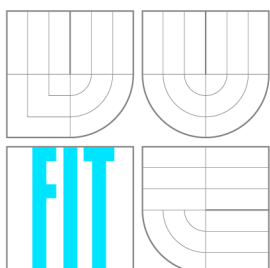


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SKRIPTY PRO PODPORU VÝUKY

SCRIPTS FOR TEACHING SUPPORT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ MOUR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAROSLAV DYTRYCH

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Mour Lukáš**

Obor: Informační technologie

Téma: **Skripty pro podporu výuky**
Scripts for Teaching Support

Kategorie: Web

Pokyny:

1. Seznamte se s jazykem Python a skriptovacím jazykem BASH.
2. Prostudujte problematiku výuky a hodnocení aktivit ve vybraném výukovém kurzu.
3. Navrhněte skripty, které vyučujícímu usnadní vybrané aktivity při výuce, zejména automatizované zpracování e-mailů s přihláškami na týmové projekty, vytváření tabulek pro hodnocení dílčích částí projektů, rozbalení a kontrolu pojmenování a kompletnosti odevzdaných souborů, slučování tabulek s hodnocením od více vyučujících, rozesílání zpráv s detaily hodnocení, export dat pro IS FIT apod.
4. Implementujte navržené řešení.
5. Zhodnoťte dosažené výsledky a srovnajte s alternativními přístupy.

Literatura:

- dle doporučení vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

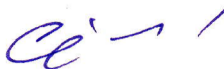
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Dytrych Jaroslav, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Tato bakalářská práce analyzuje problematiku výuky - hodnocení studentských aktivit, zejména registrace studentů na týmové projekty, tvorbu tabulek hodnocení, kontrolu kompletnosti a formální správnosti odevzdaných souborů nebo zasílání hodnocení studentům. Danou problematiku popisuje a navrhuje její automatizaci. Dále popisuje implementaci a závěrečné zhodnocení.

Abstract

This bachelor's thesis analyzes the problems of teaching - evaluation of student activities, especially the registration of students on team projects, creating rating tables, checking of completeness and formal correctness of their solutions or sending the final results back to the students. It describes the issues and proposes their automation. It also describes the implementation and final results.

Klíčová slova

Python, Bash, skripty, výuka, automatizace, studenti, hodnocení

Keywords

Python, Bash, scripts, teaching, automation, students, evaluation

Citace

MOUR, Lukáš. *Skripty pro podporu výuky*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Dytrych Jaroslav.

Skripty pro podporu výuky

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Dytrycha. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Lukáš Mour
18. května 2016

Poděkování

Velice rád bych poděkoval vedoucímu práce Ing. Jaroslavu Dytrychovi za ochotu, odborné, praktické rady a připomínky při tvorbě této bakalářské práce.

© Lukáš Mour, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Analýza současného stavu a specifikace požadavků	4
2.1	Registrace studentů na týmové projekty	4
2.2	Evidence hodnocení studentských aktivit	6
2.3	Rozbalování a kontroly odevzdaných řešení	7
2.4	Rozesílání hodnocení	7
3	Použité technologie	8
3.1	Python	8
3.2	Python - LibreOffice	8
3.2.1	LibreOffice	8
3.2.2	Python-UNO	9
3.3	Bash	9
4	Návrh řešení	10
4.1	Konceptuální model systému	10
4.2	Automatizované zpracování registračních e-mailů	11
4.3	Zpracování registrací z IS FIT	12
4.4	Tvorba tabulky hodnocení	13
4.4.1	Obecná pravidla pro zástupné sekvence/vzorce	14
4.4.2	Sumarizační list	15
4.4.3	Týmový projekt	16
4.4.4	Individuální projekt	19
4.5	Doplňující skripty	21
4.6	Rozbalení a kontrola pojmenování a kompletnosti odevzdaných souborů	22
4.6.1	Rozbalení souborů	22
4.6.2	Kontrola pojmenování a kompletnosti odevzdaných souborů	22
4.6.3	Přejmenování souborů	23
4.6.4	Slučování adresářů	23
4.7	Slučování tabulek od více vyučujících	24
4.8	Rozesílání zpráv s detaily hodnocení	24
4.8.1	Získ dat z tabulky	24
4.8.2	Odesílání e-mailů	25
4.9	Export dat pro IS FIT	25

5 Implementace	26
5.1 Automatizované zpracování registračních e-mailů	26
5.2 Zpracování registrací z IS FIT	27
5.3 Tabulka hodnocení	27
5.4 Rozbalení a kontrola pojmenování odevzdaných souborů	29
5.5 Slučování tabulek od více vyučujících	30
5.6 Rozesílání detailního hodnocení	31
5.7 Export dat do IS FIT	31
6 Testování	33
6.1 Rychlost skriptů spolupracujících s LibreOffice	33
6.2 Automatizované zpracování registračních e-mailů	34
6.3 Zpracování registrací z IS FIT	34
6.4 Tvorba tabulky hodnocení	34
6.5 Rozbalení a kontrola pojmenování a kompletnosti odevzdaných souborů	35
6.6 Rozesílání zpráv s detaily hodnocení	35
7 Závěr	36
Literatura	37

Kapitola 1

Úvod

Cílem této bakalářské práce je automatizace vybraných úkonů vyučujícího, které musí provést při hodnocení studentských aktivit ve vybraném výukovém kurzu na Fakultě informačních technologií.

Kapitola **Analýza současného stavu a specifikace požadavků** popisuje výběr referenčního výukového kurzu, detailně popisuje současné postupy jednotlivých prováděných úkonů a problémy s nimi spojené. Současně pak definuje požadavky na výsledné řešení.

V kapitole **Použité technologie** jsou stručně představeny a popsány technologie využité pro realizaci implementace.

Detailní návrh řešení daného problému, vycházející z podrobné analýzy, popisuje kapitola **Návrh řešení**.

Plynně navazuje kapitola **Implementace**, která popisuje samotnou realizaci řešení automatizace pomocí skriptů v programovacím jazyce Python. U každého skriptu jsou dostupné tabulky s parametry pro spuštění.

Kapitola **Testování** se zabývá popisem jednotlivých dosažených výsledků. Poukazuje také na problémy se staršími verzemi kancelářského balíku LibreOffice.

Závěr obsahuje konečné zhodnocení, porovnání současného a automatizovaného způsobu, včetně návrhů na další vylepšení.

Kapitola 2

Analýza současného stavu a specifikace požadavků

Prvním krokem byla volba referenčního výukového kurzu pro analýzu problematiky výuky - hodnocení studentských aktivit. Byl zvolen volitelný kurz IVS - Praktické aspekty vývoje software na Fakultě informačních technologií Vysokého učení technického v Brně¹. V tomto kurzu je potřeba zajistit časově náročnou organizaci týmového projektu. Správu, organizaci a hodnocení aktivit zde má na starosti vedoucí této práce, Ing. Jaroslav Dytrych. Požadavkem je provést maximální možnou automatizaci níže popisovaných úkonů.

2.1 Registrace studentů na týmové projekty

V kurzu IVS byl před akademickým rokem 2015/2016 používán vzorový koncept e-mailu, zveřejněný na stránce předmětu. Předepsaným pravidlem je, že registrační e-mail na adresu vyučujícího zasílá vedoucí týmu a do kopie příjemců zprávy uvádí ostatní členy svého týmu. Důvodem je informování, že vedoucí pro tým provedl registraci. Dále může vyučující podle příjemců kopie zprávy zkontrolovat loginy (jednoznačné sedmimístné identifikátory studentů FIT), které mohou být chybně uvedeny v těle zprávy. Úkolem vyučujícího je manuální procházení registračních e-mailů a tvorba seznamu týmů. Nedílnou, podstatnou součástí je zdlouhavá kontrola obsažených informací, zejména jmen a loginů registrovaných studentů.

Ukázka registračního e-mailu:

Předmět (Subject): IVS - projekt - xlogin01 xlogin02 xlogin03 ...

Obsah (E-mail text body):

Přihlášení na projekt do IVS

Tým: název týmu

Vedoucí: Jméno Příjmení xlogin01@stud.fit.vutbr.cz

Členové:

Jméno Příjmení xlogin02@stud.fit.vutbr.cz

Jméno Příjmení xlogin03@stud.fit.vutbr.cz

...

¹Praktické aspekty vývoje software - <http://ivs.fit.vutbr.cz/>

S tímto způsobem registrace se pojí velká rizika chybovosti, ve většině případů z nepozornosti studentů. Mezi nejčastější chybové stavy patří:

- ponechání části jmen studentů ze vzoru - odesílatel e-mailu zapomene obecný tvar jména nahradit za ten skutečný, uvedeno obecné *Jméno Příjmení*,
- ponechání loginů ze vzoru - místo správného loginu bývá uveden vzorový tvar *xlogin00*,
- překlepy v uvedených údajích,
- vynechání diakritiky,
- nedodržení vzorového e-mailu, odeslání registrace vlastním způsobem,
- nedodržení pokynů - například je dáno, že počet členů v týmu je možný v rozmezí 3 až 4 členů a přihlásí se pětičlenný tým. V takovém případě musí vyučující studenty na tento fakt upozornit a vyčkat na validní registraci

Počínaje akademickým rokem 2015/2016 byla tato e-mailová registrace označena jako sekundární způsob, využitý pouze při komplikacích nového způsobu a pozdním přihlašování. Náhradou je nově implementovaná funkcionalita v informačním systému fakulty². Studentům je umožněno vytvářet si týmy a nastavovat jejich vlastnosti. Tedy určit, kdo bude vedoucím, a zadat název týmu. Vedoucí má oprávnění si své členy do týmu manuálně přidat, ale i odebrat v případě, kdy se mu do týmu neoprávněně přidá nějaký student, kterého nezná - každý student má možnost se do libovolného týmu přihlásit. Vyučující je oprávněn týmy upravovat, určovat počáteční datum a čas tvorby týmů a konečné datum a čas, kdy budou registrace uzavřeny. Dále může nastavit limitní počet členů globálně pro týmy. Je tedy zaručeno, že nenastane komplikace s větším počtem studentů přihlášených do týmu, než je povoleno. Dalším nastavitelným atributem je maximální počet celkových týmů, které mohou být registrovány, nebo povolit automatické generování názvu týmů jednotlivým studentům na základě jejich loginů.

V informačním systému jsou pro vyučujícího zobrazitelné dvě tabulky - *seznam registrovaných týmů*, jehož ukázkou vyobrazuje obrázek 2.1, a *seznam všech zařazených studentů do jednotlivých týmů* viz obrázek 2.2. Pro to, aby měl vyučující všechny potřebné informace o týmu, je třeba kombinovat data z obou tabulek. V současné době není umožněn export do .csv³, či jiného, strojově zpracovatelného formátu. Vyučující je tedy stále nucen sestavovat si seznam týmů a jejich členů kopírováním/opisováním údajů. Hlavní výhodou tohoto nového způsobu přihlašování studentů je to, že není potřeba řešit chyby způsobené lidským faktorem - nepozorností. Údaje v informačním systému jsou vždy správné - co se uvedených jmen, příjmení a loginů týče. Nedostatkem je, že informační systém neumožňuje kontrolovat minimální počet členů týmu.

Požadavkem je automatizované zpracování obou způsobů registrací.

²Informační systém FIT - <https://wis.fit.vutbr.cz/FIT/>

³Comma-Separated Values (CSV) Files - <https://tools.ietf.org/html/rfc4180>

Seznam termínů, kde se vyskytuje tato skupina										
#	Datum	Typ	Název	Min	Max	Mist	Prihl	Přihlašování od	Jak	Zobr
1	2016-05-06	variantní	Veřejná prezentace a obhajoba týmových projektů	0	0	100	0	2016-04-25 20:00:00	student	ano

Seznam zařazených studentů										Seznam nezařazených studentů									
#	Název týmu	Id	Vedoucí → detail	Login	Členů	Založil	Kdy	Založit											
1	Fitáci	2524	Mour Lukáš	xmourl00	4	xmourl00	2016-02-24 19:24:10	Zrušit											

Obrázek 2.1: Ukázka seznamu týmů v systému z pohledu vyučujícího

Otevřený předmět										
Praktické aspekty vývoje software - IVS/2015L										
Název skupiny týmů					Název skupiny týmů anglicky					
Týmy na projekt z IVS										
Studenti zařazení do týmů:										
#	Jméno	Login	Fa	Typ	St	Rč	Tým	Body	Zkouška	
1	Mour Lukáš	xmourl00	FIT	B	1	3	Fitáci	0	0000-00-00	

Obrázek 2.2: Ukázka seznamu zařazených studentů v systému z pohledu vyučujícího

2.2 Evidence hodnocení studentských aktivit

Pro výpočet bodového hodnocení z jednotlivých částí za danou aktivitu je manuálním způsobem vytvářena tabulka v kancelářském programu LibreOffice Calc (ukázka tabulky viz obrázek 2.3). Přípravná fáze tvorby tabulky je pro vyučujícího časově náročná. Musí se naplnit studenty kopírováním údajů z vytvořeného seznamu týmů popisovaného výše a doplnit vzorce. V IVS se ve vzorcích používají konstanty, které jsou závislé na počtu členů týmů a vzorce je tedy nutné pozorně kontrolovat, zda jsou korektní pro daný tým. Dalším aspektem tvorby tabulek je zvýraznění součtů, tvorba formátování - vizuálního odlišení pro následnou, příjemnější práci s tabulkou. Požadovaným aspektem je vytvořit obdobnou tabulku, avšak automatizovaně. Při hodnocení rozsáhlejších aktivit nebo projektů s velkým počtem studentů je běžnou praxí, že se na hodnocení podílí větší počet lidí. Tato situace se nemusí týkat pouze projektů, ale i půlsestrálních či závěrečných zkoušek. Jednotliví vyučující zašlou svoji část vyplněné tabulky hlavní kompetentní osobě, která hodnocení musí sjednotit pro import do informačního systému. Současným stavem je neefektivní, pracné kopírování hodnot mezi tabulkami. Dalším požadavkem je tedy možnost sloučení obsahu z více tabulek.

=IF(SUM(K2:R2)>12,12,SUM(K2:R2))																					
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R			
1	Tým	Jméno a příjmení	Členové					St. celk.	Tým celk.	Plán	S.t.	p	a	u	K.k.	P.k.	V	Z	K	S	
2	1	Název týmu	Jméno vedoucího	login1	login2	login3	login4		46	5	1	1	1	1	0	1	1	0	0	0	
3		Jméno a příjmení 1	login1					26		2											
4		Jméno a příjmení 2	login2					26		2											
5		Jméno a příjmení 3	login3					26		2											
6		Jméno a příjmení 4	login4					26		2											
7	2	Název týmu	Jméno vedoucího	login1	login2	login3			42	9	1	2	2	1	1	1	1	1	0	0	
8		Jméno a příjmení 1	login1					27		3											
9		Jméno a příjmení 2	login2					27		3											
10		Jméno a příjmení 3	login3					27		3											
11	Tým	Jméno a příjmení	Členové					St. celk.	Tým celk.	Plán	S.t.	p	a	u	K.k.	P.k.	V	Z	K	S	
12																					
13																					

Obrázek 2.3: Ukázka části využívané tabulky pro hodnocení projektu v kurzu IVS

2.3 Rozbalování a kontroly odevzdaných řešení

Po vypršení doby, kterou studenti mají na vypracování zadaných aktivit a odevzdání řešení do informačního systému, si může daná řešení vyučující hromadným způsobem stáhnout ve formátu .zip. Tento soubor obsahuje jednotlivé adresáře studentů, kteří odevzdali nějaké soubory, dle jejich loginů a uvnitř každého se nachází odevzdaná řešení. Nutným úkolem vyučujícího je rozbalení všech archivů pro přímý přístup k jednotlivým souborům. Následuje kontrola formální správnosti a kompletnosti odevzdaných řešení. V tomto případě je primárním požadavkem zautomatizovat rozbalení všech archivů a provádět potřebné formální kontroly.

Při analýze této problematiky byl zjištěn fakt, že studenti často nedodržují typ odevzdaného archivu. Příkladem problému může být požadavek na odevzdání .tar. Student, který neví, jak se daný archiv vytváří, vytvoří odlišný typ archivu ve formátu .zip a pouze změní příponu, čímž však vnitřní strukturu nezmění a nastane problém při rozbalování.

V kurzu IVS je součástí odevzdání jednoho z projektů i GIT repozitář⁴, instalátor/odinstalátor aplikace, zdrojové kódy, soubory pro generování dokumentace apod. Informační systém fakulty má limit velikosti odevzdaných souborů ve výši 2 MB, což není v tomto případě dostačující. Proto je stanoveno, že co se studentům nevejde do archivu, který odevzdávají v IS FIT, uloží do určeného adresáře na serveru. Je tedy nutná kompletace souborů a adresářů odevzdaných na dvě místa.

2.4 Rozesílání hodnocení

Zveřejňování výsledného hodnocení studentům za jejich aktivity je v IVS prováděno ve dvou krocích:

- **Výsledné bodové hodnocení** - získané body vyučující vkládá do informačního systému, kdy je nutné ručně vytvářet .csv soubor z tabulky hodnocení pro import - vložení.
- **Detaily hodnocení, komentář** - studenty je vhodné informovat o jednotlivých položkách hodnocení a popisu, jak hodnocení probíhalo, včetně detailního rozepsání dílčích částí, získaných bodů a slovních komentářů hodnotícího. V současné době není možné vkládat právě toto hodnocení do informačního systému hromadným způsobem. Aktuální praxí je průběžné ukládání do externího textového souboru a poté manuální tvorba e-mailových zpráv a jejich rozesílání jednotlivým studentům.

⁴Git - distributed version control system - <https://git-scm.com/>

Kapitola 3

Použité technologie

Pro implementaci všech skriptů byl zvolen programovací jazyk Python. Skripty pro tvorbu tabulek hodnocení a spojování tabulek budou používat vývojové API kancelářského balíku LibreOffice, skripty pro kontrolu kompletnosti a formální správnosti souborů pak prvky skriptovacího jazyka Bash.

3.1 Python

Python je moderním, vysokoúrovňovým skriptovacím programovacím jazykem, mezi jehož hlavní přednosti patří dynamická kontrola datových typů a podpora objektově orientovaného programování [3]. Interpret Pythonu, společně s rozsáhlou standardní knihovnou, jsou zcela volně šiřitelné a dostupné, včetně nástrojů, modulů třetích stran a dokumentací¹. Jedná se o multiplatformní jazyk - podpora v Microsoft Windows, Unixových systémech i Apple Mac OS X [6]. Ve většině Linuxových distribucích je Python součástí standardní instalace.

Tento programovací jazyk byl vytvořen v roce 1990-1991 Guido van Rossumem v Matematickém centru Stichting v Nizozemsku jako následník jazyka nazývaného ABC. Python je vyvíjen s důrazem na pragmatičnost, vývoj jeho verzí je spíše evoluční [5].

Jednou z nejrozšířenějších verzí Pythonu je verze 2.7. Současnou aktuální verzí je Python 3, konkrétně 3.5.1. Verze 3.0 byla vydána 3. prosince 2008 a je zpětně nekompatibilní.

3.2 Python - LibreOffice

3.2.1 LibreOffice

LibreOffice je volně dostupný kancelářský balík, který sdružuje více nástrojů - mezi nejznámější patří editor textových dokumentů Writer, tabulkový procesor Calc, nástroj pro práci s prezentacemi Impress. Mezi další integrované nástroje patří Draw, Base a Math². Jedná se o multiplatformní balík nástrojů pro Microsoft Windows, Unix a Apple Mac OS X. Ve většině Linuxových distribucí je LibreOffice součástí standardní instalace.

Projekt LibreOffice založili bývalí vývojáři projektu OpenOffice, ze kterého vychází a má s ním mnoho společného.

Aktuální dostupnou verzí je LibreOffice 5.1.2.

¹Python - <https://www.python.org/>

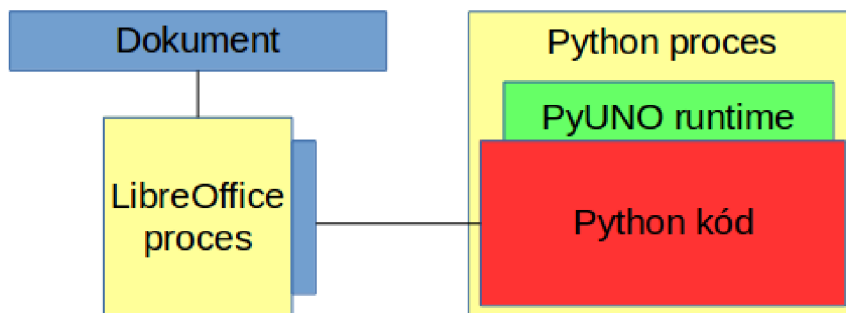
²LibreOffice - <http://www.libreoffice.org/>

3.2.2 Python-UNO

Pro spolupráci mezi programovacím jazykem Python a kancelářským balíkem LibreOffice se využívá tzv. Python-UNO (PyUNO) brigde [8], který umožňuje využívat standardní vývojové API LibreOffice - vytváří tzv. prostředníka mezi nativním jazykem LibreOffice/OpenOffice BASIC [7] a jazykem Python. Tím umožňuje využívat prvky a konstrukce tohoto jazyka pro tvorbu maker a skriptů (například pro čtení/zápis dat LibreOffice dokumentu) přímo v programovacím jazyce Python.

Pro účely implementace je vhodné využít metodu externích spustitelných skriptů v jazyce Python, mimo samotný proces kancelářského balíku, kdy si každý skript danou komunikaci zajistí.

Samotná komunikace mezi procesem Pythonu a procesem balíku LibreOffice je realizována na bázi socketů, kdy je nutné v první řadě spustit v systému proces LibreOffice, který povoluje komunikaci na daném portu jako server. Pro dané použití je třeba mít v systému nainstalovanou knihovnu Python-UNO (importovatelnou do skriptu jako modul *uno*). Znázornění komunikace popisuje obrázek 3.1. V mnoha případech je již knihovna dostupná v systému - společně s balíkem LibreOffice. Na operačních systémech založených na systému Debian lze knihovnu nainstalovat jako: *python3-uno*³.



Obrázek 3.1: Ukázka spojení LibreOffice - Python

Příklad nutného spuštění procesu LibreOffice v terminálu systému Linux:

```
soffice "--accept=socket,host=localhost,port=2002;urp;"
```

3.3 Bash

Bash (Bourne again shell) je jedním z tzv. shellů, tedy interpretů příkazového řádku (Command Line Interpreter - CLI) unixových systémů, obsahující mnoho nástrojů [10] a zároveň jedním ze skriptovacích jazyků. Jedná se, mimo jiné, o rozhraní mezi uživatelem a systémem [11]. Vývoj začal již v roce 1988. V současné době je Bash dostupný na unixových systémech, avšak na konferenci Microsoft Build 2016 bylo oznámeno, že do operačního systému Windows 10 bude v blízké aktualizaci (build 14316 [1]) přidán subsystém Linuxu, umožňující nativní podporu shellu Bash [4].

³python3-uno package details - <https://packages.debian.org/sid/python3-uno>

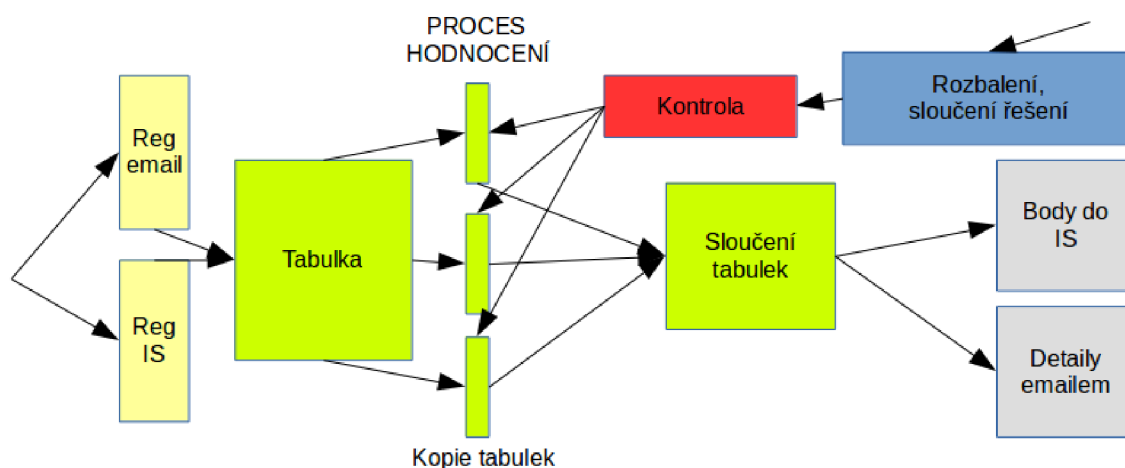
Kapitola 4

Návrh řešení

Navrhovaným způsobem řešení je tvorba sady skriptů, tvořící systém pro hodnocení studentských aktivit.

4.1 Konceptuální model systému

Počáteční fází je registrace studentů na týmový projekt, která je možná dvěma způsoby: registrace e-maily nebo přes informační systém fakulty. Oba způsoby registrací by měly tvořit jednotný, mezi sebou kompatibilní výstup. Navazující fází bude tvorba tabulky hodnocení, pro kterou bude jedním ze vstupů výstup ze zpracovaných registrací. Poté bude možné vygenerovanou tabulku rozeslat více vyučujícím, kteří po rozbalení, kontrole formální správnosti a kompletnosti odevzdaných souborů projekty zhodnotí a s příslušnou vyplněnou částí, zašlou zpět hlavní, kompetentní osobě. Následuje sloučení tabulek. Poslední část by měla být schopna pracovat právě s automatizovaně vytvořenou tabulkou hodnocení, získat z ní potřebná data pro automatické vytvoření .csv souboru pro import bodového hodnocení do informačního systému a tvorbu detailního výpisu hodnocení, který se automatizovaně rozešle studentům e-mailem. Schéma modelu systému zobrazuje obrázek 4.1.



Obrázek 4.1: Ukázka modelu systému

4.2 Automatizované zpracování registračních e-mailů

Skript pro automatizované zpracování registračních e-mailů s přihláškami na týmové projekty by měl být schopen na základě vstupu - adresáře se staženými e-maily, vytvořit výstup - seznam týmů a jejich členů v přehledném, dále strojově zpracovatelném textovém formátu. Registrační e-maily by měly být ve standardizovaném, běžně používaném formátu (Internet Message Format [9]).

Ukázka části zdrojového souboru e-mailové zprávy:

```
...
Message-ID: <20160425141755.44756tjrnatx5ng3@email.fit.vutbr.cz>
Date: Mon, 25 Apr 2016 14:17:55 +0200
From: Mour =?iso-8859-2?b?THVr4bk=?= <xmourl00@stud.fit.vutbr.cz>
To: Mour =?iso-8859-2?b?THVr4bk=?= <xmourl00@stud.fit.vutbr.cz>
Subject: Predmet zpravy
...
Content-Type: text/plain;
charset=ISO-8859-2;
...
Samotny text zpravy
```

Ukázka jednoho řádku navrhovaného výstupního formátu:

```
Navez_tymu;;Login_clen1 Login_clen2 ...
```

Výstupní formát byl takto zkonstruován z důvodu jasného odlišení části s názvem týmu a částí s loginy členů. V běžné praxi dochází ke kreativním názvům týmů včetně středníků, HTML entit, čísel apod.

Tento skript bude pracovat s e-maily od studentů, tedy s daty, u kterých není zaručena správnost, musí se počítat s chybami vzniklými z nepozornosti, které byly popisovány v kapitole 2.

Jednou z předností skriptu by měla být právě schopnost zotavovat se z takových chyb. Loginy a jména uvedené v těle zprávy je možné kontrolovat se správnými daty v informačním systému - studenti zapsaní v kurzu/u dané aktivity. Soubor je v informačním systému snadno dostupný ve formátu .csv.

Pro důkladnější kontroly lze využít logický koncept e-mailové zprávy, tedy povinnost studenta (vedoucího týmu, který e-mail zasílá) vyplnit předmět e-mailu - loginy všech členů týmu včetně svého, nebo druhý aspekt - odeslat registrační e-mail v kopii všem členům týmu. E-mailová adresa uvedená v kopii obsahuje studentský login, lze ji tedy využít taktéž pro kontrolu.

Bude vhodné sestavit algoritmus na analýzu kombinace login - jméno - příjmení. Pokud se v e-mailu, například v těle zprávy, bude vyskytovat jeden chybně uvedený údaj (login/příjmení/jméno) u člena týmu, avšak login uvedený u příjemce kopie zprávy/předmět zprávy se bude shodovat s údaji z informačního systému, lze relativně bezpečně provést autokorekci bez zásahu uživatele. Chyby ve výstupu - špatné složení týmu, nekompletní tým apod. jsou fatální a neakceptovatelné. Proto, pokud bude e-mail obsahovat dva a více rozdílných údajů u jedné registrující se osoby, které musí být korespondující, nebude možné takový e-mail úspěšně automatizovaně zpracovat a bylo by vhodné jej vytržít a oznámit uživateli, že

bude potřebný manuální zásah. Například vytvořením adresáře */NEZPRACOVANO*, kam by se zkopírovaly ty zprávy, které nemohly být zpracovány. Speciálním požadavkem pro kurz IVS je přidání přepínače (volitelného parametru) skriptu, který upraví výstupní soubor dle požadovaných konvencí pro použití k tvorbě skupin a oprávnění na školním serveru. Jedná se o úpravy názvů týmů, který může obsahovat nepovolené znaky.

Pravidla úprav:

- Odstranění diakritiky
- Odstranění apostrofů
- Odstranění znaku vykřičníku nebo bílého znaku na konci názvu týmu
- Nahrazení všech znaků mimo množinu A-Z, a-z, 0-9 a mimo znaků / (lomeno), - (pomlčka), za _ (podtržítka)
- Odebrání všech podtržitek na začátku a na konci názvu týmu

Příklad provedené úpravy:

Původní název týmu: Fiťáci A > F, upravený: Fitaci_A__F

4.3 Zpracování registrací z IS FIT

Úlohou skriptu pro zpracování registrací z IS FIT by mělo být zpracování registrací studentů z informačního systému a vytvoření identického, kompatibilního výstupu jako v případě e-mailových registrací. S tím rozdílem, že není třeba dbát na kontroly informací, protože je jejich správnost garantována. Skript by měl být schopen získat si údaje z tabulky zaregistrovaných týmů i z tabulky se seznamem zařazených studentů do týmů. Nejlepší cestou, jak získat informace, je stáhnout si webové stránky s tabulkami a ze zdrojového kódu v HTML provést extrahování potřebných dat. Zjednodušený příklad reprezentace studenta ve zdrojovém kódu stránky s tabulkou zařazených studentů je vyobrazen na obrázku 4.2. Zajímavostí, zároveň i možným potenciálním problémem tabulky seznamu týmů, je umístění názvu týmu, který je součástí atributu značky HTML `<input>` viz obrázek 4.3.

```
<tr bgcolor="#dfe7cf" valign="top"><td class="dataquery" align="right">1</td>
<td class="dataquery"><a href="https://wis.fit.vutbr...&pid=32288&cid=18">Mour Lukáš</a></td>
<td class="dataquery">xmourl00</td>
<td class="dataquery">FIT</td>
...
<td class="dataquery"><a href="https://wis.fit.vutbr.cz/FIT/db/vyuka/ucitel/course...">Fiťáci</a></td>
...
</tr>
```

Obrázek 4.2: Ukázka zdrojového kódu tabulky seznamu zařazených studentů do týmů

Aby byl zachován ekvivalentní výstup jako v případě předchozího skriptu, bude též implementován speciální přepínač pro použití k tvorbě skupin a oprávnění na školním serveru dle pravidel popisovaných v kapitole 2.

```

<td style="padding:0"><input onchange="form changed()" class="forminput" size="32"
name="form v24_input 3 2526" autocomplete="off" required=""
placeholder="vyplňte název týmu" value="Fifáci" type="text"></td>
<td>2524</td>
<td><a href="https://wis.fit.vu...">Mour Lukáš</a></td>
<td>xmourel00</td>
<td align="center">4</td>
<td>xmourel00</td>
<td>2016-02-24 19:24:10</td>

```

Obrázek 4.3: Ukázka zdrojového kódu tabulky seznamu týmů

4.4 Tvorba tabulky hodnocení

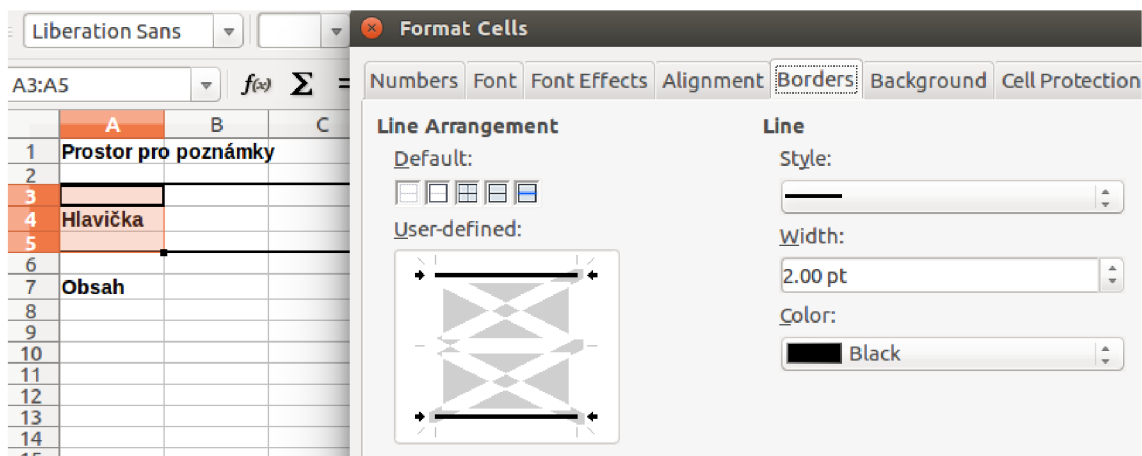
Tento skript by měl za úkol na základě šablony vygenerovat výslednou tabulku hodnocení. Šablonou se rozumí soubor LibreOffice Calc - sešit tabulkového procesoru, do kterého se umístí definované zástupné sekvence, vzorce, formátování pro první tým v případě týmového projektu, nebo pro jednotlivce v případě projektu individuálního. Na základě této šablony dojde ke generování celé tabulky hodnocení. Součástí bude načtení seznamu týmů z minulé fáze - registrací (v případě týmového projektu), seznamu studentů a šablony, její analýza a následné n-násobné zopakování pod sebe pro každý tým/jednotlivce se seznamu v závislosti na tom, zda je projekt týmový nebo individuální.

Tvorba tabulky hodnocení by měla mít univerzální charakter pro použití nejen pro projekty, ale například i pro půlsestránní/závěrečné zkoušky. Dále by bylo vhodné vyhradit jeden list (vždy ten první) sešitu LibreOffice Calc k sumarizaci hodnocení, tedy pro seznam všech zapsaných studentů na projekt/ve výukovém kurzu. Tento list by sdružoval výsledné body z jednotlivých listů hodnocení a mohl by být využit pro snadný export hodnocení zpět do informačního systému.

Speciální funkcionalitou by mělo být zvýšení komfortu při pozdější práci s tabulkou, kupříkladu automatické opakování hlavičky po určitém počtu řádků, aby byl jasný daný význam sloupce a nemuselo se pokaždé tabulkou posouvat. Dále pak podbarvování každého druhého týmu/jednotlivce v pořadí. Tvorba tabulky se drží konceptu a zvyklostí používaných při manuální tvorbě tabulky v kurzu IVS. Je patrné, že například podbarvování nemusí vyhovovat každému uživateli, proto by bylo vhodné implementovat možnost jeho vypnutí nebo změny barvy. Jako výchozí by se používala zlatožlutá barva.

Součástí návrhu je rozčlenění každého listu šablony do tří základních prvků:

- **Prostor pro poznámky** - Jedná se o volitelnou součást na úplném začátku tabulky. Ve většině případu je nutné do tabulky vložit nějaké referenční hodnoty, na které se vzorce v tabulce budou odkazovat, nebo detailnější popisky jednotlivých položek hlavičky, pokud s tabulkou bude manipulovat další osoba, která ji nevytvořila.
- **Hlavička tabulky** bude tvořit dělicí prvek mezi poznámkovým prostorem a vlastním obsahem. Pokud šablona poznámkový prostor obsahovat nebude, bude hlavička počátečním prvkem. Pravidlem pro její ohraničení bude plné ohraničení o minimální šířce 2 px z horní i spodní strany viz obrázek 4.4. Hlavička bude moci obsahovat libovolný počet řádků, ale v každém případě bude muset začínat v prvním sloupci. Bylo by vhodné implementovat kontrolu formální správnosti a umístění hlavičky.
- **Obsah tabulky** - pod hlavičkou bude následovat obsah, kam se umístí zástupné sekvence a zástupné vzorce pro definici jednoho týmu/jednotlivce, na jehož základě bude provedeno generování tabulky pro reálná data.



Obrázek 4.4: Ukázka členění listu tabulky

4.4.1 Obecná pravidla pro zástupné sekvence/vzorce

V šabloně pro generování budou umístěny definované zástupné sekvence a zástupné vzorce, podle kterých se vygenerují reálná data. Zástupnou sekvencí se rozumí například $\$PORADI_TYM$, která bude pro první vygenerovaný tým nahrazena hodnotou 1, pro druhý 2 apod. Obecným pravidlem pro vkládání řetězců do tabulky (jména, loginy, názvy týmů, ...) bude uvození znakem \$ nebo \$\$\$. V případě, kdy se daný řádek vygeneruje pouze jednou pro tým, budou všechny obsažené sekvence na tomto řádku uvozeny pouze \$. Za situace, kdy se má řádek několikrát zopakovat pod sebe (tzv. rozgenerovat), budou sekvence uvozeny \$\$\$. Jednotlivé podporované sekvence budou uvedeny níže u každého režimu použití tabulky.

Zástupné vzorce budou začínat znaky ~= nebo ~=. Využití zdvojeného uvození je identické, jako v případě zástupných sekvencí. Příkladem zástupného vzorce může být ~=A1+B1. Jedním z důvodů uvozování vzorců je ten, že pokud by začínaly samotným znakem =, LibreOffice interpretuje obsah buňky jako vzorec nebo funkci přímo a bude zobrazovat pouze výsledek. Nebylo by na první pohled patrné, co uživatel do dané buňky nadefinoval.

Vzorci většinou pracují s buňkami v tabulce. Uživatel provede obecnou definici týmu a skript bude muset dynamicky měnit číslo řádku buňky, aby daný vzorec dával smysl pro každý tým. Musí pro to existovat pravidla, která jsou popsána v tabulce 4.1. Ne všechny způsoby jsou podporovány v každém režimu tabulky, bude opět níže popsáno u každého režimu.

Zápis	Příklad	Význam
SloupecRadek	A1	jedná se o statickou buňku, hodnota řádku nikdy nebude aktualizována
Sloupec#Radek#	A#1#	hodnota řádku, která se bude inkrementovat s každým novým vygenerovaným řádkem (vždy o 1)
Sloupec&Radek&	A&1&	hodnota zůstane fixní pro tým (typicky pokud chceme provést rozpočet bodů týmu na studenta), s dalším týmem se hodnota zvýší o patřičnou hodnotu

Tabulka 4.1: Pravidla pro zápis buňek v zástupných vzorcích

V šabloně bude možné použít vzorce a funkce dle standardních konvencí tabulkového procesoru LibreOffice Calc - průměry, podmínkové konstrukce, zaokrouhlovací funkce apod. Smysluplným formátováním je tučné zvýraznění obsahu buňky, proto pokud bude v šabloně buňka tučná - každá buňka, právě z této vygenerovaná, bude též tučná. Příkladem mohou být výsledné součty.

4.4.2 Sumarizační list

Ve všech případech šablony bude první list sešitu vyhrazen pro sumarizaci hodnocení. Při vytváření šablony je to třeba zohlednit. Bude se jednat o seznam studentů a k nim odkazy na následující listy s projekty. Na každém řádku bude umístěn jeden student. Jedná se o rozgenerování stejných řádků pod sebe, je tedy nutné uvození \$\$ u zástupných sekvencí a ~~= u vzorců. Podporované zástupné sekvence jsou vyobrazeny v tabulce 4.2.

\$\$PORADI_STUD	zástupná sekvence, která bude nahrazena pořadovým číslem studenta (číslováno od hodnoty 1)
\$\$PR_JM_STUD	sekvence pro nahrazení za celé příjmení a jméno studenta
\$\$LOGIN_STUD	sekvence pro umístění uživatelského jména (loginu) studenta
\$\$BODY=nazev_list	sekvence slouží pro zobrazení odkazovaných bodů, tedy výsledných bodů, které student získal v určitém projektu, který je zanesen v tabulce v tom stejném sešitu LibreOffice Calc, ale v jiném listu Příklad: \$\$BODY=Projekt odkáže na buňku v listu Projekt, která je označena \$\$BODY=vzorec/hodnota, inteligentně dle loginu studenta

Tabulka 4.2: Tabulka podporovaných zástupných sekvencí sumarizačního listu

Vzhledem k faktu, že šablona bude moci obsahovat více listů s projekty a vždy první sumarizační list, bude možné právě do tohoto sumarizačního listu umísťovat i zástupné vzorce například pro součty apod. pro kontrolu, zda například student nepřesáhl z nějakého důvodu maximální bodovou hranici předmětu 100 bodů a jiné případy.

Podporované definice buněk zástupných vzorců viz tabulka 4.3.

Zástupná sekvence	Popis	Příklad
SloupecRadek	fixní, neměnná buňka v celé tabulce	A1
Sloupec#Radek#	buňka, která inteligentně mění hodnotu řádku	A#1# definováno na řádku 1 bude na řádku 6 rovno A6

Tabulka 4.3: Podporované definice buněk zástupných vzorců

V buňce, kde bude umístěna sekvence \$\$BODY= by bylo vhodné provádět podbarvení pozadí buňky, zda je daná reference aktivní (tedy zda je student součástí projektu na daném odkazovaném listu). Červené podbarvení může zřetelně poukázat, že se student v projektu nenachází - nebyl vygenerován - zapomněl provést registraci na projekt apod. Bílé pozadí

buňky bude použito v případě, kdy je reference aktivní. Obrázek 4.5 znázorňuje návrh šablony sumarizačního listu včetně příkladu očekávaných vygenerovaných dat:

- řádek 1-5 je definice šablony pro sumarizační list
- řádky 8 a 9 znázorňují očekávaná vygenerovaná data
- v šabloně je na 5. řádku definován zástupný vzorec $\sim=SUM(F\#5\#:I\#5\#)$, pro 8. řádek by měl být vzorec vygenerován jako $=SUM(F8:I8)$
- $$$BODY=Projekt1$ bude odkazem na list tabulky s názvem *Projekt1* - na buňku obsahující výsledné body, opět označenou $$$BODY$ patřící danému studentovi - párování dle loginů. Červené podbarvení značí, že pro daného studenta neexistuje odkaz (není součástí hodnocení projektu)

	A	B	C	D	E	F
1	Prostor pro poznámky					
2						
3						
4	Pořadí	Jméno a příjmení	Login	Body celkem		Body – list Projekt1
5	$$$PORADI_STUD$	$$$PR_JM_STUD$	$$$LOGIN_STUD$	$\sim=SUM(F\#5\#:I\#5\#)$		$$$BODY=Projekt1$
6						
7						
8		1 Prijmeni01 Jmeno01	xlogin01	0		0
9		2 Prijmeni02 Jmeno02	xlogin02	0		

Obrázek 4.5: Ukázka sumarizačního listu včetně vzorových vygenerovaných dat

Každý následující list sešitu za tím sumarizačním bude moci obsahovat definici konkrétního projektu/zkoušky.

4.4.3 Týmový projekt

Týmový projekt bude možné definovat dvěma způsoby. Automatickým rozgenerováním členů nebo zvoleným umístěním sekvencí do šablony.

a) Automatické rozgenerování členů - pokud uživatel vyžaduje konzervativní způsob, tedy jeden řádek pro tým a pod ním řádky pro jednotlivé členy (každý člen na jeden řádek) viz obrázek 4.6, bylo by možné tvorbu šablony urychlit tím způsobem, že by se řádek pro členy nadefinoval pouze jednou, a skript by tento řádek automaticky vygeneroval tolikrát, kolik členů týmu daný tým má. Pro zástupné sekvence tedy bude platit, že řádek pro tým bude definován se zástupnými sekvencemi uvozenými $$$$, vzorce $\sim=$, a řádek pro členy se bude rozgenerovávat - sekvence musí být uvozeny $$$$ respektive $\sim=$.

	A	B	C	D	E	F	G
1	Pořadí	Tým	Student celkem	Tým celkem	Ukol1	Ukol2	Ukol3
2		1 Fitáci		15	10	3	2
3		Jan Novák		5			
4		Karel Novák		5			
5		Pavel Novák		5			

Obrázek 4.6: Ukázka konzervativního způsobu používání tabulky hodnocení

Podporované zástupné sekvence popisuje tabulka 4.4. Zástupné sekvence buněk vzorců pak znázorňuje tabulka 4.5.

Zástupná sekvence	Popis
\$PORADI_TYM	sekvence bude nahrazena za pořadové číslo týmu, počínaje hodnotou 1 pro první tým
\$NAZEV_TYM	nahrazení za skutečný název týmu
\$LOGIN_VEDOUCI	určeno pro nahrazení za login vedoucího
\$PR_JM_VEDOUCI	pro nahrazení za příjmení a jméno vedoucího týmu
\$KOMENTAR	sekvence, která do dané buňky vepíše řetězec “Komentář:” pro následné doplnění slovního hodnocení týmu do dané buňky
\$POCET_CLENU	nahrazeno za číselnou hodnotu počtu členů týmu
\$\$LOGIN_CLEN	sekvence pro nahrazení za login člena
\$\$PR_JM_CLEN	sekvence pro nahrazení za příjmení a jméno člena týmu
\$\$BODY=vzorec	sekvence pro vytvoření odkazu daného studenta do sumarizačního listu tabulky (výsledné body)

Tabulka 4.4: Podporované zástupné sekvence týmového projektu definovaného rozgenerováním

Zástupná sekvence	Popis	Příklad
SloupecRadek	fixní, neměnná buňka v celé tabulce	A1 ve všech případech
Sloupec#Radek#	buňka, která inteligentně mění hodnotu řádku	A#1# definováno na řádku 1 bude na řádku 6 rovno A6
Sloupec&Radek&	buňka, která bude fixní pro jeden tým, s dalším týmem se hodnota nahradí tak, aby dávala smysl	A&1& zůstane pro prvního i druhého člena týmů stejná (typicky pro rozpočet bodů týmu na člena)

Tabulka 4.5: Zástupné sekvence buněk vzorců

Výhodou tohoto způsobu je úspora místa v tabulce, kdy nevznikají prázdné řádky, skript si bude dynamicky generovat tolik řádků pro členy, kolik jich tým má. Nevýhodou je naopak nižší možnost přizpůsobení.

Ukázka možného tvaru šablony a očekávaných vygenerovaných dat viz obrázek 4.7.

Pořadí	Název týmu	Jméno a příjmení	Login	St.celk.	Plán
\$PORADI_TYM	\$NAZEV_TYM	\$PR_JM_VEDOUCI	\$LOGIN_VEDOUCI		~=IF(SUM(K#5#:R#5#Y)\$POCET_CLENU
		\$\$PR_JM_CLEN	\$\$LOGIN_CLEN	\$\$BODY=SUM(J#6#:AG#6#)	---=ROUNDUP(J&5&/\$POCET_CLENU)
1	A	Prijmeni01 Jmeno01	xlogin01		0
		Prijmeni01 Jmeno01	xlogin01		0
		Prijmeni02 Jmeno02	xlogin02		0
		Prijmeni03 Jmeno03	xlogin03		0

Obrázek 4.7: Ukázka definice šablony týmového projektu vytvořeného rozgenerováním a očekávaná data

b) **Umístění všech sekvencí** - některým uživatelům nemusí způsob, že všichni členové mají stejnou strukturu, vyhovovat, proto je vhodné přidat možnost dané řádky si více přizpůsobit viz obrázek 4.8. Netýká se to pouze různé struktury řádků studenta, ale například i definice řádku pro tým až pod jeho členy.

	A	B	C	D	E	F	G	H	I
1	Pořadí	Tým			St celkem	Tým celkem	Ukol1	Ukol2	Ukol3
2		1 Fit áci				15	10	3	2
3		Jan Novák			5				
4			Karel Novák		5				
5				Pavel Novák	5				

Obrázek 4.8: Ukázka různorodého umístění jmen členů týmu v tabulce

Rozdílem jsou sekvence pro jednotlivé členy, které se vyznačují pouze jedním uvozením \$. Podporované sekvence znázorňuje tabulka 4.6.

Princip bude takový, že první výskyt \$LOGIN_CLEN bude nahrazen za login prvního

Zástupná sekvence	Popis
\$PORADI_TYM	sekvence bude nahrazena za pořadové číslo týmu, počínaje hodnotou 1 pro první tým
\$NAZEV_TYM	nahrazení za skutečný název týmu
\$LOGIN_VEDOUCI	určeno pro nahrazení za login vedoucího
\$PR_JM_VEDOUCI	pro nahrazení za příjmení a jméno vedoucího týmu
\$KOMENTAR	sekvence, která do dané buňky vepíše řetězec "Komentář:" pro následné doplnění slovního hodnocení týmu do dané buňky
\$POCET_CLENU	nahrazeno za číselnou hodnotu počtu členů týmu
\$LOGIN_CLEN	sekvence pro nahrazení za login člena
\$PR_JM_CLEN	sekvence pro nahrazení za příjmení a jméno člena týmu
\$BODY=vzorec	sekvence pro vytvoření odkazu daného studenta do sumarizačního listu tabulky (výsledné body)

Tabulka 4.6: Podporované zástupné sekvence týmového projektu definovaného umístěním všech sekvencí

člena týmu, \$PR_JM_CLEN za příjmení a jméno prvního člena týmu. Do šablony bude potřeba zanést tolik opakování těchto sekvencí, jaký maximální počet členů týmu může nastat. V případě IVS, kdy jsou povoleny tři, až čtyřčlenné týmy, bude v případě těch

tříčlenných docházet k plýtvání místem - tvorbě jednoho volného řádku navíc, protože šablona bude muset být zkonstruována pro podporu sekvencí pro čtyři členy. Zástupné sekvence buněk vzorců pak znázorňuje tabulka 4.7 .

Zástupná sekvence	Popis	Příklad
SloupecRadek	fixní, neměnná buňka v celé tabulce	A1 ve všech případech
Sloupec#Radek#	buňka, která inteligentně mění hodnotu řádku	A#1# definováno na řádku 1, bude na řádku 6 rovno A6

Tabulka 4.7: Zástupné sekvence buněk vzorců

Od automatického generování se vzorce liší tím, že je zmenšen počet potřebných zástupných sekvencí buněk vzorců. V tomto případě se definují všechny řádky členů týmu, tedy pro každého člena zvlášť - odpadá nutnost využívat Sloupec&Radek&.

Ukázka možného tvaru šablony a očekávaných vygenerovaných dat viz obrázek 4.9.

H15		f(x) Σ =		=SUM(J15:AG15)				
	A	B	C	D	E	F	G	H
1	Prostor pro poznámky							
2								
3								
4	Pořadí	Název týmu	Jméno a příjmení	Login				St.celk.
5	\$PORADI_TYM	\$NAZEV_TYM	\$PR_JM_VEDOUCI	\$LOGIN_VEDOUCI				
6			\$PR_JM_CLEN	\$LOGIN_CLEN				\$BODY=SUM(J#6#:AG#6#)
7			\$PR_JM_CLEN		\$LOGIN_CLEN			\$BODY=SUM(J#7#:AG#7#)
8			\$PR_JM_CLEN			\$LOGIN_CLEN		\$BODY=SUM(J#8#:AG#8#)
9			\$PR_JM_CLEN				\$LOGIN_CLEN	\$BODY=SUM(J#9#:AG#9#)
10	\$KOMENTAR							
11								
12		1 A	Prijmeni01 Jmeno01	xlogin01				
13			Prijmeni01 Jmeno01	xlogin01				0
14			Prijmeni02 Jmeno02		xlogin02			0
15			Prijmeni03 Jmeno03			xlogin03		0
16								
17	Komentář:							
18		2 B	Prijmeni04 Jmeno04	xlogin04				
19			Prijmeni04 Jmeno04	xlogin04				0
20			Prijmeni05 Jmeno05		xlogin05			0
21			Prijmeni06 Jmeno06			xlogin06		0
22			Prijmeni07 Jmeno07				xlogin07	0
23	Komentář:							

Obrázek 4.9: Ukázka možného tvaru šablony a očekávaných vygenerovaných dat při umístění všech sekvencí

4.4.4 Individuální projekt

Navrhovaný princip individuálního projektu je téměř identický, s rozdílem podporovaných sekvencí. V případě, že by se v šabloně nevyskytovaly sekvence \$\$PORADI_TYM, \$\$NAZEV_TYM, \$\$LOGIN_VEDOUCI nebo \$\$PR_JM_VEDOUCI dojde k tomu, že skript rozpozná, že se jedná o individuální projekt (pro jednotlivce) a pro generování nepoužije seznam týmů a jejich členů, ale pouze seznam zařazených studentů.

a) **Automatické rozgenerování studentů** - v případě, kdy uživateli postačuje jeden řádek na studenta, je možná definice pomocí zdvojených sekvencí \$\$, respektive ~~=.

Seznam podporovaných zástupných sekvencí viz tabulka 4.8.

Zástupná sekvence	Popis
\$\$PORADI_CLEN	sekvence bude nahrazena za pořadové číslo člena - studenta, počínaje hodnotou 1 pro prvního studenta
\$\$KOMENTAR	sekvence, která do dané buňky vepíše řetězec "Komentář:" pro následné doplnění slovního hodnocení týmu do dané buňky
\$\$LOGIN_CLEN	sekvence pro nahrazení za login člena - studenta
\$\$PR_JM_CLEN	sekvence pro nahrazení za příjmení a jméno člena - studenta
\$\$BODY=vzorec	sekvence pro vytvoření odkazu daného studenta do sumarizačního listu tabulky (výsledné body)

Tabulka 4.8: Seznam podporovaných zástupných sekvencí individuálního projektu - rozgenerování

Zástupné sekvence buněk vzorců pak znázorňuje tabulka 4.9.

Zástupná sekvence	Popis	Příklad
SloupecRadek	fixní, neměnná buňka v celé tabulce	A1 ve všech případech
Sloupec#Radek#	buňka, která inteligentně mění hodnotu řádku	A#1# definováno na řádku 1 bude na řádku 6 rovno A6

Tabulka 4.9: Ukázka zástupných sekvencí buněk vzorců

Příkladem definice šablony individuálního projektu včetně očekávaných vzorových dat viz obrázek 4.10.

Pořadí	Jméno a příjmení	Login	Body	UKOL 1	UKOL 2	Komentář
\$\$PORADI_CLEN	\$\$PR_JM_CLEN	\$\$LOGIN_CLEN	\$\$BODY=SUM(F#9#:G#9#)			\$\$KOMENTAR
1	Prijmeni01 Jmeno01	xlogin01		0		Komentář:
2	Prijmeni02 Jmeno02	xlogin02		0		Komentář:
3	Prijmeni03 Jmeno03	xlogin03		0		Komentář:
4	Prijmeni04 Jmeno04	xlogin04		0		Komentář:
5	Prijmeni05 Jmeno05	xlogin05		0		Komentář:

Obrázek 4.10: Ukázka definice šablony individuálního projektu včetně očekávaných vzorových dat

b) Umístění všech sekvencí - v případě, kdy uživateli vyžaduje více řádků na jednoho studenta, je nutná definice pomocí sekvencí \$, respektive ~= dle tabulky 4.10.

Pro zástupné sekvence buněk platí stejná pravidla, jako při rozgenerovacím způsobu, tedy pravidla viz. tabulka 4.9.

Zástupná sekvence	Popis
\$PORADI_CLEN	sekvence bude nahrazena za pořadové číslo člena - studenta, počínaje hodnotou 1 pro prvního studenta
\$KOMENTAR	sekvence, která do dané buňky vepíše řetězec "Komentář:" pro následné doplnění slovního hodnocení týmu do dané buňky
\$LOGIN_CLEN	sekvence pro nahrazení za login člena - studenta
\$PR_JM_CLEN	sekvence pro nahrazení za příjmení a jméno člena - studenta
\$BODY=vzorec	sekvence pro vytvoření odkazu daného studenta do sumarizačního listu tabulky (výsledné body)

Tabulka 4.10: Podporované zástupné sekvence

Příklad definice takovéto šablony včetně vzorových dat viz obrázek 4.11.

Pořadí	Jméno a příjmení	Login	Fáze 1		Celkem	Komentář
			Fáze 2			
\$PORADI_CLEN	\$PR_JM_CLEN	\$LOGIN_CLEN			\$BODY=D#3#+D#4#	\$KOMENTAR
1	Prijmeni01 Jmeno01	xlogin01				Komentář:
						0
2	Prijmeni02 Jmeno02	xlogin02				Komentář:
						0
3	Prijmeni03 Jmeno03	xlogin03				Komentář:
						0

Obrázek 4.11: Ukázka definice šablony včetně vzorových dat

4.5 Doplnující skripty

Dále by bylo vhodné implementovat doplňující malé skripty pro generování souborů na externí poznámky. Jednalo by se o textový soubor pro umístění slovního hodnocení studentů/týmů. Na základě vstupu (seznamu studentů) by generoval soubor s těmito studenty a u každého z nich volné místo pro vepsání libovolně dlouhého komentáře. Jednotliví studenti od sebe budou zřetelně odděleni. V případě vstupního seznamu týmů vygeneruje obdobný formát, pouze namísto jednoho loginu by zde byl umístěn název týmu a výčet loginů jeho členů. Vyplněný soubor bude využit k tvorbě e-mailů s detaily hodnocení.

Ukázka:

```
*****
xlogin00 / Nazev_tym: xlogin00 xlogin01 ...
```

```
*****
```

Další doplňující skript by měl být schopen sloučit takovéto soubory s externími poznámkami dle loginů/týmů. Pokud nastane situace, kdy projekt bude hodnotit více vyučujících, byl by nejjednodušším způsobem rozdělení souboru s externími poznámkami takový, že by se pouze daným hodnotícím osobám zaslala celá kopie, kde by každý ve své kopii vyplnil

část a následně se provedlo automatizované sloučení.

Na závěr by bylo vhodné dalším skriptem doplnit k obsahovým řádkům externích komentářů hromadně odrážky, pokud chybějí, na sjednocení stylu.

Příklad:

```
*****
xlogin00
- komentar1
- komentar2
- komentar3
*****
```

4.6 Rozbalení a kontrola pojmenování a kompletnosti odevzdaných souborů

Úlohou tohoto skriptu by měla být práce s odevzdanými soubory studentů. Tuto činnost by bylo vhodné rozčlenit do více skriptů.

4.6.1 Rozbalení souborů

Tento skript by měl být schopen rozbalit většinu známých typů archivů, tedy např.: .zip, .tar a .tar.gz. V týmovém projektu výukového kurzu IVS je požadavek i na odevzdání repozitáře GIT, a studenti často volí tvorbu archivu, který je posléze umístěn do archivu hlavního - odevzdaného. Z tohoto důvodu by bylo vhodné využít pro rozbalování rekurzivní metodu - tedy provést rozbalení všech archivů, včetně těch vnořených.

Pro správnou detekci typu archivu není vhodné se spoléhat pouze na přípony souborů. Spolehlivým řešením je detekce pomocí Bash nástroje - *file*.

4.6.2 Kontrola pojmenování a kompletnosti odevzdaných souborů

Úlohou tohoto skriptu by měla být kontrola, zda každý student odevzdal požadované soubory a mají správné jméno. Nabízí se varianta, že by uživatel vytvořil konfigurační soubor, ve kterém by byl seznam požadovaných souborů a jejich korektní pojmenování. Často se využívá pojmenování souborů pomocí studentského loginu, bude tedy potřeba jisté zobecnění na vyhledávání loginu v názvu. V tomto případě by se do konfiguračního souboru umístila zástupná sekvence *xlogin00* a skript by sám korektně kontroloval správný login dle nadřazeného adresáře, které již jsou pojmenovány loginem (struktura staženého archivu s projekty z informačního systému).

Příklad souboru s pravidly:

```
projekt.c
projekt.h
xlogin00.pdf
src/
```

Skript by měl zkontrolovat každý adresář studenta na výskyt žádaného souboru a v případě problému (např. chybné pojmenování či chybný typ souboru) vypsat chybu u daného studenta a případné kandidátní soubory, které by mohly odpovídat hledanému souboru. Pokud záznam v pravidlech končí znakem /, jedná se o adresář.

Dále by se skript mohl pokusit po minimalizaci adresářové struktury pro snazší přístup k souborům, tedy po korekci názvů, všechny soubory umístit do kořenového adresáře, nikoliv ponechat v podadresářích s loginy. Dále pak mít možnost smazat zbytečné prázdné adresáře.

Pro kurz IVS by bylo vhodné implementovat vyhledávání pouze v adresářích vedoucích týmů. Ze skriptů zpracovávající týmové registrace je dostupný seznam týmů, u kterého lze snadno určit loginy vedoucích (první v pořadí). Adresářová struktura stažených odevzdaných souborů je též dle loginů, lze tedy určit, v jakých adresářích by se mělo vyhledávat.

4.6.3 Přejmenování souborů

Tento skript by měl usnadnit vyučujícímu práci se soubory. V kurzu IVS nejsou striktně požadována jména některých odevzdávaných souborů. Příkladem může být libovolně pojmenovaná dokumentace ve formátu .pdf. Úlohou skriptu by mělo být to, že na základě konfiguračního souboru s pravidly projde adresář každého studenta (postupně pro každé pravidlo). Pokud pro dané pravidlo naleznе právě jednu shodu, provede přejmenování na nový název, který bude součástí konfiguračního souboru. Nejlepším řešením je textový formát konfiguračního souboru, ve kterém budou umístěny regulární výrazy pro nalezení shody a nový název.

Příklad:

```
.*\ .pdf DOKUMENTACE.pdf
```

Bylo by vhodné skriptu implementovat parametr hloubky zanoření adresářů. V jisté situaci může být požadavek na přejmenování obrázku .jpg - screenshotu, který je umístěn v kořenovém adresáři odevzdaného řešení. Studenti mohou odevzdat zdrojové soubory pro tvorbu dokumentace, které mohou obsahovat taktéž .jpg soubory. V tomto případě by bylo nalezeno větší množství kandidátů a přejmenování by neproběhlo. Pokud bychom v tomto případě omezili vyhledávání souborů na hloubku 1, našel by se pouze vyžadovaný soubor a přejmenování by bylo provedeno.

4.6.4 Slučování adresářů

Dalším požadavkem pro kurz IVS je schopnost sloučit více adresářů s odevzdanými soubory. V současné době je v informačním systému FIT velikostní limit pro odevzdání a pro složitější projekty je tento limit nedostačující, proto studenti odevzdávají soubory alternativní cestou - adresář na serveru. Skript by měl adresáře umět sloučit a v případě konfliktu souboru vypsat hlášení, že ho nebylo možné sloučit a uživatel bude muset takový problém vyřešit manuálním zásahem. Stavy, které mohou nastat:

- Soubor v cílové lokaci neexistuje, je tedy zkopírován
- Soubor se stejným názvem v cílové lokaci existuje. V tomto případě dochází na dotaz na velikost souboru, který je požadován na sloučení a velikost souboru v cílové destinaci.

Pokud jsou velikosti shodné, je předpokládáno, že se jedná o identický soubor a není potřebná žádná operace. V případě, že je velikost rozdílná, skript vypisuje hlášení na příkazový řádek s upozorněním na konflikt daného souboru.

4.7 Slučování tabulek od více vyučujících

Navržený skript by měl umět slučovat stejné tabulky LibreOffice Calc. Tedy v případě, kdy je do hodnocení zapojeno více vyučujících. Každý vyplní část tabulky, zašle hlavní kompetentní osobě, která všechny tabulky sloučí do jedné výsledné. V této úloze je žádoucí, aby sloučení tabulek proběhlo bez poškození hodnot, tedy v krajních chybových případech, kdy je například jeden řádek tabulky vyplněn různým způsobem ve více tabulkách, se jedná o konfliktní situaci a je třeba o tom uživatele informovat.

Možné případy, které mohou nastat:

- a) Buňka v cílové tabulce je prázdná, buňka ve zdrojové tabulce je neprázdná - dojde ke kopii této buňky
- b) Buňka v cílové tabulce je neprázdná, buňka ve zdrojové tabulce je prázdná - není provedena žádná akce
- c) Buňka v obou tabulkách má stejný obsah - opět není potřeba žádná operace, nejedná se o konflikt. V tomto případě nezáleží na formátování obsahu buňky (ohraničení, řez písma apod.).
- d) Buňka v obou tabulkách má rozdílný obsah - jedná se o konfliktní situaci - skript vypíše buňku, kterou nebylo možné sloučit pro kontrolu uživatele a zároveň vypíše úspěšně sloučené tabulky.

4.8 Rozesílání zpráv s detaily hodnocení

Vzhledem k současným možnostem informačního systému, kdy není vhodná forma pro vložení slovního komentáře hodnocení hromadným způsobem, je nejlepší variantou využít e-mailové zprávy. Tento skript by měl přímo interagovat s navrženou, automaticky generovanou tabulkou hodnocení, kdy, by se vytvořil vzor - šablona e-mailové zprávy, do které by se umístily zástupné sekvence buňky ze šablony tabulky a skript by si sám požadované položky z tabulky dokázal získat a následně odeslat správným osobám.

V této části by bylo vhodné skripty rozdělit na dva funkční celky. Hromadné odeslání e-mailu je nevratná věc, proto je žádoucí, aby měl uživatel přehled o tom, co bude odesláno.

4.8.1 Zisk dat z tabulky

Skript na extrahování požadovaných dat z tabulky. Na základě vstupu - tabulky naplněné daty, která byla vygenerovaná ze šablony tabulky a vzoru e-mailové zprávy obsahující zástupné sekvence ze šablony tabulky, by vytvořil výstup, textový soubor obsahující náhled na všechny vygenerované e-mailové zprávy oddělené určitým oddělovačem.

Obecným konceptem je šablona e-mailové zprávy pro jeden tým/jednotlivce, kdy **na prvním řádku** je pomocí zástupné sekvence definován **příjemce** (buňka šablony tabulky, kde je umístěn login - automaticky by skript doplnil @stud.fit.vutbr.cz), **na druhém předmět zprávy** a **na dalších řádcích samotný obsah**.

Příkladem:

```
{D6}
Hodnocení projektu
Dobrý den,
celkem jste získal(a): {E6} bodů.
```

V případě, kdy by se zástupná sekvence odkazovala na buňku v šabloně tabulky, která je uvozena \$\$ nebo ~~= (určená pro rozgenrování), skript by načtl obsah všech buněk patřících danému týmu - typicky celkový počet získaných bodů na člena: například: 7 7 7. Příjemcem emailu by byl vedoucí týmu a příjemci kopie ostatní členové týmu. Nastává stav, kdy by všichni členové věděli, kolik každý z nich dostal bodů. V některých případech to však není žádoucí, proto by bylo vhodné rozšířit zástupné sekvence:

```
[{D6}]
Hodnocení projektu
Dobrý den,
celkem jste získal(a): [{E6}] bodů.
```

Toto řešení by odeslalo e-mail každému zvlášť a body ostatních členů by zpráva neobsahovala. Podle loginu (adresy) příjemce by skript rozeznal, kterou správnou buňku s hodnocením má přiřadit.

Dále je vhodné navrhnout zástupnou sekvenci *{extern}*, která umožní na dané místo vložit slovní hodnocení - externí komentáře ze souboru, který byl vygenerován pomocným skriptem u tabulky hodnocení. Na základě loginu by skript vybral blok komentářů, který k danému člověku/týmu patří.

4.8.2 Odesílání e-mailů

Úkol této části by měl být jednoduchý - pouze načíst textový soubor s vygenerovanými e-maily z minulé fáze a pomocí vhodné knihovny jazyka Python provést odeslání všech zpráv na správné adresy.

4.9 Export dat pro IS FIT

Jak již bylo zmíněno v odstavci popisujícím tvorbu tabulek hodnocení, vygenerovaná tabulka bude obsahovat speciální list LibreOffice Calc sdružující seznam studentů a výsledné počty bodů. Seznam studentů je vstupem pro skript pro tvorbu tabulky, je tedy garantováno stejné pořadí studentů. Informační systém v současné době umožňuje vkládání bodového hodnocení pomocí copy & paste, tedy bude stačit daný sloupec označit a výsledné body všech studentů vložit do systému.

Alternativním způsobem je tvorba skriptu, který získá data ze sumarizačního listu tabulky hodnocení a sestaví .csv, které lze do informačního systému importovat. Skript by byl principem podobný tomu na získávání dat z tabulky pro detailní hodnocení. Definovala by se šablona .csv dle šablony tabulky a skript by daný soubor pro import do systému vytvořil.

Příkladem: 000000;{B5};{C5};{D5};ucitel;datum, kde *{B5}* je odkaz na buňku šablony tabulky s *\$\$PR_JM_CLEN*, *{C5}* na *\$\$LOGIN_CLEN* a *{D5}* na výsledné body.

Kapitola 5

Implementace

Pro implementaci byl využit převážně Python[2] ve verzi 3 (3.4.3), pro zpracování e-mailů pak starší verze Python 2.7 (2.7.10), z důvodu dostupné vhodné knihovny pro práci s e-mail¹ pro tuto verzi. Při práci se soubory byly uplatněny nástroje shellu Bash.

Implementace skriptů proběhla na operačním systému Linux Ubuntu 15.10 s kancelářským balíkem LibreOffice verze 5.

Postup samotné implementace vychází z návrhu řešení popisovaného v kapitole 4.

5.1 Automatizované zpracování registračních e-mailů

Skript byl pojmenován jako **Email_reg.py**, parametry skriptu popisuje tabulka 5.1.

Parametr zkráceně	Parametr alternativa	Popis
-a	--adresar	adresář se staženými e-mailly
-s	--studenti	seznam zařazených studentů
-v	--vystup	výstupní textový soubor
-g	--groups	speciální požadavek pro kurz IVS, vytvoří výstupní soubor pro tvorbu skupin oprávnění na serverech - provede úpravy v názvu týmu dle pravidel popisovaných v návrhu

Tabulka 5.1: Parametry skriptu pro automatizované zpracování registračních e-mailů

Skript je implementován jako jedna třída `Main()`, která obsahuje inicializační a další 4 vlastní metody. Jedna z metod má prostou funkci, zpracování parametrů, případnou detekci neplatných vstupů. Dále provádí průchod adresářem přes jednotlivé e-mailové zprávy, kde nad každým nalezeným platným souborem provede volání metod na zpracování. Metoda `Zpracovani()` tvoří funkční základ. Provádí analýzu e-mailové zprávy, extrahuje odesílatele, příjemce kopie, předmět zprávy a její obsah. Další podstatnou částí je metoda `Vystup()`, která implementuje algoritmus pro kontrolu závislostí příjmení - jméno - login popisovaný v návrhu řešení. Pokud algoritmus vyhodnotí jednu nekonzistentní informaci u jednoho člověka, provede nápravu. V případě, že jeden člověk obsahuje více nekonzistencí (např: chybné jméno nebo chybný login), e-mail je označen za automatizovaně nezpracovatelný. V takovém případě dojde ve zdrojovém adresáři k vytvoření adresáře `/NEZPRACOVANO/` a nezpracovatelný e-mail do něj bude zkopírován. Z důvodu zachování původní souborové

¹email package - <https://docs.python.org/2/library/email.html>

struktury a snadného vyhledání e-mailů pro manuální zpracování, současně dochází k vypsání hlášení na konzoli pro uživatele, že došlo k chybě u daného e-mailu. V opačném případě dojde k zápisu týmu do výstupního souboru. Pokud je zadán parametr -g, skript vytvoří druhý, upravený výstupní soubor.

5.2 Zpracování registrací z IS FIT

Zdrojový text skriptu je zapsán v souboru **ISFIT_reg.py**. Byly implementovány dva režimy použití:

- a) **Online** - vstupem je URI (URL) adresa. První pro zobrazení tabulky seznamu týmů a druhá pro přístup k tabulce studentů zařazených do týmů v informačním systému.
- b) **Offline** - v tomto režimu skript pracuje se staženými .html soubory obou stránek s tabulkami.

Parametry skriptu popisuje tabulka 5.2.

Parametr zkráceně	Parametr alternativa	Popis
-z	--zarazeni	seznam zařazených studentů (lokální .html/URL na tabulku v IS FIT)
-t	--tymy	seznam týmů (lokální .html/URL na tabulku v IS FIT)
-c	--certifikat	bezpečnostní certifikát pro přístup do IS FIT při použití online režimu
-v	--vystup	výstupní textový soubor
-g	--groups	speciální požadavek pro kurz IVS, vytvoří výstupní soubor pro tvorbu skupin oprávnění na serverech - provede úpravy v názvu týmu dle pravidel popisovaných v návrhu
-d	--debug	při použití online režimu uloží stahované .html soubory

Tabulka 5.2: Parametry skriptu pro zpracování registrací z IS FIT

Skript je implementován jako 3 třídy, z nichž počáteční `Main()` provádí zpracování parametrů příkazové řádky a načtení vstupů. Pokud jsou vstupní .html zadány odkazem (využit online režim), dojde ke stažení těchto stránek. Ze třídy `Main()` jsou vytvářeny instance tříd `Parser_studenti()` a `Parser_tymy()`. Třída `Parser_studenti()` dědí ze třídy `HTMLParser`, která je součástí importované knihovny. Funkčním principem je detekce značek HTML a na jejich základě ukládání potřebných částí zdrojového kódu stránky. Třída `Parser_tymy()` je v jádru podobná jako třída `Parser_studenti()`, proto z ní dědí. Některé metody jsou přepsány a uzpůsobeny extrakci rozdílných potřebných informací.

5.3 Tabulka hodnocení

Tabulka hodnocení je z hlediska implementace nejrozsáhlejší částí. Implementace je zapsána v souboru **Tabulka_hodnoceni.py**. Parametry skriptu znázorňuje tabulka 5.3.

Parametr zkráceně	Parametr alternativa	Popis
-z	--zarazeni	seznam zařazených studentů v kurzu/projektu (exportováno z IS FIT jako .csv)
-t	--tymy	seznam týmů (výstupní soubor skriptů pro zpracování registrací)
-s	--sablonas	šablona pro generování tabulky
-o	--obarveni_vypnout	parametr co vypne vizuální odlišení každého druhého týmu nebo jednotlivce v pořadí

Tabulka 5.3: Parametry skriptu pro generování tabulky hodnocení

Byla implementována třída `Main()`, která obsahuje metody pro zpracování parametrů příkazové řádky, dále pak metody pro zajištění komunikace mezi LibreOffice a skriptem - automatické spuštění instance LibreOffice v subshellu operačního systému. Pro připojení je nutné zajistit dokončení spuštění, je tedy implementována kontrola, zda je proces spuštěn a je možné provést spojení. Poté dochází k načtení souborů potřebných pro generování - v případě individuálního projektu pouze seznam zařazených studentů v kurzu/projektu, pro týmový projekt pak také seznam zařazených studentů v kurzu/projektu a navíc seznam týmů. Vstupní šablona tabulky je souborem tabulkového procesoru LibreOffice Calc, lze jí tedy přímo využít pro generování obsahu. Pro zachování původní vytvořené šablony je v systému ve stejném adresáři vytvořena kopie šablony s názvem:

`nazev_sablony-Generovano-datum_cas`

Právě tato kopie je využita pro generování obsahu. Jedná se o hledisko optimalizace, kdy součást tabulky - poznámkový prostor nemusí být načítána a následně vygenerována v tabulce, z důvodu že je již v šabloně obsažena.

Součástí výsledné tabulky nadále zůstává šablona, za ní se vygeneruje určitý počet volných řádků, které se skrývají. Hlavním důvodem je její následné využití pro generování emailů s detaily hodnocení. Mezi další důvody patří pozdější dogenerování dalších studentů, nebo stav, kdy si uživatel čistou verzi šablony smaže.

Třída `nacti_list()` obsahuje metodu pro analýzu jednotlivých listů (Sheets) dokumentu LibreOffice Calc. Jednotlivé listy jsou zpracovány v cyklu. Pro každý list je provedena kontrola, zda je správně definována hlavička tabulky metodou `kontrola_hlavicky()`. Pokud je list v pořádku, metoda `nacti_hlavicku()` provede její načtení a uložení. Na stejném principu je založena metoda `nacti_obsah()`, která má na starosti načítání zástupných sekvencí, vzorců apod.

Dle obsahu zástupných sekvencí skript určí, zda se jedná o individuální nebo týmový projekt. Rozhodnutí je prováděno na základě obsažených zástupných sekvencí v šabloně.

V případě týmového projektu je volána metoda `tymovy_projekt()`, která pracuje se seznamem týmů a zároveň se seznamem studentů pro vygenerování tabulky. Zde byla nutná detekce zdvojených sekvencí `$$` nebo `~~=` pro automatické generování, nebo jednoduchých `$` a `~=` pro generování standardní dle konvencí šablony.

Podobný princip má metoda `individualni_projekt()`, která pouze obsahuje jiné zástupné sekvence.

V případě referencí v tabulkách `$$BODY=` nebo `$BODY=` dochází k interaktivnímu uložení souřadnice buňky k odpovídajícímu studentovi (název listu, sloupec a řádek buňky).

Třída `Export()`, jejíž instance je vytvářena na konec a obsahuje metody, které provádí generování prvního listu v pořadí, určeného pro sumarizaci bodů pro export dat do IS FIT.

5.4 Rozbalení a kontrola pojmenování odevzdaných souborů

Implementace byla provedena dle dostupného návrhu - dodrženo rozdělení skriptů.

a) **Rekurzivní rozbalení souborů** implementováno jako `Rekurziv_rozbaleni.py`. Parametry popisuje tabulka 5.4.

Parametr zkráceně	Parametr alternativa	Popis
-a	--adresar	adresář pro aplikování rekurzivního rozbalování

Tabulka 5.4: Parametry skriptu pro rekurzivní rozbalení souborů

Implementace obsahuje třídu `Main()`, která obsahuje metodu pro průchod vstupním adresářem. Pokud je nalezen archiv, je rozbalen. Mezi podporované archivy patří běžné typy používané pro projekty na Fakultě informačních technologií: `.zip`, `.tar`, `.tar.gz`. Pro správnou detekci typu archivu není používána pouhá detekce přípony, ale spolehlivější metoda - volání Bash příkazu `file` na detekci typu souboru.

Byla implementována funkcionalita navíc, kdy se skript dotáže uživatele, zda má provést smazání `.zip` na nejvyšší úrovni, očekává se jedna z interaktivních odpovědí (ANO/NE). Jedná se o archivy, které studenti odevzdali a jsou již rozbalené.

b) **Kontrola pojmenování a kompletnosti odevzdaných souborů** zapsána ve `Vyhledavani_Kontrola.py` s parametry ke spuštění viz tabulka 5.5.

Parametr základní	Parametr alternativní	Popis
-p	--pravidla	požadované soubory
-a	--adresar	adresář pro vyhledávání
-t	--tymy	seznam týmů pro vyhledávání pouze v adresářích vedoucích týmů

Tabulka 5.5: Parametry skriptu pro kontrolu pojmenování a kompletnosti odevzdaných souborů

Skript obsahuje implementovanou základní třídu `Main()`, která provádí zpracování parametrů. Instance třídy `Hledat_vsude()` je vytvořena v případě, že nebyl zadán vstupní seznam týmů. Dle načtených pravidel (seznamu souborů) provádí průchod adresáři studentů a vyhledává. Pokud je předmětem hledání adresář, využívá se metoda `hledam_adresar()`, pokud soubor, tak metoda `hledam_soubor()`. V opačném případě, kdy je parametrem zadán seznam týmů, je tvořena instance třídy `Hledat_vedouci()`, který využívá stejné metody, liší se pouze lokací vyhledávání. Z důvodu tvorby možného duplicitního kódu třída `Hledat_vedouci()` dědí ze třídy `Hledat_vsude()`.

c) Skript pro přejmenování souborů je implementován v souboru **Prejmenovani.py**. Podporované parametry popisuje tabulka 5.6.

Parametr základní	Parametr alternativní	Popis
-z	--zanoreni	maximální hloubka průchodu adresáře
-a	--adresar	vstupní adresář
-p	--pravidla	pravidla pro přejmenování

Tabulka 5.6: Parametry skriptu pro přejmenování souborů

Podobně jako ostatní skripty obsahuje implementovanou třídu `Main()`. Tato třída obsahuje metody pro zpracování parametrů, načtení vstupů, avšak nejzajímavější je metoda `walk()`, jejíž speciální funkcí je průchod adresáři do omezené hloubky, pokud je to vyžadováno.

d) Skript pro slučování adresářů je implementovaný v souboru **Slucovani_adresaru.py** a podporuje parametry uvedené v tabulce 5.7.

Parametr základní	Parametr alternativní	Popis
-z	--zdroj	zdrojové adresáře pro sloučení (n adresářů)
-c	--cil	cílová lokace sloučení

Tabulka 5.7: Parametry skriptu pro slučování adresářů

Skript obsahuje implementovanou třídu `Main()`, která obsahuje metody pro zpracování parametrů a hlavní metodu `Sluc()`. Metoda vezme adresářovou strukturu prvního zdroje uvedeného v parametru, který zkopíruje do cílového adresáře a každý následující zdroj se pokusí sloučit do cílového adresáře.

Skript vypíše konfliktní situaci i v případě, kdy dojde k pokusu sloučení souboru a adresáře se stejným názvem.

5.5 Slučování tabulek od více vyučujících

Tento skript podporuje vstupní parametry uvedené v tabulce 5.8 a zdrojový kód je umístěný v souboru **Spojeni_tabulek.py**.

Parametr základní	Parametr alternativní	Popis
-t	--tabulka	tabulky pro spojení
-l	--list	list tabulky pro spojení (ve všech tabulkách stejný)

Tabulka 5.8: Parametry skriptu na slučování tabulek

Skript byl implementován jako třída `Main()`, kde v inicializační metodě probíhá spuštění instance LibreOffice, a následně probíhá testování, zda je LibreOffice schopen komunikovat a bylo navázáno spojení ze skriptu - obdobným způsobem jako při tvorbě tabulky hodnocení.

Metoda `nactiSoubor()` provede kopii první tabulky v pořadí a této kopii je změněn název na `nazev_tabulky-SLOUCENO-datum_cas`. Tato kopie se stává zároveň výstupem - jedná se o prvek optimalizace, kdy se netvoří nový prázdný soubor tabulkového procesoru. Jedná se o ušetření spojení jedné tabulky a neprovádí změny žádné z původních tabulek.

Hlavním logickým mechanismem je postupné slučování dvojic tabulek v pořadí, v jakém byly zadány parametrem skriptu. Metoda `Spojeni()` provádí průchod jednotlivými buňkami a provádí obsahovou kontrolu buněk na stejných souřadnicích v obou tabulkách.

5.6 Rozesílání detailního hodnocení

Při implementaci byl dodržen navrhovaný způsob řešení popisovaný v návrhu řešení.

a) **Skript pro načtení dat z tabulky** - pojmenovaný jako `Nacteni_hodnoceni.py`. Seznam parametru skriptu viz tabulka 5.9.

Parametr základní	Parametr alternativní	Popis
-t	--tabulka	tabulka s hodnocením
-s	--sablona	vstupní šablona (vzor) e-mailu
-e	--extern_komentare	soubor s extern. komentáři
-l	--list	list tabulky
-v	--vystup	výstupní soubor
-k	--konec	číslo řádku, kterým tabulka končí

Tabulka 5.9: Parametry skriptu pro načtení dat z tabulky

Skript byl implementován jako třída `Main()`, která obsahuje metody na zpracování parametrů, navázání spojení s LibreOffice Calc. Tabulka hodnocení může být vygenerována dvěma způsoby. Pokud jsou do tabulky umístěny všechny zástupné sekvence, tak je velikost šablony v počtu řádků dána staticky. V tom případě pro čtení dat z tabulky skript provádí skoky o tolik řádků, kolik daná šablona má. Tento způsob implementují metody `napln_obycejne()` a `zisk_hodnotu()`. V druhém případě, kdy je tabulka vytvořena rozgenerováváním členů, nemá tým zaručen staticky počet řádků, je tedy nutné využít jiný způsob. Metoda `napln_rozgenerovane()` implementuje tuto situaci tak, že provádí testování, zda je řádek vhodný, na základě jeho obsahu. Metoda `hlavni_operace()` provádí tvorbu a formulaci výstupního souboru.

b) **Skript pro odesílání hodnocení** je pojmenován jako `Odeslani_hodnoceni.py`. Parametry viz 5.10.

Implementace tohoto skriptu zahrnuje opět třídu `Main()` s metodou na zpracování parametrů a další dvě metody. `Nacti_email()` provádí načítání vstupního souboru s e-maily vytvořeného skriptem `Nacti_hodnoceni.py`, `Odesli_email()` pak sestaví e-maily a provede jejich odeslání. Uživatel má přehled o úspěšně odeslaných emailech z výpisu na terminálu.

5.7 Export dat do IS FIT

Zdrojový kód skriptu je zapsán v souboru `Tvorba_export_CSV.py` a má parametry uvedené v tabulce 5.11.

Parametr zkráceně	Parametr alternativa	Popis
-z	--adresar	zdrojový soubor s emaily
-s	--server	SMTP server:port
-j	--jmeno_odesilatel	zobrazované jméno odesílatele e-mailu příjemcům
-e	--email_odesilatel	e-mailová adresa odesílatele

Tabulka 5.10: Parametry skriptu pro odesílání hodnocení

Parametr zkráceně	Parametr alternativa	Popis
-t	--tabulka	tabulka s hodnocením
-s	--sablona	šablona souboru .csv
-v	--vystup	výstupní soubor
-k	--konec	číslo řádku, kterým tabulka končí

Tabulka 5.11: Parametry skriptu pro export dat do IS FIT

Jedná se o upravený skript `Nacti_hodnoceni.py` implementovaný pro načítání detailního hodnocení z tabulky. Princip je stejný, avšak není třeba rozlišovat, jakým způsobem byla tabulka vygenerována (definice sumarizačního listu tabulky, odkud skript data získává, je pouze jedna). Došlo tedy k redukci implementovaných metod a úpravě formátu výstupního souboru.

Kapitola 6

Testování

Testování proběhlo převážně na notebooku s procesorem Intel Core i5 3210M, 8GB operační paměti a s nainstalovaným systémem Linux Ubuntu 15.10 v 64 bitové verzi. Pro implementované skripty byla vytvořena testovací sada pro ověření dosažené funkcionality. Skripty byly dále testovány vedoucím práce Ing. Jaroslavem Dytrychem a některé z nich byly nasazeny v reálném provozu kurzu IVS v akademickém roce 2015/2016.

6.1 Rychlost skriptů spolupracujících s LibreOffice

V případě využití verze LibreOffice 4.x dochází k razantnímu zpomalení komunikace mezi skriptem a spuštěným procesem LibreOffice. Byl proto proveden experiment s virtualizačním nástrojem. Na počítačové sestavě s procesorem Intel Core i7 4770k, 16 GB operační paměti s 64-bitovým operačním systémem Microsoft Windows 10 Professional v nástroji Oracle VM VirtualBox byly vytvořeny dva identické virtuální počítače, kdy každý virtuální počítač měl k dispozici 2GB operační paměti.

a) Na **první virtuální počítač** byl nainstalován 64-bitový operační systém Linux Ubuntu 14.04.4 LTS a byly provedeny základní aktualizace systému. Standardní instalace tohoto systému obsahuje LibreOffice ve verzi 4.x

b) **Druhý virtuální počítač** byla nainstalována 64-bitová verze operačního systému Linux Ubuntu 15.10 se stejným postupem a provedením standardních aktualizací. Tato instalace systému obsahuje LibreOffice ve verzi 5.x.

Byl proveden pokus s generováním tabulky hodnocení. Vstupními soubory byl seznam týmů, který obsahoval 5 týmů po 3-4 členech, a seznam studentů obsahující 20 studentů. Šablona pro tabulku obsahovala jeden sumarizační list a jeden list vzorového standardního projektu. Doba generování na virtuálním počítači a) byla 57 vteřin, na počítači b) 18 vteřin. Pokud byla vytvořena rozsáhlejší sada vstupních dat - 40 týmů po 3-4 členech se seznamem studentů obsahujícím 140 studentů se stejným projektem, byly časové rozdíly markantní. Doba generování na virtuálním počítači a) byla 3 hodiny 20 minut, na počítači b) pouhé 2 minuty 12 sekund.

Součástí experimentu byla odinstalace LibreOffice verze 4.x z virtuálního počítače a), čistá instalace nejnovější LibreOffice verze 5.x a opětovné opakování testů. Výsledné časy gene-

rování se podobaly těm naměřeným na virtuálním počítači b).

Bylo ověřeno, že problém způsobuje starší verze LibreOffice, kdy při každém provedení dotazu na obsah buňky/pokusu o zápis dochází k prodlevám.

6.2 Automatizované zpracování registračních e-mailů

Při testování skriptu byl kladen důraz hlavně na validaci funkčnosti algoritmu, který je schopen kontrolovat správnost závislostí login-příjmení-jméno. Byl vytvořen fiktivní seznam studentů simulující ten stažený z informačního systému. Tento soubor obsahoval záznam pro studenta *Jana Nováka* s loginem *xnovak00* a další jména. Prvotní testování se zaměřovalo na chybu u jednoho ze 4 členů týmu (tedy ti ostatní byli správně).

Vytvořené testovací e-maily obsahovaly registrace pro *Jana Nováka (xnovak00)*:

- *Jan Novák* a login *xnovak00*, jednalo se o zcela správné informace, registrace týmu byla zaznamenána.
- *Karel Novák* a login *xnovak00*, algoritmus detekoval chybu jména, avšak příjmení a login byly v pořádku, provedl korekci registrace studenta na *Jana Nováka* a tým byl opět registrován.
- Další případy s jednou chybou: překlepem v příjmení: *Jan Novákk*, *xnovak00*, i chybou v loginu: *Jan Novák*, *xlogin00*, dopadly stejným způsobem jako výše zmíněný.
- *Pavel Novák* a login *xnovak99*, kde jsou obsaženy dvě chyby, algoritmus vyhodnotil registraci celého týmu jako automatizovaně nezpracovatelnou.

V dalších fázích byly testovány chyby u více členů týmu apod. Výstupy skriptu korespondovaly s těmi očekávanými.

6.3 Zpracování registrací z IS FIT

Součástí testování skriptu na zpracování registrací přes informační systém bylo jeho přímé nasazení do reálného provozu kurzu IVS. Byl vytvořen výstup obsahující 78 týmů. Uplatnění našel i režim skriptu, který upraví výsledný výstup pro tvorbu skupin oprávnění na školním serveru.

6.4 Tvorba tabulky hodnocení

V případě tabulky hodnocení byla vytvořena středně rozsáhlá testovací data - seznam studentů čítající 150 studentů a seznam týmů s 50 týmy právě z těchto studentů. Jedná se zhruba o poloviční rozsah dat, než je v reálném kurzu IVS. Testem prošly všechny dostupné režimy činnosti v různých modifikacích:

- Individuální projekt - s výsledným časem generování 2 minuty 32 sekund
- Týmový projekt definovaný pro automatické rozgenerování - s výsledným časem generování 2 minuty 41 sekund

- Týmový projekt definovaný rozložením zástupných sekvencí - s výsledným časem generování 2 minuty 50 sekund

Testem prošly i pomocné skripty na generování souboru pro externí komentáře a jejich slučování.

Skript pro generování tabulky hodnocení a pomocný skript pro generování externích komentářů byly též využity při letošním hodnocení projektu kurzu IVS. Ing. Jaroslav Dytrych vytvořil šablonu na míru projektu a výsledná vygenerovaná tabulka byla využita při hodnocení projektu.

6.5 Rozbalení a kontrola pojmenování a kompletnosti odevzdaných souborů

Pro otestování skriptů, které pracují s odevzdanými soubory, byly k dispozici data odevzdaná studenty v kurzu IVS minulý rok (spadají pod GNU GPL¹ licenci).

Všechny implementované skripty vykazovaly korektní požadovanou činnost.

V případě rekurzivního rozbalování souborů se nabízí porovnání s dostupným nástrojem drtx: Intelligent archiv extraction² mezi jehož výhody patří práce s mnoha archivy, kdy uživatel nemusí vědět, o jaký typ archivu se jedná (s tím souvisí i terminálový příkaz na jeho rozbalení). Nástroj drtx provádí automatickou detekci a vše potřebné pro rozbalení provede za uživatele. Další nespornou výhodou je rekurzivní rozbalování, tedy situace, kdy archiv obsahuje vnořený archiv. Zápor a zároveň hlediskem nepoužitelnosti tohoto nástroje je fakt, že v mnoha případech provádí dotaz na uživatele na vícero rozhodnutí (zda je požadováno rozbalit vnořené archivy nebo zda se má rozbalený obsah přejmenovat apod.).

V kurzu IVS v akademickém roce 2014/2015 bylo studenty odevzdáno cca 70 archivů, pokud by byl prováděn takovýto dotaz například nad každým druhým archivem, časová náročnost pro uživatele potřebná pro zpracování by byla nižší, než v případě plného manuálního rozbalování, ale současně stále vyšší než v případě plné automatizace. Řešení implementované v rámci této práce vykazuje identickou činnost bez dotazování, je tedy pro použití pro větší množství archivu mnohem lepší.

6.6 Rozesílání zpráv s detaily hodnocení

Pro skript, který načítá data z tabulky hodnocení, byla sestavena testovací sada skládající se z anonymizované reálné tabulky hodnocení, souboru s externími komentáři a vzoru emailu, které se v IVS využívají. Výsledný výstup finální verze skriptu byl zkontrolován a výsledek dopadl dle očekávání. Skript na odeslání hodnocení byl otestován pro odeslání 100 vzorových e-mailů na mojí adresu (xmourl00@stud.fit.vutbr.cz). Jako server byl využit studentský poštovní server FIT (eva.fit.vutbr.cz s portem 587), kde bylo dosaženo povoleného limitu odeslání 10 zpráv. Více zpráv server nepovolil poslat, avšak těch prvních 10 odeslal bez problémů dle očekávání. Lze tedy předpokládat, že bez limitu by došlo k úspěšnému odeslání všech e-mailových zpráv.

V letošním akademickém roce 2015/2016 bylo detailní hodnocení projektu IVS odesláno těmito skripty.

¹GNU General Public License (GPL) - <http://www.gnu.org/licenses/gpl-faq.cs.html>

²drtx: Intelligent archiv extraction - <http://brettcsmith.org/2007/dtrx/>

Kapitola 7

Závěr

Analyzoval jsem postupy vyučujícího Ing. Jaroslava Dytrycha ve volitelném kurzu IVS (Praktické aspekty vývoje software) při hodnocení studentských aktivit. Následně byl navržen způsob automatizace vybraných postupů.

Samotná implementace proběhla dle vypracovaného návrhu řešení. Implementačním jazykem byl Python - převážně ve verzi 3. Pro skript zpracovávající e-mailové registrace byl z důvodu dostupné knihovny využit Python 2.7.

Všechny skripty byly celkově otestovány formou uživatelského testování na různě rozsáhlé množině vstupů. Některé z nich byly nasazeny do reálného provozu - kurzu IVS.

Pokud porovnáme původní, manuální způsob provádění úkonů potřebných pro hodnocení studentských aktivit a výsledné, implementované řešení, je zjevné, že došlo k několikanásobnému snížení časové náročnosti včetně úsilí, které je s tím spojené.

Vzhledem k rozdělení skriptů do více částí nebude v budoucnu problém některou z částí upravit, nahradit nebo rozšířit.

V dalších fázích vývoje by bylo možné rozšířit skript pro vyhledávání a kontrolu odevzdaných souborů o možnost kontrolovat soubory zadané regulárními výrazy. V současnosti jsou podporovány pouze soubory s pevným názvem. Dále by skripty bylo možné optimalizovat pro zvýšení rychlosti se zaměřením na ty, které pracují s tabulkovým procesorem LibreOffice Calc.

Literatura

- [1] Aul, G.: *Announcing Windows 10 Insider Preview Build 14316*. [Online; navštíveno 15.5.2016].
URL <https://blogs.windows.com/windowsexperience/2016/04/06/announcing-windows-10-insider-preview-build-14316/>
- [2] Beazley, D. M.: *Python - Podrobná referenční příručka pro programovací jazyk Python*. Neocortex, 2002, ISBN 80-86330-05-2.
- [3] Daryl Harms, K. M.: *Začínáme programovat v jazyce Python*. Computer Press, 2008, ISBN 80-7226-799-X.
- [4] Harsh, M.: *Run Bash on Ubuntu on Windows*. [Online; navštíveno 15.5.2016].
URL <https://blogs.windows.com/buildingapps/2016/03/30/run-bash-on-ubuntu-on-windows/>
- [5] Kosina, P.: *Python - popis jazyka*. [Online; navštíveno 15.5.2016].
URL <http://programujte.com/clanek/1970010106-python-popis-jazyka/>
- [6] Mark Lutz, D. A.: *Naučte se Python - Pohotová příručka*. Grada, 2003, ISBN 80-247-0367-X.
- [7] Oracle: *OpenOffice.org BASIC Programming Guide*. [Online; navštíveno 15.5.2016].
URL https://wiki.openoffice.org/wiki/Documentation/BASIC_Guide
- [8] Oracle: *PyUNO bridge*. [Online; navštíveno 15.5.2016].
URL https://wiki.openoffice.org/wiki/PyUNO_bridge
- [9] Resnick, P.: *Internet Message Format*. [Online; navštíveno 15.5.2016].
URL <https://tools.ietf.org/html/rfc5322>
- [10] Sheppard, S.: *An A-Z Index of the Bash command line for Linux*. [Online; navštíveno 15.5.2016].
URL <http://ss64.com/bash/>
- [11] Sobell, M. G.: *Linux: Mistrovství v Linuxu*. Computer Press, 2007, ISBN 80-251-1726-X.