



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Návrh a implementace modulu Antiplagiát pro Moodle FIT
Student:	Bc. Michal Sklenár
Vedoucí:	Ing. David Buchtela, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Cílem práce je návrh a implementace modulu Antiplagiát pro e-learningový systém FIT Moodle. Modul bude umožňovat kontrolovat plagiátorství v odevzdaných semestrálních pracích (běžných typ textových souborů v rámci modulu Úkol systému Moodle) pomocí spojení s portálem Odevzdej.cz (provozovaném Masarykovou univerzitou v Brně).

1. Formou rešerše se seznamte s modulární architekturou systému Moodle FIT a aplikací ním rozhraním portálu Odevzdej.cz.
2. Proveďte analýzu funkčních a nefunkčních uživatelských požadavků na modul Antiplagiát z pohledu studenta, vyučujícího a systému Odevzdej.cz.
3. Na základě analýzy navrhnete úpravu stávajícího modulu Úkol nebo vlastní modul v souladu s metodami softwarového inženýrství.
4. V koordinaci s vedoucím práce implementujte do systému FIT Moodle (PHP, MySQL) a realizujte propojení na Odevzdej.cz (SSL, CURL).
5. Modul otestujte a nasazte do užívání v rámci FIT.
6. Vypracujte uživatelskou a administrátorskou dokumentaci k modulu.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
ředitel katedry

V Praze dne 13. listopadu 2015

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



Diplomová práce

Návrh a implementace modulu Antiplagiát pro Moodle FIT

Bc. Michal Sklenár

Vedúci práce: Ing. David Buchtela, Ph.D.

5. mája 2016

Pod'akovanie

Týmto by som chcel poďakovať ľuďom, bez ktorých by úspešné dokončenie tohto projektu nebolo možné. Rád by som poďakoval vedúcemu mojej práce Ing. Davidovi Buchtelovi, Ph.D. za vedenie, poskytnutie materiálov potrebných pre prácu na projekte a užitočné rady. Ďalej by som rád poďakoval môjmu teamleaderovi v práci Ing. Martinovi Cajthamlovi za poskytnutie hostingu pre testovanie mojej práce spolu s užitočnými radami a Ing. Jiřímu Machovi za umožnenie testovať komunikáciu so službou Odevzdej.cz pod účtom ČZU. Moje poďakovanie patrí takisto oponentovi mojej práce Ing. Josefovi Vogelovi, Csc., za to, že súhlasil s oponentúrou.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 5. mája 2016

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2016 Michal Sklenár. Všetky práva vyhradené.

Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Sklenár, Michal. *Návrh a implementace modulu Antiplagiát pro Moodle FIT*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Obsahom mojej diplomovej práce je analýza, návrh, popis implementácie a testovania modulu pre systém Moodle. Navyše v závere ponúkam niekoľko návrhov na vylepšenia a prípadné úpravy výsledku mojej práce. Tento modul bol vyvinutý za účelom zefektívnenia a zjednodušenia práce vyučujúcich na fakulte.

Kľúčová slova Moodle, plagiátorstvo, modul, Antiplagiát, Odevzdej.cz

Abstract

The content of my magister thesis is analysis, design, description of implementation and testing of module for system Moodle. Moreover at the end of my work I am outlining possible extensions which could improve the module. This module has been developed in purpose of simplification increasing the effectiveness of faculty's teachers work.

Keywords Moodle, plagiarism, module, Antiplag, Odevzdej.cz

Obsah

Úvod	1
1 Cieľ práce	3
1.1 Vymedzenie problematiky	3
1.2 Ciele práce	3
1.3 Funkčné a nefunkčné požiadavky	4
2 Analýza a návrh	7
2.1 Súčasný stav riešenia danej problematiky	7
2.2 Možnosti riešenia	7
2.3 Zvolené riešenie	8
2.4 Použité technológie a nástroje	9
2.5 Popis systému Odevzdej.cz	10
2.6 Analýza alternatív k službe Odevzdej.cz	11
2.7 Popis systému Moodle	14
2.8 Návrh systému	17
2.9 Analýza systému z užívateľského hľadiska	22
3 Realizácia	25
3.1 Všeobecný popis fungovania modulu z hľadiska implementácie .	25
3.2 Užívateľské role a práva	26
3.3 Prehľad odovzdaných semestrálnych prác	28
3.4 Stiahnutie odovzdaných semestrálnych prác	33
3.5 Komunikácia so službou Odevzdej.cz a proces kontroly práce .	34
4 Testovanie	41
4.1 Popis procesu testovania	41
4.2 Testovacie prostredie systému Moodle	43
4.3 Testy	44

5	Aktuálny stav a výhľad	51
5.1	Aktuálny stav riešenia	51
5.2	Potenciálne chyby a nedostatky systému	52
5.3	Návrhy na vylepšenie modulu	53
5.4	Výhľad do budúcnosti	55
	Záver	57
	Literatúra	59
A	Zoznam použitých skratiek	61
B	Obsah priloženého CD	63

Zoznam obrázkov

2.1	Diagram pohľadu na kľúčové funkcie systému Moodle	16
2.2	Konceptuálny model modulu Antiplagiát	18
2.3	Databázový model modulu Antiplagiát	21
3.1	Prostredie systému Moodle FIT - tabuľka s prehľadom odovzda- ných semestrálnych prác	32
3.2	Kontrolný XML súbor pre komunikáciu so službou Odevzdej.cz . .	37
3.3	Výsledkový protokol kontroly plagiátorstva službou Odevzdej.cz - 1. strana	39
3.4	Výsledkový protokol kontroly plagiátorstva službou Odevzdej.cz - 2. strana	40

Úvod

Problém plagiátorstva je už dlho súčasťou výukového procesu na školách. Študenti sa namiesto toho, aby práce vypracovali samostatne, pokúšajú ísť tou najľahšou cestou, a teda cestou plagiátorstva s vedomím, že nie je v silách vyučujúceho skontrolovať ručne všetky existujúce zdroje a odhaliť ich podvod. V dnešnej dobe, kedy informačné technológie zasahujú takmer do všetkých oblastí ľudského života, je však možné tento problém efektívne riešiť a znížiť tak počet nekvalitných, neoriginálnych prác, teda plagiátov. Za týmto účelom boli vytvorené rôzne služby a systémy, ktoré sa procesom kontroly plagiátorstva zaoberajú. Niektoré z nich poskytujú možnosť ich využitia v iných systémoch a aplikáciách. Príkladom takéhoto systému je aj e-learningový systém Moodle [1].

Ciel' práce

1.1 Vymedzenie problematiky

Témou mojej diplomovej práce je navrhnuť a následne implementovať modul s názvom Antiplagiát pre e-learningový systém FIT Moodle.

Tento modul by mal umožňovať kontrolovanie plagiátorstva v odovzdaných semestrálnych prácach pomocou spojenia s portálom Odevzdej.cz [2], ktorý je prevádzkovaný Masarykovou univerzitou v Brne.

Výsledkom mojej diplomovej práce je funkčný, riadne otestovaný a v praxi použiteľný modul, ktorý je možné nasadiť do produkčnej verzie systému FIT Moodle.

Dôvodom pre vznik spomínaného modulu a vlastne celej tejto práce je potreba efektívne a jednoducho skontrolovať odovzdávané súbory obsahujúce vypracované semestrálne práce z rôznych predmetov. Ručná kontrola je totiž veľmi neefektívna a časovo nesmierne náročná, dokonca v niektorých prípadoch je v podstate nemožná. Navyše v prípade ručnej kontroly zohráva dôležitú úlohu aj ľudský faktor, ktorý býva často príčinou množstva chýb. Je teda dôležité zefektívniť a zjednodušiť proces kontroly tak, aby urýchlil a maximálne zjednodušil prácu vyučujúcich a tým zároveň skrátil i čakanie študentov na vyhodnotenie ich semestrálnych prác.

1.2 Ciele práce

Medzi ciele tejto diplomovej práce patrí zoznámenie sa s modulárnou architektúrou systému Moodle FIT a aplikačným rozhraním portálu Odevzdej.cz. Bez dostatočných vedomostí o spomínaných systémoch a bez dôkladnej analýzy by nebolo možné prácu vôbec realizovať. Z toho dôvodu sa venujem v celej jednej kapitole práve rozboru e-learningového systému Moodle a portálu Odevzdej.cz.

Ďalšou dôležitou súčasťou tejto diplomovej práce je dôkladná analýza všetkých funkčných a nefunkčných užívateľských požiadaviek kladených na vznik

kajúci modul Antiplagiát z pohľadu študenta, vyučujúceho a systému Odevzdej.cz. Na základe tejto analýzy bol navrhnutý a následne naimplementovaný a otestovaný požadovaný modul Antiplagiát.

1.3 Funkčné a nefunkčné požiadavky

Na základe dokumentácie a zdrojového kódu funkčne identického, avšak zastaraného a v praxi už nepoužiteľného modulu, ktoré mi poskytol vedúci mojej práce a na základe vlastnej analýzy som stanovil tieto požiadavky kladené na vznikajúci modul:

- **Funkčné požiadavky:**

Študent:

1. Jednoduché a rýchle nahranie súboru s obsahom semestrálnej práce
2. Prístup k vlastnej známke a prípadnému komentáru od vyučujúceho

Učiteľ:

1. Jednoduché a rýchle vytvorenie novej inštancie modulu v rámci kurzu
2. Jednoduchá konfigurácia a nastavenie všetkých potrebných vlastností inštancie modulu
3. Prístup k prehľadu všetkých inštancií modulu v rámci kurzu
4. Prístup k zoznamu všetkých odovzdaných prác s možnosťou kontroly a známkovania
5. Spustenie kontroly práce portálom Odevzdej.cz
6. Možnosť spustenia hromadnej kontroly prác portálom Odevzdej.cz
7. Možnosť zasielať e-mailom informácie o získaní výsledku kontroly vyučujúcim
8. Možnosť zasielať e-mailom informácie o pridaní známky, alebo komentáru k práci študentom
9. Zobrazenie výsledku kontroly práce portálom Odevzdej.cz spolu s informáciou o percentuálnej hodnote podobnosti kontrolovanej práce
10. Prístup k nedávnej aktivite v module
11. Možnosť stiahnuť dokument s výsledkom kontroly obsahujúcim vyznačené podobnosti v práci
12. Možnosť stiahnuť všetky odovzdané projekty v rámci inštancie modulu

Odevzdej.cz:

1. Modul musí byť schopný vygenerovať validný xml kontrolný súbor používaný službou Odevzdej.cz pri importe súboru
2. Modul musí implementovať cURL príkazy potrebné pre komunikáciu so službou Odevzdej.cz

Ostatné:

1. Rozdelenie právomocí na základe užívateľských rolí

• **Nefunkčné požiadavky:**

1. Pre komunikáciu so službou Odevzdej.cz používať neosobný účet (účet platiacej inštitúcie)
2. Zachovať zásady modulárnej architektúry používanej e-learningovým systémom Moodle a zaistiť tak správnosť návrhu
3. Doména, na ktorej pobeží modul Antiplagiát musí byť medzi doménami, ktoré majú povolenú komunikáciu so službou Odevzdej.cz
4. Použitie rovnakých technológií pri vývoji aplikácie, aké sú použité v systéme Moodle
5. Využitie funkcií a knižníc, ktoré poskytuje systém Moodle pre zaisťovanie kompatibility a správneho návrhu modulu
6. Podpora v internetových prehliadačoch Mozilla Firefox, Opera, Google Chrome a Internet Explorer
7. Čo najvyššia dostupnosť a spoľahlivosť systému
8. Navrhnutý a následne naimplementovaný modul pre e-learningový systém Moodle FIT musí byť schopný nasadenia na produkčnú verziu systému, bez nutnosti zásahu do samotného systému Moodle
9. Vypracovanie dokumentácie k navrhovanému modulu zahŕňajúcej vypracovanie užívateľského manuálu, inštaláčnej príručky a programátorskej príručky

Analýza a návrh

2.1 Súčasný stav riešenia danej problematiky

V súčasnosti neexistuje žiadny v praxi použiteľný modul, ktorý by pokrýval všetky požiadavky stanovené v predchádzajúcej kapitole. V minulosti existoval modul Antiplagiát, ktorý implementoval vedúci mojej práce pre Moodle ČZU. Tento modul je založený na module Úkol [3], ktorý je súčasťou systému Moodle FIT, avšak tento modul bol používaný na oveľa staršej verzii Moodlu, než je verzia používaná v súčasnosti a v praxi je momentálne naprosto nepoužiteľný.

Mnou navrhnutý a následne naimplementovaný modul Antiplagiát je funkčne takmer identický s pôvodnou verziou modulu, avšak využíva výhody a funkcie súčasnej verzie moodlu a mnoho vecí zjednodušuje a zefektívňuje.

2.2 Možnosti riešenia

Existuje viacero rôznych spôsobov, ktorými by bolo možné daný problém riešiť. Všetky tieto riešenia sa navzájom líšia časovou zložitou a obtiažnosťou, s akou by bolo možné zabezpečiť pokrytie všetkých požiadaviek kladených na navrhovaný modul.

Na základe vykonanej dôkladnej analýzy je možné pre implementáciu vybrať jednu z nasledujúcich možností:

- Je možné použiť už existujúci spomínaný modul Antiplagiát naimplementovaný vedúcim mojej práce a prepísať ho tak, aby bol použiteľný na v súčasnosti používanej verzii systému Moodle FIT
- Druhou možnosťou je použitie nejakého iného modulu, ktorý aspoň z časti rieši danú problematiku a pokrýva tak aspoň časť z požiadaviek kladených na navrhovaný modul

- Ďalšou z možností je postaviť modul na module Úkol, ktorý bol použitý aj pri pôvodnej verzii modulu a ktorý rieši väčšinu požiadaviek a doplní ho o prepojenie a proces kontroly práce na portáli Odevzdej.cz
- Ďalšou možnosťou je vytvorenie samostatne použiteľného modulu od počiatku a použitie niektorých funkcií z pôvodnej verzie modulu Antiplagiát a z modulu Úkol, na ktorom je tento modul postavený
- Poslednou možnosťou je vyvinúť celý modul nezávisle na systéme module, teda bez využitia knižníc a funkcií, ktoré ponúka

2.3 Zvolené riešenie

Riešenie znamenajúce použitie už existujúceho modulu Antiplagiát som po krátkej úvahe a dôkladnejšej analýze modulu zavrhol. Väčšina funkcií použitých v starej verzii modulu je označených ako deprecated (zastarané), čo znamená, že buď budú v niektorej z ďalších verzií systému moodle odstránené a nahradené inými funkciami, alebo už dávno odstránené boli. Niektoré z nich dokonca nemajú v súčasne používanej verzii systému ani náhradu, čím sa pre mňa stali úplne nepoužiteľnými. Použitie tejto možnosti by v konečnom dôsledku znamenalo naimplementovať modul od začiatku, pretože snaha o znovupoužiteľnosť tohto modulu by bola až príliš časovo náročná.

Druhá možnosť sa ukázala takisto nepoužiteľná, pretože v súčasnosti neexistuje žiadny modul okrem modulu Úkol, ktorý by pokrýval také množstvo požiadaviek, aby stálo za to postaviť na ňom navrhovaný modul.

Nad možnosťou postaviť modul na module Úkol som uvažoval asi najdlhšie. Tento modul rieši odovzdávanie a známkovanie odovzdaných prác. Tieto dve funkcie patria k hlavným funkciám navrhovaného modulu. Modul Úkol je však príliš robustný a väčšina ďalších funkcií, ktoré poskytuje je pre navrhovaný modul zbytočná. Za najväčší zápor tohto riešenia, ktorý nakoniec rozhodol o jeho nepoužití však považujem skutočnosť, že priame použitie tohto modulu a jeho prípadné rozšírenie by bolo až príliš neisté vzhľadom k ďalšiemu vývoju. Modul Úkol je totiž aj v súčasnosti vyvíjaný a upravovaný a je tu teda reálna možnosť, že v niektorej z novších verzií bude uskutočnená zmena, ktorá by mohla ovplyvniť môj modul a urobiť ho nepoužiteľným.

Posledná možnosť síce poskytuje relatívne veľkú voľnosť pri výbere riešenia konkrétnych problémov a odstraňuje problém s prípadnou nekompatibilitou s prechodom na novšiu verziu systému, ale zároveň je časovo najnáročnejšia a teda aj najmenej efektívna. Okrem toho by použitie tejto možnosti znamenalo to, že obrovské množstvo funkcií, ktoré systém moodle už obsahuje by som musel implementovať sám. Navyše po dôkladnejšej analýze som prišiel na to, že v niektorých prípadoch je využitie funkcií obsiahnutých v systéme moodle nevyhnutné, pretože moodle niektoré veci jednoducho nedovolí robiť iným spôsobom, než tým, ktorý sám odporúča v dokumentácii [4].

Po podrobnej analýze všetkých stanovených požiadaviek na navrhovaný modul a po dôkladnom preskúmaní výhod a nevýhod všetkých dostupných možností som zvolil použitie predposlednej možnosti riešenia. Rozhodol som sa teda vytvoriť samostatne použiteľný modul, nezávislý na iných moduloch, ktorý bude používať niektoré vlastnoručne upravené funkcie obsiahnuté v module Úkol a v pôvodnej verzii modulu Antiplagiát. Môj modul som navrhol tak, aby neporušoval modulárnu architektúru systému Moodle FIT. Databázová vrstva je realizovaná pomocou jazyka SQL. K vytvoreniu modulu som použil technológie PHP, HTML, CSS a Javascript. Pre komunikáciu so službou Odevzdej.cz používam vo svojom module knižnicu libcurl.

2.4 Použité technológie a nástroje

Ako som už spomínal, rozhodol som sa pre implementáciu celej aplikácie použiť technológiu PHP.

PHP je skriptovací programovací jazyk určený predovšetkým pre programovanie dynamických internetových stránok a webových aplikácií. Použitie PHP bolo jednou z nefunkčných požiadaviek stanovených hneď v úvode projektu. Celý systém Keďže celý systém Moodle FIT je naprogramovaný v jazyku PHP, táto požiadavka bola samozrejmosťou. Navyše PHP je jazyk nezávislý na platforme, takže jeho použitím sa odstráni problém možnej neprenositelnosti aplikácie medzi rôznymi platformami. Podporuje mnoho knižníc pre rôzne účely a poskytuje možnosť prepojenia s databázovým systémom. Okrem iného podporuje aj knižnicu libcurl, ktorá je jedinou možnosťou pre komunikáciu so službou Odevzdej.cz.

Celá implementácia prebiehala v známom editore od spoločnosti JetBrains s názvom PhpStorm [5], ktorý používa množstvo spoločností v Českej republike i v zahraničí a je obľúbený kvôli množstvu funkcií, ktoré poskytuje. Obsahuje vlastný debugger, poskytuje možnosť priameho prepojenia s verzovacím systémom a mnoho ďalšieho.

Keďže PHP potrebuje pre svoje fungovanie nejaký server a ja som až donedávna nedisponoval žiadnym priestorom, kde by som svoju prácu mohol umiestniť, po väčšinu času stráveného implementáciou som používal aplikáciu MAMP PRO [6]. Táto aplikácia umožňuje vytvorenie lokálneho serveru, celkom intuitívnu správu virtualhostov a pomáha mi pri každodennej práci.

Samozrejmosťou bolo použitie nejakého verzovacieho systému, ktorý by mi slúžil ako úložisko pre môj projekt v prípade, že by nastal nejaký problém s mojím počítačom a prišiel by som o všetky dáta a zároveň by poskytoval možnosť vrátiť aplikáciu do nejakého konkrétneho stavu. Keďže sa jedná o projekt pre školu, rozhodol som sa využiť fakultný GitLab.

Ako bolo už skôr spomenuté, jedinou možnosťou pre zabezpečenie komunikácie so službou Odevzdej.cz je v súčasnosti použitie cURL, respektíve knižnice libcurl, ktorá umožňuje pripojenie a komunikáciu s mnohými rozličnými

typmi servrov s mnohými rozličnými typmi protokolov. Využíva sa hlavne pri prenose dát, komunikácii s rôznymi službami (napr. Odevzdej.cz), ako základ pre komunikáciu v rôznych API (napr. Mailchimp), dokáže zistiť, či nejaká stránka existuje, či je presmerovaná na inú adresu a ak áno, tak na akú. Možnosť využitia existuje samozrejme oveľa viac, ale nebudem ich vymenovávať všetky. Veľkou výhodou knižnice libcurl je skutočnosť, že je zadarmo a je prenositeľná, čo znamená, že funguje identicky na mnohých platformách. Medzi platformy, ktoré stoja za zmienku patria napríklad Linux, Windows, Android, iOS a OS X. Libcurl je navyše kompatibilná s IPv6 a podporuje SSL/TLS prostredníctvom OpenSSL. Okrem toho je knižnica libcurl thread-safe, čo znamená, že manipuluje so zdieľanými dátovými štruktúrami spôsobom, ktorý garantuje bezpečnú exekúciu viacerými vláknami v tom istom čase. Toto je využívané pri viacvláknových programoch.

2.5 Popis systému Odevzdej.cz

Služba Odevzdej.cz je systém pre odhaľovanie plagiátov v seminárnych, alebo iných prácach, ktorý je prevádzkovaný vývojovým tímom systému Masarykovej univerzity v Brne. Tento systém je vlastne e-learningovým nástrojom pre zber, kontrolu a hodnotenie prác učiteľom. V roku 2009 a 2010 bol vznik tohto systému finančne podporený z centralizovaného rozvojového projektu MŠMT C13/2009 “Odhaľovanie plagiátů v seminárních pracích” a MŠMT C20/2010 “Rozvoj infrastruktur pro využívání hledání podobností mezi studentskými pracemi a zdroji na Internetu”.

Tento systém je jedným z trojice systémov prevádzkovaných Masarykovou univerzitou v Brne. Ďalším z tejto trojice je systém Theses.cz, systém pre odhaľovanie plagiátov medzi záverečnými prácami, ktorý okrem toho slúži vysokým školám a univerzitám nielen v Českej republike ako národný register záverečných prác a ako úložisko prác pre vyhľadávanie plagiátov. Posledným z trojice služieb je systém Repozitar.cz, systém, ktorý si za ciele kladie zber metadát a plných zamestnaneckých diel, kontrolu podobností (potenciálnych plagiátov), zverejňovanie plných textov v režime Open-Access (v súlade s licenčnými podmienkami), automatizované prepojenie s inými systémami a poskytuje pokročilé vyhľadávacie nástroje.

Služba Odevzdej.cz okrem iného poskytuje kontrolu prác pre verejnosť. Takáto kontrola spočíva v tom, že užívateľ, ktorý chce skontrolovať nejakú prácu a zistiť tak, či sa jedná o plagiát, jednoducho nahrá súbor s prácou priamo vo webovom rozhraní služby. Výsledok kontroly je užívateľovi zhruba za jeden až dva dni odoslaný priamo na e-mailovú adresu, ktorú vyplnil pri nahrávaní súboru s prácou.

Služba funguje na princípe porovnávania podobností naprieč databázami porovnávaných dokumentov už spomínanej trojice systémov Odevzdej.cz, Theses.cz a Repozitar.cz a ďalších zdrojov vrátane vyhľadávania podobností na

Internetu. Ďalšou funkciou tejto služby je možnosť napojenia systému na lokálny systém školy a vzdialené využívanie služieb vyhľadávania podobností. Na tejto funkcii je postavený celý proces kontroly a porovnávania podobností v semestrálnych prácach, ktorý je hlavnou funkciou a dôležitou súčasťou navrhovaného modulu Antiplagiát. Obrovskou výhodou a podľa môjho názoru aj hlavným dôvodom pre zvolenie tejto služby vedúcim mojej diplomovej práce, a teda autorom pôvodného modulu Antiplagiát využívajúceho túto službu je práve skutočnosť, že táto služba disponuje tromi databázami s veľkým množstvom rôznych prác, s ktorými je možné semestrálnu prácu porovnávať. Pri vypracovávaní analýzy som sa rozhodol urobiť si menší prieskum a nenašiel som žiadnu službu, ktorá by fungovala na tomto princípe a zároveň pokrývala všetky požiadavky modulu Antiplagiát. Všetky služby, ktoré som skúmal vyhľadávali podobnosti ku kontrolovanej semestrálnej práci prostredníctvom vyhľadávania rôznych špecifických výrazov na internete, hlavne prostredníctvom vyhľadávača Google a väčšina z nich vôbec neposkytovala možnosť vzdialeného využívania služby. Aj keď nerád, nakoniec som musel priznať, že výber služby Odevzdej.cz bol správnu voľbou, ktorá pokrýva požiadavky kladené na navrhovaný modul Antiplagiát.

Vzdialené využívanie služieb vyhľadávania podobností funguje za použitia cURL, respektíve v prípade PHP knižnice libcurl, ktorej fungovanie a využitie som podrobne popísal v predchádzajúcej sekcii. Užívateľ sa najprv pomocou jednoduchého príkazu autentizuje voči službe Odevzdej.cz pomocou identifikátoru, nazývaného “neosobné učo” a hesla, ktoré obdržal. Následne môže užívateľ, respektíve systém zasílať príkazy pre import súborov s prácami do systému, alebo príkazy pre zistenie existencie výsledku kontroly už nainportovaných súborov.

Služba Odevzdej.cz poskytuje na svojich webových stránkach nápovedu, kde sú uverejnené príklady všetkých povolených príkazov, ktoré je možné použiť pri komunikácii so službou spolu s popisom.

2.6 Analýza alternatív k službe Odevzdej.cz

V tejto časti by som rád popísal služby, ktoré sú svojou funkčnosťou viac, či menej podobné fungovaniu služby Odevzdej.cz. V zadaní síce mám explicitne uvedené, že modul má využívať komunikáciu práve s touto službou, ale niekoľko vecí ma od používania už od začiatku odrádzalo. V priebehu projektu pribúdali ďalšie problémy a v istom bode nebolo vôbec isté, či je správna komunikácia so zvolenou službou vôbec uskutočniteľná. Všetko toto boli dôvody pre vykonanie mešieho prieskumu, ktorý spomínam pri popise služby Odevzdej.cz a ktorého cieľom bolo nájsť záložnú alternatívu k službe, ktorá by svojím fungovaním pokrývala požiadavky kladené na navrhovaný modul tak, ako ich pokrýva zvolená služba Odevzdej.cz. Preto v tejto časti popíšem alternatívy k zvolenej službe Odevzdej.cz a zároveň aj dôvody, kvôli ktorým vôbec

vznikla potreba tohto prieskumu.

Prvým dôvodom, ktorý ma odrádzal od používania zvolenej služby Odevzdej.cz bol celkový dizajn webového rozhrania tejto služby. Služba pripomína systém, o ktorý sa už dlhší čas nikto nestará, jeho vývoj sa zastavil pred pár rokmi a nič nenasvedčuje tomu, že by sa plánovala nejaká modernizácia, alebo aspoň nejaká zmena, ktorá by potenciálneho užívateľa presvedčila, alebo mu aspoň nejakým spôsobom indikovala, že služba je vôbec ešte funkčná a použiteľná v praxi.

Druhým dôvodom, ktorý ma vlastne najprv skôr prekvapil je skutočnosť, že technická podpora služby Odevzdej.cz zrejme odpovedá len na otázky položené prostredníctvom kontaktného formuláru umiestneného na internetových stránkach služby. Otázky položené prostredníctvom tohto formuláru sú totiž spoplatnené. V priebehu projektu som narazil na problém, keď mi služba stále poskytovala podľa dokumentácie validný popis stavu, v ktorom sa nachádzal kontrolovaný súbor, avšak tento stav sa ani po niekoľkých dňoch nijak nezmenil, pričom celý proces kontroly by mal trvať maximálne dva dni. Kontaktoval som teda podporu služby Odevzdej.cz na kontaktnej adrese uvedenej na ich webových stránkach. Ani po pár dňoch mi neprišla žiadna reakcia na mnou popisovaný problém a žiadosť o radu. Od známeho študujúceho na Masarykovej univerzite v Brne sa mi podarilo získať kontakt priamo na hlavného vývojára, ktorý má na starosti projekt Odevzdej.cz. Keď som ho kontaktoval, po pár dňoch mi prišla odpoveď, že mám kontaktovať podporu na e-mailovej adrese, ktorú som už predtým použil. Nakoniec som bol nútený využiť spomínaný spoplatnený kontaktný formulár. Tu som dostal odpoveď do pár hodín po odoslaní formuláru. Príčinu problému sa mi tak podarilo zistiť a nakoniec problém aj úspešne vyriešiť, avšak do toho momentu nebolo vôbec jasné, či sa mi podarí zabezpečiť správnu komunikáciu so službou, a teda či vôbec budem schopný dokončiť túto prácu.

Popisované problémy v priebehu vývoja modulu Antiplagiát viedli k vzniku potreby nájsť záložnú alternatívu k službe Odevzdej.cz, ktorú som mal v pláne použiť, ak by problém so zvolenou službou bol nevyriešiteľný.

Po menšom prieskume som vybral dvoch kandidátov, ktorí ma zaujali a stoja za zmienku:

- **Crot Plagiarism Checker** - Crot [7] je plugin pre e-learningový systém Moodle, ktorého úlohou je kontrola plagiátorstva v súboroch. Pracuje v dvoch základných módoch:
 1. Lokálna detekcia plagiátorstva: vyhľadávanie podobností medzi prácami odovzdanými priamo systéme Moodle.
 2. Internetová detekcia plagiátorstva: vyhľadávanie podobných dokumentov na internete, ich stiahnutie a zobrazenie podobností.

Použitie tohto pluginu by bolo veľmi jednoduché, keďže je navrhnutý priamo pre Moodle. Pri podrobnejšom hľadaní som však narazil na sku-

točnosť, že vo verzii, ktorá by bola kompatibilná s aktuálne používanou verziou systému Moodle FIT momentálne nefunguje detekcia plagiátorstva cez internet. Zostáva tak len funkcia lokálnej detekcie plagiátorstva, čo znamená, že podobnosti by bolo možné vyhľadávať len medzi prácami odovzdanými priamo v systéme Moodle FIT. Bohužiaľ, tým sa rapídne znižuje pravdepodobnosť, že bude plagiát skutočne odhalený. Každopádne podľa informácií, ktoré uverejnili vývojári, ktorí stoja za pluginom Crot Plagiarism Checker, na tomto projekte sa stále pracuje a vyriešenie problémov s kontrolou plagiátorstva na internete je tak pravdepodobne otázka času.

- **Turnitin** - služba turnitin [8] je využívaná viac ako tridsiatimi miliónmi študentov a pätnástimi tisícmi inštitúcií po celom svete. Služba disponuje vlastným API, na ktoré je možné sa pripojiť a využívať tak jej funkcie. Okrem toho existuje už niekoľko pluginov, respektíve modulov pre e-learningový systém Moodle, ktoré už využívajú službu turnitin pre kontrolu plagiátorstva. Implementácia by vďaka tomu nebola taká zložitá a je tu veľký priestor pre inšpiráciu z týchto už existujúcich modulov. Navyše by sa výrazne znížilo riziko možnosti, že pri implementácii zostanem visieť na mŕtvom bode. Viacero vývojárov totiž prepojenie systému Moodle a služby turnitin už riešilo, a teda na množstvo prípadných problémov je v priebehu pár minút možné nájsť uspokojivé riešenie. Táto služba vyzerala zo všetkých skúmaných asi najslubnejšie a najvhodnejšie. Navyše tým, čo poskytuje by pokryla stanovené požiadavky a otvorila by ďalšie možnosti pre vylepšenia a rôzne úpravy. Veľkým plusom je skutočnosť, že turnitin porovnáva práce s databázou podobne ako služba Odevzdej.cz len s tým rozdielom, že podľa všetkých dostupných informácií databáza, ktorú používa služba turnitin je najväčšia svojho druhu. Existuje teda veľké množstvo kladov, ktoré hovoria v prospech výberu tejto služby.

Na druhej strane táto služba je tiež spoplatnená. Konkrétnu cenu sa mi nepodarilo zistiť, pretože sa vytvára na mieru pre konkrétnu inštitúciu, prípadné individuálneho jedinca. Avšak podľa informácií dostupných na internete sa cena pohybuje zhruba okolo niekoľko tisíc dolárov. Najlacnejšiu variantu som našiel zhruba za tristo dolárov, avšak tá bola spojená s obmedzením počtu skontrolovaných súborov na zhruba stopäťdesiat. Keďže som si vedomý toho, že sa jedná o naozaj vysokú cenu, prípadné využitie tejto služby zostalo otáznou a táto možnosť sa stala skôr návrhom na prípadné vylepšenie modulu v budúcnosti.

Pri mojom prieskume som narazil na množstvo služieb, modulov a rôznych rozšírení, ktoré riešia detekciu plagiátorstva. Drvivá väčšina z nich však prehľadáva len internet s využitím API od spoločnosti Google a nedisponuje vlastnou databázou. Ďalšou nevýhodou služieb fungujúcich na tomto princípe

je skutočnosť, že kontrola trvá naozaj dlho a značne by zťažovala server. Bohužiaľ, okrem služby Turnitin sa mi nepodarilo nájsť žiadnu inú službu, ktorá by pokrývala požiadavky stanovené na modul Antiplagiát ekvivalentne so službou Odevzdej.cz. Jediným ďalším riešením by bolo vytvorenie vlastnej služby pre odhaľovanie plagiátorstva v semestrálnych prácach. Toto riešenie poskytuje najväčšiu voľnosť a bolo by teda bez problémov možné navrhnúť tento systém tak, aby pokrýval všetky požiadavky stanovené na modul Antiplagiát. Na druhej strane je táto možnosť značne časovo náročná a pre moju prácu nerealizovateľná. Je to však nápad, ktorý ma zaujal a myslím si, že v budúcnosti by sa mohol použiť ako zadanie pre inú záverečnú prácu, alebo študentský projekt.

2.7 Popis systému Moodle

Moodle je výuková platforma navrhnutá tak, aby poskytovala pedagógom, administrátorom a študentom odolný, bezpečný a integrovaný systém pre vytváranie personalizovaných vzdelávacích prostredí. Moodle môžeme označiť ako Learning Management System (LMS), Course Management System (CMS), alebo Virtual Learning Environment v závislosti na tom, aký termín preferujete. Jeho cieľom je poskytnúť vyučujúcim a študentom nástroje, ktoré potrebujú pre výuku a učenie sa. Tieto nástroje sa okrem iného snažia aj o zefektívnenie a zjednodušenie výukového procesu.

Slovo Moodle, bolo pôvodne akronymom pre Modular Object-Oriented Dynamic Learning Environment (Modulárne objektovo orientované dynamické prostredie pre výuku). Pôvodným autorom programu Moodle je Martin Dougiamas, austrálsky pedagóg a vývojár, ktorý koordinuje jeho vývoj dodnes. Moodle funguje v rámci projektu Moodle, ktorý je vedený a koordinovaný Moodle Pty Ltd, austrálskou spoločnosťou tridsiatich developerov, ktorá je finančne podporovaná sieťou vyše šesťdesiatich servisných spoločností po celom svete. Niektoré z nich sú nositeľom certifikátu Moodle Partner a môžu tak používať značku MoodleTM. Moodle Partner pre Českú republiku je spoločnosť PragoData Consulting.

Existujú ďalšie typy softvérových systémov, ktoré sú dôležité pre výukové inštitúcie, medzi ktoré patrí napríklad ePortfolios. Moodle sa nesnaží o opätovné objavenie týchto oblastí funkcionality, naopak, cieľom Moodlu je byť tak dobrým Learning Management System ako je to možné, byť v týchto oblastiach funkcionality čo najlepší a interoperovať s ďalšími systémami s cieľom sprostredkovať nové oblasti funkcionality. Veľkým pozitívom Moodlu je skutočnosť, že je možné ho používať ako samostatnú aplikáciu bez nutnosti integrácie s iným systémom.

Prvé prototypy systému Moodle boli napísané v jazyku Python nad aplikačným serverom Zope. Potom bol systém prepísaný do jazyka PHP a prvá verzia bola zverejnená v roku 2002. Na českej lokalizácii sa pracuje od 10.1.2003.

Moodle podporuje viacero typov databáz ako PostgreSQL, MySQL, Microsoft SQL Server, MariaDB či Oracle. Databáze Moodle obsahuje niečo vyše 250 databázových tabuliek. Tieto tabuľky sa v databáze vytvoria automaticky pri inštalácii systému.

Tak ako mnoho úspešných open source systémov, i Moodle je štruktúrovaný ako jadro aplikácie, obklopené mnohými pluginmi, s cieľom poskytovať špecifické funkcie. Moodle je navrhnutý tak, aby bol rozšíriteľný a modifikovateľný, bez nutnosti úpravy kľúčových systémových knižníc. Takéto zmeny by totiž znamenali často nevyriešiteľné problémy pri aktualizácii systému Moodle na novšiu verziu. V prípade modifikovania, alebo rozširovania inštalácie Moodle by všetky úpravy mali byť vykonávané priamo v architektúre pluginov.

Za základnú stavebnú jednotku systému Moodle môžeme považovať kurz. Pre účely tejto práce môžeme kurz chápať ako nejaký školský predmet a užívateľov prihlásených do kurzu ako študentov tohto predmetu. Kurz potom môže obsahovať rôzne pluginy, moduly, aktivity a rozšírenia, ktoré pomáhajú pri výuke. Patria sem napríklad moduly pre odovzdávanie súborov ako už spomínaný modul “Úkol”, rôzne moduly a pluginy pre rozšírenie možností známkovania, tvorbu a využívanie ankiet, alebo aj kontrolu plagiátorstva v semestrálnych prácach, čo je funkciou mnou navrhnutého a následne naimplementovaného modulu Antiplagiát. Okrem toho je možné systém Moodle použiť aj na vytváranie rôznych kvízov, čo je užitočné v prípade nejakej školskej inštitúcie. Tieto kvízy môžeme totiž chápať aj ako elektronické testy. Takéto testy sú výhodou pre vyučujúcich, ktorí nemusia tráviť veľké množstvo času opravovaním testov a zároveň aj pre študentov, ktorí nemusia dlho čakať na výsledky testu. Výsledok testu sa dozvedia hneď po odovzdaní vyplneného zadania.

Inštalácia systému Moodle obsahuje Moodle kód spustiteľný v PHP na webovom serveri, databázu spravovanú MySQL, PostgreSQL, Microsoft SQL Server, MariaDB, alebo Oracle a úložisko súborov pre nahrané a vygenerované súbory (zložka moodledata). Všetky tieto tri časti môžu bežať na jednom serveri, alebo naopak môžu byť oddelené a bežať na viacerých serveroch kvôli vyrovnávaniu záťaže. Moodle je navrhnutý tak, aby bol jednoducho nainštalovateľný na ľubovoľnom serveri, ktorý spĺňa základné podmienky, ktoré sú užívateľovi zobrazené pred inštaláciou spolu s informáciami o tom, či sú dané podmienky splnené, alebo nie. Moodle sa akonáhle je skopírovanie kódu na webový server úspešne dokončené a prázdna databáza vytvorená následne sám nainštaluje. Podobne dokáže Moodle vykonať aktualizáciu seba samého z jednej verzie na druhú. Na vytvorenie výukového prostredia založeného na systéme Moodle, teda nie sú potrebné žiadne extra skúsenosti, inštalácia je rýchla a jednoduchá a zvládne ju aj užívateľ, ktorý nie je programátorom.

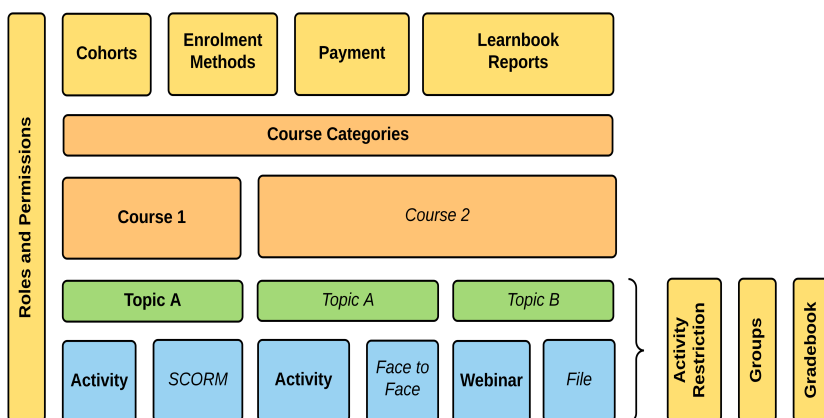
Štandardná inštalácia Moodle obsahuje jadro Moodle a celú radu pluginov každého typu. Moodle je tak ihneď po nainštalovaní možné začať bez ďalších úprav používať. Fyzicky je plugin v systéme Moodle len zložka obsahujúca PHP skripty (a CSS, Javascript ak je to nutné). Jadro Moodle komunikuje s pluginmi hľadaním prístupových bodov, často definovaných v súbore lib.php

2. ANALÝZA A NÁVRH

vo vnútri pluginu.

Plugins v rámci Moodle sú vždy určitých špecifických typov. To znamená, že napríklad plugin pre autentifikáciu a modul aktivity komunikujú s jadrom Moodle prostredníctvom odlišných API. Funkcie, ktoré sú pre pluginy spoločné (inštalácia, aktualizácia, oprávnenia, konfigurácia...) sú obsluhované konzistentne naprieč všetkými typmi pluginov. Každý správne navrhnutý plugin by mal implementovať niektoré funkcie využívané jadrom Moodle. Ak tieto funkcie nebudú naimplementované, nedôjde k chybe a systém bude fungovať správne, avšak niektoré podľa môjho názoru základné funkcie, nebudú s dátami pluginu vôbec počítané. Jedná sa o funkcie využívané pri zálohovaní, resetovaní, alebo obnovovaní kurzu, o výpis nedávnej aktivity v rámci pluginu, výpis nedávnej aktivity konkrétneho užívateľa, sprostredkovanie nahraných súborov, alebo pomocné funkcie využívané pri známkovaní. Tieto funkcie by mali byť typicky implementované v súbore lib.php. Avšak tento súbor by nemal obsahovať implementáciu žiadnej ďalšej logiky v module. Implementácia logiky modulu, respektíve pluginu, by mala byť umiestnená v súbore local-lib.php, prípadne ďalších súboroch, využívaných hlavne kvôli prehľadnosti a vytvoreniu nejakej rozumnej a prehľadnej štruktúry.

V závere tejto časti venovanej popisu e-learningového systému Moodle by som chcel aspoň stručne popísať core funkcionality tohto systému. Nasledujúci diagram, ktorý môžete vidieť na obrázku číslo 2.1 predstavuje vysokoúrovňový pohľad na funkcie Moodle používané administrátormi a študentmi.



Obr. 2.1: Diagram pohľadu na kľúčové funkcie systému Moodle

- Users (užívatelia) - fyzické užívateľské účty, každý s užívateľským menom a heslom
- Cohorts (globálne skupiny) - skupiny užívateľov naprieč systémom. Môžu byť automaticky vygenerované z polí v užívateľskom profile

- Enrolment Methods (metódy zápisu) - riadi, do ktorých kurzov je užívateľ zapísaný
- Payment (platby) - platby sú uskutočňované prostredníctvom špeciálnych metód zápisu
- Learnbook Reports (reporty) - reportovanie naprieč webom napríklad prostredníctvom kurzov
- Course Categories (kategórie kurzov) - hierarchia kurzov
- Courses (kurzy) - kurzy sú priestormi, kde vyučujúci pridávajú učebné materiály (aktivity) pre svojich študentov
- Topics (témy) - sekcie v rámci kurzu
- Activities (aktivity) - aktivita (alebo činnosť) je Moodle funkcia ako napríklad modul “Úkol”, alebo “Kvíz”, do ktorých sa môžu študenti zapojiť
- Activity Restrictions (obmedzenia aktivít) - konfigurovateľné pravidlá, ktoré riadia aké aktivity sú dostupné pre študentov. Sú vysoko flexibilné a môžu byť použité pre obmedzenie užívateľa dokončením iných aktivít, časom, alebo nejakými poľami v užívateľskom profile
- Groups (skupiny) - skupiny užívateľov v rámci nejakého konkrétneho špecifického kurzu
- Gradebook (kniha známok) - obsahuje uložené známky, ktoré užívatelia získali za určité aktivity
- Roles and Permissions (role a oprávnenia) - pravidlá, ktoré kontrolujú úroveň administrácie, ktorú je užívateľ oprávnený vykonávať. Nezamieňať s Activity Restrictions, ktoré sa používajú pre riadenie prístupu k aktivitám kurzu.

Týmto by som ukončil sekciu venovanú popisu e-learningového systému Moodle. Myslím, že informácie obsiahnuté v tejto časti sú dostatočné pre vytvorenie si aspoň základnej predstavy o fungovaní, význame a využití tohto systému.

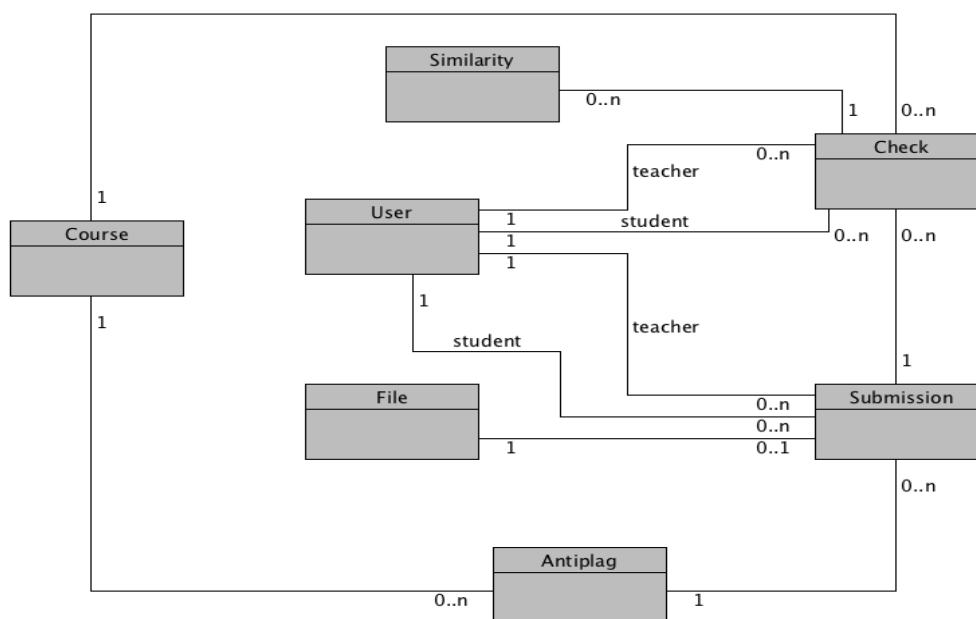
2.8 Návrh systému

V tejto časti by som rád popísal mnou navrhovaný modul z hľadiska architektúry, použitých technológií a podobne.

2. ANALÝZA A NÁVRH

Na obrázku číslo 2.2 môžete vidieť konceptuálny model modulu Antiplagiát. Pomocou tohto modelu by som rád vysvetlil požadované fungovanie navrhovaného modulu. Vo svojej podstate je jeho štruktúra skutočne jednoduchá a ako môžete vidieť v diagrame, pozostáva len zo štyroch hlavných entít, ktoré majú ďalej väzby na entity systému Moodle. Hlavnými entitami tvoriacimi jadro modulu Antiplagiát sú:

1. Antiplag - entita reprezentujúca inštanciu modulu Antiplagiát
2. Submission - entita reprezentujúca záznam o odovzdaní semestrálnej práce
3. Check - entita reprezentujúca záznam o kontrole semestrálnej práce službou Odevzdej.cz
4. Similarity - entita reprezentujúca záznam o dokumente podobnom so súborom kontrolovaným prostredníctvom služby Odevzdej.cz



Obr. 2.2: Konceptuálny model modulu Antiplagiát

Ďalšími entitami, ktoré môžete vidieť v modeli sú Course, entita reprezentujúca kurz v rámci systému Moodle, ďalej entita User, reprezentujúca užívateľa (študent, vyučujúci) a entita File, reprezentujúca záznam o nahranom súbore so semestrálnou prácou. Toto sú entity systému Moodle, na ktoré má mnou navrhovaný modul väzby.

V tomto modeli sú zachytené len tie najdôležitejšie entity reprezentujúce len samotný navrhovaný modul a dôležité entity, s ktorými má nejaký vzťah. Model nezachytáva celý systém Moodle. Konceptuálny model celého systému by bol omnoho väčší, zložitejší a pre účely tejto práce absolútne nezmyselný a zbytočný. Pre potreby vysvetlenia princípu fungovania pripravovaného modulu v rámci systému Moodle nám úplne postačia uvedené entity. Atribúty jednotlivých entít v tomto modeli nie sú uvedené. Mnou vytvorený model slúži hlavne na popísanie vzťahov medzi jednotlivými časťami modulu a vzťahov medzi modulom a systémom Moodle ako celkom.

Za najdôležitejšiu, alebo ústrednú entitu, od ktorej sa všetko odvíja môžeme pokladať entitu s anglickým názvom Antiplag. Táto entita je vlastne základom navrhovaného a následne implementovaného modulu Antiplagiát. K jednému záznamu o inštancii modulu môže byť v databáze priradených až N záznamov, ktoré sú reprezentované entitou Submission. Ako som už spomínal táto entita reprezentuje záznam o odovzdaní semestrálnej práce. Zároveň však platí, že záznamy typu Submission vôbec existovať nemusia. Jedná sa napríklad o stav po vytvorení inštancie modulu, alebo o stav, keď skutočne žiadny študent zatiaľ neodovzdal semestrálnu prácu. Naopak však platí, že záznam o odovzdaní semestrálnej práce nemôže existovať bez existencie záznamu o inštancii modulu. Entita Antiplag sa ďalej viaže entita kurz. Typicky platí, že v rámci jedného kurzu môže existovať viacero inštancií modulu Antiplagiát, avšak naopak platí, že jedna inštancia modulu sa musí viazať práve na jeden kurz.

Ďalšou entitou je už viackrát spomínaná entita Submission, ktorá predstavuje záznam o odovzdaní semestrálnej práce. Táto entita má okrem už tiež spomínanej väzby na entitu Antiplag niekoľko ďalších väzieb. Entita má viacnásobnú väzbu na entitu User. Jedna väzba predstavuje študenta, respektíve autora semestrálnej práce, ktorý vykonal odovzdanie. Druhý záznam predstavuje vyučujúceho. Záznam o vyučujúcom nie je priradený až do okamihu, keď vyučujúci oznámkuje prácu, alebo pridá k práci nejaký komentár. Samozrejme z druhej strany je väzba nepovinná. Užívateľ nemusí mať totiž priradený žiadny záznam o odovzdaní semestrálnej práce. Ďalšou väzbou je väzba na entitu File. Je to entita v rámci systému Moodle, ktorá reprezentuje skutočný fyzický súbor, v tomto prípade so semestrálnou prácou. Posledný vzťah týkajúci sa tejto entity je vzťah s pravdepodobne najvýznamnejšou entitou modulu Antiplagiát, a to s entitou s anglickým názvom Check. Táto entita, ako už názov napovedá, reprezentuje záznam o kontrole semestrálnej práce službou Odevzdej.cz. Takýchto záznamov môže byť k entite priradených až N. Je to kvôli skutočnosti, že v prípade povoleného viacnásobného

odovzdania práce je možné a logicky aj žiadané upravenú a opätovne nahranú semestrálnu prácu znovu skontrolovať.

Ďalej by som rád popísal vzťahy spojené s entitou s anglickým názvom Check. Táto entita má podobne ako aj entita Submission viacnásobnú väzbu na entitu User. Rovnako ako pri entite Submission sa jedná o vzťah reprezentujúci študenta, ktorého práca je kontrolovaná a o vzťah reprezentujúci vyučujúceho, ktorý žiada o kontrolu práce službou Odevzdej.cz. Samozrejmosťou je skutočnosť, že väzba na entitu Submission je povinná. Záznam o kontrole semestrálnej práce logicky nemôže existovať bez toho, aby existoval záznam o odovzdaní danej semestrálnej práce.

Ako posledné by som rád popísal entitu Similarity, ktorá predstavuje záznam o dokumente, ktorý je podobný s testovaným dokumentom. Záznamov typu Similarity môže byť k jednému záznamu o kontrole ľubovoľné množstvo. Zároveň platí, že nemusí existovať pre daný záznam o kontrole žiadny záznam typu Similarity. Tento prípad nastáva logicky v momente, keď je kontrolovaný súbor unikátny a služba Odevzdej.cz sa nepodarí k nemu nájsť žiadne dokumenty, ktoré by s ním boli podobné.

Pri navrhovaní vzťahov a celkovo pri návrhu systému a jeho štruktúry som sa snažil zachovať čo možno najviac vzťahov, ktoré boli použité v rámci starého modulu Antiplagiát kvôli tomu, že sa jedná o modul s v podstate identickou funkcionalitou.

Týmto boli podľa mojej mienky v stručnosti vysvetlené všetky vzťahy nutné pre pochopenie princípu, na ktorom sú procesy existujúce v tomto module založené.

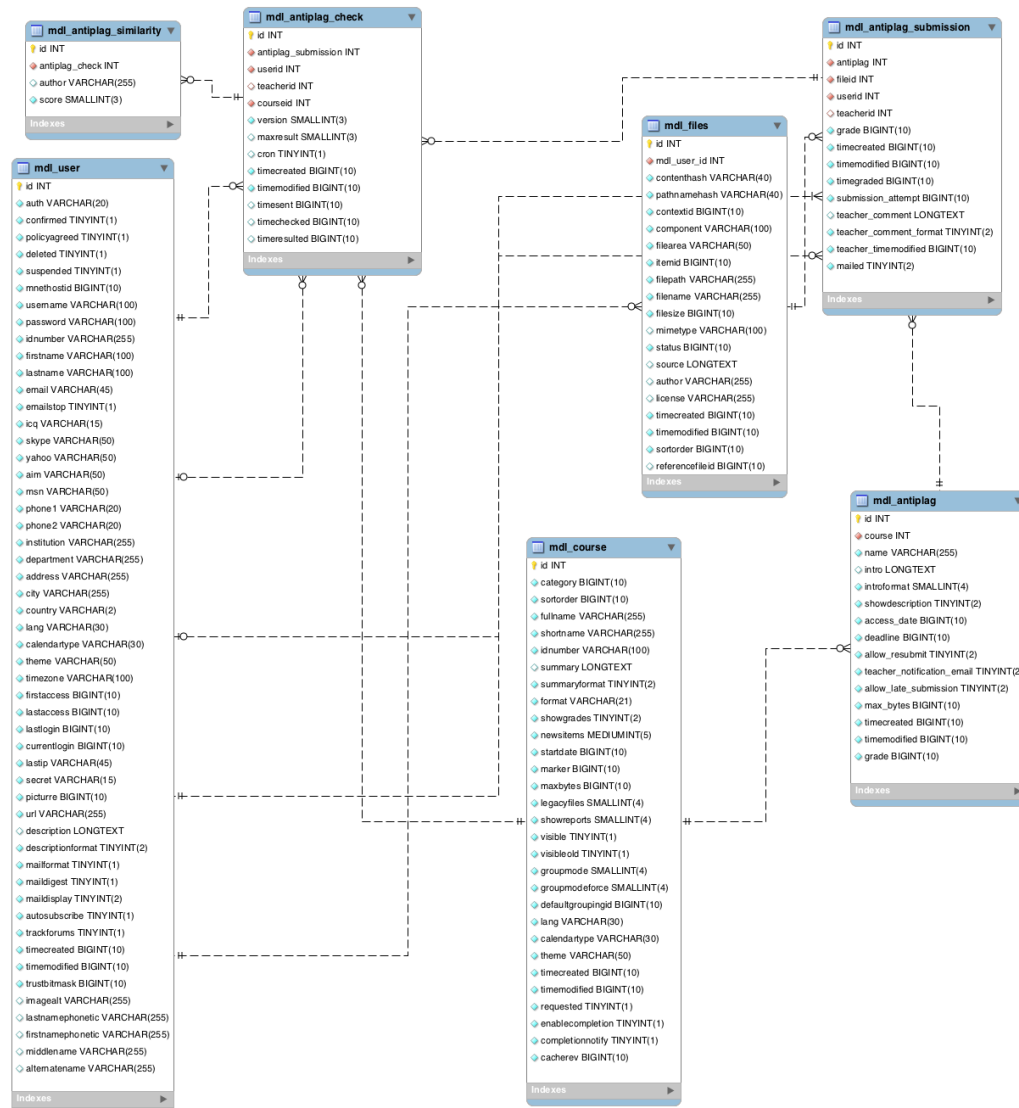
Databázová vrstva navrhovaného modulu Antiplagiát a vlastne aj celého e-learningového systému Moodle je realizovaná pomocou jazyka SQL. V tomto konkrétnom prípade je využívaný databázový systém MySQL. Databáza samotného systému Moodle obsahuje niečo vyše 250 databázových tabuliek. Mnou navrhovaný modul pridáva do tejto databázy len štyri databázové tabuľky:

1. **mdl_antiplag** - databázová tabuľka reprezentujúca inštanciu modulu Antiplagiát
2. **mdl_antiplag_submission** - databázová tabuľka reprezentujúca záznam o odovzdaní semestrálnej práci, informácie o známke a prípadný komentár od vyučujúceho
3. **mdl_antiplag_check** - databázová tabuľka reprezentujúca záznam o kontrole semestrálnej práce službou Odevzdej.cz
4. **mdl_antiplag_similarity** - databázová tabuľka reprezentujúca záznam o dokumente podobnom s testovaným dokumentom

Rovnako ako pri konceptuálnom modeli, aj tu som sa rozhodol zobrazit v diagrame ďalšie tri tabuľky, ktoré sú súčasťou databázy systému Moodle,

ale mnou navrhovaný modul má na ne väzby a pre lepšiu predstavu o štruktúre databázy sú podľa môjho názoru dôležité. Jedná sa o tabuľky mdl_files, mdl_course a mdl_user. Na rozdiel od konceptuálneho modelu sa nebudem venovať popisu a vysvetľovaniu významu väzieb a vzťahov medzi tabuľkami v databáze, pretože popis väzieb pri konceptuálnom modeli považujem za dostatočný pre pochopenie fungovania modulu.

Štruktúru databázy navrhovaného modulu spolu s väzbami na niektoré databázové tabuľky systému Moodle (podobne ako pri konceptuálnom modeli) môžete vidieť na obrázku číslo 2.3



Obr. 2.3: Databázový model modulu Antiplagiát

2.9 Analýza systému z užívateľského hľadiska

V tejto sekcii sa budem venovať popisu modulu Antiplagiát z užívateľského hľadiska. Keďže v systéme Moodle a takisto aj v mnou navrhnutom a následne naimplementovanom module sú právomoci rozdelené na základe užívateľských rolí, bude aj tento popis rozdelený na popis z pohľadu týchto rolí. V module sa používajú dve základné užívateľské role:

1. študent
2. vyučujúci

Samozrejme, v systéme je zadaných viac užívateľských rolí, ale pre účely tejto analýzy nám úplne postačia vyššie zmienené role. Tieto a pravdaže aj ostatné užívateľské role prítomné a využívané v systéme popíšem v kapitole venujúcej sa popisu implementácie modulu Antiplagiát.

Podľa požiadaviek stanovených po preskúmaní pôvodnej verzie modulu Antiplagiát, vytvorenej vedúcim tejto diplomovej práce, modul obsahuje tieto dve základné sekcie:

- sekcia pre odovzdávanie nových semestrálnych prác
- sekcia pre správu odovzdaných semestrálnych prác

Po otvorení konkrétnej aktivity modulu Antiplagiát v rámci nejakého kurzu v systéme Moodle, sa užívateľ dostáva na obrazovku s detailnými informáciami o tejto aktivite. Oba typy užívateľov tu nájdu informácie o názve aktivity, jej popise, ktorý môže byť napríklad podrobnejšou špecifikáciou zadania požadovanej semestrálnej práce a informácie o dátume začiatku a konca možnosti odovzdávať tieto semestrálne práce. Užívateľ v roli klasického študenta okrem toho navyše vidí formulár pre nahranie svojho vypracovaného projektu a v prípade, že jeho projekt bol už ohodnotený, nájde tu informácie o hodnotení svojho projektu spolu s prípadným komentárom od vyučujúceho. Táto obrazovka je v podstate jediná obrazovka, na ktorej má študent v roli študenta možnosť vykonávať nejaké akcie v rámci tohto modulu. Užívateľ v roli vyučujúceho vidí okrem informácií spoločných pre obe užívateľské role navyše odkaz vedúci na obrazovku s prehľadom všetkých odovzdaných semestrálnych prác.

Ďalšie popisované obrazovky a sekcie sú určené výhradne pre užívateľov v roli vyučujúceho, pretože ako som už spomenul, užívateľ v roli klasického študenta má pre svoju prácu k dispozícii len jednu už popísanú obrazovku. Nasledujúcou sekciou, ktorú by som chcel popísať je sekcia pre správu odovzdaných semestrálnych prác. Obrazovka s prehľadom odovzdaných semestrálnych prác slúži vyučujúcim hlavne ako možnosť oznámkovať jednotlivé semestrálne práce, prípadne k nim napísať nejaký komentár ak je to potrebné. Táto sekcia ponúka dve možnosti, ktoré môže vyučujúci použiť pre oznámkovanie semestrálnej práce, prípadne pre pridanie nejakého komentára k danej práci:

1. Výber známky z ponuky a pridanie komentáru priamo v tabuľke so semestrálnymi prácami
2. Výber známky z ponuky a pridanie komentáru prostredníctvom samostatného okna

Každá z týchto dvoch možností je vhodná pre iné situácie. Prvá možnosť umožňuje oznámkovať a okomentovať viacero semestrálnych prác naraz. Vyučujúci si z ponuky v riadku tabuľky pri danej práci vyberie známku, prípadne pridá jednoduchý komentár. Tento postup opakuje pri každej semestrálnej práci, ktorú chce ohodnotiť. Nakoniec prostredníctvom tlačítka umiestneného pod tabuľkou uloží všetky svoje hodnotenia. Nevýhodou je skutočnosť, že táto možnosť neumožňuje žiadne pokročilejšie formátovanie textu v komentároch.

Druhá z popisovaných dvoch možností známkovania spočíva v otvorení samostatného okna, kde vyučujúci znovu vidí možnosť vybrať z ponuky známku, skontrolovať, či je práca plagiátom prostredníctvom služby Odevzdej.cz a pridanie komentáru. Táto varianta ponúka pokročilejšie možnosti formátovania textu pre komentár. Táto funkcia je sprostredkovaná použitím WYSIWYG editoru s názvom CKEditor. Druhá možnosť je teda vhodné použiť najmä v prípadoch, že vyučujúci chce pridať obsiahlejší komentár, ovplyvniť formátovanie textu v komentári a zariadiť tak jeho lepší vzhľad a väčšiu prehľadnosť.

Asi najdôležitejšou funkciou, ktorá je prístupná z tejto obrazovky a zároveň je dôvodom pre vznik modulu Antiplagiát je možnosť poslať požiadavku na kontrolu plagiátorstva v semestrálnej práci službou Odevzdej.cz.

Tak ako aj pri známkovaní, aj pri kontrole semestrálnej práce je viac možností, ako túto kontrolu vykonať. Opäť je prítomná možnosť hromadnej kontroly, kde vyučujúci označí semestrálne práce, v ktorých chce skontrolovať plagiátorstvo prostredníctvom služby Odevzdej.cz a následne vyberie z ponuky možnosť odoslania súborov na kontrolu.

Druhou možnosťou je kliknúť na odkaz pre spustenie kontroly, ktorý je umiestnený vždy pri danej semestrálnej práci. Pri tejto možnosti sa užívateľovi zobrazí obrazovka s informáciami o autorovi semestrálnej práce a bližšími informáciami o odovzdanej semestrálnej práci. V prípade, že kontrola už prebehla, táto obrazovka obsahuje navyše informácie o výsledku kontroly plagiátorstva v danej semestrálnej práci. Okrem informácií o podobných dokumentoch a percentuálnej hodnote podobnosti s týmito dokumentami uvidí užívateľ odkaz na dokument, ktorý je výstupom kontroly plagiátorstva službou Odevzdej.cz a obsahuje podrobné informácie o všetkých nájdených podobnostiach kontrolovanej semestrálnej práce.

Tento popis modulu z užívateľského hľadiska by mal byť dostatočný pre získanie nejakej konkrétnejšej predstavy o tom, čo všetko prostredie modulu Antiplagiát užívateľom ponúka. Týmto popisom by som zároveň chcel ukončiť kapitolu venujúcu sa analýze mnou navrhovaného a následne naimplementovaného modulu Antiplagiát pre e-learningový systém Moodle FIT.

Realizácia

V tejto kapitole by som rád popísal mnou navrhovaný modul Antiplagiát z hľadiska implementácie. Okrem všeobecného popisu implementácie celého modulu Antiplagiát, sa zameriam v prvom rade na neštandardné časti modulu, pri ktorých je dôležité popísať podrobnejšie samotnú implementáciu, alebo sú niečím zaujímavé a stojí za to venovať sa popisu ich implementácie.

Po dlhšej úvahe som pre podrobnejší popis implementácie vybral tieto časti modulu Antiplagiát:

- Užívateľské role a práva
- Prehľad odovzdaných semestrálnych prác
- Stiahnutie odovzdaných semestrálnych prác
- Komunikácia so službou Odevzdej.cz a proces kontroly práce

Popis implementácie poslednej vybranej časti bude asi najobsiahlejší, pretože komunikácia so službou Odevzdej.cz je kľúčová funkcionálna časť modulu Antiplagiát.

3.1 Všeobecný popis fungovania modulu z hľadiska implementácie

V tejto časti sa zameriam na veľmi stručný popis princípu, na ktorom je postavené fungovanie celého systému z implementačného hľadiska.

Implementácia celého modulu bola uskutočnená v súlade s oficiálnymi pravidlami a odporúčaniami pre tvorbu modulov v rámci e-learningového systému Moodle. Implementácia logiky týkajúcej sa modulov všeobecne je obsiahnutá v súbore lib.php. Tu sa nachádza implementácia funkcií, ktoré od modulu vyžaduje samotný systém Moodle. Logika, ktorá je špecifická pre konkrétny

modul Antiplagiát je naimplementovaná v súbore locallib.php v triede s názvom antiplag. Každá z obrazoviek, s ktorými prichádzajú užívatelia v module do styku má svoj vlastný súbor obsahujúci nejaké základné nastavenie atribútov zobrazovanej stránky a spracovanie prípadných parametrov. V každom z týchto súborov je následne použitá metóda view triedy antiplag, ktorá ako argument prijíma názov sekcie, ktorá sa má zobraziť. Táto metóda na základe prijatého argumentu následne určí, aký obsah sa užívateľovi skutočne zobrazí.

3.2 Užívateľské role a práva

V systéme Moodle FIT sú prítomné tieto užívateľské role:

- **Student** - študenti zvyčajne majú v rámci kurzu menej oprávnení
- **Vedúci tímu** - vedúci tímu študentov
- **Asistent** - asistenti bez práva upravovať môžu učiť v kurzoch a známkovať študentov, nemôžu však meniť činnosti
- **Vyučujúci** - vyučujúci môžu v rámci kurzu robiť v podstate všetko, vrátane zmien aktivít a známkovania študentov
- **Garant kurzu** - garanti kurzu môžu v rámci kurzu robiť v podstate všetko, vrátane zmien aktivít a známkovania študentov. Navyše môžu meniť nastavenia jednotlivých kurzov.
- **Manager** - manažéri môžu pristupovať ku kurzom a upravovať ich, väčšinou sa sami však žiadneho kurzu nezúčastňujú
- **Superadministrátor** - superadministrátori, alebo správcovia zvyčajne môžu na stránkach a v kurzoch robiť čokoľvek

Tieto užívateľské role reprezentujú archetypy používané pri definícii jednotlivých oprávnení. Implementácia mnou navrhnutého a následne naimplementovaného a riadne otestovaného modulu Antiplagiát pre systém Moodle FIT pridáva do systému štyri nové oprávnenia:

1. **mod/antiplag:addinstance** - právo pre vytvorenie novej inštancie modulu Antiplagiát v rámci systému Moodle
2. **mod/antiplag:view** - právo prístupu do vytvorenej inštancie modulu Antiplagiát v rámci systému Moodle
3. **mod/antiplag:grade** - oprávnenie známkovať projekty, prezerať si udeľené známky a vykonávať proces kontroly plagiátorstva v semestrálnych prácach
4. **mod/antiplag:submit** - oprávnenie odovzdať novú semestrálnu prácu

Definícia jednotlivých oprávnení sa nachádza v adresári db v súbore access.php. Pri každej definícii oprávnenia je uvedený typ oprávnenia. Hodnota typu určuje, či sa jedná o právo pre čítanie, respektíve prezeranie obsahu obmedzeného týmto oprávnením, alebo o právo pre zápis, respektíve editáciu obsahu, ku ktorému sa vzťahuje povinnosť disponovať týmto oprávnením. Ďalej je tu definovaná úroveň kontextu, v ktorom sa nachádza toto oprávnenie. Môže sa jednať o tieto tri úrovne kontextu:

1. Oprávnenie v kontexte na úrovni modulu
2. Oprávnenie v kontexte na úrovni kurzu
3. Oprávnenie v kontexte na úrovni systému

Oprávnenie na úrovni modulu v praxi znamená, že užívateľ môže vykonávať akcie, ku ktorým sa dané oprávnenie vzťahuje len v rámci konkrétnej inštancie modulu a k ostatným inštanciam nemá prístup. Pri oprávnení na úrovni kurzu môže užívateľ vykonávať týmto oprávnením povolené akcie v rámci celého kurzu, to znamená v rámci ktorejkoľvek inštancie modulu Antiplagiát vytvorenej v danom kurze. Ako je už asi jasné, pri oprávnení na úrovni systému môže užívateľ disponujúci týmto oprávnením vykonávať jednotlivé akcie v rámci celého systému, teda v každom kurze, do ktorého má prístup. Okrem toho sú v súbore access.php definované jednotlivé už spomínané archetypy vzťahujúce sa k užívateľským rolám.

Skoro všetky akcie vykonávané v rámci modulu Antiplagiát sa na tieto oprávnenia odkazujú, každá z nich vždy v úvode kontroluje, či má užívateľ, ktorý sa pokúša o vykonanie danej akcie skutočne pridelené oprávnenie potrebné pre jej úspešné vykonanie. Každý užívateľ má totiž priradenú jednu, alebo viacero užívateľských rolí. Tým pádom je možné jednoznačne na základe archetypu, ktorý sa viaže k danej užívateľskej roli určiť, či je požadované oprávnenie definované pre tento archetyp. V prípade zistenia systému, že užívateľ sa pokúša o vykonanie akcie, na ktorú nemá právo, systém Moodle zobrazí chybovú hlášku, kde informuje užívateľa o tom, že pre vykonanie danej akcie nemá dostatočné práva. Väčšinou tiež zobrazuje informáciu o tom, aká užívateľská rola disponuje požadovanými oprávneniami.

Pri pokuse o zovšeobecnenie kvôli lepšej predstave môžeme prehlásiť, že v rámci implementácie modulu Antiplagiát sa pracuje s dvomi užívateľskými rolami, ktorými sú študent a vyučujúci. Študent typicky disponuje len oprávnením pre prístup k vytvorenej inštancii modulu Antiplagiát a oprávnením pre odovzdávanie súborov obsahujúcich jeho semestrálne práce v rámci tejto inštancie modulu.

3.3 Prehľad odovzdaných semestrálnych prác

Dôležitou súčasťou modulu Antiplagiát a zároveň sekciou, ktorá je v rámci modulu asi najviac využívaná je sekcia s prehľadom odovzdaných semestrálnych prác. V rámci tejto sekcie sú zobrazovaní aj užívatelia, ktorí zatiaľ nemajú odovzdanú svoju semestrálnu prácu, ale hlavnou funkciou tejto stránky je práve manipulácia s už odovzdanými prácami. Keďže väčšina akcií vykonávaných v module je vykonávaná práve v prostredí tejto obrazovky, považujem za dôležité popísať implementáciu tejto časti.

Implementácia základného nastavenia stránky s prehľadom semestrálnych prác je umiestnená v súbore `submissions.php`. V tomto súbore je zavolaná metóda `view` triedy `antiplag` s príslušným argumentom, ktorý určuje, že sa má pre zobrazenie obsahu stránky použiť metóda `view_submissions_table`. Hneď v úvode tejto metódy je vyžadované, aby užívateľ pokúšajúci sa prístupit k obsahu stránky mal pridelené oprávnenie `“mod/antiplag:grade”`. To znamená, že táto sekcia je prístupná len pre vyučujúcich, ktorí majú oprávnenie známkovať semestrálne práce. V prípade, že užívateľ takýmto oprávnením nedisponuje, systém Moodle vypíše chybovú hlášku s informáciami o tom, že na zobrazenie požadovaného obsahu nemá potrebné oprávnenie.

Samotný obsah, ktorý tvoria jednotlivé odovzdané semestrálne práce spolu s informáciami o nich je zobrazovaný vo forme tabuľky. V tomto prípade som sa po preštudovaní dokumentácie k systému Moodle rozhodol použiť tabuľku reprezentovanú triedou `flexible_table`, ktorá je súčasťou systému Moodle, konkrétne knižnice `tablelib`. Tento typ tabuľky poskytuje veľké množstvo funkcií a mne tým značne zjednodušil implementáciu, pretože inak by som využité funkcie musel implementovať sám a okrem iného by sa tým pravdaže zvýšila časová náročnosť.

Trieda `flexible_table`, ktorá reprezentuje tabuľku použitú pre zobrazenie obsahu popisovanej sekcie disponuje okrem iného týmito funkciami:

- možnosť definovania stĺpcov, podľa ktorých majú byť dáta umiestnené v tabuľke zoradené
- možnosť nastaviť stĺpce tabuľky ako `“collapsible”`, čo v preklade znamená, že sa jednotlivé stĺpce dajú v prípade potreby skrývať
- zdefinovanie atribútov ovplyvňujúcich vzhľad jednotlivých elementov tabuľky
- stránkovanie záznamov v tabuľke a s tým spojené nastavenie maximálneho počtu záznamov, ktoré je možné zobrazit na jednej stránke
- filtrovanie záznamov v tabuľke
- definícia vlastných SQL príkazov použitých pre zoradenie záznamov v tabuľke

- možnosť definovať obsah v stĺpcoch ako obsah určený na stiahnutie spolu s pridaním tlačítka pre samotné stiahnutie
- automatické vypisovanie informácie o prázdnom obsahu tabuľky v prípade, že nie sú dostupné žiadne dáta

Práve kvôli týmto vymenovaným možnostiam, ktoré trieda `flexible_table` poskytuje bol pre účely tejto práce ten typ tabuľky zvolený ako najvhodnejší. Niektoré z týchto funkcií sú zložitejšie na implementáciu, a preto som samozrejme uvítal možnosť využiť túto triedu poskytovanú priamo systémom Moodle pre implementáciu popisovanej sekcie modulu Antiplagiát.

Vďaka triede `flexible_table` si môže užívateľ jednoducho meniť, podľa akého stĺpca sa majú dáta zoradiť a zvoliť si, či majú byť zoradené vzostupne, alebo zostupne. Veľkou výhodou je spomínaná podpora stránkovania. Pri väčšom počte užívateľov sa výpis stáva neprehľadným, preto je lepšie v prípade, že počítame s podobným množstvom dát použiť stránkovanie a obmedziť počet v jednej chvíli zobrazovaných záznamov na nejaké rozumné číslo. Značne sa tým zvyšuje prehľadnosť celej stránky. Okrem toho ma príjemne prekvapila možnosť zobrazenia filtrov, ktoré dovoľujú užívateľovi vyfiltrovať si záznamy podľa určitého kritéria, v tomto prípade podľa začiatočného písmena krstného mena, alebo priezviska.

Tabuľka s obsahom popisovanej sekcie obsahuje tieto stĺpce:

- profilový obrázok užívateľa
- krstné meno a priezvisko
- známka
- komentár
- posledná zmena vykonaná študentom
- posledná zmena vykonaná vyučujúcim
- plagiát (maximálna zhoda)
- výber
- stav hodnotenia
- výsledná známka

Ďalšou časťou implementácie tejto sekcie je získanie samotných dát, ktoré sa majú užívateľovi skutočne zobraziť. Keďže chceme zobraziť v tabuľke všetkých užívateľov bez ohľadu na to, či už svoju semestrálnu prácu odovzdali, bolo potrebné ošetriť, aby sa do tabuľky dostali aj záznamy o užívateľoch,

3. REALIZÁCIA

ktorí práce zatiaľ neodovzdali. Keďže v celej tabuľke sa pracuje so záznamami o odovzdaní tejto práce a pre týchto užívateľov žiadny podobný záznam neexistuje, rozhodol som sa pre zjednodušenie implementovať metódu `create_submission_if_not_exists`. Táto metóda vytvorí záznam o odovzdaní semestrálnej práce s nastavením len minima informácií. Nejedná sa tak o skutočné odovzdanie. Od ostatných záznamov sa takýto záznam líši hlavne tým, že neobsahuje identifikátor fyzického súboru s obsahom semestrálnej práce, ktorý logicky nemôže bez skutočného odovzdania existovať. Záznam o takomto užívateľovi sa teda zobrazí v tabuľke spolu s ostatnými, ale z hľadiska jednotlivých stĺpcov tabuľky pre tento záznam je jednoznačne vidieť, že sa jedná o typ užívateľa, ktorý zatiaľ svoju prácu neodovzdal. Skutočné odovzdanie semestrálnej práce nepredstavuje potom samozrejme žiadny problém. Jediným rozdielom oproti klasickému prípadu je to, že sa v tomto prípade nevytvorí nový záznam o odovzdaní semestrálnej práce, ale aktualizuje sa ten, ktorý bol vytvorený kvôli zahrnutiu užívateľa do výpisu v tabuľke.

Okrem existencie záznamu o odovzdaní semestrálnej práce ovplyvňuje to, ktoré dáta sa majú skutočne zobraziť ešte niekoľko faktorov. Jedným z týchto faktorov je maximálny počet záznamov, ktorý je možné na jednej stránke zobrazovať a číslo aktuálnej stránky. Tým sa určí rozsah záznamov, ktoré sa majú užívateľovi zobraziť. Samozrejme to závisí aj od spôsobu zoradenia záznamov a od použitia, či nepoužitia niektorého z filtrov. Navyše, úlohu tu zohráva aj prípadná existencia užívateľských skupín. Štandardne sa v databáze hľadajú záznamy o užívateľoch, ktorí majú právo odovzdávať semestrálne práce v rámci inštalácie modulu Antiplagiát. Ak je zvolené nastavenie, ktoré určuje, že celá práca v rámci modulu by mala prebiehať len pre určitú definovanú skupinu užívateľov, získajú sa užívatelia patriaci do požadovanej skupiny. Výslednou množinou požadovaných záznamov o užívateľoch a ich semestrálnych prácach je prienik množiny užívateľov s právom odovzdávať semestrálne práce a množiny užívateľov patriacich do povolenej skupiny užívateľov.

Následujúcim krokom implementácie popisovanej sekcie je prechádzanie získaných záznamov o užívateľoch, ktoré sa majú vyučujúcemu zobraziť. Systém prechádza tieto záznamy a postupne vytvára príslušné hodnoty pre jednotlivé stĺpce tabuľky.

Pre získanie profilového obrázku užívateľa je použitá funkcia `user_picture`, ktorá ako argumenty prijíma objekt reprezentujúci daného užívateľa a veľkosť v akej chceme obrázok zobrazovať. Meno a priezvisko užívateľa sa získava štandardným spôsobom, keďže obe hodnoty sú atribútmi objektu reprezentujúceho tohto užívateľa. Obe hodnoty sú následne spojené do odkazu, ktorý vedie na stránku s užívateľovým profilom. Stĺpce pre výber známky a vloženie komentáru sú vlastne jednoduchým formulárom. Tento formulár poskytuje vyučujúcemu možnosť takzvaného "rýchleho známkovania". Nesporne najväčšou výhodou tohto formulára je skutočnosť, že vyučujúci môže týmto spôsobom známkovať hromadne. Z ponuky pri každom užívateľovi vyberie požadovanú známku a vloží komentár. Následne všetky hodnotenia uloží. Čas poslednej

zmeny vykonanej študentom reprezentuje čas, kedy študent svoju prácu odovzdal, alebo nahral jej novšiu aktualizovanú verziu. Stĺpec s časom poslednej zmeny vykonanej vyučujúcim reprezentuje čas, kedy vyučujúci prácu oznámkoval, alebo komentoval. V stĺpci plagiát je zobrazovaný obsah závislý na procese kontroly plagiátorstva.

Tu môžu nastať tri situácie, ktoré ovplyvňujú konkrétny obsah stĺpca zobrazovaný vyučujúcemu. Tieto tri prípady sú metódou `view_submission_table` rozlišované nasledovne. Metóda skontroluje existenciu záznamu o požiadavke vykonať kontrolu plagiátorstva v danej semestrálnej práci. V prípade, že takýto záznam existuje, metóda sa ho pokúsi získať z databázovej tabuľky `mdl_antiplag_check`. Po úspešnom získaní nasleduje kontrola, či táto kontrola bola už úspešne ukončená, alebo nie. Túto skutočnosť dokáže môj algoritmus zistiť prostredníctvom hodnoty dosiahnutej maximálnej zhody. Pri existencii výsledku je táto hodnota väčšia, alebo rovná nule. Od toho závisí, ktorá možnosť sa vyučujúcemu zobrazí. Jednou možnosťou je, že vyučujúci uvidí odkaz s informáciou o tom, že kontrola stále prebieha. Po kliknutí na tento odkaz je znovu spustená komunikácia so službou `Odevzdej.cz` za účelom zistenia existencie výsledku kontroly. Druhou možnosťou je, že kontrola už úspešne prebehla. V tomto prípade vyučujúci uvidí informáciu o percentuálnej hodnote reprezentujúcej maximálnu zhodu s nejakým porovnávaným dokumentom a odkaz, ktorý vyučujúceho zavedie na stránku s podrobnými informáciami o výsledku kontroly. V prípade, že záznam neexistuje, vyučujúcemu sa zobrazí odkaz s textom, ktorý ho informuje, že po kliknutí na daný odkaz bude spustená komunikácia so službou `Odevzdej.cz` a odoslaná požiadavka na skontrolovanie konkrétnej práce. Podrobnejšiemu popisu fungovania komunikácie so službou `Odevzdej.cz` a celkovo popisu venovanému implementácii celého procesu kontroly plagiátorstva v semestrálnych prácach popisujem v samostatnej sekcii, ktorá vznikla práve za týmto účelom.

Stĺpec výber umožňuje vyučujúcemu označiť riadok tabuľky s prácou, ktorá ešte nebola skontrolovaná. Takto môže označiť ľubovoľný počet riadkov a následne ich hromadne zaslať do služby `Odevzdej.cz` na kontrolu plagiátorstva. Ďalším stĺpcom je stĺpec so stavom hodnotenia. Tu je prítomný len jeden odkaz, ktorý vyučujúcemu otvorí samostatné okno s pokročilými možnosťami známkovania a komentovania konkrétnej práce. Tento odkaz a vlastne celé okno je definovaný priamo v tele metódy pomocou funkcie `action_link`, ktorá ako argumenty prijíma adresu odkazu, text odkazu a akciu, ktorá sa má vykonať po otvorení nového okna. V poslednom stĺpci vyučujúci vidí výslednú známku, ktorá bola užívateľovi udelená. Táto hodnota je metódou získaná z takzvanej “knihy známok”.

Prostredie systému Moodle FIT a konkrétne prostredie tejto popisovanej sekcii môžete vidieť na obrázku číslo 3.1.

3.4 Stiahnutie odovzdaných semestrálnych prác

Ďalšou zaujímavou funkcionalitou modulu Antiplagiát, ktorá podľa môjho názoru stojí za zmienku je určite možnosť stiahnuť všetky odovzdané semestrálne práce.

Podľa mojej mienky situácia, keď vyučujúci potrebuje stiahnuť odovzdané semestrálne práce môže teoreticky nastať dosť často. Vyučujúci si môže chcieť tieto práce uchovať napríklad pre účely daného predmetu ako vzorové práce. Okrem toho kontrola plagiátorstva v týchto prácach nie je samozrejme jediným kritériom pre udelenie známky. Vyučujúci určite kontroluje odovzdané práce aj osobne po obsahovej stránke a na základe tejto kontroly a výsledku kontroly plagiátorstva potom určuje výslednú známku. Všetky práce je samozrejme možné stiahnuť priamo na stránke s prehľadom odovzdaných semestrálnych prác. Takéto sťahovanie je však pri väčšom počte študentov nesmierne zdĺhavé a pravdaže tým pádom neefektívne.

Z týchto popisovaných dôvodov som sa rozhodol poskytnúť vyučujúcim možnosť stiahnuť si všetky odovzdané semestrálne práce hromadne, len jediným jednoduchým kliknutím. Za účelom naimplementovania tejto funkcionality som vytvoril funkciu `download_all_submissions`.

Hneď v úvode tejto funkcie sa pre istotu zvyšuje časový limit serveru, ktorý určuje ako dlho môže byť nejaký program spracovávaný. Je to z toho dôvodu, že v prípade skutočne veľkého množstva študentov je istá pravdepodobnosť, že celá príprava všetkých požadovaných súborov do podoby vhodnej pre stiahnutie by mohla zabráť viac času, než je štandardne povolené.

Druhým krokom tejto implementácie je získanie záznamov o užívateľoch v rámci konkrétneho kontextu v systéme Moodle, ktorí majú právo odovzdávať semestrálne práce. Následne funkcia prechádza tieto záznamy a zisťuje, či daný užívateľ má evidovanú nejakú odovzdanú semestrálnu prácu. V prípade, že áno, funkcia sa pokúsi z úložiska súborov systému Moodle získať konkrétny odovzdaný súbor obsahujúci evidovanú prácu a vloží ho do predpripraveného pola určeného práve pre tieto súbory. Pre každý z týchto súborov je vytvorený unikátny názov, aby bolo možné jednoznačne priradiť daný súbor ku konkrétnemu užívateľovi.

Po prejdení všetkých záznamov o užívateľoch a získaní ich prípadných odovzdaných semestrálnych prác sa implementácia tejto funkcionality dostáva do záverečnej fázy. V tejto fáze funkcia overí, či boli skutočne nájdené a získané nejaké súbory, ktoré by mohli byť poskytnuté na stiahnutie. V prípade, že pole vytvorené pre uchovávanie týchto súborov je prázdne, dostane sa užívateľ, v tomto prípade vyučujúci na stránku s informáciou o tom, že k dispozícii nie sú žiadne odovzdané semestrálne práce. V opačnom prípade, sú tieto súbory zaarchivované do jedného výstupného súboru s príponou `zip` a odoslané užívateľovi, ktorý si tento súbor môže stiahnuť.

3.5 Komunikácia so službou Odevzdej.cz a proces kontroly práce

Komunikácia so službou Odevzdej.cz, ktorá v mnou navrhovanom a následne naimplementovanom module Antiplagiát pre e-learningový systém Moodle FIT zabezpečuje kontrolu plagiátorstva v odovzdaných semestrálnych prácach je hlavnou funkciou celého tohto modulu. Zároveň táto funkcionality bola príčinou vzniku potreby vytvoriť samotný modul Antiplagiát a teda aj príčinou vzniku mojej diplomovej práce. Hlavne z toho dôvodu si myslím, že popis samotnej implementácie komunikácie s touto službou je dôležitý a kvôli tomu som sa ho rozhodol zahrnúť do svojej práce.

V rámci kapitoly venovanej podrobnej analýze celého modulu, všetkých funkčných i nefunkčných požiadaviek, systému Moodle a služby Odevzdej.cz som už túto službu popísal a spomenul som aj spôsob, ktorým je možné s ňou komunikovať. Môj modul teda používa pre komunikáciu knižnicu libcurl, bez ktorej by žiadna komunikácia so službou nebola možná.

V pôvodnej verzii modulu Antiplagiát bola implementácia komunikácie so službou Odevzdej.cz a dokonca implementácia väčšiny častí celého modulu trochu neprehľadná a z mnohých hľadísk aj neefektívna. Prekvapilo ma napríklad, že pri jednom type požiadavky bola použitá knižnica libcurl a pri druhom type sa zas spúšťal príkaz, ktorý sa normálne spúšťa z príkazového riadku. Rozhodol som sa zjednotiť implementáciu všetkých čiastkových úkonov komunikácie so službou. Okrem toho som celú túto časť implementácie oddelil od zvyšku systému pre zvýšenie prehľadnosti kódu a zabezpečenie nejakej rozumnej logickej štruktúry modulu.

Celá implementácia logiky, ktorá stojí za komunikáciou so službou Odevzdej.cz sa nachádza v triede s názvom `antiplag_check_manager`. Metódy tejto triedy sú využívané pre zabezpečenie celého procesu kontroly plagiátorstva v odovzdaných semestrálnych prácach. V implementácii tejto časti modulu v pôvodnej verzii modulu Antiplagiát som si všimol, že vo všetkých metódach využívaných pri komunikácii sa zbytočne opakuje takmer identický kód zamyslel som sa nad tým, či je niečo také skutočne potrebné. Keďže všetky časti komunikácie so službou v tomto module využívajú v podstate úplne identické cURL príkazy, vytvoril som metódu `do_curl_command`, ktorá ako argumenty dostáva adresu konkrétnej časti služby, cestu k výstupnému súboru, kde sa po vykonaní príkazu zapisuje jeho výstup a dáta, ktoré pri komunikácii službe zasiela. Tieto tri argumenty sú skutočne jediné odlišnosti medzi použitými príkazmi. Vytvorenie tejto metódy a jej používanie ostatnými metódami som zvolil z dôvodu, že som chcel zabezpečiť, aby sa zbytočne neopakoval identický kus kódu vo viacerých metódach, pretože je to proti jednej z hlavných zásad správneho programovania a je to aj strašne neefektívne.

Pred začatím každého pokusu o komunikáciu so službou Odevzdej.cz je potrebné autorizovať svoj prístup voči tejto službe. Na to sa používa jedno-

duchý cURL príkaz, ktorý zasiela službe identifikačné číslo užívateľa, alebo v tomto prípade platiacej inštitúcie, ktorá sa pokúša o komunikáciu so službou a heslo. Po vykonaní tohto autorizačného príkazu je možné ľubovoľne komunikovať so službou. Tento príkaz sa v mojej implementácii nachádza v metóde s názvom `authorize`.

Prvým krokom celého procesu kontroly plagiátorstva v semestrálnych prácach je hneď po autorizovaní nainportovanie súboru obsahujúceho semestrálnu prácu, ktorú chceme skontrolovať do systému. Import súboru funguje na princípe vytvorenia kontrolného XML súboru, ktorý okrem iného obsahuje odkaz na verejne dostupný súbor určený pre kontrolu a jeho následné zaslanie službe Odevzdej.cz. Služba si ho potom stiahne, vytvorí z neho textovú verziu a tú porovnáva s dokumentami, ktoré má uložené v databázach využívaných pri hľadaní podobností v nainportovaných dokumentoch. Pred samotným importom je teda nutné takýto kontrolný súbor vygenerovať. Ešte predtým je však potrebné zabezpečiť, aby bol súbor určený pre kontrolu skutočne dostupný z internetu a služba si ho mohla stiahnuť a skontrolovať, či sa nejedná o plagiát. Pri inom systéme by to nebol vôbec žiadny problém a súbor by sa okamžite pri nahrávaní uložil do adresára, z ktorého by bol prístupný. Avšak systém Moodle pristupuje k nahrávaniu súborov trochu inak.

Pri nahraní súboru v rámci systému Moodle si systém tento súbor uloží do adresára `moodledata` pod zakódovaným názvom. Tento adresár nie je prístupný z internetu a v prípade, že je potrebný, systém Moodle zabezpečuje prístup k nemu prostredníctvom jednej z funkcií. Je to samozrejme bezpečný spôsob a za normálnych okolností by som k nemu nemal absolútne žiadne výhrady, ale v tomto prípade bolo naozaj potrebné, aby si daný súbor služba Odevzdej.cz mohla bez najmenších problémov stiahnuť. Preto ešte pred vygenerovaním kontrolného súboru a jeho následným zaslaním službe bolo potrebné sprístupniť kontrolovaný súbor. Toto zabezpečuje mnou vytvorená metóda `get_uploaded_file_public_url`. Tá prijíma ako argument záznam o súbore, ktorý sa nachádza v databáze systému Moodle. Pomocou tohto databázového záznamu je následne lokalizovaný súbor a jeho obsah je skopírovaný do adresára, ktorý je prístupný z internetu. Odkaz na takto dostupný súbor je potom použitý v metóde pre vygenerovanie kontrolného XML súboru.

Pre generovanie kontrolného XML súboru sa používa metóda triedy `antiplag_check_manager` s názvom `generateXML`. Tá dostáva ako argument už spomínaný odkaz na kontrolovaný súbor. Pre vytvorenie tohto kontrolného XML súboru som sa rozhodol využiť triedu `SimpleXMLElement`, ktorá sa v jazyku PHP často používa práve pre prácu s týmito typov súborov, keďže jej hlavnou úlohou je práve reprezentovať element XML dokumentu. Kód, ktorý bol potrebný pre vytvorenie požadovaného XML súboru je približne o polovicu kratší ako kód, ktorý bol za týmto účelom použitý pri generovaní kontrolného súboru v pôvodnej verzii modulu `Antiplagiát`. Okrem toho považujem využitie triedy `SimpleXMLElement` za efektívnejší, prehľadnejší a čistejší spôsob implementácie požadovanej funkcionality. Takto vygenerovaný XML súbor je

3. REALIZÁCIA

následne uložený do adresára s názvom import. Príklad štruktúry tohto kontrolného XML súboru, ktorý sa používa pri požiadavke o kontrolu práce službou Odevzdej.cz môžete vidieť na obrázku číslo 3.2.

Rád by som teraz aspoň stručne popísal význam jednotlivých parametrov, ktoré tento kontrolný súbor obsahuje, pretože samotný obrázok so zachytením štruktúry tohto súboru je podľa môjho názoru bez ďalšieho popisu úplne zbytočný:

- **sender.id** - identifikátor odosielateľa
- **textworks.id** - identifikátor odovzdávateľne, kam je súbor naimportovaný
- **title** - názov odovzdávateľne
- **subject** - názov platiacej inštitúcie, pod ktorou sa so službou komunikuje
- **teacher** - informácie o vyučujúcom (identifikátor, meno, e-mail)
- **creator** - informácie o autorovi (identifikátor, meno, e-mail)
- **get.file** - odkaz na kontrolovaný súbor a identifikátor autora
- **type.download** - príznak toho, či sa má súbor pridať, alebo prepísať, ak už existuje

Pre všetky súbory, či už sa jedná o samotný súbor určený na kontrolu, alebo o generovaný kontrolný XML súbor sa vytvárajú unikátne názvy, vďaka ktorým je možné vždy jednoznačne priradiť daný súbor ku konkrétnemu užívateľovi, k záznamu o semestrálnej práci, či k záznamu o kontrole plagiátorstva v tejto práci. Všetky parametre používané pri vytváraní takéhoto unikátneho názvu sú súčasťou poľa parametrov, ktoré sa pri vytváraní inštancie trieda `antiplag_check_manager` posielajú do konštruktoru triedy, prípadne je ich možné neskôr nastaviť, alebo zmeniť využitím na to určenej metódy s názvom `set_param`.

Keď už je všetko pripravené pre samotný import súboru pre službu Odevzdej.cz, použije sa metóda `file_import`, ktorá pošle požiadavku službe spolu s už popisovaným kontrolným XML súborom. Zároveň sa tiež zavolá metóda `create_new_check_record`, ktorá zabezpečuje vytvorenie záznamu o kontrole odovzdanej semestrálnej práce a jeho uloženie do databázy. Tým je prvá časť procesu kontroly ukončená. Nasleduje čakanie, kým prebehne celý proces skontrolovania odovzdanej semestrálnej práce na strane služby Odevzdej.cz. Tento proces trvá zvyčajne jeden až dva dni, v závislosti na počte požiadaviek, čakajúcich na spracovanie.

Ďalším krokom v procese kontroly plagiátorstva v semestrálnych prácach je získanie výsledku kontroly od služby Odevzdej.cz po dokončení procesu kontroly na strane služby. Kontrola výsledku prebieha dvomi spôsobmi.

```

<pts:metadata xmlns:pts="http://theses.cz/pts/elements/1.0/" xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <pts:textworks xmlns:pts="http://theses.cz/pts/elements/1.0/">
    <pts:sender.id>541</pts:sender.id>
    <pts:textworks.id>mczu52a1u4346v1</pts:textworks.id>
    <dc:title xmlns:dc="http://purl.org/dc/elements/1.1/">Moodle_CZU_mczu52a1u4346v1</dc:title>
    <dc:subject xmlns:dc="http://purl.org/dc/elements/1.1/">Moodle;CZU;</dc:subject>
    <pts:teacher pts:id="2318">
      <pts:foreName>David</pts:foreName>
      <pts:surName>Buchtela</pts:surName>
      <pts:email>buchtela@pef.czu.cz</pts:email>
      <pts:key>srtho2a</pts:key>
    </pts:teacher>
    <pts:creator pts:id="4346">
      <pts:foreName>Test</pts:foreName>
      <pts:surName>Test</pts:surName>
    </pts:creator>
    <pts:get.file>
      <pts:url>
        https://moodle.fit.cvut.cz/mod/antiplag/check_files/antiplag_source_c52_s1_u4346_v1.pdf
      </pts:url>
      <pts:author>4346</pts:author>
    </pts:get.file>
    <pts:type.download>w</pts:type.download>
  </pts:textworks>
</pts:metadata>

```

Obr. 3.2: Kontrolný XML súbor pre komunikáciu so službou Odevzdej.cz

Prvým spôsobom je ručné skontrolovanie existencie výsledku po kliknutí na odkaz priamo v sekcii s tabuľkou obsahujúcou všetky odovzdané semestrálne práce. V tabuľke vyučujúci môže vidieť informáciu o tom, že kontrola ešte prebieha. Po kliknutí na tento odkaz je odvedený na obrazovku s informáciami o priebehu kontroly. Zároveň je vykonaná požiadavka pre kontrolu existencie výsledku na strane služby Odevzdej.cz. V prípade, že výsledok procesu kontroly skutočne existuje sú vyučujúcemu okamžite zobrazené informácie o tomto výsledku.

Druhým spôsobom je automatická kontrola existencie výsledku kontroly plagiátorstva, ktorá prebieha prostredníctvom automatizovanej úlohy vytvorenej za týmto účelom. Táto úloha sa v určitých časových intervaloch automaticky spúšťa a kontroluje existenciu výsledkov kontroly pre všetky semestrálne práce, kde je evidovaný záznam o kontrole bez získania výsledku tohto procesu. Okrem toho táto automatizovaná úloha zabezpečuje rozoslanie notifikačných e-mailov o hodnotení, odovzdaní novej semestrálnej práce, či práve získaní výsledku kontroly. Vyučujúci tak nemusí osobne kontrolovať existenciu výsledku kontroly, ale v prípade jeho získania je o tom informovaný prostredníctvom e-mailu. Ručné kontrolovanie tohto výsledku pre väčší počet študentov by bolo


3. REALIZÁCIA

veľmi zdĺhavé a neefektívne. Účelom tejto diplomovej práce je predovšetkým zjednodušiť prácu vyučujúcich a nie ju zbytočne viac komplikovať, preto bola naimplementovaná popisovaná automatizovaná úloha. Interval spúšťania tejto automatizovanej úlohy je možné nastaviť priamo v administrácii systému Moodle.

Oba spôsoby používajú pre kontrolu existencie výsledku kontroly metódu `check_for_result`. Táto metóda má tri kroky. Prvým krokom je vykonanie `cURL` príkazu, ktorý sa pokúsi zistiť existenciu samotného výsledku. Výstupom tohto príkazu je XML súbor. Existenciu výsledku, respektíve stav procesu kontroly určuje hodnota uzlu `status`. Hodnota päť, alebo šesť znamená, že kontrola bola úspešne dokončená a jej výsledok je dostupný. Prvá hodnota znamená, že ku kontrolovanému súboru neboli nájdené žiadne podobnosti. Druhá hodnota zas znamená, že v kontrolovanom súbore boli zistené podobnosti s inými dokumentami. Druhou časťou je práve v tomto prípade spracovávanie tohto XML súboru a vyextrahovanie informácií o týchto dokumentoch. Tieto informácie sú nasledne uložené do databázy. Tretou časťou je potom aktualizovanie samotného záznamu o kontrole a uloženie týchto zmien do databázy.

Oba spôsoby získania výsledku kontroly plagiátorstva zapríčinia v konečnom dôsledku to, že užívateľ, v tomto prípade vyučujúci, uvidí v tabuľke s prehľadom odovzdaných semestrálnych prác pri skontrolovaní práce informáciu o tom, že kontrola bola dokončená spolu s percentuálnou hodnotou najvyššej dosiahnutej podobnosti a odkazom na stránku s podrobnými informáciami o výsledku kontroly. Vyučujúci tak po kliknutí na tento odkaz uvidí okrem informácií o kontrolovanom súbore aj zoznam podobných dokumentov spolu s percentuálnou hodnotou podobnosti medzi konkrétnym dokumentom a testovanou semestrálnou prácou. Okrem toho sa vyučujúcemu zobrazí aj odkaz na stiahnutie výsledného protokolu o výsledku kontroly vo formáte PDF. Po dlhšej úvahe som zvolil zobrazenie len tohto jediného úplného výsledného protokolu. Služba `Odevzdej.cz` totiž poskytuje možnosť vytvoriť takýto protokol pre akékoľvek dva, či viac dokumentov, pri ktorých chceme porovnať podobnosti. Avšak každý z týchto dokumentov by znamenal ďalšiu požiadavku vykonanú prostredníctvom `cURL` príkazu smerovaného na službu `Odevzdej.cz`. Keďže som si vedomý toho, že môže nastať situácia, kde ku kontrolovaní semestrálnej práci bude existovať veľké množstvo podobných dokumentov rozhodol som sa nakoniec po dlhšej úvahe nezískavať pre každý z týchto dokumentov samostatný protokol obsahujúci podobnosti. Hlavným dôvodom bola možnosť, že v tejto situácii by to veľké množstvo vykonávaných príkazov znamenalo značnú záťaž pre server a nepriaznivo tak mohlo ovplyvniť správne fungovanie celého systému Moodle FIT.

Príklad toho ako vyzerá takýto výsledkový protokol môžete vidieť na obrázkoch 3.3 a 3.4.



Podobnosti s vybranými dokumenty

100 %

Zkoumaný dokument

Agenda	Odevzdávárny
Jméno souboru	antiplag_source_c52_s1_u4346_v1.pdf
Změněno	20160424005247
Informace o práci	Moodle_CZU_mczu52a1u4346v1
URL	Test Test, Česká zemědělská univerzita v Praze
Informace o souboru	https://odevzdej.cz/auth/doc/czu/gczrs15/antiplag_source_c52_s1_u4346_v1.pdf https://odevzdej.cz/auth/plag/4aa163b749bc8d3b/

24. dubna 2016 23.53

Odevzdej.cz provozuje Fakulta informatiky Masarykovy univerzity.
Podobnosti se v průběhu času mění, doporučuje se kontrola on-line.

Obr. 3.3: Výsledkový protokol kontroly plagiátorstva službou Odevzdej.cz - 1. strana

3. REALIZÁCIA

Systém na odhalování plagiátů v seminárních pracích		odevzdej.cz
Seznam vybraných podobných dokumentů:		
100 %	Agenda Jméno souboru Změněno Informace o práci Informace o souboru Podobnosti	Odevzdávárny antiplag_source_c2_s1_u2_v1.pdf 20160422141941 Moodle_CZU_c2a3u3v1, test test Česká zemědělská univerzita v Praze https://odevzdej.cz/auth/plag/5ee076e6d8ce9aad/?noplag=1 https://odevzdej.cz/auth/podob/4aa163b749bc8d3b/5ee076e6d8ce9aad/
100 %	Agenda Jméno souboru Změněno Informace o práci Informace o souboru Podobnosti	Odevzdávárny antiplag_source_c2_s1_u2_v1.pdf 20160422141935 Moodle_CZU_mczu2a1u2v1, Admin User Česká zemědělská univerzita v Praze https://odevzdej.cz/auth/plag/8431744c7218e356/?noplag=1 https://odevzdej.cz/auth/podob/4aa163b749bc8d3b/8431744c7218e356/
100 %	Agenda Jméno souboru Změněno Informace o práci Informace o souboru Podobnosti	Odevzdávárny antiplag_source_c2_s1_u2_v1.pdf 20160422141935 2a1u2v1, Admin User Česká zemědělská univerzita v Praze https://odevzdej.cz/auth/plag/e09eddf7fa8c366a/?noplag=1 https://odevzdej.cz/auth/podob/4aa163b749bc8d3b/e09eddf7fa8c366a/
11 %	Agenda Jméno souboru Změněno Informace o práci Informace o souboru Podobnosti	Závěrečná práce bakalarka.pdf 21. 5. 2008 22:23, Bc. Marek Trgjiňa Penetračné testovanie, Bc. Marek Trgjiňa Masarykova univerzita https://odevzdej.cz/auth/plag/130096bd6ee9a60e/?noplag=1 https://odevzdej.cz/auth/podob/4aa163b749bc8d3b/130096bd6ee9a60e/
Podobné pasáže s výše uvedenými dokumenty jsou vyznačeny červeně		
PENETRAČNÉ TESTOVANIE Semestrálna práca z predmetu MI-IBE Bc. Michal Sklenár 2 Abstrakt Náplňou a obsahom tejto semestrálnej práce je popis penetračného testovania. V mojej práci čitateľa oboznámim s tým, čo to vlastne penetračné testovanie je a čo je jeho cieľom, prečo sa vôbec používa, aké typy testov poznáme, aké metodiky a aké nástroje sa pri testovaní		
2 z 8		

Obr. 3.4: Výsledkový protokol kontroly plagiátorstva službou Odevzdej.cz - 2. strana

Na týchto obrázkoch môžete vidieť prvé dve strany výsledkového protokolu. Na prvej strane môžete vidieť informácie o testovanom súbore spolu s percentuálnou hodnotou najvyššej dosiahnutej podobnosti s vybranými dokumentami. Na druhej strane zas vidíte zoznam vybraných podobných dokumentov. Na ďalších stranách sú potom červenou farbou vyznačené podobnosti nájdené v texte testovanej práce.

Týmto by som ukončil popis tejto časti implementácie a vlastne celého procesu kontroly plagiátorstva v semestrálnych prácach.

Testovanie

4.1 Popis procesu testovania

V nasledujúcej časti by som rád popísal proces a spôsoby testovania mnou navrhnutého a následne naimplementovaného modulu Antiplagiát pre e-learningový systém Moodle FIT. Ako snáď pri každom projekte softvérového typu, aj pri tomto projekte proces testovania zohráva dôležitú úlohu. Zároveň jednou z požiadaviek v zadaní tejto práce je riadne otestovanie modulu a jeho následné nasadenie na produkčnú verziu systému Moodle FIT.

Počas celého procesu implementácie tohto modulu som ho priebežne testoval na lokálnom serveri, kde mám nasadený systém Moodle, ktorý je ako open source bezplatne dostupný na stiahnutie priamo na internetových stránkach projektu Moodle. Testovanie na tomto serveri však mohlo prebiehať len do určitej fázy implementácie. Služba Odevzdej.cz totiž funguje tak, že pri požiadavke o kontrolu plagiátorstva v semestrálnej práci sa službe zasiela kontrolný XML súbor s verejne dostupným odkazom na súbor, ktorý sa má skontrolovať. Samozrejme súbor nahraný na mojom lokálnom serveri nie je verejne prístupný, a teda pre otestovanie funkcií spojených so službou Odevzdej.cz bolo nutný nejaký testovací server, z ktorého by tento súbor bol dostupný a služba by si ho odtiaľ mohla prevziať. Pôvodným plánom bolo vytvorenie testovacieho prostredia na školskom serveri, to sa však nakoniec nepodarilo zariadiť. Bohužiaľ, bezplatné hostings majú určité obmedzenia, ktoré bránili nainštalovať systém Moodle. Keďže som sa chcel vyhnúť plateniu za nejaký server, na ktorom by bolo možné systém Moodle nainštalovať, pokúsil som sa zariadiť, aby som mohol systém nainštalovať a testovať na serveri v práci. To mi bolo bez problémov umožnené a mohol som tak posunúť testovanie do ďalšej fázy.

Testovanie práce službou Odevzdej.cz trvá približne jeden až dva dni v závislosti od počtu požiadaviek čakajúcich na spracovanie. Avšak aj po niekoľkých pokusoch sa mi stále nedarilo dosiahnuť úspešné otestovanie súboru službou Odevzdej.cz. Po kontaktovaní vývojového tímu, ktorý má chod tejto

služby na starosti prostredníctvom spoplatneného kontaktného formuláru sa mi podarilo zistiť v čom je problém. Pre komunikáciu so službou sa v súčasnosti používa účet, ktorý je zaregistrovaný na Českou zemědělskou univerzitu, pre ktorú bol vyvinutý aj pôvodný modul Antiplagiát. Prístupové údaje k tomuto účtu mi poskytol vedúci mojej práce. Bohužiaľ ani on nevedel, že komunikácia so službou Odevzdej.cz s použitím tohto účtu je obmedzená len na určité domény a ich správu má v kompetencii len jedna osoba na Českej zemědělskej univerzite. To v praxi znamenalo, že som nemohol z mojej testovacej domény otestovať funkčnosť komunikácie so službou.

Keďže vytvorenie nového účtu na Odevzdej.cz pre platiacu inštitúciu podľa informácii trvá v priemere päť až šesť týždňov a znamenalo by to neúspech celého projektu, vedúci mojej práce sa pokúsil zariadiť pridanie domény, na ktorej beží produkčná verzia systému Moodle FIT. To sa mu po dohode s Ing. Machom, ktorý je jedinou kompetentnou osobou pre správu domén podarilo s podmienkou, že novú verziu modulu poskytneme aj ich univerzite. S touto dohodou obe strany súhlasili a vedúci práce sa následne pokúsil zariadiť mi prístup na produkčnú verziu systému Moodle FIT. To sa mu aj keď s menšími problémami nakoniec podarilo a ja som teda mohol začať konečne testovať správnosť implementácie komunikácie so službou.

Záverečné fázy testovania teda prebiehali už na produkčnej verzii systému Moodle FIT, kde mal byť mnou navrhnutý a následne naimplementovaný modul Antiplagiát nasadený pôvodne až po otestovaní. Testoval som hlavne komunikáciu so službou Odevzdej.cz. Tu bolo treba rozlíšiť dva prípady:

1. Ku kontrolovanému súboru existujú nejaké dokumenty, ktoré sú aspoň trochu podobné
2. Ku kontrolovanému súboru neexistujú žiadne podobné dokumenty a súbor je teda unikátny

Oba prípady bolo treba brať samozrejme do úvahy minimálne z dôvodu, že výstup s výsledkom kontroly službou Odevzdej.cz sa v oboch prípadoch trochu líši. Otestoval som teda prácu, ktorú som testoval už v minulosti pod iným účtom a mal som teda istotu, že bude vyhodnotená ako plagiát a vo výstupe z kontroly plagiátorstva nájdem minimálne jeden podobný dokument. Na otestovanie druhého prípadu som použil moju semestrálnu prácu z jedného z predmetov, ktorú som vypracoval v priebehu svojho štúdia na fakulte informačných technológií. Tu som mal zas istotu, že práca je unikátna a nastane požadovaná situácia, ktorú som chcel otestovať. Obe kontroly plagiátorstva prebehli na Odevzdej.cz v priebehu jedného dňa a poskytli mi uspokojivé výsledky, ktoré mi umožnili posunúť sa do ďalšej fázy testovania.

Ďalšou fázou testovania bolo napísanie testov, ktoré by potvrdili správnu funkčnosť ostatných častí systému. Nie všetky časti totiž bolo možné otestovať prostredníctvom takto napísaných testov. Jedná sa napríklad o už spomí-

nanú komunikáciu so službou Odevzdej.cz, kde proces kontroly trvá v priemere jeden až dva dni a testy by teda nič nepreukázali.

4.2 Testovacie prostredie systému Moodle

Po preskúmaní dokumentácie e-learningového systému Moodle som zistil, že tento systém umožňuje vytvorenie testovacieho prostredia a spúšťanie vlastných testov. Pri podrobnejšej analýze som sa dozvedel, že systém Moodle ponúka možnosť využitia konkrétne dvoch typov testov:

1. Akceptačné testy
2. Unit testy

Akceptačné testy sú označované ako testy kvality, na základe ktorých sa potom určuje, či bude produkt, v informatike väčšinou nejaký systém, alebo aplikácia akceptovaný. To znamená, že sa zisťuje, či výsledný produkt, respektíve systém spĺňa požadované špecifikácie a kritériá. Pôvodne sa tento typ testov označoval ako funkcionálne testy, pretože každý akceptačný test sa snaží otestovať funkcionality užívateľského príbehu.

Unit testy sú testy slúžiace na otestovanie a overenie fungovania a korektnosti implementácie systému. Anglické slovo unit môžeme preložiť ako slovo jednotka, čo znamená, že cieľom týchto testov je nezávislé a individuálne overovanie správnej funkčnosti čiastkových častí, alebo teda jednotiek zdrojového kódu. Za jednotku môžeme považovať nejakú samostatne testovateľnú časť aplikačného programu. Môže to byť program, trieda, funkcia, procedúra, premenná a podobne. Za výhodu oproti akceptačným testom považujem za skutočnosť, že v prípade, že zlyhá akceptačný test, tak síce vieme, že niečo nefunguje, ale nevieme to tak presne ako pri unit teste. Jedná sa o to, že akceptačné testy testujú veľké množstvo jednotiek funkcionality, a teda nedostaneme presnú informáciu. Unit testy sa však spúšťajú individuálne a nezávisle a ak jeden z nich zlyhá, dostaneme presnú informáciu o tom, ktorý test zlyhal a vo väčšine prípadov spolu s dôvodom zlyhania.

Pre mňa najjednoduchšou voľbou, s ktorou mám aj najviac skúseností bolo práve využitie zmieneného druhého typu testov, a to unit testov. Moodle umožňuje pomerne jednoduché zostavenie a následnú konfiguráciu testovacieho prostredia, v ktorom je potom možné dané testy spúšťať. Priamo v administrácii tohto e-learningového systému môžete nájsť dokonca stručný a jasný manuál k testovaniu v rámci systému Moodle s využitím práve unit testov. Okrem iného obsahuje aj odkaz na dokumentáciu k integrácii PHPUnit [9] do systému Moodle a odkaz priamo na dokumentáciu PHPUnit.

PHPUnit je testovací framework pre programovací jazyk PHP vyvinutý práve pre písanie jednotkových testov v tomto jazyku. V PHPUnit by mala každá testovaná trieda mať svoj takzvaný "TestCase", čo je trieda plniaca funkciu akéhosi kontajneru pre jednotlivé testovacie metódy.

Nakonfigurovanie testovacieho prostredia systému Moodle pozostáva v podstate z dvoch krokov.

Prvým je použitie Composeru pre nainštalovanie potrebných závislostí. Composer je nástroj pre správu závislostí v jazyku PHP. Umožňuje deklarovať knižnice, na ktorých je nejaký projekt závislý a spravuje ich, čo znamená ich inštaláciu, alebo prípadnú aktualizáciu. Je využívaný pri množstve projektov, dokonca ho používa napríklad aj v programátorskej komunite známy PHP framework Symfony [10] od francúzskej spoločnosti SensioLabs.

Druhým krokom je konfigurácia samotného serveru, na ktorom beží inštaláciu systému Moodle. Tá spočíva v tom, že sa rovnako ako pri inštalácii samotného Moodle nastavujú určité parametre, napríklad parametre umožňujúce prístup k databáze, zadefinovanie cesty k zložke s dočasnými súborami a podobne. Teoreticky by asi malo byť možné využiť aj databázu používanú priamo pre produkčné prostredie systému Moodle, avšak osobne to nepovažujem za dobrý nápad. Jedným z dôvodov je skutočnosť, že databáza systému Moodle obsahuje približne 250 databázových tabuliek a v prípade využitia tej istej databázy, len s inými názvami tabuliek by sa táto veľkosť zdvojnásobila. Druhým dôvodom je, že ak by sme použili rovnakú databázu, dokonca aj s rovnakými tabuľkami, dostali by sme produkčnú databázu zaplnenú množstvom rôznych testovacích dát, prípadne by sme mohli prísť o už existujúce dáta, alebo by mohli byť nežiadúcim spôsobom upravené, či inak znehodnotené.

Zostáva už len spustenie inicializačného skriptu, ktorý nainicializuje testovacie prostredie. Po úspešnom ukončení tohto skriptu je možné spúšťať unit testy v rámci systému Moodle.

4.3 Testy

V tejto sekcii by som rád popísal mnou vytvorené testy, ktoré boli použité pri testovaní správnosti implementácie modulu Antiplagiát. Napísal som niekoľko Unit testov, ktoré pokrývajú kľúčové časti modulu. Keďže v rámci pôvodnej verzie modulu Antiplagiát vytvorenej vedúcim tejto diplomovej práce neboli vytvorené žiadne testy, ktoré by som mohol použiť, inšpiroval som sa testami, ktoré sú dostupné v rámci modulu “Úkol”. Tento modul obsahuje veľké množstvo testov. Podľa všetkých odporúčaní pre správnu implementáciu modulu pre systém Moodle by každý modul mal obsahovať implementáciu niektorých funkcií, ktoré sú systémom Moodle používané. To v tomto prípade znamená, že v module “Úkol” existujú aj testy pre kontrolu správneho fungovania týchto funkcií, ktoré sú samozrejme súčasťou aj implementácie môjho modulu. Preto som sa pri vytváraní týchto konkrétnych testov pre môj modul inšpiroval spomínanými testami z modulu “Úkol”. Pravdaže nebolo možné proste ich prevziať a použiť. Aj keď sa môj modul v niektorých veciach podobá tomuto modulu, implementácia týchto funkcií v mojom module je predsa len trochu odlišná. Prispôbil som teda tieto testy mojej implementácii funkcií požado-

vaných systémom Moodle. Okrem spomínaných funkcií pravdaže testujem aj základné akcie, ktoré môžu byť v module vykonávané ako napríklad vytvorenie záznamu o nahranom súbore so semestrálnou prácou, alebo známkovanie projektu vyučujúcim.

Aktuálne obsahuje modul Antiplagiát tieto jednotkové testy, ktoré boli použité pre overenie správneho fungovania modulu:

- **test_create_instance** - tento test je možné nájsť v súbore `base_test.php`. Je to jednoduchý test, ktorý kontroluje, či sa skutočne vytvorila nová inštancia triedy `testable_antiplag`, triedy, ktorá je potomkom triedy `antiplag` a je využívaná pri všetkých ostatných testoch
- **test_update_instance** - test sa nachádza v súbore `lib_test.php`. Úlohou tohto testu je overiť správne fungovanie aktualizácie inštancie triedy `antiplag`. V rámci testu je vytvorená nová inštancia triedy `testable_antiplag`, potomka triedy `antiplag`. Následne je upravený jeden z parametrov, konkrétne dátum odovzdania práce. Takto upravená inštancia je potom použitá ako argument pre funkciu `antiplag_update_instance`, ktorá vykoná aktualizovanie záznamu v databáze. Nakoniec sa z databázy vyberie záznam o tejto inštancii triedy `antiplag` a skontroluje sa, či dátum odovzdania práce súhlasí so zmeneným dátumom odovzdania. Ak áno, test je úspešný, v opačnom prípade sa test považuje za neúspešný.
- **test_delete_instance** - test sa opäť nachádza v súbore `lib_test.php`. Úlohou testu je overenie správnosti fungovania odstránenia inštancie triedy `antiplag`. V rámci tohto testu je opäť vytvorená nová inštancia triedy `testable_antiplag`. Následne sa simuluje odovzdanie súboru so semestrálnou prácou a jej oznámkovanie. Nakoniec sa zavolá funkcia `antiplag_delete_instance`, ktorá ako argument dostane identifikátor vytvorenej inštancie. Pri vrátení hodnoty "true" je test považovaný za úspešný. Návratová hodnota "false" znamená neúspech celého testu.
- **test_print_recent_activity** - tento test ako aj predchádzajúce je možné nájsť v súbore `lib_test.php`. Jeho úlohou je otestovanie funkcie, ktorá je systémom Moodle používaná pre výpis nedávnej aktivity v rámci modulu Antiplagiát. V tomto teste sa používa inštancia, ktorá bola vytvorená ešte pred spustením testov. Používa sa tu funkcia `antiplag_print_recent_activity`, ktorá ako argumenty prijíma inštanciu kurzu, informácia o tom, či je povolené zobrazovať celé mená užívateľov a dátum začiatku obdobia, od ktorého sa má vypisovať aktivita. Výstup sa následne kontroluje na existenciu určitého regulárneho výrazu, reprezentujúceho text, ktorý by mal správny výstup obsahovať. Od toho potom závisí úspech tohto testu.
- **test_antiplag_get_recent_mod_activity** - funkciu s názvom `antiplag_get_recent_mod_activity` používa systém Moodle pre získanie

dát reprezentujúcich nedávne aktivity užívateľa v rámci kurzu. Opäť je možné tento test nájsť v súbore `lib_test.php`. V rámci toho testu je znovu vytvorená nová inštancia triedy `testable_antiplag`, potomka triedy `antiplag`. Následne je nasimulované odovzdanie súboru so semestrálnou prácou. Známkovanie odovzdanej semestrálnej práce nie je treba simulovať, pre účely tohto testu je zbytočné. Ďalším krokom je vytvorenie pola aktivít, ktoré na indexe s číslom nula obsahuje aktivitu typu “activity”. Nasleduje zavolanie funkcie `antiplag_get_recent_mod_activity`, ktorá dostáva ako argumenty predtým vytvorené pole aktivít, do ktorého sa následne zapisujú aj ďalšie aktivity, index, na ktorý sa má ďalší záznam o aktivite zapísať, dátum začiatku obdobia, od ktorého sa majú hľadať aktivity, identifikátor kurzu, identifikátor modulu v rámci kurzu a identifikátor užívateľa, ktorého aktivity chceme získať. Za úspech testu sa považuje situácia, keď vo výslednom poli s aktivitami je na indexe s číslom jedna aktivita typu “antiplag”. V opačnom prípade sa pravdaže jedná o neúspech testu.

- **test_antiplag_user_complete** - tento test je predposledným testom zo súboru `lib_test.php`. Funkcia `antiplag_user_complete` je ďalšou z funkcií využívaných systémom Moodle. Táto funkcia konkrétne slúži na získanie reprezentácie toho, čo užívateľ robil v rámci nejakej konkrétnej inštancie modulu a používa sa v reportoch užívateľských aktivít. Test začína opäť vytvorením novej inštancie triedy `testable_antiplag` a nasimulovaním odovzdania semestrálnej práce. Následne sa zavolá už spomínaná funkcia `antiplag_user_complete`, ktorej sa ako argumenty nastaví objekt reprezentujúci kurz, objekt reprezentujúci užívateľa, o ktorého aktivite chceme získať informácie, objekt reprezentujúci modul v rámci kurzu a objekt s inštanciou triedy `antiplag`. Vo výstupe tejto funkcie sa následne kontroluje existencia regulárneho výrazu reprezentujúceho určitý text, ktorý má tento výstup obsahovať. Keďže simulujeme len vytvorenie záznamu so semestrálnou prácou, ale nenahrávame žiadny fyzický súbor obsahujúci túto prácu, hľadá sa vo výstupe výraz s informáciou o tom, že nebol nahraný zatiaľ žiadny skutočný súbor. Existencia tohto výrazu rozhoduje o úspechu, alebo prípadnom neúspechu testu.
- **test_antiplag_user_outline** - test `test_antiplag_user_outline` je posledným testom, ktorý môžete nájsť v súbore `lib_test.php`. Úlohou tohto testu je overenie správneho fungovania funkcie `antiplag_user_outline`, ktorú tiež využíva systém Moodle. Táto funkcia vracia malý objekt so stručnými súhrnnými informáciami o tom, čo užívateľ robil v rámci nejakej konkrétnej inštancie modulu, a tak ako funkcia, ktorá bola predmetom predchádzajúceho popisovaného testu je používaná v reportoch užívateľských aktivít. Tento test využíva inštanciu triedy `testable_antiplag`,

ktorá bola vytvorená ešte pred spustením testov a v rámci ktorej bolo nasimulované odovzdanie a následné oznámkovanie semestrálnej práce. Test zavolá funkciu `antiplag_user_outline` s niekoľkými argumentami, ktorými sú objekt reprezentujúci kurz, objekt reprezentujúci užívateľa, o ktorého aktivite chceme získať informácie, objekt reprezentujúci modul v rámci kurzu a objekt s inštanciou triedy `antiplag`. Úspech testu závisí potom na tom, či sa vo výstupe tejto funkcie nachádza hľadaný regulárny výraz. V tomto prípade hľadaný výraz reprezentuje text s informáciou o známke. Ak sa hodnota známky zhoduje s hľadanou známkou, test je úspešný. V opačnom prípade považujeme tento test za neúspešný.

- **test_reset_userdata** - tento test je prvým popisovaným testom zo súboru `locallib_test.php`. Jeho úlohou je overiť správne fungovanie funkcie `reset_userdata`, ktorá sa používa pri zovaní funkcie pre resetovanie dát kurzu v rámci systému Moodle. Funkcia `reset_userdata` v module `Antiplagiát` slúži na odstránenie všetkých záznamov o odovzdaných semestrálnych prácach, fyzických súborov, ku ktorým sa tieto záznamy vzťahujú, záznamov o kontrole semestrálnych prác službou `Odevzdej.cz` a záznamoch o známkach udelených v tomto module v rámci daného konkrétneho kurzu. Test začína vytvorením novej inštancie triedy `testable_antiplag` so správnym nastavením dátumu povolenia prístupu k modulu a dátumu ukončenia odovzdávania prác. Následne je nasimulované odovzdanie súboru so semestrálnou prácou a jeho oznámkovanie. Ďalším krokom je vytvorenie objektu, ktorý obsahuje informácie o tom, ktoré akcie vykonané v rámci modulu chceme z kurzu odstrániť a informácia o posune časov pre modul v rámci kurzu. Nasleduje zovanie funkcie `reset_userdata`, ktorá dostane ako argument spomínaný objekt s dátami. Prvou podmienkou, ktorá musí byť pre prípadný úspech testu splnená je skutočnosť, že po resete funkcia `testable_has_submission`, ktorá vracia informáciu o tom, či inštancia modulu obsahuje nejaké odovzdané semestrálne práce vráti hodnotu "false", ktorá znamená, že dáta z modulu boli úspešne odstránené. Po tejto kontrole sa v databáze nájde obnovený záznam o tejto inštancii triedy `antiplag` a overí sa, či sa pri resete čas správne posunul o hodnotu, ktorá bola obsiahnutá v argumente resetovacej funkcie. Ďalšou situáciou testovanou v rámci tohto testu je otestovanie funkcie `antiplag_reset_userdata`, ktorá vykonáva reset pre všetky inštancie modulu `Antiplagiát` s použitím funkcie `reset_userdata`. Znovu po resete obnovíme záznam s inštanciou triedy `antiplag` a overíme, či sa správne posunul dátum odovzdania. Do tohto momentu sa všetky akcie pre tento test vykonávali len s jednou inštanciou. Pre lepšie otestovanie je teda v ďalšom kroku vytvorená ďalšia testovacia inštancia a znovu sa volá funkcia pre reset dát. Test končí poslednými kontrolami správnosti nových dátumov odovzdania pre obe inštancie. Ak test úspešne prešiel cez všetky testované podmienky, je vyhodnotený ako

úspešný. Opačný prípad znamená neúspech.

- **test_can_grade** - tento test je ďalším z testov umiestnených v súbore `locallib_test.php`. Ako už názov napovedá, jeho úlohou je otestovať správne fungovanie určenia, či má užívateľ právo oznámkovať projekt v rámci modulu Antiplagiát. Tento test je vo svojej podstate skutočne veľmi jednoduchý a nie je tu nič zložité, čo by stálo za podrobnejšie vysvetlenie. Celý test funguje na princípe nastavenia aktuálneho užívateľa a otestovanie, či má právo hodnotiť semestrálnu prácu. Testuje sa užívateľ, ktorý reprezentuje klasického študenta, užívateľ, ktorý je klasickým vyučujúcim a užívateľ, ktorý má práva editujúceho vyučujúceho, teda o niečo vyššie ako klasický vyučujúci. Vždy sa jeden z týchto užívateľov nastaví ako aktuálny užívateľ a následne sa otestuje, či tento užívateľ má udelené potrebné právo na známkovanie semestrálnej práce. Pri oboch typoch vyučujúcich je správnym výsledkom hodnota označujúca, že užívateľ môže práce hodnotiť. Pri klasickom študentovi je správnym výstupom samozrejme informácia o tom, že tento užívateľ nemá práva potrebné pre hodnotenie semestrálnych prác. Ak všetky tieto čiastkové testy vrátia správne hodnoty, test môžeme považovať za úspešný. V prípade, že aspoň jeden z týchto čiastkových testov nevráti správnu hodnotu, je celý test považovaný za neúspešný.
- **test_can_view_submission** - ako aj predchádzajúce dva testy, i tento test nájdete v súbore `locallib_test.php`. Takisto aj v tomto prípade názov napovedá, že úlohou tohto testu je otestovať správne fungovanie určenia, či má užívateľ právo zobrazíť si semestrálne práce odovzdané v rámci modulu Antiplagiát. Znova tak ako aj v predchádzajúcom teste, aj tu celé testovanie prebieha na princípe nastavovania rôznych typov užívateľov do úlohy aktuálneho užívateľa a následné testovanie správneho vyhodnotenia toho, či má užívateľ právo na to, aby si odovzdané semestrálne práce zobrazil. V úvode testu je znovu vytvorená nová testovacia inštancia triedy `testable_antiplag`. Znovu sa tu pracuje s tromi typmi užívateľov, a to s klasickým študentom, klasickým vyučujúcim a editujúcim vyučujúcim. Následne sa v čiastkových testoch kontroluje či má daný užívateľ skutočne právo pre zobrazenie prác odovzdaných v rámci modulu Antiplagiát. Správnosť informácií o tom, či má daný užívateľ dostatočné práva pre zobrazovanie prác rozhoduje v konečnom dôsledku o úspechu, alebo neúspechu celého testu.
- **test_is_graded** - úlohou tohto testu je otestovať správne fungovanie známkovanie, respektíve správnosti informácie, či má užívateľ oznámkovanú svoju semestrálnu prácu. Test je možné nájsť opäť v súbore `locallib_test.php`. Test začína vytvorením novej inštancie triedy `testable_antiplag`. V tomto teste sa pracuje s dvomi užívateľmi. Obaja títo užívatelia reprezentujú klasických študentov. Test pokračuje nasimulova-

ním odovzdania semestrálnej práce a jej následného oznámkovania pre prvého z dvoch využívaných užívateľov. Následne sa spustí pre oboch užívateľov čiastkový test, ktorý má zistiť, či je ich projekt oznámkovaný. Pre prvého užívateľa, pre ktorého bolo nasimulované odovzdanie práce a jej oznámkovanie má tento čiastkový test správne skončiť informáciou, že jeho semestrálna práca je oznámkovaná. Pre druhého užívateľa má čiastkový test skončiť správne informáciou o tom, že nebol nájdený žiadny jeho oznámkovaný projekt. Pri úspechu oboch týchto čiastkových testov pre testovaných užívateľov je celý test vyhodnotený ako úspešný. V opačnom prípade, je test považovaný za neúspešný.

- **test_count_submissions** - test `test_count_submissions` je predposledným z testov zo súboru `locallib_test.php`. Náplňou testovania v rámci tohto konkrétneho testu je overiť, či po odovzdaní určitého počtu semestrálnych prác zodpovedá počet záznamov v databáze o odovzdaných prácach počtu skutočne odovzdaných pri simulácii odovzdania. Test znovu začína vytvorením novej inštancie reprezentujúcej potomka triedy `antiplag`, ktorá je využívaná vo všetkých popisovaných testoch. Pokračuje sa nasimulovaním odovzdania semestrálnej práce a jej následného oznámkovania. Ďalej sa simuluje odovzdanie ďalších dvoch semestrálnych prác, ale bez simulácie známkovania. Nakoniec sa nasimuluje odovzdanie ešte jednej semestrálnej práce znovu spolu so simuláciou oznámkovania tejto práce. Pri simulácii odovzdania každej z týchto testovacích semestrálnych prác je využívaný vždy iný užívateľ, reprezentujúci vždy iného študenta. Keď sú pripravené všetky potrebné testovacie dáta, pristupuje sa k samotnému testovaniu. Spustia sa tri čiastkové testy, z ktorých prvý overuje správnosť počtu oznámkovaných semestrálnych prác, druhý overuje správnosť počtu semestrálnych prác čakajúcich na oznámkovanie a tretí overuje správnosť počtu všetkých odovzdaných semestrálnych prác. Po úspešnom prejdení cez tieto tri testy je celý test vyhodnotený ako úspešný. Pri neúspechu v ľubovoľnom z týchto troch čiastkových testov, celý test končí neúspechom.
- **test_empty_submission_upload** - toto je posledný z testov nachádzajúcich sa v súbore `locallib_test.php` a zároveň posledný z popisovaných testov. Ako už názov možno napovedá, v rámci testu je po vytvorení novej inštancie triedy `testable_antiplag` simulovaný pokus o odovzdanie semestrálnej práce. Simuluje sa však odovzdanie, respektíve nahranie prázdneho súboru. Správnym výstupom tohto testu je hodnota “false”, ktorá správne udáva, že takýto záznam obsahujúci prázdny súbor sa nepodarilo nahráť. Toto predstavuje úspech testu. V prípade, že by test ukázal, že nahranie prázdneho súboru je povolené, môžeme tento test považovať za neúspešný.

Aktuálny stav a výhľad

V tejto kapitole sa budem zaoberať popisom aktuálneho stavu navrhovaného a vyvíjaného modulu Antiplagiát pre e-learningový systém Moodle FIT, určeného na zefektívnenie a zjednodušenie práce vyučujúcich pri kontrole možného plagiátorstva v semestrálnych prácach. Okrem toho poukážem na slabé miesta a potenciálne chyby v aktuálnom riešení a navrhmem rôzne vylepšenia, ktoré by na projekte mohli byť v budúcnosti vykonané.

5.1 Aktuálny stav riešenia

Po dlhých mesiacoch strávených návrhom a následnou implementáciou modulu Antiplagiát pre e-learningový systém Moodle FIT, sa mi podarilo dostať tento modul do stavu umožňujúceho nasadenie a začlenenie do systému Moodle FIT a tým pádom i jeho používanie vyučujúcimi pri kontrole plagiátorstva v semestrálnych prácach.

Po otestovaní systému vedúcim mojej práce na produkčnej verzii systému Moodle FIT bola finálna verzia modulu Antiplagiát umiestnená na fakultný GitLab a odovzdaná vedúcemu mojej diplomovej práce Ing. Buchtelovi. V súčasnosti je tak mnou vyvinutý modul už použiteľný pri kontrole semestrálnych prác vyučujúcimi. Osobne dúfam, že im tento modul maximálne zjednoduší a zefektívni prácu a budú s jeho fungovaním spokojní.

Keďže modul Antiplagiát je funkčný a v súčasnej dobe zároveň používaný pri reálnej práci vyučujúcich, ciele mojej diplomovej práce, ktoré boli stanovené v úvode projektu považujem týmto za splnené. Bola vypracovaná podrobná analýza a návrh modulu a následne bol tento modul naimplementovaný, boli naimplementované testy na overenie funkčnosti, modul bol riadne otestovaný a úspešne nasadený na produkčnú verziu e-learningového systému Moodle FIT.

Všetky v úvode mojej práce stanovené funkčné i nefunkčné požiadavky na navrhovaný modul Antiplagiát sú vo finálnom produkte pokryté. Pri návrhu boli využívané funkcie systému Moodle a dodržiavané rady a odporúčania z ofi-

ciálnej dokumentácie systému. To by malo zabezpečiť, že modul bude funkčný a zostane neporušený aj pri aktualizácii systému Moodle FIT na vyššiu verziu minimálne dovedy, kým sa Moodle nerozhodne úplne odstrániť, prípadne nahradiť niektorú z funkcií využívaných mnou navrhnutým a následne naimplementovaným modulom. V tomto prípade bude samozrejme potrebná úprava modulu, aby zodpovedal aktuálnym štandardom Moodlu. Počas celej práce som dbal na to, aby som postupoval v súlade s odporúčaniami a pravidlami pre vývoj modulov v rámci systému Moodle a zabezpečil tak neporušenie modúlárnej architektúry systému ako celku. Systém bol otestovaný vo všetkých požadovaných internetových prehliadačoch, pričom nebola detekovaná žiadna chyba ovplyvňujúca funkčnosť modulu, prípadne systému Moodle ako celku. Keďže pri testovaní a používaní modulu Antiplagiát nebol zatiaľ zaznamenaný žiadny výpadok, požiadavku na spoľahlivý a dostupný systém považujem takisto za splnenú.

Napriek splneniu všetkých cieľov stanovených v úvode projektu vidím stále priestor pre návrh rôznych vylepšení, ktoré by mohli ešte viac zpríjemniť, zjednodušiť a zefektívniť prácu vyučujúcich.

Za najproblémovejšiu časť práce na tomto projekte považujem komunikáciu so službou Odevzdej.cz. Ako som už v predchádzajúcich častiach spomínal, v priebehu projektu som narazil na pár problémov s touto službou, ktoré som už podrobnejšie popísal a najväčším problém pre mňa znamenala skutočnosť, že do momentu, kým som neposlal podpore správu prostredníctvom platného kontaktného formuláru, nikto nebol so mnou ochotný komunikovať a pri pokuse kontaktovať priamo hlavného vývojára zodpovedného za projekt Odevzdej.cz som bol odkázaný znova na podporu, ktorá nejavila záujem so mnou komunikovať, dokiaľ za položenie otázky nezaplatím. Našťastie sa problém nakoniec podarilo úspešne vyriešiť a moju diplomovú prácu, a teda modul Antiplagiát úspešne dokončiť, otestovať a nasadiť.

5.2 Potenciálne chyby a nedostatky systému

Pri navrhovaní a následnej implementácii modulu Antiplagiát pre e-learningový systém Moodle FIT som si všimol pár potenciálnych nedostatkov, ktorými disponuje mnou vytvorený modul. Za najväčší nedostatok považujem skutočnosť, že služba Odevzdej.cz dokáže kontrolovať plagiátorstvo len v textových typoch súborov. Keďže modul bol vyvíjaný pre fakultu informačných technológií, väčšina zo semestrálnych prác tu má formu nejakého zdrojového kódu. Teoreticky by malo byť možné odovzdať takýto zdrojový kód v textovej podobe a skontrolovať ho službou Odevzdej.cz, prípadne ho pri odovzdávaní previesť do podoby, v ktorej je ho možné skontrolovať na Odevzdej.cz. Avšak stačí, aby študent zmenil názvy premenných, trochu zahladil stopy po kopírovaní a výsledná podobnosť bude okamžite výrazne nižšia, dokonca možno až nulová. Odevzdej.cz kontroluje čisto podobnosti v texte, neberie do úvahy žiadnu logiku, alebo fun-

govanie aplikácie. V tomto prípade by odovzdaná aplikácia bola kontrolovaná skutočne len po textovej stránke. Táto služba nedokáže otestovať fungovanie nejakej aplikácie a detekovať podobnosť v jej fungovaní s ostatnými odovzdanými prácami. Túto skutočnosť vnímam ako veľký nedostatok tohto modulu.

Ďalším veľkým nedostatkom modulu Antiplagiát považujem problém, ktorý v podstate zapríčinil aj zánik pôvodnej verzie modulu vytvorenej vedúcim mojej práce a vytvoril potrebu tohto nového modulu. Jedná sa o to, že samotný systém Moodle sa stále vyvíja a stále vznikajú nové verzie systému. V každej z týchto verzií je viac či menej upravované fungovanie systému, niektoré funkcie sú úplne odstránené, ďalšie nahradené. Mojm pôvodným zámerom bolo vytvoriť modul, ktorý by bol prakticky nezávislý na systéme Moodle. Bohužiaľ, neskôr som zistil, že to prakticky nie je možné. Využitiu niektorých funkcií Moodle som sa totiž nedokázal vyhnúť, prípadne by ich nevyužitie znamenalo značné navýšenie časovej náročnosti a takýto postup by bol na úkor efektivity systému. Okrem toho by samozrejme boli porušené princípy, na ktorých samotný systém Moodle funguje, a ktorými by sa vývojár pri návrhu nového modulu mal riadiť. Navyše toto je štandardné správanie väčšiny systémov. Nie je to teda chyba v pravom zmysle slova, skôr vec, ktorá vytvára isté riziko, že ak sa modul pravidelne v prípade prechodu na nové verzie systému Moodle nebude upravovať, onedlho sa stane znova nepoužiteľným. Riešením by mohlo byť zafixovanie systému Moodle FIT na verziu, kde všetko funguje požadovaným spôsobom. V tomto prípade by sa však škola pripravila o možnosti využívať nové moduly a funkcie systému, ktoré by fungovali len na novších verziách systému. Druhým možným riešením je, ako som už spomenul, pri prípadnom prechode na novú verziu systému vždy skontrolovať a prípadne upraviť modul, aby sa zachovala kompatibilita s novou verziou.

5.3 Návrhy na vylepšenie modulu

Počas práce na projekte a po preskúmaní množstva rôznych pluginov do systému Moodle som narazil na viaceré potenciálne možnosti vylepšenia modulu v aktuálnej podobe.

Asi najzásadnejšou a najvýznamnejšou zmenou by bolo založenie vlastného účtu pre službu Odevzdej.cz v rámci fakulty, prípadne školy, keďže fakulta informačných technológií nie je jedinou fakultou využívajúcou e-learningový systém Moodle. Ako som už spomínal, aktuálna verzia modulu Antiplagiát beží pod účtom, ktorý založila a v súčasnosti platí Česká zemědělská univerzita v Prahe. Tým pádom je používanie tohto modulu v rámci systému Moodle FIT podmienené dohodou s touto univerzitou, ktorá sa prípadne v budúcnosti môže zmeniť a takáto zmena by mohla znemožniť využívanie mnou navrhnutého a naimplementovaného modulu Antiplagiát. V rámci neosobného účtu, ktorý je momentálne používaný pre komunikáciu so službou Odevzdej.cz existuje totiž obmedzenie, ktoré umožňuje komunikáciu so službou len z určitých

povolených domén. Česká zemědělská univerzita tak může hocikedy zrušit doménu moodle.fit.cvut.cz a znefunkčnit tak navrhnutý modul. Preto zastávam názor, že by bolo správne v budúcnosti založiť neosobný účet aj pre České vysoké učení technické v Prahe.

Ďalším návrhom na prípadné vylepšenie tohto modulu by mohla byť výmena zvolenej a používanej služby Odevzdej.cz za inú službu. Za najvhodnejšiu z tých, ktoré som pri svojom hľadaní objavil považujem službu turnitin. Táto služba by uspokojivo pokryla všetky stanovené požiadavky, disponuje vlastným API a databázou prác pre vyhľadávanie podobností, ktorá je podľa všetkého najväčšia svojho druhu, tým pádom väčšia než databáza, ktorá je používaná službou Odevzdej.cz. Bohužiaľ, podľa informácií, ktoré sa mi podarilo zhromaždiť je licencia pre používanie tejto služby skutočne drahá a mám pochybnosti o tom, či by škola, respektíve fakulta bola ochotná zaplatiť tak vysokú sumu. Konkrétnu sumu sa mi nepodarilo vypátrať, našiel som len útržkovité informácie o možnej cene. Tá sa pohybovala v rozmedzí niekoľkých tisíc dolárov. V prípade, že by sa našli ďalšie možné spôsoby využitia tejto služby a škola by bola ochotná zaplatiť si jej používanie, bolo by to určite vítané zlepšenie a v tom prípade by som nasadenie služby turnitin odporučil. Momentálne sa však využíva kontrola plagiátorstva len v module Antiplagiát a dokiaľ služba Odevzdej.cz uspokojivo plní svoju funkciu, asi nie je dôvod ju meniť. Zaujímavou možnosťou, o ktorej som sa zmienil už pri popise alternatív k tejto službe, by bolo vytvorenie vlastného systému pre kontrolu plagiátorstva. Škola by si tak postupne vypracovávala databázu rôznych semestrálnych a záverečných prác podobne ako služba Odevzdej.cz. Okrem iného by škola nemusela vynakladať žiadne finančné prostriedky na využívanie podobného systému a mala by možnosť prispôbiť si celý systém svojim potrebám. Navrhnutie takéhoto systému však nebolo predmetom mojej práce. Okrem toho myslím, že rozsahovo by podobný projekt bol na úrovni samostatnej záverečnej práce.

Posledným návrhom na vylepšenie navrhnutého a naimplementovaného modulu, ktorý považujem za možný prínos pre projekt a myslím, že je dôležité ho spomenúť je možnosť kontroly plagiátorstva v zdrojovom kóde, o ktorej som písal už v predchádzajúcej sekcii. Modul Antiplagiát v súčasnej dobe umožňuje len kontrolu plagiátorstva textových typov súborov, respektíve služba Odevzdej.cz, ktorá sa tu používa pre kontrolu plagiátorstva umožňuje len kontrolu textových typov súborov. Pri hľadaní možných alternatív k tejto službe som narazil na pár modulov a služieb, ktoré riešia kontrolu plagiátorstva zdrojového kódu a zaujala ma myšlienka doplnenia modulu Antiplagiát o túto novú funkcionality, ktorá by ho urobila komplexnejším a všestrannejším. Napadlo ma preskúmanie možnosti využiť algoritmus, ktorý je používaný v systéme Progtest, ďalšom školskom systéme, o ktorom viem, že disponuje okrem iných funkcií i funkciou kontroly plagiátorstva v zdrojovom kóde. Osobne som nemal možnosť preskúmať funkciu kontroly plagiátorstva zdrojového kódu v rámci systému Progtest, a tak neviem na akom princípe presne funguje, a

teda či by bola v tomto prípade vôbec použiteľná. V prípade, že by bolo jej fungovanie prijateľné pre účely tohto modulu a túto funkcionality by bola možné jednoducho preniesť a zahrnúť ju do modulu Antiplagiát, určite by som túto možnosť vylepšenia v budúcnosti odporučil. V opačnom prípade by som navrhoval zvážiť použitie nejakej inej dostupnej služby, knižnice, alebo už existujúceho pluginu pre systém Moodle, prípadne zadanie vytvorenia tejto funkcionality nejakému študentovi v rámci nejakého školského predmetu.

5.4 Výhľad do budúcnosti

Ako som už v predchádzajúcej časti spomenul, pri tomto module stále existuje pomerne veľký priestor na prípadne rozširovanie, alebo vylepšovanie. Rôzne rozšírenia, alebo vylepšenia by mohli vyučujúcim ešte viac zjednodušiť a spríjemniť ich prácu.

Aj napriek tomu, že ja svoje štúdium na tejto škole už končím a nebudem sa tak môcť už tomuto modulu venovať a prípadné odsúhlasené vylepšenia zapracovať, existuje stále niekoľko možností ako zabezpečiť zapracovanie navrhnutých úprav:

- Zapracovanie úprav a prípadných vylepšení si vezme na starosť vedúci mojej práce.
- Zadanie úprav v rámci nejakého školského predmetu.
- Zadanie úprav v rámci záverečnej práce.

Keďže vedúci mojej práce je zároveň správcom systému Moodle FIT a autorom pôvodného modulu Antiplagiát, mohla by to byť najjednoduchšia cesta. Z vlastnej skúsenosti môžem potvrdiť, že samotné preštudovanie dokumentácie, zorientovanie sa v prostredí systému a osvojenie si nejakých základných návykov a postupov v systéme Moodle istý čas zaberie. Pri použití prvej možnosti by to nebolo vďaka predchádzajúcim skúsenostiam s týmto systémom nutné.

Pri druhej možnosti by popisované vylepšenia mohli byť vypracované napríklad v rámci semestrálnej práce. Táto možnosť by síce mohla byť prínosom pre konkrétny predmet, avšak osobne neviem o žiadnom predmete na našej fakulte, v rámci ktorého by bolo použiteľné takto špecifické zadanie.

Poslednou možnosťou je vytvorenie zadania pre záverečnú prácu podobne, ako bolo vytvorené zadanie pre túto prácu. Nemyslím si však, že by zapracovanie mnou popisovaných vylepšení a úprav bolo rozsahovo dostatočné pre diplomovú prácu. Určite by však podľa mňa mohlo byť dostačujúce pre bakalársku prácu. Avšak v prípade, že by sa došpecifikovalo a upresnilo zadanie pre doplnenie kontroly plagiátorstva v zdrojovom kóde, je možné, že by rozsah práce mohol zodpovedať i samostatnej diplomovej práci.

Záver

V tejto záverečnej časti by som rád zhrnul výsledky a popísal prínosy mojej diplomovej práce.

Mnou navrhovaný a následne naimplementovaný a riadne otestovaný modul Antiplagiát pre e-learningový systém Moodle FIT, ktorý bol predmetom tejto diplomovej práce je v súčasnosti úspešne nasadený a pripravený pre používanie zamestnancami fakulty informačných technológií. Práca vyučujúcich by mala byť vďaka môjmu modulu v mnohých ohľadoch jednoduchšia, rýchlejšia a tým pádom i efektívnejšia. Vyvinutím tohto modulu bola zvýšená pravdepodobnosť, že v rámci konkrétnych predmetov budú prijímané semestrálne práce, ktoré sú v skutočnosti plagiátmi, ale bežný vyučujúci by to nemusel zistiť. Vyučujúci môže pohodlne len pár kliknutiami skontrolovať požadované práce a oznámkovať ich. Okrem toho tu vidím i prínos pre študentov. Z vlastnej skúsenosti viem, aké nepríjemné je čakanie na ohodnotenie odovzdanej práce, ktoré môže byť vďaka môjmu modulu skrátené.

Okrem popisovaných prínosov znamenajúcich výhody pre vyučujúcich a študentov je tu určite prínos i pre mňa ako programátora. Pri tomto projekte som prenikol do systému Moodle a naučil sa niečo o jeho modulárnej architektúre, s ktorou neprichádzam do styku skoro vôbec. Keďže celú komunikáciu s vývojovým tímom služby Odevzdej.cz som obstarával ja, vyskúšal som si trochu samostatnejšiu prácu na projekte. Zvyčajne pracujem na projektoch spolu s tímom ľudí, kde každý má svoju úlohu. V rámci tohto projektu som musel všetky tieto úlohy zastúpiť sám. Tento projekt pre mňa znamenal obrovský prínos vo forme získania nových znalostí a vedomostí a bol pre mňa určite dobrou a zaujímavou skúsenosťou.

Literatúra

- [1] M. Drlík, P. Švec, J. Kapusta, M. Mesárošová: *Moodle - Kompletní průvodce tvorbou a správou elektronických kurzů*. Computer Press, 2013, ISBN 978-80-251-3759-8.
- [2] Fakulta informatiky Masarykovy univerzity: Odevzdej.cz: seminární a školní práce. Máj 2016. Dostupné z: <http://odevzdej.cz/>
- [3] Moodle Pty Ltd: Assignment activity. Apríl 2016. Dostupné z: https://docs.moodle.org/29/en/Assignment_activity
- [4] Moodle Pty Ltd: Moodle: Developer Documentation. Máj 2016. Dostupné z: https://docs.moodle.org/dev/Main_Page
- [5] JetBrains s.r.o.: PhpStorm: Lightning-smart PHP IDE. Apríl 2016. Dostupné z: <https://www.jetbrains.com/phpstorm/>
- [6] appsolute GmbH: MAMP & MAMP PRO. Máj 2016. Dostupné z: <https://www.mamp.info/en/>
- [7] Crot Software: Crot Plagiarism Checker. Apríl 2016. Dostupné z: <http://www.crotsoftware.com/>
- [8] Turnitin, LLC.: Turnitin. Máj 2016. Dostupné z: <http://turnitin.com/>
- [9] Bergmann, S.: PHPUnit - The PHP Testing Framework. Máj 2016. Dostupné z: <https://phpunit.de/>
- [10] SensioLabs: High Performance PHP Framework for Web Development - Symfony. Máj 2016. Dostupné z: <http://symfony.com/>

Zoznam použitých skratiek

- PHP** Hypertext Preprocessor - skriptovací programovací jazyk určený predovšetkým pre programovanie dynamických internetových stránok
- HTML** HyperText Markup Language - značkovací jazyk pre hypertext, ktorý sa používa pre vytváranie internetových stránok
- SQL** Structured Query Language - jazyk používaný pre prácu s dátami v relačnej databáze
- MySQL** Volne šíriteľný databázový systém, ktorý komunikuje pomocou jazyka SQL
- PDF** Portable Document Format - súborový formát pre ukladanie dokumentov nezávisle na platforme
- CSS** Cascade Style Sheets - jazyk pre popis spôsobu zobrazenia internetových stránok
- JavaScript** - multiplatformný, objektovo orientovaný skriptovací jazyk, používaný pri tvorbe internetových stránok
- WYSIWYG** What You See Is What You Get - spôsob editácie dokumentov v počítači, pri ktorom je verzia zobrazená na obrazovke vzhľadovo totožná s výslednou verziou dokumentu
- framework** - softwarová štruktúra, ktorá slúži ako podpora pri programovaní a vývoji a organizácii iných softwarových projektov
- CKEditor** - WYSIWYG editor
- cURL** Client URL Library - softwarový projekt poskytujúci knižnicu nástroj pre príkazový riadok pre prenos dát s použitím rôznych protokolov
- ČZU** Česká zemědělská univerzita - univerzita v Prahe

A. ZOZNAM POUŽITÝCH SKRATIEK

OpenSSL - open source implementácia protokolov SSL a TLS

XML eXtensible Markup Language - rozšíriteľný značkovací jazyk vyvinutý ako pokračovanie jazyka SGML, využívaný pre serializáciu dát

Unit test - jednotkový test slúžiaci na otestovanie čiastkovej časti (jednotky) zdrojového kódu

FIT Fakulta informačních technologií - fakulta na univerzite České vysoké učení technické

GitLab - webový Git repozitár s podporou sledovania chýb

API - Application Programming Interface - označuje rozhranie pre programovanie aplikácií

Obsah priloženého CD

readme.txt.....	stručný popis obsahu CD
manuals.....	adresár obsahujúci všetky požadované príručky
├─ installation_manual.pdf	inštalačná príručka
├─ user_manual.pdf	užívateľská príručka
├─ admin_manual.pdf	administrátorská príručka
└─ documentation	programátorská príručka vygenerovaná pomocou aplikácie DoxyGen
src	
├─ antiplag.....	zdrojové kódy implementácie
├─ thesis.....	zdrojová forma práce vo formáte L ^A T _E X
text	text práce
└─ DP_Sklenár_Micha1_2016.pdf	text práce vo formáte PDF