

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Generátor a parser formulářů recenzí příspěvků na konferenci TSD

Místo této strany bude
zadání práce.

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V bakalářské práci jsou použity názvy programových produktů, firem apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

V Plzni dne 24. dubna 2019

Vojtěch Danišík

Poděkování

Děkuji panu Ing. Kamilu Ekšteinovi, Ph.D. za ochotu při vedení bakalářské práce a rady s jejím vypracováním.

Abstract

Generator and Parser of Submission Review Forms for the TSD Conference. The goal of this thesis is to create easily integrable PHP module into already existing informational system for TSD conference management. Task of the module is to create evaluative form in the PDF format and process it after loading it back into the system. First part of thesis deeply explains standard format PDF and forms created with PDF. Existing libraries for processing and generating the scientific contribution are explained afterward. Second part of thesis focuses on implementation of mentioned libraries into the web portal of TSD conference. Module has been tested by users of conference system. There has been used multiple PDF explorers during testing. Results of the tests are part of the thesis.

Abstrakt

Cílem bakalářské práce je vytvořit jednoduše integrovatelný PHP modul do již existujícího informačního systému pro správu konference TSD. Úkolem modulu bude vytvořit hodnotící formulář ve formátu PDF a následně ho po nahrání do systému zpracovat. První část práce důkladně vysvětluje standardní formát PDF a formuláře vytvořené v PDF. Následně jsou popsány existující PHP knihovny pro generování a zpracování PDF formuláře daného vědeckého příspěvku. Druhá část práce se věnuje implementaci vybraných knihoven do webového portálu konference TSD. Modul byl otestován uživateli konferenčního systému. Při testování bylo použito více PDF prohlížečů. Výsledky testování jsou součástí této práce.

Obsah

1	Úvod	1
2	Formát PDF	2
2.1	Objekty	2
2.1.1	Základní objekty	2
2.1.2	Složené objekty	3
2.1.3	Linkovací objekty	4
2.2	Kompresi dat v PDF	4
2.3	Vnitřní struktura PDF	5
2.4	PDF formuláře	7
2.4.1	Základní prvky	8
3	Knihovny	10
3.1	PHP knihovny pro generování PDF	11
3.1.1	FPDF	11
3.1.2	dompdf	11
3.1.3	TCPDF	11
3.1.4	HTML2FDPF	11
3.1.5	mPDF	12
3.2	PHP Knihovny pro zpracování PDF	12
3.2.1	pdf-to-html	12
3.2.2	TCPDF parser	12
3.2.3	PDF Parser	13
3.2.4	php-pdfkit	13
3.2.5	pdftotext	13
3.3	Závěr průzkumu	14
4	Návrh modulu	15
4.1	Vzhled dokumentu PDF	15
4.1.1	Záhlaví	15
4.1.2	Titulek	15
4.1.3	Formulář	15
4.1.4	Hodnocený vědecký příspěvek	16
4.1.5	Vodoznak	16
4.1.6	Fonty	16
4.2	Hlavní funkce modulu	18

4.2.1	Funkce pro generování	18
4.2.2	Funkce pro zpracování	19
5	Implementace modulu	20
5.1	Adresářová struktura modulu	20
5.2	Implementované třídy	20
5.2.1	Výčtové typy	20
5.2.2	Elements	21
5.2.3	TextConverter	21
5.2.4	ConfigurationData	22
5.3	Generátor	22
5.3.1	TCPDF versus mPDF	22
5.3.2	Popis vytvoření dokumentu	23
5.3.3	Nedostatky v mPDF	24
5.3.4	Nová knihovna pro slučování souborů PDF	25
5.3.5	Změna knihovny pro generování souborů PDF	26
5.4	Parser	26
5.4.1	Popis zpracování dokumentu	26
5.4.2	Extrakce formulářových prvků z předpřipravených dat	28
5.5	Výsledný vzhled PDF formuláře	29
5.6	Technické požadavky	29
6	Rozšiřitelnost modulu	30
6.1	Podpora zbylých formulářových prvků	30
6.2	Změna fontu	31
6.3	Načtení nově přidáných dat z konfiguračního souboru	32
7	Ověření kvality software	33
7.1	Testování modulu	33
7.1.1	Generování souboru PDF	33
7.1.2	Vyplnění a zpracování souboru PDF	34
7.1.3	Nalezené chyby při testování	35
7.2	Testovací scénář	35
7.3	Výsledky uživatelského testování	36
7.3.1	Vzhled dokumentu PDF	36
7.3.2	Nalezené chyby a připomínky	36
8	Závěr	38
	Literatura	39

A	Uživatelská dokumentace	40
B	Testovací reporty	42
B.1	Tester 1	42
B.2	Tester 2	43
C	Vzhled PDF formuláře	44

1 Úvod

TSD (**T**ext, **S**peech and **D**ialogue) je mezinárodní konference zabývající se například problémy zpracování, překladu a rozpoznávání přirozeného jazyka nebo analýzou řeči. Mezi nejčastěji probíraná témata se řadí například rozpoznávání řeči, modelování řeči, textové korpusy, značkování textu a mnoho dalších. Konference se koná každý rok v září a místo konání se střídá mezi Brnem (pořadatelem je Fakulta informatiky Masarykovy Univerzity) a Plzní (pořadatelem je Fakulta aplikovaných věd Západočeské univerzity v Plzni). Tento rok bude konference organizována právě Západočeskou univerzitou, a poprvé se bude konat za hranicemi České Republiky, přesněji ve Slovinsku ve městě Ljubljana.

Ke konferenci existuje webový portál, na nějž jsou od uživatelů nahrávány vědecké příspěvky. Tyto příspěvky jsou poté hodnoceny recenzenty (převážně členy programového výboru) formou online formuláře a na základě konečného hodnocení jednotlivých parametrů a na doporučení recenzentů jsou tyto příspěvky schváleny organizátorem a mohou být prezentovány na konferenci. Modul, vytvářený autorem, bude implementován do webového portálu konference TSD.

Cílem této práce je prostudovat strukturu formátu PDF, který je pro vytváření editovatelných formulářů nejvhodnější a byl vybrán zadávajícím jako standard, tak i funkcionalitu volně dostupných PHP knihoven pro generování a parsování souborů PDF obsahujících editovatelný formulář, aby existovala možnost ohodnocení daného vědeckého příspěvku i v místech, kde není dostupné internetové připojení, neboli off-line. Vytvořený soubor PDF musí obsahovat hodnotící formulář se všemi hodnotícími parametry doplněný o text vědeckého příspěvku. Pro generování a parsování musí být použity výhradně knihovny v jazyce PHP, jelikož není vhodné využívat aplikace třetích stran spustitelné z terminálu. Modul musí být nezávislý na platformě a lze ho upravovat v jakémkoliv prohlížeči PDF. Před vytvořením modulu na testovací verzi webového portálu bude potřeba projít zdrojové soubory webového portálu pro seznámení s již existujícími funkcionalitami a zařadit do portálu i náš modul. Z dřívějších let je zde naimplementován totožný modul pro generování a parsování souborů PDF, bohužel tento modul nesplňuje veškeré body zadání právě z důvodu použití nevhodného parseru.

2 Formát PDF

Formát **PDF** (**P**ortable **D**ocument **F**ormat) je souborový formát vyvinutý společností Adobe v roce 1992. Formát PDF byl vyvinut za účelem konzistentní prezentace dokumentů na různých platformách. Díky konzistenci lze dosáhnout toho, že soubor PDF vytvořený a uložený v systému Windows bude zobrazen totožně na systémech Mac, na všech distribucích Linuxu nezávisle na použitém prohlížeči PDF (Adobe Reader, Foxit a další).

V souboru PDF lze uchovávat velice širokou škálu dat, včetně formátovaného textu, vektorové grafiky a rastrových obrazů, nebo například informace o rozložení, velikosti a tvaru stránky. Informace definující umístění jednotlivých položek (jsou zde zahrnuty i editovací objekty pro formuláře) na stránce jsou zde uloženy též. Do dokumentu lze ukládat i metadata. Metadata jsou informace uložené v hlavičce souboru a lze do nich uložit název dokumentu, autora dokumentu, předmět a klíčová slova. Je zde možnost uložit heslo, aby byl dokument přístupný pouze autorizovaným uživatelům. Všechny tyto informace jsou uloženy ve standardním formátu [4, 8].

2.1 Objekty

PDF objekty jsou základním stavebním kamenem pro uchovávání dat v dokumentu. Množinou PDF objektů lze reprezentovat bitmapové a vektorové objekty, barevné prostory, text, fonty aj. [9].

2.1.1 Základní objekty

V PDF můžeme najít celkem pět základních objektů:

- **Celá a reálná čísla** – Přesnost a rozsah celých a reálných čísel je definován jednotlivými implementacemi PDF. V některých implementacích platí pravidlo, které přetypuje celé číslo na reálné po přesáhnutí předem daného rozsahu.
- **Řetězce** – Řetězec je reprezentován jako množina po sobě jdoucích bytů vepsaných mezi jednoduché závorky. Jako příklad lze uvést: (*Hello, World!*). Pro zobrazení zpětného lomítka a jednoduchých závorek je potřeba před tyto znaky přidat zpětné lomítko pro jejich správné zobrazení v dokumentu. V tabulce 2.1 lze vidět využití zpětného lomítka pro zobrazení odřádkovacích znaků:

Sekvence znaků	Význam
<code>\n</code>	<i>Line feed (LF)</i>
<code>\r</code>	<i>Carriage return (CR)</i>
<code>\t</code>	<i>Tab</i>
<code>\b</code>	<i>Backspace</i>

Tabulka 2.1: Odřádkovací sekvence znaků

Řetězce můžou být reprezentovány i jako sekvence hexadecimálních čísel vložených mezi znaky `<` a `>`.

Jako příklad lze uvést: `<4F6EFF00>` – `0x4F`, `0x6E`, `0xFF`, `0x00`.

- **Jména** – Jméno je reprezentováno jako sloučení lomítka a řetězce (př. `/Jmeno`). Za jméno se pokládá i zpětné lomítko bez řetězce. Pokud bychom potřebovali nadefinovat v dokumentu jméno, jenž bude obsahovat mezery, musíme do řetězce přidat i sekvenci znaků **#20** (v tabulce ASCII je hexadecimální hodnota dvacet vyjádřena jako prázdný znak). U jmen se rozlišují velká a malá písmena, proto `/Jmeno` a `/jmeno` jsou dvě různá jména. Jeho využití v PDF je prosté, slouží jako klíče ve slovnících a pro definice složitějších (vícehodnotových) objektů.
- **Boolean (pravdivostní) hodnoty** – Logické hodnoty **true/false** a vyskytuje se v jednotlivých záznamech ve slovníku jako příznak.
- **Hodnota null** – Nabývá hodnot *f* (free) nebo *n* (use) a vyjadřuje, zda je objekt vyobrazen v dokumentu.

2.1.2 Složené objekty

Složený objekt je takový objekt, který obsahuje seřazenou/neseřazenou množinu základních objektů i množinu složených objektů.

- **Pole** – Pole je v PDF reprezentováno jako seřazená množina základních i složených PDF objektů (v poli může být uložen například i slovník nebo pole) nezávisle na typech (v poli lze uchovávat například řetězec a číslo zároveň). Hodnoty pole jsou vloženy mezi znaky `[` a `]`.
- **Slovníky** – Slovník se skládá z množiny dvou prvků: klíče a hodnoty, pomocí kterých se slovník namapuje. Klíč je reprezentován pomocí **jména**, zatímco hodnota může být kterýkoliv PDF objekt, povoleny jsou i slovníky nebo pole. Slovníky jsou uloženy mezi znaky `<<` a `>>`.

- **Datové proudy** – Datové proudy slouží především pro uložení binárních dat a skoro ve všech případech jsou zkomprimovány různými kombinacemi algoritmů, které jsou popsány v kapitole 2.2, proto datové proudy musí být zároveň i nepřímým odkazem (odkaz na objekt obsahující data). Datové proudy se skládají ze slovníků a části binárních dat. Slovník je využit pro ukládání parametrů binárních dat, jako například délka binárních dat aj.

2.1.3 Linkovací objekty

PDF objekty mohou být různě velké. Pokud je objekt až příliš veliký, pak jsou v kódu dokumentu využity nepřímé odkazy. Na obrázku 2.1 si lze všimnout využití nepřímých odkazů ve slovníku.

```
<<
/Resources 10 0 R <--- znak R reprezentuje nepřímý odkaz na objekt s ID 10 a gen. číslem 0
/Contents [4 0 R]
>>
```

Obrázek 2.1: Ukázka nepřímého odkazu

2.2 Komprese dat v PDF

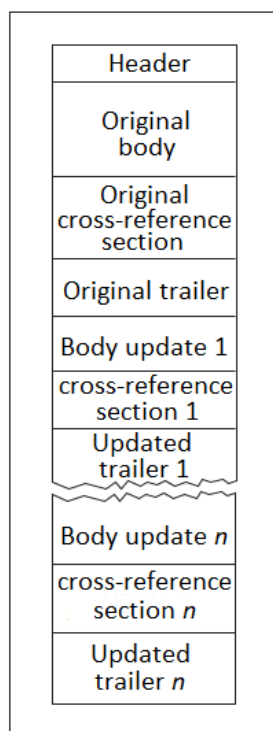
Soubory PDF mohou být poměrně kompaktní, o mnoho menší než ekvivalentní soubory vytvořené v **PostScriptu** (programovací jazyk určený ke grafickému popisu tisknutelných dokumentů vyvinutý v roce 1985 firmou Adobe Systems Incorporated). Tato vlastnost je dosažena nejen lepší strukturou dat, ale i díky kompresním algoritmům, které jsou efektivní. Typ komprese dat souboru PDF lze zjistit pomocí textového editoru, který dokáže zpracovat binární data, vyhledáním klíčového slova **/Filter**. Níže jsou popsány kompresní algoritmy využívané v PDF [3].

- **CCITT G3/G4** – Algoritmus je bezztrátový a využívá se pro vykreslení černobílých obrázků.
- **JPEG** – JPEG algoritmus může být jak ztrátový, tak i bezztrátový. V Acrobatu se využívá pouze ztrátový s pěti stupni komprese. Využívá se pro barevné a šedotónové obrázky.
- **JPEG2000** – Rychlejší algoritmus na bázi JPEGu. Víceméně se nepoužívá, jelikož není kompatibilní se staršími systémy a má vysoké nároky na procesor.

- **Flate** – Bezeztrátový algoritmus, vychází z kompresních algoritmů LZ77 a Huffmanova kódování.
- **JBIG2** – Alternativní k CCITT. V Dnešní době se nevyužívá z důvodu pomalejší komprese než je u jeho protějšku.
- **LZW** – Komprimací LZW algoritmem lze dosáhnout až o polovinu menší velikosti díky komprimaci veškerého textu a operátorů v souboru.
- **RLE** – Bezeztrátový algoritmus pro vykreslování černobílých obrázků. Nahrazen efektivnějším algoritmem CCITT.

2.3 Vnitřní struktura PDF

Vnitřní reprezentace souboru PDF je rozdělena na sekce, které jsou znázorňeny na obrázku 2.2.



Obrázek 2.2: Interní struktura souboru PDF

Z obrázku lze vyčíst, že se zde vyskytují čtyři hlavní sekce: *Header*, *Body*, *Cross-reference* a *Trailer*. Díky jedné z vlastností formátu PDF se při úpravě souboru staré sekce neodstraní, místo toho se pouze na jeho konci vytvoří nové sekce [6].

- **Header** – Hlavička souboru je uložena na první řádce, obsahující primárně použitou verzi PDF.

```
%PDF-1.4%âãäåÖ <--- hlavička souboru
```

Obrázek 2.3: Ukázka hlavičky

- **Body** – V těle dokumentu jsou uložena veškerá data objektů reprezentující celý dokument. Objekty jsou referencovány v tabulce Cross-reference z důvodu rozprostření částí dat patřících k danému objektu po celé sekci. Pokud se v dokumentu vyskytuje jeden obrázek/zvukový záznam vícekrát než jednou, tak se poté všechny objekty reprezentující obrázky odkazují na jednu množinu dat [5].

```
4 0 obj      <--- start objektu
<</Filter /FlateDecode /Length 1882>>
stream      <--- start dat
xśíśBoŮ6ÇąWzi 0†ŔĂŠööD
iD CÔČŘÖ=ŘŽăx“[Všv ýŽŠ-Drě,>#0#',QũñiH~
(S'ũµ-ĽŇRk□□□lžĚtv+ß'□\□.^□žj9□Ř÷{Nä}
endstream   <--- konec dat
endobj      <--- konec objektu
```

Obrázek 2.4: Ukázka dat objektu

- **Cross-reference table** – Jinak nazývána **xref** je tabulka obsahující reference na veškeré objekty uložené v těle a v kódu začíná řetězcem *xref*. Reference uložená v tabulce je reprezentována na dvou řádcích pomocí řetězce a skládá se z pěti částí o celkové velikosti dvacet bytů včetně oddělovačů *CRLF* (Windows), *CR* (Mac OS), *LF* (Unix, Linux):
 - *Číslo objektu* – Jednoznačný číselný identifikátor objektu.
 - *Počet subobjektů* – Počet částí daného objektu vyskytujícího se v dokumentu.
 - *Začátek objektu* – Tvoří většinu řetězce (prvních deset bytů) a určuje offset od začátku dokumentu PDF až po začátek daného objektu.
 - *Generační číslo objektu* – Vyjadřuje, jak často byl objekt vymazán při úpravě dokumentu.
 - *Identifikátor využití* – Nabývá hodnot *f* (free) nebo *n* (use) a vyjadřuje, zda je objekt vyobrazen v dokumentu.

```
xref <--- start tabulky
0 1 <--- ID objektu a počet subobjektů
0000000001 65535 f
31 1
0000423765 00000 n
```

Obrázek 2.5: Ukázka jednoduché xref tabulky

- **Trailer** – Trailer je seznam informací, ze kterých lze snadno zjistit například velikost nebo umístění xref tabulky. Trailer může obsahovat tyto elementy:
 - *Size* – Udává počet objektů referencovaných v xref tabulce.
 - *Prev* – Offset od začátku dokumentu k předchozí xref tabulce.
 - *Root* – Odkazuje na objekt obsahující informace ohledně katalogu xref tabulek.
 - *Encrypt* – Specifikuje komprimační algoritmus použitý pro daný dokument.
 - *Info* – Obsahuje dodatečné informace ohledně katalogu xref tabulek.
 - *ID* – 2-bytový identifikátor dokumentu PDF.
 - *XrefStm* – Offset od začátku dokumentu až k dekodovanému xref streamu. Využívá se pouze u hybridně-referencovaných (v raw kódu jsou využity přímé i nepřímé odkazy) souborů za předpokladu že hledaný objekt není nalezen v xref tabulce (před tím, než se volá element *Prev*).

```
trailer <--- start traileru
<<
/Size 742 <--- velikost xref tabulky
/Root 741 0 R <--- odkaz na objekt odkazující na objekt katalogu xref tabulek
/Info 740 0 R <--- odkaz na informační slovník xref tabulek
/ID [<009feb05c3e899ac1d26612f86bb56aa> <009feb05c3e899ac1d26612f86bb56aa>] --->
<--- identifikátor souboru
>>
startxref
408764 <--- offset tabulky xref
%%EOF
```

Obrázek 2.6: Ukázka traileru

2.4 PDF formuláře

Pod pojmem formulář si lze představit dokumenty, které od svých uživatelů vyžadují vyplnění určitých údajů. Mezi nejznámější dokumenty lze napří-

klad uvést daňová přiznání, oznamovací tiskopisy a dotazníky. Ruční vyplňování i jejich následné zpracování bývá obvykle pracné a zdlouhavé, proto je v dnešní době výhodnější využívat interaktivní elektronické formuláře. Základní výhoda těchto formulářů spočívá ve zrychlené komunikaci mezi objekty a subjekty. Díky elektronické podobě dochází k úspoře financí. Běžný občas přijde nejčastěji do styku s přihlašovacími formuláři a různými dotazníky, které jsou ve formátu HTML. Nevýhoda těchto formulářů je v jejich závislosti na internetovém připojení.

Proto firma Adobe přišla se svým řešením, interaktivním formulářem PDF, který lze vyplňovat kdekoliv nezávisle na internetovém připojení. Mezi další výhody formulářů PDF patří elektronický podpis (lze s ním potvrzovat smlouvy z domova), zabezpečení (dokument se otevře až po zadání správného hesla, neautorizovaným uživatelům je přístup zamítnut) aj. Tyto formuláře obsahují stejné interaktivní prvky jako mají formuláře HTML, viz kapitola 2.4.1. Pro generování formulářů PDF lze využít kterýkoliv programovací jazyk, který podporuje práci se soubory PDF (například *PHP*, *Java*), produkty firmy Adobe (například *Adobe Acrobat*) nebo lze použít i nekomerční aplikace typu *TeX* nebo *pdfmarks*.

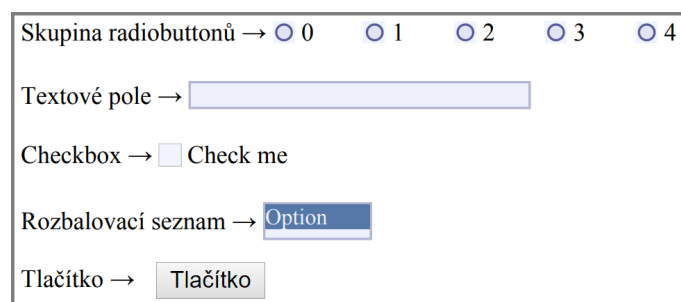
Tvorba formulářů je jedna věc, druhá věc je jejich zpracování (získání dat vyplněných od uživatele). Mezi nejznámější nástroje pro zpracování vyplněných dat patří určitě nástroj *FDF Toolkit* od firmy Adobe. Tento nástroj je zcela zdarma a umožňuje vytvářet orientovaná řešení pro zpracování dat v jazycích *C/C++*, *ActiveX*, *Java* a *Perl*. Jsou-li data odeslána v HTML, lze k jejich zpracování využít nástroje určené pro formáty *CGI*, *PHP* aj. [1].

2.4.1 Základní prvky

Jednotlivé formulářové prvky mohou mít přiřazeny nejrůznější atributy a jsou reprezentovány jako PDF objekty. Tyto atributy lze rozdělit do následujících skupin: **Vzhled** (definovaný vzhled prvku), **Akce** (po kliknutí na prvek se provede daná akce), **Formát** (typ fontu textu aj.), **Ověřování dat** (akceptovatelný formát vstupu) a **Výpočty** (matematické operace použité při práci se vstupy z jiných prvků) [2].

Ve formuláři se může vyskytovat až sedm různých prvků viz obrázek 2.7:

- **Textové pole** – Slouží k vyplnění textu. Jako příklad lze uvést například klasický přihlašovací formulář, který obsahuje dvě textové pole, jedno pro zadání uživatelského jména a druhé (upravené, místo textu se zobrazují pouze speciální znaky pro zakrytí zadaného textu) pro zadání hesla. Při vytváření lze předvyplnit toto pole výchozím textem,



Obrázek 2.7: Základní prvky vyskytující se v PDF

lze omezit maximální počet znaků vkládaných do pole a jejich formát. Pole může být uzamčeno a může sloužit i jako informační položka.

- **Tlačítko** – Účel tohoto prvku je spouštění zvolených akcí, které se po kliknutí na tlačítko mají provést, tudíž se označují jako hlavní řídicí prvek každého formuláře. Tlačítko se skládá převážně z ikonky a textu, případně mu může být nastaven externí obrázek.
- **Seznam** – Zobrazuje seznam položek, ze kterého lze současně označit jednu nebo více položek (s využitím klávesy *Shift* nebo *Ctrl*). Pro seznamy lze nastavit filtry, které budou seznam třídit podle předem daných parametrů a zobrazí položky na základě těchto filtrů.
- **Kombinované pole** – Kombinované pole je ve své podstatě seznam prvků, ale liší se ve výběru položek. V kombinovaném poli lze vybrat pouze jeden aktivní prvek, ostatní budou zakázány. Platí zde pravidla s tříděním prvků podle filtrů.
- **Přepínací tlačítka** – Je seznam tlačítek, ve kterém uživatel vybírá pouze jednu z nabízených hodnot.
- **Zaškrtávací pole** – Jedná se o indikační prvek umožňující současný výběr více položek.
- **Podpis** – Pomocí tohoto prvku lze do dokumentu vložit elektronický podpis.

3 Knihovny

V programování můžeme knihovnu definovat jako kolekci předem zkompilovaných procedur, funkcí (v objektovém programování i třídy a objekty), konstant a datových typů. Knihovna by měla být následně i dobře zdokumentována pro její snadnější zakomponování do již existujících modulů (při používání nezdokumentovaných knihoven se musí provádět takzvaný reverse engineering pro zjištění všech procedur a funkcí, nebo vyhledávat už hotová řešení na internetu).

Knihovny jsou z technického hlediska rozděleny do dvou skupin, které se následně rozdělují do dvou podskupin:

- Rozdělení z hlediska způsobu propojení s programem:
 - **Statická knihovna** – Zdrojový kód knihovny je v průběhu překládání zkopírován do výsledného programu pomocí kompilátoru. Největší výhoda statických knihoven spočívá v jistotě, že všechny potřebné knihovny budou přítomny ve výsledném programu, proto nikdy nemůže nastat situace nazvaná *dependency hell* (*DLL Hell*), která značí nepřítomnost jedné nebo více knihoven, které jsou využívány jinou knihovnou, nebo také může značit nadbytečné závislosti knihoven, které nejsou ve výsledku využity.
 - **Dynamická knihovna** – Oproti statickým knihovnám nejsou zdrojové kódy dynamických knihoven zakomponovány ve výsledném programu, ale pomocí linkeru jsou vytvořeny záznamy na funkce použité v programu, které jsou následně uloženy do tabulky symbolů vyskytující se ve výsledném programu.
- Rozdělení z hlediska sdílení kódu mezi programy:
 - **Sdílená knihovna** – Zdrojový kód sdílených knihoven je možné sdílet mezi více programy. Tímto způsobem jsou efektivně sníženy nároky na velikost operační paměti, protože úseky kódu využívané více procesy jsou uloženy ve sdílené paměti (namapovány do adresních prostorů všech procesů, které ji využívají).
 - **Nesdílená knihovna** – Nesdílené knihovny neumožňují sdílet úseky kódu více procesům.

3.1 PHP knihovny pro generování PDF

3.1.1 FPDF

Free PDF (zkráceně FPDF) [<http://www.fpdf.org/>] je knihovna psaná v jazyce PHP a slouží pro generování souborů PDF bez využití externích programů. Díky volně dostupným zdrojovým kódům lze na veřejných stránkách nalézt velice užitečná rozšíření této knihovny. Mezi hlavní funkce patří například automatické zalamování stránek, komprese stránek, hyperlinky a mnoho dalších. Bohužel zde nejdou vytvářet interaktivní formuláře, proto nelze tuto knihovnu použít pro vyvíjený modul.

3.1.2 dompdf

PHP knihovna **dompdf** [<https://dompdf.github.io/>] má za úkol převést HTML kód do souboru PDF. Své funkce dosáhne za pomoci externí knihovny *PDFlib* (placená) nebo pomocí třídy *R&OS CPDF* (autorem je *Wayne Munro*). Mezi hlavní funkce patří podpora 8/24/32-bitových obrázků (bitmapové a JPEG), externí CSS styly uložené na jiných stránkách/FTP, podpora atributů v HTML 4.0 aj. Bohužel **dompdf** není vhodná pro vyvíjený modul z důvodu neschopnosti vytvářet interaktivní formuláře (za podmínky využití přiložené pomocné třídy místo knihovny PDFlib).

3.1.3 TCPDF

Knihovna **TCPDF** [<https://tcpdf.org/>] je otevřená knihovna PHP sloužící pro práci se soubory PDF. Její vývoj odstartoval už v roce 2002, kdy vznikla jako odnož knihovny FPDF. Díky rozmanitosti jejich funkcí pro vytváření souborů PDF si ji oblíbilo mnoho uživatelů a je využívána i na mnoha webových portálech. Mezi hlavní funkce lze zařadit například podporu kódování UTF-8, kompresi stránek, vkládání zdrojových souborů, šifrování celého dokumentu, vkládání čárových kódů aj. Protože je psána pouze v jazyce PHP a nevyužívá žádné externí knihovny, pak ji lze brát jako vhodnou knihovnu pro vyvíjený modul.

3.1.4 HTML2FPDF

HTML2FPDF [<https://www.html2pdf.fr/>] vychází z již existující knihovny FPDF a má za úkol převést HTML kód a vytvořit z něj soubor PDF. Bohužel tato knihovna už není dále vyvíjena, ale stále funguje na všech verzích PHP.

Mezi hlavní nevýhody lze zařadit nemožnost vytvářet interaktivní formuláře, proto ji nelze využít pro vyvíjený modul.

3.1.5 mPDF

mPDF [<https://mpdf.github.io/>] je další PHP knihovna pro generování souborů PDF, která obsahuje velké množství užitečných funkcí. Byla vyvinuta z již existujících knihoven *FPDF* a *HTML2FPDF*. Převádí HTML kód a vytváří z něj soubor PDF se všemi prvky HTML (až na výjimky jako například nemožnost zobrazit tlačítko). Oproti knihovnám, ze kterých *mPDF* vychází, je tvorba PDF výrazně pomalejší. Podle autora knihovny zpomalení způsobuje užití Unicode fontů. Na druhou stranu je nespornou výhodou možnost využít kaskádové styly. Mezi hlavní funkce se řadí vytváření interaktivních formulářů, kódování UTF-8 HTML kódu, vkládání vodoznaku do stránek a mnoho dalšího. Z důvodu možnosti vytvářet interaktivní formuláře a nezávislosti na externích programech lze tuto knihovnu využít pro vyvíjený modul.

3.2 PHP Knihovny pro zpracování PDF

3.2.1 pdf-to-html

Knihovna **pdf-to-html** [<https://github.com/mgufrone/pdf-to-html>] má za úkol překonvertovat veškerý obsah souboru PDF do struktury HTML, ze které lze snadno extrahovat obsah souboru a předat ho ke zpracování. Pro správné fungování této knihovny musí být v konfiguraci PHP povolen přístup k příkazové řádce systému a na serveru musí být nainstalovaný *Poppler* (knihovna napsaná v jazyce C++ sloužící k renderování dokumentů PDF) [7]. Protože je tato knihovna závislá na knihovně (*Poppler*), nelze ji brát jako vhodnou pro vyvíjený modul.

3.2.2 TCPDF parser

TCPDF parser [<https://tcpdf.org/>] je součástí knihovny **TCPDF** (viz 3.1.3), která zpracovává soubor PDF. Pro svůj běh nepotřebuje žádné externí knihovny a je psána pouze v jazyce PHP, ale stále se nachází ve fázi vývoje, proto jsem zvolil jinou knihovnu.

3.2.3 PDF Parser

PDF Parser [<https://pdfparser.org/>] je další z mnoha knihoven sloužících pro zpracování souborů PDF. Tato knihovna je založena na již existující knihovně **TCPDF parser**, která je navíc doplněna o nové funkce jako je například extrakce metadat a komprimovaných souborů aj. Na stránkách PDF Parseru lze najít demo verzi, která demonstruje funkčnost, kdy po nahraní jakéhokoliv souboru PDF se na stránkách zobrazí data extrahovaná z nahraného souboru. Vzhledem k tomu, že PDF Parser je velice obsáhlá knihovna, využívá jí mnoho webových portálů pro zpracování souborů PDF, pak ji lze brát jako vhodnou knihovnu pro vyvíjený modul.

3.2.4 php-pdftk

Nástroj **PDF Toolkit** (zkráceně **pdftk**) [<https://www.drupal.org/project/phpdpdftk>] je multiplatformní nástroj pro manipulaci s soubory PDF, který navazuje na starší verzi nástroje **iText library**. PDF Toolkit lze najít ve třech verzích. Mezi neplacené verze patří *PDFtk Server*, což je otevřený nástroj v příkazové řádce a verze *PDFtk Free*, která je úplně zdarma, zatímco mezi placené verze patří verze *PDFtk Pro* (patří mezi proprietární software, jehož zdrojové soubory nejsou volně dostupné). Pomocí tohoto nástroje lze například oddělovat/spojovat/šifrovat soubory PDF, měnit vlastnosti, metadata, vyplňovat formuláře *FDF daty* (Forms Data Format). Díky velkému množství funkcí PDF Toolkitu byla vyvinuta knihovna v PHP s názvem **php-pdftk**, pomocí které lze využívat veškeré funkce tohoto nástroje v jazyce PHP. Bohužel díky závislosti na externím programu ji nelze brát jako vhodnou pro vyvíjený modul.

3.2.5 pdftotext

pdftotext [<https://pdftotext.com/>] je otevřený nástroj spouštěný v příkazové řádce využívaný k převodu souboru PDF do prostého textu využívající knihovnu *Poppler*. Je volně dostupný v linuxových distribucích (v některých distribucích je součástí systému), zatímco pro Windows ho lze nalézt jako součást programu *Xpdf*. Belgická firma *Spatie* vyvinula otevřenou PHP knihovnu využívající tento nástroj, aby byl dostupný i v jazyce PHP. Protože tato knihovna stejně jako **pdf-to-html** využívá *Poppler*, pak ji nelze brát jako vhodnou pro vyvíjený modul.

3.3 Závěr průzkumu

Autor této práce provedl rozsáhlý průzkum zaměřující se na volně dostupné PHP knihovny pro generování a zpracování souborů PDF. Co se týče PHP knihoven pro generování interaktivních formulářů, pak zde existují dvě vyhovující knihovny **mPDF** a **TCPDF**, které dokážou splnit veškeré požadavky zadavatele. Proto při vývoji modulu budou použity obě dvě a následně bude vybrána ta nejvíce vyhovující zadání. U knihoven zpracovávajících soubory PDF to tak není, většina knihoven využívá pro své fungování externí programy/knihovny psané v jiném programovacím jazyku a jsou převážně spouštěny z příkazové řádky, což silně odporuje požadavkům zadavatele. Jediná knihovna splňující tyto požadavky byla **PDF Parser**, proto bude použita při vývoji modulu.

4 Návrh modulu

4.1 Vzhled dokumentu PDF

Při návrhu výsledného vzhledu celého dokumentu je potřeba klást důraz hlavně na co nejpřesnější transformaci webového formuláře do souboru PDF. Vzhled webového formuláře je zobrazen na obrázku 4.1.

4.1.1 Záhloví

Záhloví dokumentu by mělo obsahovat jednoznačný identifikátor recenzního příspěvku doplněný o název vědeckého příspěvku, který bude hodnocen. Rozhodně by zde nemělo chybět ani logo konference TSD (ideálně ve vektorovém formátu). V případě, že název vědeckého příspěvku bude zasahovat do loga, bude název zkrácen na pevnou velikost.

4.1.2 Titulek

Titulek dokumentu by měl uživateli jednoznačně říct, který vědecký příspěvek hodnotí (ideálně zobrazit název i identifikátor). Každý příspěvek má v systému vlastní identifikátor (S-ID#), stejně jako každý uživatel (U-ID#) a samotné recenze (R-ID#). V titulku by nemělo chybět ani jméno recenzenta a doplňující informace týkající se vyplňování formuláře, případně recenzenta informovat o nedostatecích a omezeních aktuálně vygenerovaného dokumentu PDF.

4.1.3 Formulář

Vzhled formuláře by se měl v ideálním případě shodovat s webovým formulářem. První část formuláře obsahuje stupnicové hodnocení, zatímco druhá je spíše slovní formou. Po konzultaci s vedoucím práce jsem se rozhodl, že kombinované pole reprezentující stupnicové hodnocení parametru bude nahrazeno skupinou přepínacích tlačítek. Důvody tohoto rozhodnutí spočívaly v lepším zobrazení hodnot na stupnici a zároveň neschopností PHP generátorů zobrazit uživatelsky přívětivé kombinované pole. Pro uložení doplňujícího textu bylo použito textové pro případné poznámky ohledně stavu a obsahu vědeckého příspěvku. Element popisující *Review state* (stav recenzního příspěvku) a tlačítko *Save review* (uložit recenzní příspěvek) nebudou

do formuláře vloženy, jejich funkce není potřebná pro modulem vytvářený formulář.

4.1.4 Hodnocený vědecký příspěvek

Vygenerovaný dokument by měl hlavně sloužit pro vyplňování hodnotícího formuláře off-line a ideálně by měl obsahovat i veškerý obsah hodnoceného vědeckého příspěvku, aby měl recenzent možnost kdykoliv nahlédnout na jeho obsah. Tento příspěvek bude vložen na konec souboru PDF.

4.1.5 Vodoznak

Často se stává, že se neoprávněně kopírují již hotová díla a vydávají se pod cizím jménem, proto je vhodné do celého dokumentu vložit vodoznak, který bude jasně říkat, že se jedná pouze o soubor v recenzním řízení, nikoliv o plnohodnotné dílo.

4.1.6 Fonty

V prostředí PDF a PostScript se lze setkat s pojmem čtrnáct standardních/-základních fontů. Tento pojem byl odvozen ze standardních třinácti PostScript fontů a vyjadřuje základní fonty používané při vytváření veškerých souborů PDF. Všechny základní fonty lze nalézt v tabulce 4.1.

Rodina fontů	Fonty
<i>Times</i>	Times-Roman Times-Italic Times-Bold Times-BoldItalic
<i>Helvetica</i>	Helvetica Helvetica-Oblique Helvetica-Bold Helvetica-BoldOblique
<i>Courier</i>	Courier Courier-Oblique Courier-Bold Courier-BoldOblique
<i>Symbol</i>	Symbol
<i>ZapfDingbats</i>	ZapfDingbats

Tabulka 4.1: Tabulka základních fontů v souborech PDF

Originality:	0	▼
Significance:	0	▼
Relevance:	0	▼
Presentation:	0	▼
Technical quality:	0	▼
Overall rating:	0	▼
Amount of rewriting:	0	▼
Reviewer's expertise:	0	▼

Main contributions:	<div></div>
Positive aspects:	<div></div>
Negative aspects:	<div></div>
Comment:	<div></div>
Internal comment:	<div></div>

Review state:	Unfinished	▼
---------------	------------	---

Save review
(autosave in 15:02)

Obrázek 4.1: Webový formulář používaný pro hodnocení vědeckých příspěvků na portálu konference TSD

4.2 Hlavní funkce modulu

Vyvíjený modul musí být napsán stejným stylem jako je celý webový portál konferenčního systému TSD. Jelikož se na tomto portálu vyskytuje modul, který má stejnou funkcionalitu jako modul vyvíjený autorem bakalářské práce, tak je návrh funkcí jednodušší. Modul vyskytující se na portálu implementuje tři důležité funkce, které zajišťují veškerou funkcionalitu i přes to, že pro zpracovávání souborů PDF je použit externí program *PDFtk*. Po konzultaci s vedoucím práce bylo rozhodnuto, že se původní názvy funkcí zachovají a budou pouze změněny jejich parametry. Autorův modul bude tedy ve výsledku obsahovat, stejně jako starý modul, tři důležité funkce doplněny o pomocné funkce a konstanty. Všechny hlavní funkce jsou popsány níže.

4.2.1 Funkce pro generování

Pro generování souboru PDF byla navržena jedna funkce. Tato funkce má za úkol nejdříve nastavit veškeré fonty a styly pro vzhled dokumentu, následně využít vhodný generátor souborů PDF, který vytvoří všechny části dokumentu (vypsání v kapitole 4.1) a nabídne uživateli možnost stáhnout si výsledný soubor PDF v recenzním řízení. Návrh hlavičky funkce viz výpis 4.1.

```
function generate_offline_review_form($rid,  
    $reviewer_name, $sid, $submission_name,  
    $submission_filename)
```

Listing 4.1: Návrh hlavičky funkce pro generování souboru PDF

Popis vstupních parametrů funkce:

- `$rid` – ID hodnotícího příspěvku
- `$reviewer_name` – celé jméno recenzenta
- `$sid` – ID vědeckého příspěvku
- `$submission_name` – celý název vědeckého příspěvku
- `$submission_filename` – celý název souboru PDF vědeckého příspěvku

4.2.2 Funkce pro zpracování

Pro zpracování souboru PDF byly navrženy dvě funkce, kdy první z nich má za úkol nahrát celý soubor do konferenčního souboru. Po nahrání souboru začne extrakce dat pomocí vhodných parserů a zpracování vstupních souborů. Následně se vše uloží do odpovídajících struktur vícerozměrných polí a všechny extrahované hodnotící parametry se předají do následující funkce. Návrh hlavičky funkce viz výpis 4.2.

```
function process_offline_review_form($rid, $sid,  
    $revform_filename)
```

Listing 4.2: Návrh hlavičky funkce pro extrakci dat

Popis vstupních parametrů funkce:

- `$rid` – ID příspěvku v recenzním řízení
- `$sid` – ID vědeckého příspěvku
- `$revform_filename` – celý název souboru PDF příspěvku v recenzním řízení

Druhá funkce pro zpracování souboru PDF má za úkol uložit již extrahované hodnotící parametry do databáze konferenčního systému. Návrh hlavičky funkce viz výpis 4.3.

```
function upload_to_DB_offline_review_form($rid, $values)
```

Listing 4.3: Návrh hlavičky funkce pro uložení dat do databáze

Popis vstupních parametrů funkce:

- `$rid` – ID příspěvku v recenzním řízení
- `$values` – seznam všech hodnotících parametrů

5 Implementace modulu

Během analýzy celého programu bylo potřeba vybrat nejvhodnější knihovnu pro generování souborů PDF a tu následně integrovat s předem vybraným parserem. Pro snadnější integraci byly vytvořeny nové třídy.

5.1 Adresářová struktura modulu

Adresářová struktura modulu na serveru vypadá následovně:

- `config` – Adresář obsahující konfigurační soubor `configuration.xml`, ve kterém jsou uložena často měněná data (rok konference aj.).
- `img` – Adresář obsahující obrázky použité v dokumentu (logo konference TSD).
- `lib` – Adresář obsahující zdrojové kódy knihoven třetích stran (generátor a parser).
- `src` – Adresář obsahující zdrojové kódy vytvořené autorem bakalářské práce.
- `orlib.php` – Hlavní soubor modulu obsahující všechny tři stěžejní funkce modulu.

5.2 Implementované třídy

Pro modul jsem vytvořil sedm tříd, které integrují pomocné knihovny pro generování a zpracování souborů PDF (vytváření formulářových prvků, konstanty aj.).

5.2.1 Výčtové typy

Výčtový typ (neboli **Enum**) je datový typ určený pro uložení konstant programu, kdy každé z těchto konstant je přiřazena jedna instance výčtu. Ve vytvářeném modulu byly použity čtyři výčtové typy.

Výčtový typ **Instruction** uchovává konstanty využívané při generování titulu a informací během vyplňování formuláře. Tyto konstanty reprezentují celkem čtyři části dokumentu (záhlaví, titulek dokumentu, jméno recenzenta a instrukční text pro vyplňování formuláře).

Výčtový typ **FormElements** slouží pouze pro rozlišení použitých objektů na základní prvky formuláře. Zde byly použity výběrové tlačítko a textové pole.

Ve výčtovém typu **TextareaInfo** jsou uloženy veškeré hodnotící parametry, které jsou reprezentovány jako textová pole, a pomocné funkce. Každý hodnotící parametr je zde určen třemi konstantami (jednoznačný identifikátor, název a jeho popis). Dále se tu vyskytují dvě konstanty využívané při vytváření formulářového prvku pomocí HTML kódu. Tyto konstanty jsou využity i při následném zpracování dokumentu pro všechny formulářové prvky reprezentované jako textová pole. Byla zde vytvořena i funkce *getNotNeededConstants* pro získání nepovinných hodnotících parametrů.

Výčtový typ **RadiobuttonInfo** je téměř totožná s třídou **TextareaInfo** s tím rozdílem, že hodnotící parametry jsou reprezentovány jako skupina výběrových tlačítek.

5.2.2 Elements

Hlavním důvodem vzniku této třídy byla snaha nevytvářet formulářové prvky přímo v hlavní funkci *generate_offline_review_form*, ale použít nově vytvořené metody. Pomocí implementovaných metod lze vytvářet textová pole, výběrová tlačítka, textové části dokumentu a načítat vědecký příspěvek.

5.2.3 TextConverter

V některých případech jsou název vědeckého příspěvku nebo jméno recenzenta příliš dlouhé, a proto narušuje vzhled výsledného dokumentu. Problém může nastat i při chybě programátora, pokud by byl instrukční text příliš rozsáhlý. Proto byla vytvořena třída **TextConverter**, která má za úkol nejdříve zkontrolovat předaný text a porovnat ho se stanovenými konstantami určujícími maximální délku textu. Pokud je rozsah textu delší než stanovená délka, vypočte se následně potřebný font pro vykreslení celého textu pomocí vzorce (5.1). Délka se porovnává se stanovenými konstantami určujícími minimální velikost fontu.

$$h_n = \left\lceil \frac{l_{\max}}{l} \cdot h \right\rceil, \quad (5.1)$$

kde h_n je nově vypočtená velikost fontu, l_{\max} je maximální délka kontrolovaného textu, l je aktuální délka kontrolovaného textu a h je aktuální velikost fontu.

Pokud je vypočtený font je menší než předem stanovený minimální font, je text zkrácen na velikost vypočtenou pomocí vzorce (5.2) a doplněn třemi tečkami na jeho konci.

$$l_n = \left\lfloor \frac{h_p}{h_{\min}} \cdot l \right\rfloor, \quad (5.2)$$

kde l_n je nově vypočtená délka kontrolovaného textu, h_p je původní velikost fontu, h_{\min} je minimální velikost fontu a l je délka kontrolovaného textu.

5.2.4 ConfigurationData

Třída načítá veškerý obsah konfiguračního souboru, který následně ukládá do svých proměnných. Data uložená v konfiguračním souboru slouží pro nastavení textu vodoznaku a jako informační text pro uživatele.

5.3 Generátor

Generátor by měl být při vytváření dokumentu PDF rychlý, vykreslit co nej-přesněji prvky webového formuláře do vygenerovaného dokumentu a nebyť implementačně náročný.

5.3.1 TCPDF versus mPDF

Při analyzování dostupných PHP knihoven pro generování souborů PDF byly nalezeny dvě vyhovující knihovny, které splňují potřebnou funkcionalitu. Po vytvoření jednoduchého souboru obsahujícího základní formulářové prvky jsem se rozhodl, že použiji knihovnu **mPDF** pro generování souborů PDF. Důvody této volby jsou popsány níže.

Za jeden z důležitých faktorů lze označit podporu *CSS3* (Cascading Style Sheets 3) u **mPDF**, díky čemuž lze dosáhnout perfektního nastavení stylů pro jednotlivé objekty v dokumentu. Naproti tomu **TCPDF** nepodporuje značné množství CSS parametrů, například parametr určující šířku vnějšího okraje prvku, a pro dosažení obdobného výsledku je zapotřebí značné množství jiných parametrů definujících styl prvku.

Důležitým faktorem při vytváření dokumentu PDF je rychlost generování a paměťová náročnost. V tabulce 5.1 lze vidět porovnání knihoven pro dva soubory PDF, kdy první PDF obsahovalo hlavně kaskádové styly, zatímco v druhém PDF byla vytvořena tabulka s více jak tisíci záznamy.

Posledním a zároveň rozhodujícím faktorem je psaní PHP kódu pro vykreslování obsahu, kdy při vytváření kódu u **mPDF** se využívá minimum

Název	Komplexní PDF		Dlouhé PDF	
	Paměť [MB]	Čas [ms]	Paměť [MB]	Čas [ms]
TCPDF (v6.2.13)	74	35944	2,3	96350
mPDF (v7.1.6)	14	11316	22,5	4120

Tabulka 5.1: Tabulka časové náročnosti a využití paměti při generování

funkcí pro nastavení parametrů souboru PDF jako jsou například metadata, zatímco veškeré zobrazené elementy a text jsou psány v jazyce HTML, se kterým se snadno pracuje. V mPDF lze snadno měnit parametry jednotlivých elementů, což bude oceněno hlavně u parseru. U **TCPDF** se zobrazovaný obsah vkládá pomocí předem vytvořených funkcí, přičemž tyto funkce mohou obsahovat mnoho parametrů, které si uživatel obtížně zapamatuje a vždy bude potřebovat patřičnou dokumentaci pro správné použití, což bude zabírat mnoho času při vyvíjení nových modulů.

Na závěr porovnání lze říci, že ve většině případů je vhodné využít pro generování souborů PDF knihovnu **mPDF**. Pokud by bylo nutné vytvořit dokument například ve stylu knihy s nulovým využitím CSS stylů a potřebou kvalitního vysázení textu, pak je lepší použít knihovnu **TCPDF**.

5.3.2 Popis vytvoření dokumentu

Na samotném začátku generování jsou vytvořeny instance tříd. U instance třídy **ConfigurationData**, proběhne i načtení dat z konfiguračního souboru XML. Dále jsou vytvořeny proměnné reprezentující název vybraného dokumentu a informace o nahrání vyplněného dokumentu do webového portálu konference TSD. Před samotným začátkem generování je do modulu importováno CSS nastavení pro vzhled celého dokumentu.

V první části generování probíhá vytvoření záhlaví. V celém dokumentu je použit font *Helvetica*, pouze výjimečně je zařazen font *Times New Roman*, například pro titulek dokumentu a text se stylem *Bold*. Do záhlaví je vložen identifikátor recenze doplněn o název hodnoceného vědeckého příspěvku, který je případně zkrácen na určitou délku, pokud nesplňuje limity nastavené ve třídě **TextConverter**, a logo konference TSD.

V druhé části generování probíhá vložení vodoznaku do celého dokumentu, uložení jednoznačného identifikátoru jak vědeckého příspěvku v recenzním řízení, tak i hodnotícího příspěvku do příslušných metadat dokumentu. Na první stránce dokumentu je vykreslen titulek s identifikátorem hodnoceného vědeckého příspěvku, název hodnoceného vědeckého příspěvku (případně zkrácen stejně jako u záhlaví), jméno recenzenta a doprovodný

text při vyplňování hodnotícího formuláře. Pod tímto textem je vykreslena první část hodnotícího formuláře, která obsahuje osm skupin výběrových tlačítek a jedno textové pole.

V poslední části probíhá vykreslování zbylých čtyř textových polí, kde dvě poslední z nich jsou nepovinná. Za hodnotícím formulářem je vložen kompletně celý hodnocený vědecký příspěvek.

5.3.3 Nedostatky v mPDF

Při vytváření dokumentu byly nalezeny dvě chyby znemožňující úplné vykreslení celého dokumentu. Níže jsou tyto chyby popsány i s návrhem jejich řešení.

První nedostatek byl zjištěn na úplném začátku implementace generátoru, kdy při vkládání textových polí do formuláře se po přeložení kódu nevytvořil žádný dokument. Při zkoumání zdrojového kódu knihovny a vytvoření testovacích dokumentů bylo zjištěno, že knihovna neumožňuje použít textové pole, pokud se při jeho vytvoření nezadá vkládaný text. Proto bylo nutné upravit kód knihovny, konkrétně ve vykreslování textového pole. Aby bylo možné takto upravovat zdrojový kód knihovny, nesmí být knihovna pod licenci a naopak musí být alespoň pod licenci dovolující úpravy, například *GNU General Public License* verze 2, pod kterou je licencována i mPDF. Pro vyřešení tohoto problému byl přidán mechanismus, který při vytváření prázdného textového pole přidá znak „a“ (viz 5.1) a posléze je v knihovně při vykreslování textového pole tento znak odstraněn, což nemá vliv na jakýkoliv jiný znak či slova než zmiňovaný znak „a“, viz obrázek 5.2.

```
if($textarea_text == '') $textarea_text = 'a';
```

Listing 5.1: Dočasné přiřazení znaku „a“ do textového pole (Elements.php)

```
if (isset($objattr['text']) && $objattr['text'] != 'a')
{
    $texto = $objattr['text'];
}
else $texto = '';
```

Listing 5.2: Odstranění znaku „a“ z textového pole (Mpdf.php)

Druhý nedostatek byl nalezen při testování zkracování délky textu titulku, pokud překročí nastavenou mez. V aktuální verzi PHP se vyskytuje problém, který zneplatňuje některé znaky v kódování UTF-8. Pokud je například vytvořen nový uživatel se jménem obsahující například znak „ř“, pak

se tento znak nepřeveďe správně a bude vykreslen jako neznámý znak. Bohužel generování dokumentu neprobíhalo správně, protože knihovna mPDF tyto znaky nerozpoznala, a proto výsledek vždy skončil chybou. Ze všech vyzkoušených možností, jako například změna kódování textu titulku nebo nahrazení neplatných znaků prázdnými, fungovala pouze jedna, a to nastavení atributu `ignore_invalid_utf8` na `true` u proměnné třídy *Mpdf* (viz obrázek 5.3).

```
$mpdf->ignore_invalid_utf8 = true;
```

Listing 5.3: Nastavení atributu `ignore_invalid_utf8` (orlib.php)

Nejzávažnější nedostatek knihovny mPDF byl objeven na samém konci testování. Bylo testováno především slučování hodnotícího formuláře s vědeckými příspěvky. Vědecké příspěvky byly uloženy ve formátu PDF v různých verzích, nejčastěji verze 1.4 až 1.6. Testování probíhalo perfektně pro PDF verze 1.4, bohužel pro novější verze už slučování neprobíhalo správně a modul generoval výjimku. Důkladným zkoumáním mPDF knihovny bylo zjištěno, že pro slučování souborů PDF se využívá podpůrná knihovna **FPDI**, která funguje pouze pro soubory PDF do verze 1.4. V kapitole 5.3.4 je popsán postup řešení tohoto nedostatku, který zapříčiňuje nefunkční generování hodnotícího dokumentu PDF.

5.3.4 Nová knihovna pro slučování souborů PDF

Jedna z podmínek pro generující knihovnu je umět vkládat hodnocený vědecký příspěvek do výsledného hodnotícího dokumentu PDF. Protože již implementovaná knihovna FPDÍ tuto funkci neumožňuje, bude potřeba najít jinou knihovnu splňující slučování dokumentů PDF. Nově zvolená knihovna by měla být kompatibilní s knihovnou mPDF.

Po rozsáhlém průzkumu byla nalezena pouze jedna PHP knihovna, která dokáže slučovat dokumenty PDF verze 1.4. Je jí knihovna **TCPDI parser**, která je součástí knihovny **TCPDI**, která dokáže slučovat PDF do verze 1.7.

Následně byla knihovna FPDÍ nahrazena TCPDI parserem a otestována na několika testovacích souborech PDF s rozdílnou verzí PDF. Výsledek sloučení nebyl dostatečně kvalitní. Všechny použité styly nebyly přeneseny do výsledného PDF, odsazení textu bylo natolik špatné, že občas byl text posunutý mimo stránku. Bohužel veškeré přílohy, jako jsou například obrázky, komentáře nebo vzorce, nebyly přítomny ve výsledném dokumentu PDF. Na základě těchto problémů bylo rozhodnuto změnit generovací knihovnu.

5.3.5 Změna knihovny pro generování souborů PDF

Změna knihovny pro generování souborů PDF byla nutná z důvodu nedokonalosti mPDF při slučování více souborů PDF. Při testování byla objevena nová knihovna TCPDI, která rozšiřuje již existující knihovnu TCPDF o nové funkce při slučování dvou či více souborů PDF.

Pro zprovoznění TCPDI je nutné, aby byla na serveru přítomna knihovna TCPDF, do které bude následně vložen zdrojový kód TCPDI. Zdrojový kód TCPDI se skládá ze tří souborů. První soubor `tcpdi_parser.php` obsahuje třídu `tcpdi_parser`, která načte data ze souboru PDF a uloží je do předem stanovených struktur. Druhý soubor `tcpdi.php` obsahuje třídu `TCPDI`, která rozšiřuje stávající třídu `TCPDF` a umožňuje slučovat data souborů PDF, ze kterého následně zobrazí výsledný soubor PDF. Třetí a zároveň poslední soubor `fpdf_tpl.php` obsahuje třídu `fpdf_tpl`, která vytváří základy pro znovupoužití PDF objektů v souboru PDF.

Pro generování obsahu souboru PDF se nevytváří HTML kód, ale využívají se předem vytvořené metody. Jako příklad lze uvést metodu *TextField* třídy `TCPDI` pomocí které se do souboru vloží textové pole. `TCPDI` obsahuje metodu, která umožňuje generovat soubor PDF i pomocí HTML kódu, ale z důvodu nedostatečné podpory kaskádových stylů je tento způsob generování nedoporučován. Proto bylo nutné kompletně přepsat již vytvořený zdrojový kód generátoru a co nejvíce se přiblížit vzhledu souboru PDF jako tomu bylo u mPDF. Starý zdrojový kód využívající knihovnu mPDF byl zachován pro případ, že bude vytvořena nová knihovna, která bude schopna slučovat soubory PDF nehlédě na verzi PDF a bude kompatibilní s mPDF.

5.4 Parser

Z analýzy knihoven pro zpracování dokumentů PDF splnil nutné požadavky pouze **PDF Parser**. Při implementování parseru bylo zjištěno, že jednotlivé prohlížeče PDF při uložení dokumentu PDF využívají jiné komprimační metody, pracující s novějšími verzemi PDF pro určité funkce a některé prohlížeče ukládají objekty v dokumentu na více místech (duplikace, jednou komprimovaně, jednou nekomprimovaně).

5.4.1 Popis zpracování dokumentu

Zpracování dokumentu začíná ihned po jeho nahrání do webového portálu konferenčního systému, kdy se veškerá data předají do PDF Parseru. Před samotnou extrakcí dat jsou pomocí TCPDF parseru, který je součástí PDF

Parseru, data rozdělena na objekty pomocí *traileru* a *xref tabulky* (viz 2.3). Následně jsou objekty dekodovány a předány PDF Parseru, který s nimi dále pracuje. Ihned po předání jsou tyto objekty dále zpracovávány podle specifických znaků, které se vyskytují v datech, například v objektu *Slovník* se na samém začátku vyskytují znaky <<. Zároveň se kontroluje název objektu, podle kterého lze zjistit, o jaký typ formulářového prvku se jedná. Každý prvek formuláře má přiřazený jednoznačný název. Jako příklad lze uvést textové pole, které má při generování přiřazen název „textareaID“, kde ID je jednoznačný identifikátor textového pole, který je deklarován ve výčtovém typu **TextareaInfo**. Pokud je název totožný se specifickým názvem jakéhokoliv formulářového prvku, který je modulem podporován, je potom ihned uložen do struktury, která je po dokončení parsování poslána dále ke zpracování.

Po zpracování celého dokumentu jsou požadované objekty roztříděny na základě jejich jednoznačných identifikátorů (čísla na konci slovního identifikátoru, například „*textarea0*“, kde nula popisuje v modulu hodnotící parametr *Originality*). Následně jsou z objektů extrahovány hodnoty, které se otestují, zda jsou nebo nejsou vyplněny. Zpracování probíhá pro každý základní prvek formuláře samostatně pomocí cyklu a switche. V případě, že všechna povinná pole jsou vyplněna a neproběhla žádná chyba ve zpracování, jsou všechny hodnoty hodnotících parametrů uloženy do databáze webového portálu konference TSD.

Při nahrávání dokumentu může dojít k několika chybám, kterých se recenzent může dopustit, a proto jsou náležitě ošetřeny.

- **Neplatné PDF** – Recenzent při nahrávání dokumentu zvolí nevalidní dokument PDF (nevygenerovaný webovým portálem).
- **Neplatný identifikátor příspěvku v~recenzním řízení** – Recenzent může při nahrávání zvolit dokument PDF patřící k jiné recenzi (kontrola identifikátoru recenzního příspěvku a vědeckého příspěvku).
- **Nevyplněné požadované parametry** – Nahrávaný dokument PDF obsahuje nevyplněné povinné hodnotící parametry. Tyto parametry jsou vypsány v chybovém hlášení zobrazeném po pokusu nahrát dokument PDF do webového portálu.
- **Databázové chyby** – Při ukládání dat do databáze webového portálu může dojít k neočekávané chybě, která zapříčiní nesprávné uložení dat.
- **Uzavřené hodnocení příspěvku** – Recenzent nahraje hodnotící dokument PDF do webového portálu v době, kdy hodnocení vědeckého

příspěvku je uzavřeno.

5.4.2 Extrakce formulářových prvků z předpřipravených dat

TCPDF parser předpřipraví data, která jsou následně předána PDF Parseru ke konečnému zpracování. PDF Parser rozdělí data na základě datového typu vyplněného obsahu, jako je například prostý text, datum nebo číselná hodnota. Příklad struktury jednotlivých objektů lze vidět na obrázku 5.1. Tento obrázek ukazuje pouze část struktury, která je mnohem rozsáhlejší a každým krokem parseru se rozkládá na menší díly.

```
Smalot\PdfParser\Header Object ( ...  
  [T] => Smalot\PdfParser\Element\ElementString Object (  
    [document:protected] => [value:protected] => group0 )  
  [V] => Smalot\PdfParser\Element\ElementName Object (  
    [document:protected] => [value:protected] => 2  
  ... )
```

Obrázek 5.1: Část dat PDF objektu

K extrahování dat a zjištění typu formulářového prvku byla vyvinuta metoda *extractElement*, viz výpis 5.4.

```
protected function extractElement($header) {  
    $elementKey = $header->getElements()['T'];  
    if ($elementKey != null) {  
        if (strpos($elementKey->getContent(), 'group') !==  
            false) $type = 'groups';  
        else if (strpos($elementKey->getContent(), '  
            textarea') !== false) $type = 'textareas';  
  
        if ($type != null) {  
            $key = $elementKey->getContent();  
            $elementValue = $header->getElements()['V'];  
            if ($elementValue != null) {  
                $value = $elementValue->getContent();  
            }  
        }  
    }  
}
```

Listing 5.4: Funkční kód pro uložení formulářových prvků z PDF objektů

Kód na samém začátku kontroluje, zda je aktuálně zpracováváný objekt pojmenován. To je zjištěno na základě indexu „T“ (T - Type) v poli elementů.

Pokud název existuje, zjišťuje se, zda se jedná o textové pole nebo skupinu výběrových tlačítek. Za předpokladu, že typ objektu je validní, je kontrolována hodnota na základě indexu „V“ (V - Value) v poli elementů. Jakmile objekt obsahuje hodnotu, jsou do parseru vráceny všechny tři hodnoty (název formulářového prvku, typ formulářového prvku a jeho hodnota), které parser uloží do pole všech extrahovaných prvků.

5.5 Výsledný vzhled PDF formuláře

Výsledný vzhled PDF formuláře lze vidět na přiloženém souboru PDF v příloze C. Před formulářem je instrukční text vysvětlující hodnocení prvních osm hodnotících parametrů. Protože se v některých případech stávalo, že uživatelé otevřeli tento dokument ve webovém prohlížeči, který umožňuje pouze otvírat soubory, nikoliv ukládat, byl zde přidán pokyn využívat prohlížeč PDF Adobe Acrobat, ale je možné využívat i jiné prohlížeče, které podporují vyplňování formulářů. Na konec formuláře byl přidán informační text popisující, jak má uživatel vyplněný formulář nahrát do webového portálu konference TSD.

5.6 Technické požadavky

Technické požadavky pro bezproblémové fungování TCPDI a PDF Parser jsou:

- **PHP verze** – TCPDI momentálně funguje na všech verzích PHP, zatímco u PDF Parseru je potřeba minimální verze 5.3. Proto je nutné mít na serveru PHP verzi alespoň 5.3.0.
- **Podpůrné knihovny** – Při kompresi stránek souborů PDF pomocí knihovny TCPDI je potřeba mít na serveru povoleno rozšíření **php-zlib**.
- **FPDF_TPL verze** – Aktuální verze TCPDI je kompatibilní pouze s verzí 1.2.3 knihovny FPDF_TPL.

6 Rozšiřitelnost modulu

Modul je naimplementovaný tak, aby byl snadno rozšiřitelný o nové funkce. Zadání bakalářské práce sice splněno bylo, ale v blízké budoucnosti mohou být požadavky na modul změněny. Jako příklad lze uvést potřebu změnit typy hodnotících formulářových prvků, například změna výběrových tlačítek na zaškrťávající tlačítka, změnu typu fontu ze základního na speciální, který není přítomen v modulu a mnoho dalšího. V této kapitole jsou popsány 3 možné návrhy na rozšíření modulu, především generátoru.

6.1 Podpora zbylých formulářových prvků

V modulu jsou momentálně podporovány dva formulářové prvky, a to textové pole a výběrové tlačítko. Pro přidání podpory jakéhokoliv formulářového prvku je potřeba splnit několik implementačních kroků. Na ukázkou bude uvedeno vytvoření podpory pro prvek zaškrťávající tlačítko.

Nejprve je potřeba vytvořit nový výčetový typ s názvem **CheckboxInfo**. Jeho instancemi bude poté možné vytvořit nové hodnotící parametry. Tyto instance budou definovány jednoznačným identifikátorem, názvem a popisem. Dále se vytvoří metoda *getConstants*, která bude jako mít návratovou hodnotu pole všech hodnotících parametrů. Důležité při vytváření je nadefinovat konstanty reprezentující název prvku při jeho vytváření pomocí TCPDF metod. Pro zaškrťávající tlačítka bude název například *checkbox* a *checkboxes* (použito při ukládání hodnot u parsování).

Pro generování kódu reprezentující určitý formulářový prvek je potřeba vytvořit novou metodu ve třídě **CheckboxInfo**. Tato metoda bude mít za úkol vložit do výsledného PDF jeden formulářový prvek daného typu, pomocí kterého bude – v našem případě – vykresleno zaškrťávající tlačítko v dokumentu. Tento formulářový prvek bude vložen jako text „checkbox“ (lze použít i pro skupinu, záleží na programátorovi) a bude doplněn o identifikátor určující počet již vytvořených skupin zaškrťávajících tlačítek. Tento identifikátor je vhodné vytvořit na začátku třídy **CheckboxInfo**.

Při parsování je potřeba přidat tento typ do podporovaných prvků v nově implementované metodě *extractElement*, viz 5.4, kde se při kontrole typu přidá nový příkaz *else if*.

Po parsování je potřeba vytvořit nové pole, do kterého se uloží extrahované hodnoty spolu s novým polem, které bude obsahovat data roztríděná

na základě jména hodnotícího parametru. Pole extrahovaných hodnot se následně roztrídí a nevalidní hodnoty se uloží do společného pole nevalidních prvků (pouze u povinných prvků). Pokud bude zvoleno kritérium povinný prvek/nepovinný prvek, je nutno tyto parametry zkontrolovat nad rámec běžné kontroly. Při nevyplnění nebo nezvolení hodnoty hodnotícího parametru nebude tento parametr reprezentován v poli všech prvků daného typu formulářového prvku a při ukládání hodnoty do databáze nebude stávající hodnota přepsána. Pokud bude jeden z povinných prvků nevalidní, je potřeba zjistit který a přidat ho do chybového hlášení pro uživatele.

Pokud jsou všechny povinné hodnoty extrahovány a uloženy, jsou následně zaneseny do databáze.

6.2 Změna fontu

V celém dokumentu jsou využity dva typy fontů – hlavní font *Helvetica* pro veškerý netučný text, zatímco font *Times New Roman* pro veškeré tučné písmo.

Při případné změně fontu pro celý dokument je potřeba změnit rodinu fontů ve třídě **Elements**. Mezi podporované základní fonty u knihovny TCPDF patří *courier* (Courier), *helvetica* (Helvetica), *times* (Times New Roman), *symbol* (Symbol) a *zapfdingbats* (Symbol). Také lze použít i fonty které nepatří mezi základní a jsou již obsaženy v knihovně TCPDF. Mezi tyto fonty patří například *freeserif*. Pro změnu normálního nebo tučného písma je potřeba přepsat hodnotu proměnné *\$normal_font* definující font využitý pro normální písmo, zatímco *\$bold_font_tcpdf* definuje typ fontu pro tučné písmo. Nachází se zde i proměnná *\$bold_font_html*, kterou je nutné přepsat taktéž při změně fontu pro tučné písmo.

V případě, že je potřeba využít fonty třetích stran, které nejsou přidány v knihovně TCPDF, je nutné je definovat. To se provede pomocí metody *addTTFfont* viz 6.1.

```
$pdf->addTTFfont('/path-to-font/font.ttf', '
    TrueTypeUnicode', '', 32);
```

Listing 6.1: Nový font vložený do knihovny TCPDF

Před zavoláním metody je potřeba získat knihovnu definující vzhled a velikost jednotlivých znaků daného fontu ve standardu *TrueType* (koncovka *.ttf*). Cesta k souboru bude použita jako první parametr metody.

6.3 Načtení nově přidaných dat z konfiguračního souboru

Konfigurační soubor *configuration.xml* obsahuje data, která se mohou častěji měnit v průběhu hodnocení bez nutnosti přepisovat zdrojový kód modulu. Pro přidání nového záznamu je nutné dodržovat stanovený postup:

1. Vytvoření nového elementu a přiřazení textu.
2. Vytvoření nové proměnné ve třídě *ConfigurationData*.
3. Načtení dat pomocí XML readeru implementovaného v PHP. Při získávání dat elementu je nutné dodržovat styl *\$reader->nazev_elementu*.

7 Ověření kvality software

Po vytvoření modulu je nutné ověřit kvalitu řešení. Testování bylo zaměřeno hlavně na kvalitu vygenerovaného dokumentu PDF a jeho následného zpracování. Pro tento účel byl vytvořen testovací scénář pro recenzenta vědeckých příspěvků. Důležitým faktorem při testování vytvořeného modulu je otestovat kompatibilitu s nejvíce využívanými webovými a prohlížeči PDF. Vyplněné testovací scénáře se nachází v příloze bakalářské práce.

7.1 Testování modulu

7.1.1 Generování souboru PDF

Pro generování a stažení dokumentu bylo použito šest nejčastěji využívaných webových prohlížečů.

Jako první testované prohlížeče lze zmínit *Microsoft Edge (v42.17134.1.0)* a *Internet Explorer (v11)* firmy Microsoft, které jsou již předinstalované na všech operačních systémech Windows od verze 10. Stažení proběhlo zcela v pořádku, při stahování je potřeba zvolit možnost stažení souboru do počítače, nikoliv pouze otevření souboru. Vygenerovaný soubor PDF obsahoval všechny potřebné části pro hodnocení vědeckého příspěvku. Formulář bylo možné editovat.

Generování bylo testováno i na webových prohlížečích *Mozilla Firefox (v66.0.2)* a *Google Chrome (v73.0.3683.103)*. Testování probíhalo na dvou operačních systémech Windows 10 a Linux (Ubuntu v18.04.1 LTS, Kernel 4.15.0–44-generic. I zde proběhlo generování a stažení dokumentu bez sebemenších problémů, kdy Google Chrome ihned po stažení otevřel výchozí prohlížeč souborů PDF, zatímco Mozilla Firefox nabídl možnost stažení či pouze otevření souboru PDF. Stejně jako tomu bylo u prvních dvou webových prohlížečů, bylo nutno soubor stáhnout, nikoliv pouze otevřít. Vygenerovaný soubor obsahoval editovatelný formulář i vědecký příspěvek, který je posuzován.

Poslední dva webové prohlížeče, na kterých bylo generování souboru PDF testováno, jsou *Safari (v5.1.7)* a *Opera (v58)*. Testování probíhalo pouze na systému Windows. Po stažení byl soubor PDF ihned otevřen výchozím prohlížečem PDF. Formulář byl editovatelný, veškeré části hodnotícího souboru PDF zde byly přítomny.

Protože chytré telefony má dnes skoro každý, bylo proto nutné otestovat

generování souboru PDF i na operačních systémech *Android (v4.2.1 – Jelly Bean)* a *iOS (v12.2)*. Bez sebemenších problémů byl hodnotící soubor PDF vygenerován. Hodnotící formulář byl editovatelný.

7.1.2 Vyplnění a zpracování souboru PDF

Při vyplnění vygenerovaného hodnotícího souboru PDF bylo použito sedm prohlížečů PDF, viz 7.1. K nahrání vyplněného souboru PDF byl použit webový prohlížeč Google Chrome.

Prohlížeč PDF	Verze PDF	Interaktivní formulář	Uložení hodnot
Adobe Acrobat Reader	1.6.0	Ano	Ano
Adobe Acrobat Pro	1.6.0	Ano	Ano
Evince	1.4.0	Ano	Ano
PDFElements	1.4.0	Ano	Ano
Nitro Pro	1.4.0	Ano	Ano
Sumatra	X	Ne	Ne
Foxit - Linux	1.4.0	Ano	Ano
Evince - Linux	1.4.0	Ano	Ano

Tabulka 7.1: Prohlížeče PDF použité při testování

Nejčastěji využívané prohlížeče PDF jsou *Adobe Acrobat Reader* a *Adobe Acrobat Pro* ve verzi v19.01.020091. Hodnotící formulář zde byl editovatelný, obrázky se vyskytují v perfektní kvalitě a při ukládání byla použita verze PDF 1.6.0. Po nahrání souboru PDF v recenzním řízení do webového portálu konference TSD byly všechny vyplněné hodnoty extrahovány a úspěšně uloženy do databáze.

Mezi známější prohlížeče PDF lze zařadit i *Evince (v3.32)*, který byl použit pro testování na operačních systémech *Windows 10* a *Linux*. Vzhled formulářových prvků zde vypadal trochu jinak než jak tomu bylo u Adobe prohlížečů, při ukládání souboru PDF byla použita verze PDF 1.4.0. Nahrání souboru a extrahování potřebných hodnot proběhlo bez sebemenších problémů na obou operačních systémech.

Mezi méně známé prohlížeče PDF lze zařadit *PDFElements (v6)*, *Nitro Pro (v12)* a *Foxit (v9.4.1.16828)*. Všechny tyto prohlížeče bezproblémově uložily všechny vyplněné hodnoty ve verzi PDF 1.4.0. Stejně jako tomu bylo u *Evince*, i zde se nevyskytl žádný problém při nahrávání souboru a extrakci požadovaných hodnot.

Jako poslední prohlížeč PDF byl použit prohlížeč *Sumatra (v3.1.2)*. Sumatra nepodporuje editovatelné formuláře, slouží pouze k prohlížení souborů

PDF. Proto z tohoto důvodu nelze použít prohlížeč Sumatra pro vygenerovaný soubor PDF v recenzním řízení.

7.1.3 Nalezené chyby při testování

Při testování modulu se objevily dvě větší chyby, které nebyly zapříčiněné použitou knihovnou. Jednalo se o chyby při zpracovávání nahraného souboru PDF.

První chyba byla objevena při testování knihovny TCPDI. Při kontrole extrahovaných hodnot z nahraného souboru PDF se u každé skupiny přepínačích tlačítek kontroluje, zda obsahuje hodnotu „Off“. Pokud obsahuje, pak uživatel nezvolil ani jednu možnost v dané skupině přepínačích tlačítek. Bohužel tento postup fungoval pouze u formulářů vygenerovaných knihovnou mPDF, nikoliv u formulářů vygenerovaných knihovnou TCPDI. U TCPDI se při nezvolené možnosti ve skupině přepínačích tlačítek se daný hodnotící prvek ani neextrahuje, tudíž není dostupný a nelze ho nijak zkontrolovat. Proto byla přidána kontrola, zda extrahovaná data obsahují všechny hodnotící parametry u všech formulářových prvků.

Druhá chyba byla nalezena při ukládání hodnot do databáze. V hodnotícím formuláři se vyskytují parametry, které uživatel nemusí vyplňovat. Pokud tyto položky nejsou vyplněné, nejsou při nahrávání extrahovány a následně uloženy do databáze, tudíž při nahrání nového souboru PDF se stávalo, že zde zůstali z předchozího hodnocení komentáře. Proto bylo nutné zjistit zda tyto nepovinné parametry jsou vyplněny a pokud ne, byly explicitně vytvořeny v modulu s přiřazenou hodnotou, která byla prázdný řetězec. Tímto způsobem byly přepsány všechny parametry z minulého hodnocení.

7.2 Testovací scénář

Testovací scénář je orientován především na vzhled celého dokumentu a rozložení jeho jednotlivých částí. Dále byly položeny otázky na bezproblémové ovládání modulu (vygenerování a nahrání dokumentu PDF). Poslední otázka byla zaměřena na poznámky a připomínky ohledně vytvořeného modulu, které by do budoucna mohly posloužit při případném rozšiřování modulu. Výsledný testovací scénář předložený uživatelům konferenčního systému TSD obsahuje tyto otázky:

- Vyskytly se nějaké problémy při stahování či nahrávání dokumentu PDF?
- Jak by jste ohodnotil vzhled dokumentu jako celek?

- Jak hodnotíte vzhled formuláře a použité prvky reprezentující jednotlivé hodnotící parametry?
- Byla velikost textového pole dostatečně velká pro případně komentáře ohledně vědeckého příspěvku?
- Jak hodnotíte rychlost stažení a nahrání dokumentu PDF?
- Byly Vámi vyplněné hodnoty správně nahrány do webového portálu?
- Děkuji za vyplnění dotazníku. Pokud máte jakékoliv další poznámky, připomínky či návrhy, uveďte je prosím zde:

7.3 Výsledky uživatelského testování

Funkčnost modulu byla otestována dvěma uživateli webového portálu konference TSD. Níže jsou popsány souhrny jednotlivých částí testovacího reportu. V kapitole B jsou podrobně rozepsány všechny odpovědi uživatelu.

7.3.1 Vzhled dokumentu PDF

Vzhled hodnotícího formuláře byl podle uživatelů vcelku přehledný a věcný, ale svým vzhledem je ani neurazil, ani nenadchnul. Velikost textových polí byla dostatečně velká pro zadání všech informací. Ohledně celkového vzhledu dokumentu měli uživatelé námitku na odsazení výběrových tlačítek. Ocenili by především větší odsazení skupin výběrových tlačítek od ostatních sekcí.

7.3.2 Nalezené chyby a připomínky

Při stahování a nahrávání dokumentů se nevyskytly žádné problémy a vše proběhlo bez výraznějších prodlev mezi jedna až dvěmi vteřinami. Při nahrávání vyplněného souboru PDF v recenzním řízení do webového portálu se u prvního uživatele vyskytly problémy, zatímco druhý neměl žádné připomínky. Připomínky prvního uživatele byly: portál nezná háčky, čárky a opravuje je na nesmyslné znaky, je schopný nahrát i nesmyslný text ve stylu samých teček a problém při ukládání hodnot do databáze, který spočíval v přepisování starých hodnot novými. Co se týče špatného zobrazování háčků a čárek tak tento problém byl vysvětlen v kapitole 5.3.3. Ohledně nesmyslného textu, většina hodnotících parametrů je povinná a proto musí být i v off-line formuláři náležitě vyplněny všechny povinné parametry. Poslední

připomínka byla zjištěna ke konci autorova testování, kdy už byly rozeslány testovací reporty všem uživatelům, viz kapitola 7.1.3.

8 Závěr

V rámci této práce se autor seznámil s konferenčním systémem TSD a prostudoval již existující PHP knihovny pro práci s formátem PDF.

Na základě nabytých znalostí byl navržen a implementován modul umožňující vygenerování hodnotícího souboru PDF a jeho zpracování. Modul využívá dvě PHP knihovny třetích stran, které zajišťují veškerou funkcionalitu.

Pro generování hodnotícího souboru PDF byla nejdříve použita knihovna mPDF, která ale měla jeden zásadní nedostatek, který spočíval v nezpůsobilosti sloučit soubory PDF ve verzi PDF nad 1.4. Po zjištění tohoto nedostatku byla vybrána knihovna TCPDI, se kterou lze tyto soubory PDF slučovat i přes to, že vzhled výsledného vygenerovaného souboru byl horší.

Pro zpracování hodnotícího souboru PDF byla použita knihovna PDF Parser, která měla za úkol zpracovat nahraný soubor a následně z něj extrahovat hodnotící parametry. Protože tato knihovna pouze zpracovává veškerá data a nerozlišuje zda je daný objekt formulářový prvek, obrázek nebo text, musela být upravena. Úprava spočívala v přidání kontroly objektu při zpracovávání, zda se jedná o daný hodnotící parametr. PDF Parser je distribuován pod licencí GPLv2, která umožňuje takovéto úpravy zdrojového kódu.

Zadání práce bylo splněno ve všech bodech. Výsledný modul umožňuje recenzentům nechat si vygenerovat hodnotící soubor PDF pro posuzování vědeckého příspěvku off-line a následně ho nahrát zpět na webový portál konference TSD. Veškerá funkcionalita byla zajištěna pomocí PHP knihoven třetích stran, které autor práce zakomponoval do systému s využitím pomocného zdrojového kódu.

Literatura

- [1] *PDF formuláře: obecný úvod* [online]. 2002. [cit. 19.03.2019]. Dostupné z: <http://www.grafika.cz/rubriky/pdf---adobe-acrobat/pdf-formulare-obecny-uvod-130460cz>.
- [2] *PDF formuláře: Popis formulářových prvků* [online]. 2002. [cit. 19.03.2019]. Dostupné z: <http://www.grafika.cz/rubriky/pdf---adobe-acrobat/pdf-formulare-popis-formularovych-prvku-130502cz>.
- [3] *Compression in PDF files* [online]. Prepressure, 2017. [cit. 16.03.2019]. Dostupné z: <https://www.prepressure.com/pdf/basics/compression>.
- [4] CHRISTENSSON, P. *PDF Definition* [online]. TechTerms. Sharpened Productions, 2018. [cit. 15.03.2019]. The Tech Terms Dictionary. Dostupné z: <https://techterms.com/definition/pdf>.
- [5] KING, J. *Introduction to the Insides of PDF* [online]. Adobe, 2005. [cit. 17.03.2019]. Dostupné z: <https://www.adobe.com/technology/pdfs/presentations/KingPDFTutorial.pdf>.
- [6] LUKAN, D. *PDF File Format: Basic Structure* [online]. InfoSec, 2018. [cit. 26.03.2019]. Dostupné z: <https://resources.infosecinstitute.com/pdf-file-format-basic-structure>.
- [7] NIKOLAEV, A. *PHP PDF to HTML: Convert PDF to HTML using Poppler* [online]. 2018. [cit. 21.03.2019]. Dostupné z: <https://www.phpclasses.org/package/9423-PHP-Convert-PDF-to-HTML-using-Poppler.html>.
- [8] ROUSE, M. *Portable Document Format (PDF)* [online]. TechTarget, 2010. [cit. 16.03.2019]. Dostupné z: <https://whatis.techtarget.com/definition/Portable-Document-Format-PDF>.
- [9] WHITINGTON, J. *PDF Explained, Chapter 3. File Structure*. O'Reilly Media, Inc., 2011. ISBN 9781449310028.

A Uživatelská dokumentace

Pro nasazení modulu na webový server je potřeba provést tyto kroky:

1. V kořenovém adresáři serveru nebo v aktuálním pracovním adresáři je potřeba vytvořit složky **php** a **php/offline-review**.
2. Přesunout celý modul do adresáře **php/offline-review**.
3. V souboru *orlib.php* přepsat konstantu „DATA_ROOT“ na aktuální pracovní adresář či kořenový adresář (přednastavený pracovní adresář je definován v konfiguračním souboru webového portálu konference TSD).



Postup pro uživatele při využití tohoto modulu:

1. Přihlášení do webového portálu konference TSD.
2. Přesunout se do záložky „My Reviews“.
3. Vybrat si ze seznamu jeden vědecký příspěvek a kliknout na příslušné tlačítko pro otevření hodnotícího formuláře („Review“ tlačítko).
4. Pod textem „Offline review form“ (v pravé horní části webového prohlížeče) lze vidět 2 tlačítka:
 - Levé tlačítko (obrázek bez zelené šipky) má za úkol vygenerovat příslušný hodnotící soubor k aktuálně hodnocenému vědeckému příspěvku, viz A.1.
 - Pravé tlačítko (obrázek se zelenou šipkou) má za úkol zpracovat příslušný hodnotící soubor a nahrát do databáze extrahované hodnotící parametry, viz A.2.

REVIEW ID# 386

BY VOJTĚCH DANIŠÍK (UID# 10)

Reviewed submission ID#: 129
Reviewed submission title: Titulek vedeckeho príspevku

Online review form:
 

General instructions for the assessment: The better assessment, the higher mark, i.e. 0 = the **poorest** mark, 10 = the **best** mark (the only exception is the 'Amount of rewriting' field where 0 means 'no rewriting necessary' and 10 means 'the paper must be entirely rewritten'). **Please, use your common sense or ask the organizers if unsure.** If you don't feel like filling in the web review form, you may download the **offline review form**. Fill it in without any stress and then upload it back using the two buttons above.

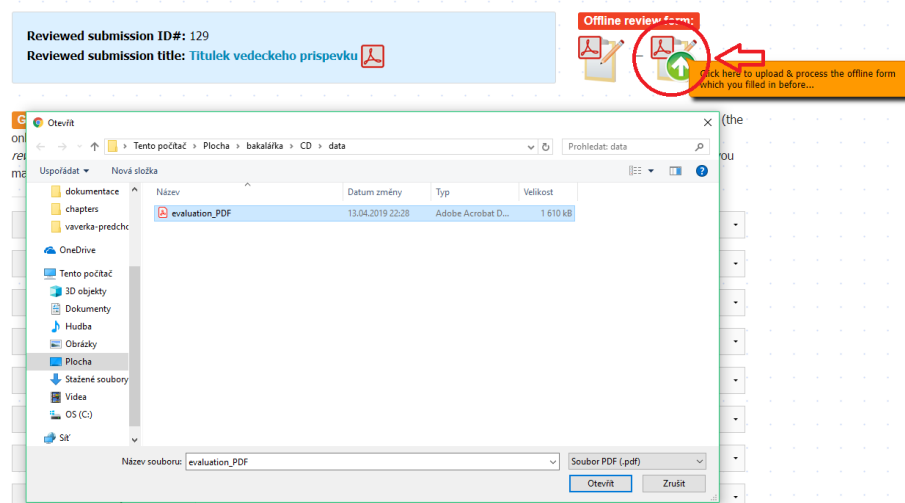
(AUTOSAVED)

Originality:	10
Significance:	9
Relevance:	8

Obrázek A.1: Ukázka stažení hodnotícího souboru PDF

REVIEW ID# 386

BY VOJTĚCH DANIŠÍK (UID# 10)



Obrázek A.2: Ukázka nahrání hodnotícího souboru PDF

B Testovací reporty

B.1 Tester 1

Vyskytly se nějaké problémy při stahování či nahrávání dokumentu PDF?

Ne, vše se nahrálo i stáhlo v pořádku a včas.

Jak by jste ohodnotil vzhled dokumentu jako celek?

Běžný formulář, který neurazí, ale ani nenadchne, každopádně je vcelku přehledný.

Jak hodnotíte vzhled formuláře a použité prvky reprezentující jednotlivé hodnotící parametry?

Oceňuji prostor pro delší odpovědi, ale zaškrťovací políčka jsou příliš u sebe a je problém rozpoznat, co je vyplněno a co ne tím, že jsou jednotlivé body od sebe vcelku vzdáleny.

Byla velikost textového pole dostatečně velká pro případně komentáře ohledně vědeckého příspěvku?

Ano.

Jak hodnotíte rychlost stažení a nahrání dokumentu PDF?

Nepozorovala jsem výraznější prodlevy.

Byly Vámi vyplněné hodnoty správně nahrány do webového portálu?

Bohužel nikoli. Portál nezná háčky, čárky apod. a opravuje je na nesmyslné znaky. Je schopný nahrát i nesmyslný text, např.: samé tečky apod. Pokud nahraji první verz souboru vyplněnou a druhou vyplněnou jen zčásti, soubory se spojí a vyskytuje se vždy něco z prvního i z druhého souboru. Možná by nebylo od věci zvýraznit opravené věci z přehrávání.

Děkuji za vyplnění dotazníku. Pokud máte jakékoliv další poznámky, připomínky či návrhy, uveďte je prosím zde:

Nevyplněno

B.2 Tester 2

Vyskytly se nějaké problémy při stahování či nahrávání dokumentu PDF?

Žádné problémy nenastaly během testování.

Jak by jste ohodnotil vzhled dokumentu jako celek?

Dokument byl přehledný a věcný.

Jak hodnotíte vzhled formuláře a použité prvky reprezentující jednotlivé hodnotící parametry?

Každá sekce s výběrovými tlačítky bodového ohodnocení by mohla být více odsazena od ostatních sekcí.

Byla velikost textového pole dostatečně velká pro případně komentáře ohledně vědeckého příspěvku?

Textová pole byla dostatečně velká pro zadání všech informací.

Jak hodnotíte rychlost stažení a nahrání dokumentu PDF?

Vše proběhlo rychle v rámci 1-2 vteřin.

Byly Vámi vyplněné hodnoty správně nahrány do webového portálu?


Všechny zadané informace byly správně rozeznány a nahrány na portál včetně diakritiky a speciálních znaků.

Děkuji za vyplnění dotazníku. Pokud máte jakékoliv další poznámky, připomínky či návrhy, uveďte je prosím zde:

Nevyplněno

C Vzhled PDF formuláře

REVIEW ID #386 : Titulek vedeckeho prispevku



Offline Review Form for Submission S-ID #129

Titulek vedeckeho prispevku

Review by Vojtěch Danišík

General instructions for the assessment: The better assessment, the higher mark, i.e. 0 = the poorest mark, 10 = the best mark (the only exception is the 'Amount of rewriting' field where 0 means 'no rewriting necessary' and 10 means 'the paper must be entirely rewritten'). Please, use your common sense or ask the organizers if unsure. **This form should be filled in using Adobe Acrobat - please, do not use the built-in PDF viewer in your browser.**

Originality - Rate how original the work is:
☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Significance - Rate how significant the work is:
☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Relevance - Rate how relevant the work is:
☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Presentation - Rate the presentation of the work:
☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Technical quality - Rate the technical quality of the work:
☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Overall rating - Rate the work as a whole:
☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Amount of rewriting - Express how much of the work should be rewritten:
☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Reviewer's expertise - Rate how confident you are about the above rating:
☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Main contributions - Summarise main contributions:

REVIEW ID #386 : Titulek vedeckeho prispievku



Positive aspects - Recapitulate the positive aspects:

Negative aspects - Recapitulate the negative aspects:

Comment (optional) - A message for the **author(s)**:

Internal comment (optional) - An internal message for the **organizers**:

After filling the form in, please, upload it to the TSD2019 web review application: Go to URL <https://www.kiv.zcu.cz/tsd2019> and after logging in, please, proceed to section 'My Reviews', select the corresponding submission and press the 'Review' button. There, you'll be able to upload this PDF file.