2.6.11 UC 11 - Zobrazit vlastní publikace

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu publikací

Hlavní scénář

1. Případ užití začíná, když Zadavatel publikací chce zobrazit vlastní publikace.

2. Zadavatel publikací zvolí filtr „zobrazit vlastní publikace“.
3. Systém zobrazí stránkovaný seznam publikací daného uživatele. Uživatel může publikace dále filtrovat podle dalších kritérií.

2.6.12 UC 12 - Zobrazit vlastní štítky

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací je přihlášen do systému

Hlavní scénář

1. Příklad užití začíná, když Zadavatel publikací chce zobrazit seznam vlastních štítků.
2. Zadavatel publikací zvolí funkci „zobrazit seznam vlastních štítků“.
3. Systém zobrazí štítky přihlášeného zadavatele publikací.

2.6.13 UC 13 - Vytvořit vlastní štítek

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu vlastních štítků

Hlavní scénář

1. Příklad užití začíná, když Zadavatel publikací chce vytvořit štítek.
2. Zadavatel publikací zvolí funkci „vytvořit štítek“.
3. Systém nabídne možnost zadání názvu štítku a volbu, zda je privátní nebo globální.
4. Zadavatel publikací vyplní název štítku, jeho viditelnost a zvolí „uložit“.
5. Systém uloží štítek.

Alternativní scénář

4. Zadavatel publikací zvolí možnost „zrušit“.
5. Systém změny neuloží.

2.6.14 UC 14 - Úpravit vlastní štítek

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu vlastních štítků

Hlavní scénář

1. Případ užití začíná, když Zadavatel publikací chce upravit štítek.
2. Zadavatel publikací zvolí funkci „upravit štítek“ u konkrétního štítku.
3. Systém nabídne možnost změny názvu štítku a volby, zda je privátní nebo globální.
4. Zadavatel publikací vyplní název štítku, jeho viditelnost a zvolí „uložit“.
5. Systém uloží štítek.

Alternativní scénář

4. Zadavatel publikací zvolí možnost „zrušit“.
5. Systém změny neuloží.

2.6.15 UC 15 - Smazat vlastní štítek

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu vlastních štítků

Hlavní scénář

1. Případ užití začíná, když Zadavatel publikací chce smazat štítek.
2. Zadavatel publikací zvolí funkci „smazat štítek“ u konkrétního štítku.
3. Systém se zeptá na potvrzení smazání štítku.
4. Zadavatel publikací potvrdí smazání štítku volbou „potvrdit“.
5. Systém vymaže štítek.

Alternativní scénář

4. Zadavatel publikací zvolí možnost „zrušit“.
5. Systém neprovede vymazání štítku.

2.6.16 UC 16 - Zobrazit štítky u publikace

Aktéři Čtenář publikací, systém

Počáteční podmínky Čtenář publikací se nachází v detailu publikace

Hlavní scénář

1. Příklad užití začíná, když Čtenář publikací chce zobrazit seznam štítků u publikace.
2. Čtenář publikací zvolí funkci „zobrazit štítky“.
3. Systém zobrazí všechny globální štítky k dané publikaci a privátní štítky aktuálního uživatele k dané publikaci.

2.6.17 UC 17 - Přidat štítek k publikaci

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu štítků u publikace

Hlavní scénář

1. Příklad užití začíná, když Zadavatel publikací chce přidat štítek k publikaci.
2. Zadavatel publikací zvolí funkci „přidat štítek“.
3. Systém nabídne seznam štítků uživatele.
4. Zadavatel publikací vybere štítek a zvolí „uložit“.
5. Systém uloží štítek k publikaci.

Alternativní scénář

4. Zadavatel publikací zvolí možnost „zrušit“.
5. Systém změny neuloží.

2.6.18 UC 18 - Odebrat štítek od publikace

Aktéři Zadavatel publikací, systém

Počáteční podmínky Zadavatel publikací se nachází v seznamu štítků u publikace

Hlavní scénář

1. Případ užití začíná, když Zadavatel publikací chce odebrat štítek od publikace.
2. Zadavatel publikací zvolí funkci „odebrat štítek“ u konkrétního štítku.
3. Systém se zeptá na potvrzení odebrání štítku.
4. Zadavatel publikací potvrdí odebrání štítku volbou „potvrdit“.
5. Systém odebere štítek od publikace.

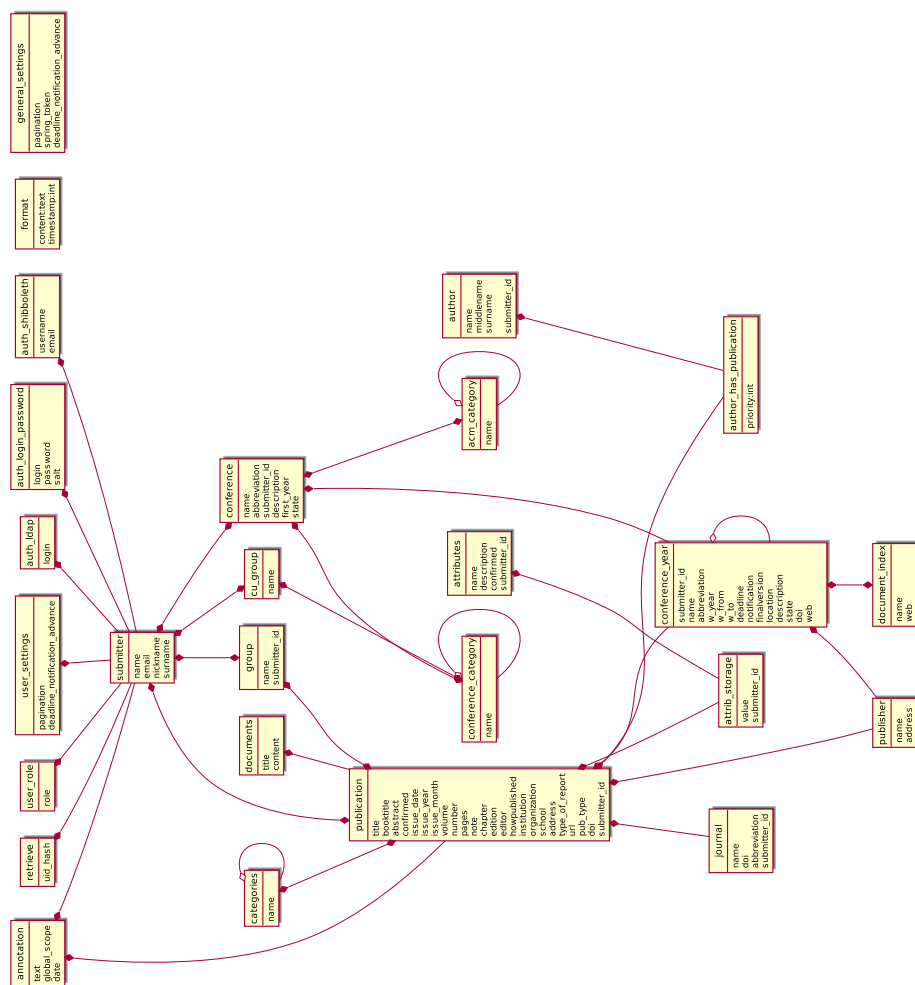
Alternativní scénář

4. Zadavatel publikací zvolí možnost „zrušit“.
5. Systém neprovede odebrání štítku od publikace.

2.7 Stručný popis datového modelu původní aplikace

Obrázek 2.3 ukazuje stav datového modelu na počátku vývoje. Aby nebyly rozměry obrázku neúnosně velké, obrázek neobsahuje ty vazební entity, jejichž účelem je výhradně vazba mezi objekty a samy o sobě nenesou žádnou hodnotu. Dále v obrázku kvůli přehlednosti nejsou zobrazeny ty vazby na entitu `submitter`, které nesouvisí s business logikou aplikace a jsou tam jen z důvodu auditování. V takovém případě je v dané tabulce uveden atribut `submitter_id`.

2. ANALÝZA A NÁVRH



Obrázek 2.3: UML Class diagram: původní stav

2.7.1 Tabulka `acm_category`

Tabulka obsahuje ACM kategorie [3]. Tyto kategorie mají stromovou strukturu, tabulka tedy kromě názvu obsahuje cizí klíč na nadřazenou kategorii.

2.7.2 Tabulka `annotation`

Tabulka obsahuje uživatelské poznámky k publikacím. Poznámka patří konkrétnímu uživateli a je veřejná nebo soukromá.

2.7.3 Tabulka `attributes`

Tabulka obsahuje definici atributů k publikacím.

2.7.4 Tabulka `attrib_storage`

Tabulka slouží k ukládání hodnot atributů publikací. Jedná se vlastně o vazební tabulku mezi tabulkami `attributes` a `attrib_storage`, ale navíc obsahuje hodnotu atributu.

2.7.5 Tabulka `author`

Tabulka obsahuje autory publikací. Autor má jméno, prostřední jméno a příjmení. Tabulka dále obsahuje cizí klíč na uživatele, který záznam vytvořil.

2.7.6 Tabulka `author_has_publication`

Vazební tabulka mezi autory a publikacemi.

2.7.7 Tabulka `auth_ldap`

Tabulka slouží k autentizaci uživatelů přes LDAP. Kromě cizího klíče na uživatele obsahuje uživatelské jméno v rámci LDAP. Protože se místo autentizace přes LDAP využívá Shibboleth, je tabulka prázdná.

2.7.8 Tabulka `auth_login_password`

Tabulka slouží k autentizaci uživatelů přes uživatelské jméno a heslo. Kromě cizího klíče na uživatele obsahuje uživatelské jméno, hash hesla a salt pro zvýšení bezpečnosti hesla.

2.7.9 Tabulka `auth_shibboleth`

Tabulka slouží k autentizaci uživatelů přes Shibboleth. Kromě cizího klíče na uživatele obsahuje uživatelské jméno a e-mail uživatele.

2.7.10 Tabulka `categories`

Tabulka obsahuje kategorie publikací. Kategorie mohou tvořit stromovou strukturu, proto jeden ze sloupců tabulky obsahuje cizí klíč na tabulku samotnou (na nadřazenou kategorii). Dále tabulka obsahuje název kategorie a cizí klíč na uživatele, který kategorii vytvořil.

2.7.11 Tabulka `categories_has_publication`

Vazební tabulka mezi kategoriemi publikací a publikacemi.

2.7.12 Tabulka `conference`

Tabulka konferencí obsahuje název konference, zkratku, cizí klíč na uživatele, který konferenci vytvořil, popis, první rok a stav konference.

2.7.13 Tabulka `conference_category`

Tabulka kategorií konferencí. Protože kategorie konferencí mohou tvořit stromovou strukturu, obsahuje tabulka kromě názvu také cizí klíč na tabulku samotnou (na nadřazenou kategorii).

2.7.14 Tabulka `conference_has_acm_category`

Vazební tabulka mezi konferencemi a ACM kategoriemi.

2.7.15 Tabulka `conference_has_category`

Vazební tabulka mezi konferencemi a kategoriemi konferencí.

2.7.16 Tabulka `conference_year`

Tabulka ročníků konferencí. Kromě cizího klíče na konferenci, na uživatele, který ročník konference vytvořil, a na tabulku samotnou (nadřazený ročník konference) obsahuje tabulka ještě název, zkratku, rok, datum začátku, datum konce, datum deadline, datum notifikace, datum finální verze, místo, popis, stav, ISBN, ISSN, DOI a odkaz na web.

2.7.17 Tabulka `conference_year_is_indexed`

Vazební tabulka mezi ročníkem konference a informací o indexaci sborníků konferencí ve veřejných databázích publikací.

2.7.18 Tabulka `cu_group`

Tabulka pro skupiny kategorií konferencí. Obsahuje název kategorie.

2.7.19 Tabulka `cu_group_has_conference_category`

Vazební tabulka mezi skupinami kategorií konferencí a kategoriemi konferencí.

2.7.20 Tabulka `documents`

Tabulka slouží pro dokumenty nahrané k publikacím. Kromě cizího klíče na publikaci obsahuje název dokumentu a obsah dokumentu, pokud se jej podařilo z dokumentu načíst.

2.7.21 Tabulka `document_index`

Tabulka obsahuje veřejné databáze publikací (název a odkaz na web).

2.7.22 Tabulka `format`

Tabulka obsahuje formáty citací pro export z detailu publikace. Kromě názvu se jedná o šablonu samotnou a timestamp.

2.7.23 Tabulka `general_settings`

Tabulka obsahuje obecná nastavení aplikace. Jedná se o počet položek na stránce při stránkování, `spring_token` a počet dnů pro notifikaci s předstihem.

2.7.24 Tabulka `group`

Tabulka slouží pro skupiny publikací. Kromě názvu skupiny obsahuje cizí klíč na uživatele, který skupinu vytvořil.

2.7.25 Tabulka `group_has_publication`

Vazební tabulka mezi skupinami publikací a publikacemi.

2.7.26 Tabulka `journal`

Tabulka časopisů. Obsahuje název časopisu, ISSN, DOI, zkratku a cizí klíč na uživatele, který časopis přidal.

2.7.27 Tabulka `publication`

Tabulka publikací. Obsahuje cizí klíče na časopis, vydavatele, ročník konference a uživatele, který publikaci vytvořil. Dále řadu textových položek, příznak, zda je publikace schválena administrátorem, a datum vydání.

2.7.28 Tabulka `publisher`

Tabulka vydavatelů. Obsahuje název a adresu vydavatele a cizí klíč na uživatele, který záznam vytvořil.

2.7.29 Tabulka `retrieve`

Tabulka slouží k možnosti resetování hesla. Obsahuje cizí klíč na uživatele a hash pro potvrzení resetování hesla.

2.7.30 Tabulka `submitter`

Tabulka obsahuje uživatele aplikace. Poněkud neobvyklý název je dán historicky. Obsahuje jméno, příjmení, uživatelské jméno a e-mail.

2.7.31 Tabulka `submitter_favourite_conference`

Vazební tabulka mezi uživateli a konferencemi slouží k zaznamenání oblíbených konferencí uživatele (označení hvězdičkou).

2.7.32 Tabulka `submitter_has_cu_group`

Vazební tabulka mezi uživateli a skupinami kategorie konferencí.

2.7.33 Tabulka `submitter_has_group`

Vazební tabulka mezi uživateli a skupinami publikací.

2.7.34 Tabulka `submitter_has_publication`

Vazební tabulka mezi uživateli a publikacemi slouží k zaznamenání oblíbených publikací uživatele (označení hvězdičkou).

2.7.35 Tabulka `user_role`

Tabulka obsahuje přiřazení uživatelů k rolím. Kromě cizího klíče na uživatele obsahuje název role, která je uživateli přiřazena.

2.7.36 Tabulka `user_settings`

Tabulka obsahuje uživatelská nastavení jednotlivých uživatelů. Kromě cizího klíče na uživatele obsahuje počet položek na stránce při stránkování a počet dnů pro notifikaci s předstihem.

2.8 Návrh změn datového modelu

Data z původního datového modelu musí zůstat zachována. Je však nutné přidat některé nové entitní typy a přidat atributy do jiných entitních typů. V souvislosti s tím bude třeba provést transformaci dat. Další entitní typy a atributy je možné odstranit.

Obrázek 2.4 ukazuje tabulky, které jsem nově vytvářel, prováděl v nich změny nebo které s takovými tabulkami úzce souvisí. Níže popisují změny podrobněji.

2.8.1 Změny v tabulce `publication`

Dosud byl rok a měsíc u publikace v tabulce uložen jako datový typ `datetime`. Přestože řadě případů je tento formát pro uložení data vhodný, v tomto případě se ukázal opak. U některých publikací je totiž vyplněn pouze rok, u jiných rok a měsíc. Den se nevyplňuje. Proto se pro uložení roku a měsíce místo jednoho sloupce typu `datetime` použijí dva sloupce typu `integer`.

2.8.2 Změny v tabulce `author`

Z funkčních požadavků vyplývá, že je třeba propojit uživatele aplikace a autory publikací. Jednou z možností by bylo zcela spojit tabulky autorů a uživatelů. To se ukazuje jako problematické, protože počet autorů je řádově vyšší, než počet uživatelů. Zvolil jsem tedy řešení formou přidání sloupce do `user_id` do tabulky `author`. Tím je možné každému autorovi přiřadit uživatele. Je možné přiřadit jednoho uživatele i více autorů, pokud by byly v aplikaci duplicitní. V takovém případě je však vhodnější záznamy sloučit.

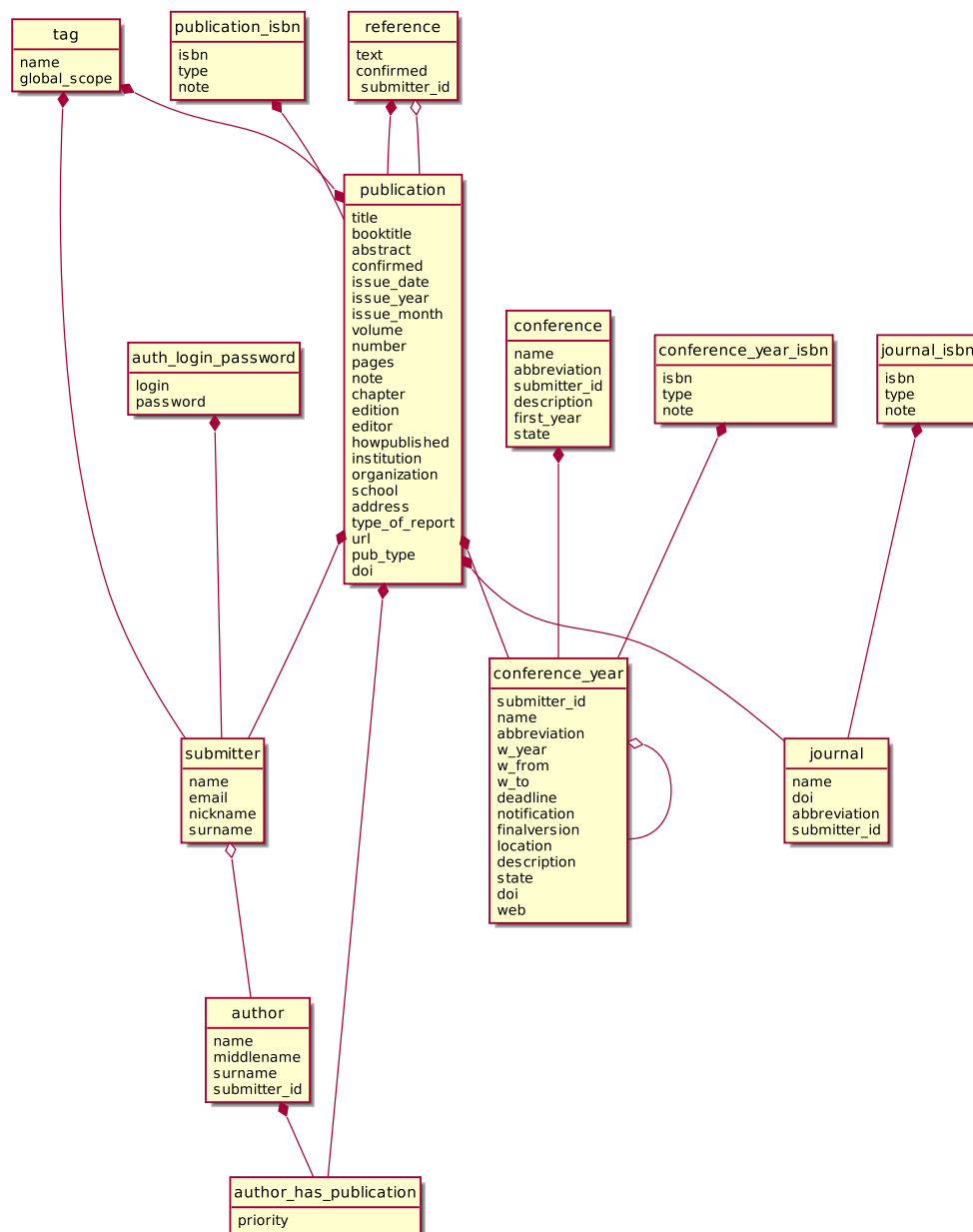
2.8.3 Vytvoření tabulek `conference_year_isbn`, `journal_isbn` a `publication_isbn`

Nově přidané tabulky `conference_year_isbn`, `journal_isbn` a `publication_isbn` umožní ke každému ročníku konference, časopisu a publikaci přiřadit několik ISBN/ISSN. Dosud bylo možné mít pouze jedno ISBN a jedno ISSN u ročníku konference, jedno ISSN u časopisu a jedno ISBN u publikace. Navíc je ke každému ISBN/ISSN možné přidat poznámku.

2.8.4 Vytvoření tabulky `tag`

Z funkčních požadavků vyplývá potřeba zaznamenávat štítky k publikacím. Nově vytvořená tabulka štítků bude obsahovat název štítku, vlastníka štítku a informaci o tom, zda se jedná o globální nebo privátní štítek.

2. ANALÝZA A NÁVRH



Obrázek 2.4: UML Class diagram: návrh změn

2.8.5 Vytvoření tabulky `publication_has_tag`

Nově přidaná vazební tabulka mezi publikacemi a štítky.

2.8.6 Odstranění tabulky `format`

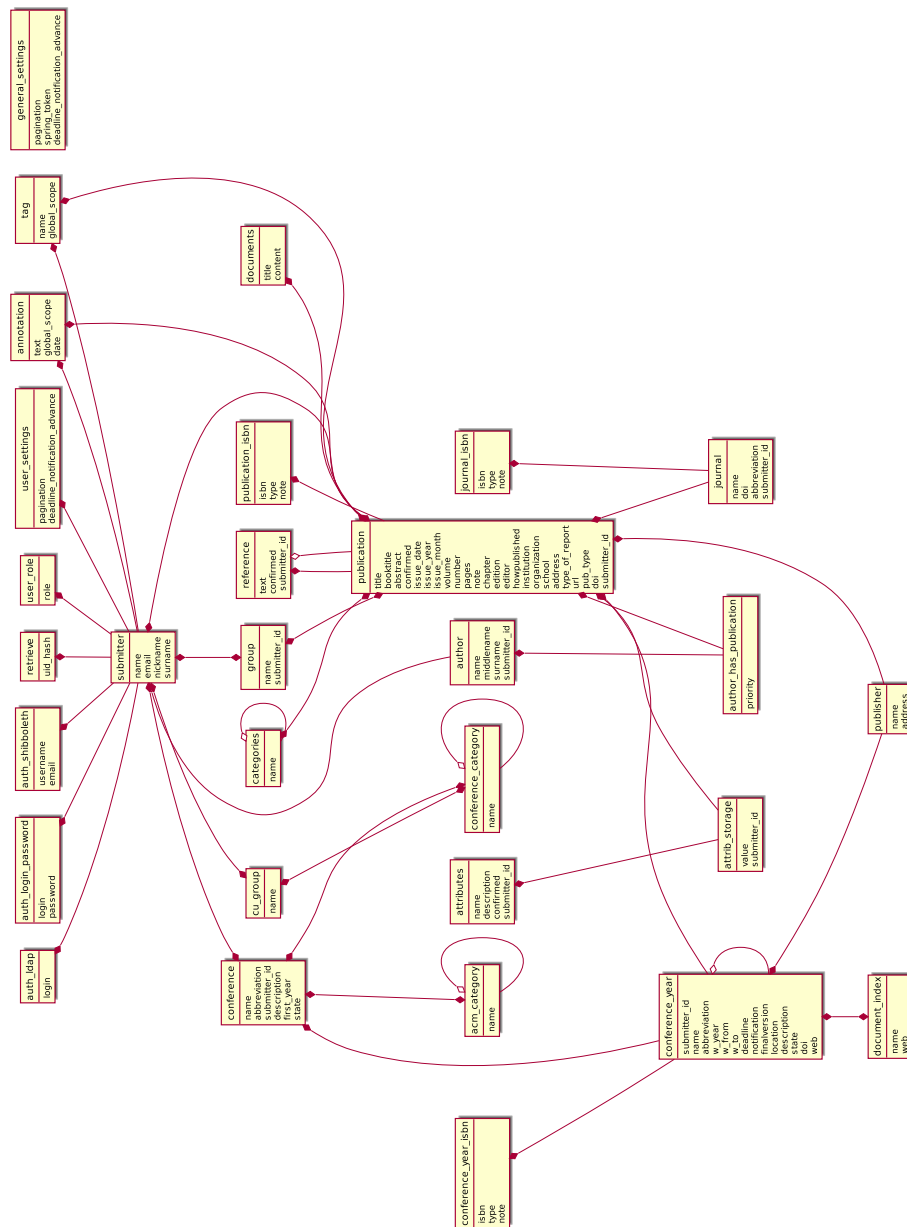
Tabulka `format` obsahovala formáty citací. Vytvoření tabulky bylo součástí zadání jedné z předchozích prací, ale v této práci jsem ji odstranil. Detailům se věnuji v další kapitole.

2.8.7 Přidání tabulky `reference`

Nově přidaná tabulka slouží buď jako vazební tabulka mezi dvěma publikacemi (citující a citovanou), nebo jako textová podoba citace k dané publikaci. V takovém případě obsahuje ještě rozpoznatý titulek reference, který umožní snazší párování, pokud publikace odkazovaná v textové referenci již v databázi existuje. Vedle toho se v tabulce bude ukládat id uživatele, který referenci zadal (`submitter_id`).

Obrázek 2.5 ukazuje konečný stav databáze. Aby nebyly rozměry obrázku neúnosně velké, obrázek neobsahuje ty vazební tabulky, jejichž účelem je výhradně vazba mezi objekty a samy o sobě nenesou žádnou hodnotu. Dále v obrázku kvůli přehlednosti nejsou zobrazeny ty vazby na tabulku, které nesouvisí s business logikou aplikace a jsou tam jen z důvodu auditování. V takovém případě je v dané tabulce uveden atribut `submitter_id`.

2. ANALÝZA A NÁVRH



Obrázek 2.5: UML Class diagram: finální stav

Realizace

3.1 Implementace nových funkcí

Nové funkce jsem implementoval iterativně. Po každé iteraci následovalo lokální otestování funkčnosti. Po dokončení většího celku jsem provedl nahrání tohoto celku do repozitáře a na produkční server, kde ho mohl vyzkoušet vedoucí práce a další uživatelé.

3.2 Nové CRUD komponenty

CRUD operace jsou operace vytváření, editace, načítání a mazání záznamů. Tato zkratka představuje čtveřici slov: create, read, update, delete. [1] V rámci této práce několik CRUD komponent upravuji a přidávám 4 nové:

CitationCrud slouží k práci s referencí na citované straně. Zobrazuje seznam citací a umožňuje je po potvrzení smazat. Neumožňuje však reference přidávat nebo jakkoliv upravovat.

ReferenceCrud slouží pro práci s referencí na citující straně. Zobrazuje seznam referencí, umožňuje je přidávat formou výběru nebo formou textových referencí, upravovat i mazat.

TagCrud slouží ke správě uživatelských štítků. Umožňuje jejich vytváření, editaci i mazání.

PublicationTagCrud slouží ke správě štítků u publikací. Umožňuje zobrazit seznam štítků u publikace a tyto štítky přidávat i odebrat.

3.3 Rozpoznávání referencí

3.3.1 Parsování referencí

Při zadávání referencí formou textu je nutné zadat každou referenci na právě jeden řádek. V opačném případě by bylo velmi obtížné oddělit jednotlivé reference od sebe, a proto by nebylo možné je rozpoznat.

K parsování referencí slouží třída `ReferenceParser`. Konstruktor třídy přijímá text reference. Samotné parsování se provádí metodou `parse()`. Po úspěšném parsování je třída metodou `getAuthors()` schopna vrátit pole autoru, metodou `getTitle()` titulku publikace a metodou `getYear()` rok publikace.

Při implementaci parsování referencí jsem postupoval jakousi obdobou test-driven development. Vybral jsem si řetězec, stanovil, jaké mají být výstupy, a postupně upravoval třídu `ReferenceParser` tak, abych dostal požadovaný výsledek. Po přidání několik různých řetězců již úspěšné parsování dalšího řetězce nevyžadovalo úpravu algoritmu.

Při implementaci parsování jsem předpokládal, že budu rozpoznávat citace podle více kritérií. V praxi se ukázalo, že pro úspěšné rozpoznání citované publikace stačí znát její název.

3.3.2 Rozpoznávání referencí formou přesné shody

Podle názvu, který jsem si uložil do databáze při zadávání textových referencí na základě jejich parsování, mohu tyto reference spojit s publikacemi pomocí přesné shody názvu s názvem. Tato metoda má výhodu v tom, že prakticky nedochází k chybnému rozpoznání. Na druhou stranu je rozpoznána jen část publikací, které ve skutečnosti jsou v databázi. Tuto metodu označuji v dalším textu jako metodu A.

3.3.3 Rozpoznávání referencí přes FULLTEXT index

Při této metodě vyhledávám publikaci pomocí MySQL Fulltext v Boolean módu [7]. To provádím ve dvou variantách:

- vyhledávání celé textové reference v názvech publikací (dále označeno jako metoda B1),
- vyhledávání parsovaného titulku z reference v názvech publikací (dále označeno jako metoda B2).

3.3.4 Porovnávání metod rozpoznávání referencí

Cílem tohoto porovnání je zjistit, jaký způsob rozpoznávání referencí je nejvhodnější. V době provedení tohoto měření měla databáze 619 publikací, z toho 281 bylo vydáno po roce 2000, 199 před rokem 2000 a u 139 publikací nebyl

Tabulka 3.1: Úspěšnost rozpoznávání referencí

Metoda	A	B1	B2
Správně nalezeno	56	132	117
Správně nenalezeno	526	13	73
Chybně nalezeno	0	3	8
Falešně pozitivní	0	513	12
Falešně negativní	81	2	453

rok vydání vyplněn. Z publikací vydaných po roce 2000 jsem náhodně vybral 30 publikací. Publikace vydané pro roce 2000 jsem zvolil proto, že je u nich větší pravděpodobnost, že citované publikace budou v databázi.

U takto 30 vybraných publikací jsem se pokusil v databázi ručně dohledat všechny reference. Jednu z publikací nebylo možné použít, protože se jednalo o knihu, která měla stovky citací a nebylo tedy reálné všechny dohledat. Další z publikací nebylo možné použít proto, že v systému neměla vůbec nahrané pdf. Další z publikací měla pdf v podobě, která vůbec neumožňovala kopírování textu referencí. Tyto 3 publikace nebyly do zkoumání zahrnuty. U několika dalších publikací se vyskytly problémy při kopírování seznamu referencí, nicméně většinu se s drobnými úpravami podařilo zkopírovat, a do zkoumání jsem je tedy zařadil.

Výše uvedených 27 pulikací obsahuje celkem 668 řádků referencí. Zkopírovat se jich podařilo 663, zbývajících 5 bylo nečitelných. Z těchto 663 referencí databáze obsahuje 137 publikací.

Metoda A dokázala správně rozpoznat 56 referencí. 526 referencí nebylo v databázi nalezeno, přičemž se v databázi skutečně nenacházejí. Neobsahovala žádné falešně pozitivní ani chybné výsledky, ale 81 výsledků bylo falešně negativních.

Metoda B1 dokázala správně rozpoznat 132 referencí. 13 referencí nebylo v databázi nalezeno, přičemž se v databázi skutečně nenacházejí. 3 reference byly rozpoznány chybně, 513 bylo falešně pozitivních a 2 reference byly falešně negativní

Metoda B2 dokázala správně rozpoznat 117 referencí. 73 referencí nebylo v databázi nalezeno, přičemž se tam skutečně nenacházejí. 8 referencí bylo rozpoznáno chybně, 12 referencí bylo falešně negativních a 453 bylo falešně pozitivních.

Jako správně rozpoznané označuji ty rozpoznané reference, kde byla rozpoznána publikace shodná s ručně dohledanou publikací. Jako falešně negativní označuji ty, kde algoritmus nerozpoznal publikaci přesto, že se v systému nachází. Jako chybné označuji ty výsledky, kde algoritmus našel jinou publikaci, než jaká byla dohledána ručně. Jako falešně pozitivní označuji situaci, kdy systém našel publikaci přesto, že správná publikace se v databázi vůbec nenachází. Výsledky zkoumání obsahuje tabulka 3.1.

Tabulka 3.2: Shrnutí úspěšnosti rozpoznávání referencí

Metoda	A	B1	B2
Správně	88 %	22 %	29 %
Chybně	12 %	78 %	71 %

V souhrnné tabulce 3.2 jsou jako správné rozpoznání uvedeny případy, kdy:

- je publikace v databázi přítomna a byla správně rozpoznána,
- není publikace v databázi přítomna a metoda žádnou publikaci v databázi nenašla.

Všechny ostatní případy jsou uvedeny jako chybné.

Podle těchto kritérií je jednoznačně nejlepší z uvedených metod metoda A a tato metoda se tedy používá pro navrhování rozpoznávaných referencí administrátorovi. Přesto metody B1 a B2 zcela nezavrhuji. Lze je použít v případě, kdy je mnohem důležitější správné rozpoznání publikací, které se v databázi nacházejí, než zjištění faktu, že se publikace v databázi nenachází. Proto metodu B1 používám pro navrhování publikací uživateli.

3.4 Dependency Injection

Jednou z dobrých praxí ve frameworku Nette je využívání Dependency Injection [8]

3.4.1 Sjedení předávání závislostí

Jak uvádí manuál [9], v Nette Frameworku existují v zásadě 4 způsoby předávání závislostí v souladu s principem Dependency Injection:

Konstruktor je základní metodou předávání závislostí. Závislosti se předávají v okamžiku vytváření objektu. V Nette se správné parametry na základě hintů předají automaticky.

Setter znamená, že závislosti jsou předány po vytvoření objektu metodou `set*`. Metodu je ovšem nutné zavolat ručně.

Metoda `inject*` je speciálním případem setteru. Nette dokáže v případě presenterů předat parametry automaticky.

Anotace `@inject` znamená, že do public proměnné s touto anotací a typem závislosti v dokumentačním komentáři se proměnná předá automaticky. Funguje to však jen v presenterech.

Dalším způsobem je využití rozšíření Kdyby/Autowired [10]. Používá anotaci `@autowired`. Jedná se o obdobu anotace `@inject`, která ovšem funguje i mimo presentery a vyžaduje mít instalované rozšíření.

Nevýhodou anotací `@inject` a `@autowired` je, že proměnná, do které se závislost vkládá, musí být `public`. Tím může dojít k porušení zapouzdření. Výhodou je zásadní zvýšení čitelnosti kódu v presenterech.

V důsledku toho, že na aplikaci pracovalo postupně více autorů, využívají se snad s výjimkou setterů všechny výše uvedené metody. Mou snahou bylo dosáhnout v celé aplikaci využívání pouze těchto dvou metod:

- anotace `@inject` pro presentery,
- konstruktor pro všechny ostatní třídy.

3.4.2 Vytváření instancí tříd pomocí továrniček

Tato praxe již byla v aplikaci použita, pokouším se ji však rozšířit. Vytváření tříd pomocí továrniček je alternativou k vytváření instancí tříd pomocí `new` přímo v presenterech.

3.4.3 Odstranění přístupů ke kontextu z presenteru

V aplikaci existovala místa, kde se princip Dependency Injection nevyužíval vůbec. Jedná se například o přístup z presenteru přímo ke kontextu. V rámci úprav jsem tyto přístupy nahrazoval předáváním závislostí v pomoci Dependency Injection.

3.5 Zabezpečení

3.5.1 Odstranění šablon formátů

Jak ukázala první část analýzy v této práci, ukládání šablon do databáze znamená riziko v podobě možnosti spuštění libovolného kódu na serveru ze strany zadavatele publikací (či spíše uživatele, kterému by se oprávnění zadavatele publikací podařilo neoprávněně získat). Navíc uživatelská přívětivost editace formátů citace nebyla o mnoho vyšší než editace přímo ve zdrojovém kódu. Proto jsem tabulku po dohodě s vedoucím práce odstranil. Šablony je možné editovat přímo ve zdrojovém kódu v adresáři `app/templates/Publication/`.

3.5.2 Algoritmus pro hashování hesel

V aplikaci jsem nahradil dosavadní algoritmus hashování hesel za `bcrypt`. Přestože heslo pro přihlášení do aplikace používá minimum uživatelů a tato změna tedy není příliš významná, považuji to za správný krok jak s ohledem na bezpečnost, tak vzhledem k tomu, že `bcrypt` je de-facto standard v PHP [11].

3.5.3 Doplnění hlaviček pro ochranu proti CSRF

Přestože nemohu zasahovat do konfigurace serveru, doporučil bych správci serveru zavést Content Security Policy HTTP hlavičky [12], což uživatele pomůže ochránit proti CSRF. Je to mnohem efektivnější řešení, než zavádět nějakou formu ochrany pomocí tokenů do celé aplikace, což by bylo velmi časově náročné a lze snadno na ochranu některé funkce zapomenout.

3.6 Správa závislostí pomocí Composeru

Composer je nástroj pro správu závislostí v PHP [13]. V tomto projektu byl již použit, ale protože neměl ošetřené závislosti na úrovni rozšíření PHP, nebylo úplně snadné zprovoznit vlastní instanci aplikace. Do závislostí v `composer.json` jsem přidal některá rozšíření PHP.

3.7 Aktualizace knihoven

Základní framework Nette jsem aktualizoval na verzi 2.3 a provedl jsem některé úpravy, které v budoucnu usnadní aktualizaci na verzi 2.4, resp. na verze řady 3. To vyžadovalo řadu úprav kódu.

Testování

Zadání požaduje důkladné otestování těch částí aplikace, ve kterých jsem prováděl úpravy. Zadání dále požaduje provedení uživatelského testování. Případně je možné zavést automatizované testování.

V průběhu celého vývoje docházelo k průběžnému testování funkcionality mnou, vedoucím práce a následně ostatními uživateli.

K zavedení automatizovaného testování, ať už formou integračních testů nebo unit testů, jsem se z časových důvodů bohužel nedostal.

Závěr

Cílem diplomové práce bylo navázat na předchozí bakalářskou a diplomovou práci, které se zabývaly problematikou databáze konferencí a publikací. Úkol spočíval ve vylepšení uživatelské přívětivosti, zavedení evidencí citací publikací, automatického rozpoznávání referencí publikací, správy vlastních (autorských) publikací, uživatelských tagů u publikací, vylepšení vyhledávání publikací a odstranění drobných chyb aplikace.

V práci byl proveden obecný a technický popis aplikace a zhodnoceny některé aspekty jejího zabezpečení. Zároveň jsem popsal požadavky na úpravu aplikace a navrhnul změny datového modelu.

Podařilo se mi iterativně implementovat nové funkce a zároveň po každé iteraci provést lokální otestování funkčnosti. Tyto celky byly k dispozici uživatelům aplikace. Pro vylepšení fungování aplikace jsem zavedl nové CRUD komponenty. Pro rozpoznávání referencí jsou v práci využity metody parsování referencí, rozpoznávání názvů publikací formou přesné shody a pomocí FULLTEXT indexu, které jsou podrobněji popsány a vyhodnoceny.

Nepodařilo se zavést automatizované testování, ke kterému jsem se nedostal v předpokládané šíři z časových důvodů. Také rozšíření vyhledávání se nepodařilo provést v plné šíři. Provedl jsem však rozsáhlou údržbu kódu, která byla nezbytná pro zachování možnosti pokračovat ve vývoji aplikace do budoucna. Odstranil jsem také několik bezpečnostních problémů.

Je možné v tomto projektu pokračovat dopracováním unit a integračních testů a následně přechodem na ORM Doctrine místo stávající knihovny Nette Database. Další možností je zabývat se podrobněji zabezpečením aplikace.

Literatura

- [1] Svoboda, P.: *Webová databáze konferencí*. Bakalářská práce, Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.
- [2] Kubálek, J.: *Webová databáze referencí*. Diplomová práce, Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.
- [3] Journal of Universal Computer Science: *ACM Categories*. [cit. 2018-02-14]. Dostupné z: http://www.jucs.org/jucs_info/acm_categories
- [4] Universität Bremen: *BibTeX*. [cit. 2018-02-14]. Dostupné z: <http://www.fb10.uni-bremen.de/anglistik/langpro/bibliographies/jacobsen-bibtex.html>
- [5] OWASP: *Application Security Verification Standard Project*. [cit. 2018-02-14]. Dostupné z: https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project
- [6] Nette Foundation: *Latte - amazing template engine for PHP*. [cit. 2018-02-14]. Dostupné z: <https://latte.nette.org/>
- [7] Oracle Corporation and/or its affiliates: *Boolean Full-Text Searches*. [cit. 2018-02-14]. Dostupné z: <https://dev.mysql.com/doc/refman/5.6/en/fulltext-boolean.html>
- [8] Nette Foundation: *Dependency Injection [online]*. [cit. 2018-02-03]. Dostupné z: <https://doc.nette.org/cs/2.4/dependency-injection>
- [9] Nette Foundation: *DI: Ziskávání závislostí [online]*. [cit. 2018-02-03]. Dostupné z: <https://doc.nette.org/cs/2.4/di-usage>
- [10] Filip Procházka: *Kdyby/Autowired [online]*. [cit. 2018-02-03]. Dostupné z: <https://github.com/Kdyby/Autowired>

LITERATURA

- [11] The PHP Group: *password_hash*. [cit. 2018-02-14]. Dostupné z: <https://secure.php.net/manual/en/function.password-hash.php>
- [12] Foundeo Inc.: *Content Security Policy*. [cit. 2018-02-14]. Dostupné z: <https://content-security-policy.com/>
- [13] Nils Adermann, Jordi Boggiano and many community contributions: *Dependency Manager for PHP*. [cit. 2018-02-14]. Dostupné z: <https://getcomposer.org/>

Seznam použitých zkratk

HTTP Hypertext Transfer Protocol

PHP PHP: Hypertext Preprocessor

XSS Cross-site scripting

CSRF Cross-site request forgery

CRUD Create, Read, Update, Delete

SSO Single sign-on

LDAP Lightweight Directory Access Protocol

SHA Secure Hash Algorithm

JS Java Script

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	src	
	impl.....	zdrojové kódy implementace
	thesis.....	zdrojová forma práce ve formátu \LaTeX
	text.....	text práce
	thesis.pdf.....	text práce ve formátu PDF