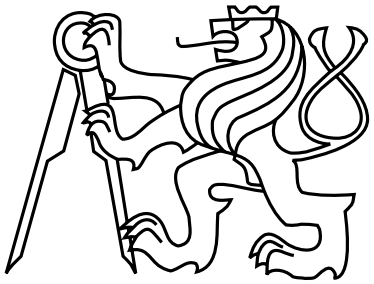
České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra kybernetiky



Bakalářská práce

PHP modul pro práci s PDF formuláři a podepisování PDF

Jan Kubát

Vedoucí práce: Ing. Radomír Polách

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Inteligentní systémy

27. května 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Jan K u b á t

Studijní program: Softwarové technologie a management

Obor: Inteligentní systémy

Název tématu: PHP modul pro práci s PDF formuláři a podepisování PDF

Pokyny pro vypracování:

1. Proveďte analýzu dostupných knihoven pro vyplňování formulářových polí v PDF.
2. Proveďte analýzu dostupných knihoven pro podepisování PDF.
3. Navrhněte a implementujte PHP modul pro práci s PDF s následující funkcionalitou:
   - vyplňování formulářových polí
   - čtení hodnot
   - zápis hodnot
   - zamčení pole
   - získání povolených hodnot u selectů a radio buttonů
   - digitální podepsání souboru
   - přidání digitálního podpisu k souboru
   - přidání vodoznaku
   - přidání vodoznaku na jednotlivé stránky nebo na celý dokument
4. Implementaci otestujte.

Seznam odborné literatury: Dodá vedoucí práce.

Vedoucí bakalářské práce: Ing. Radomír Polách

Platnost zadání: do konce letního semestru 2013/2014

L.S.

prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 24. 6. 2013

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Radomíru Poláchovi za jeho trpělivost, cenné rady a konzultace.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 23.5.2014 .....

Abstract

The objective of this Bachelor Thesis is to introduce existing solutions for work with Adobe PDF Documents, in particular with their AcroForm component, which allows fillable form fields in documents. Following with design and implementation of an extension module for PHP that would allow to work with these forms in PHP language.

Abstrakt

Předmětem této práce je rešerše současných řešení pro práci s PDF dokumenty, konkrétně jejich komponentou AcroForm, která umožňuje práci s formuláři. Dále pak vytvoření rozšiřujícího modulu pro PHP o funkcionalitu, která by umožnila práci s těmito formuláři v jazyce PHP.

Obsah

Seznam obrázků

Seznam tabulek

Kapitola 1

Úvod

Portable Document Format (dále jen PDF) je dnes jedním z nejrozšířenějších univerzálních formátů. Soubory v tomto formátu lze otevřít na většině zařízení, včetně chytrých mobilních telefonů nebo elektronických čteček, a na všech těchto platformách se zobrazí stejně. Příkladů využití PDF je mnoho - internetový obchod tento formát použije k odeslání faktury zákazníkovi, dodavatel energií, banka či pojišťovna v něm odešlou návrh smlouvy nebo leták s reklamním materiálem. Ať už je výsledné použití jakékoliv, téměř každý se dnes s těmito dokumenty někde střetne. V této práci se budu zabývat jednou součástí tohoto formátu, konkrétně komponentou AcroForm, která slouží k vyplňování formulářů.

Jak jsem již zmínil, jedním z možných využití dokumentů ve formátu PDF jsou například návrhy smluv. Takový dokument může obsahovat i relativně velké množství informací o klientovi, které dotyčný může chtít upravit. V souboru vytvořeném dle původních specifikací PDF z roku 1993 by něco takového udělat nemohl, veškeré informace byly pevně dané a v běžném prohlížeči PDF dokumentů neupravitelné. Autor PDF formátu, společnost Adobe Systems, proto specifikaci roku 1996 rozšířila o komponentu AcroForms, která umožňuje práci s interaktivními formuláři.

Toto vylepšení umožňuje využití nových objektů, jako jsou "Select box", "Text box", "Radio button"známé z formulářů ve formátu HTML a další, viz. Typy objektů (2.1.1). Validaci formulářů lze zajistit například pomocí jazyka JavaScript. [9] [7]

Ve své práci jsem se zaměřil právě na práci s touto komponentou, a to sice prostřednictvím programovacího jazyka PHP. Jedná se totiž podle mého názoru o funkčnost, která může velmi ulehčit práci programátorovi i uživateli (klientovi) a zatím jsem nenarazil na žádné řešení, které by poskytovalo kombinaci dobré dokumentace, jednoduché práce a v případě komerčních produktů i přijatelné ceny - rozbor existujících řešení je rovněž součástí mé práce.

Kapitola 2

Rešerše

V této části práce blíže vysvětlím, jaké možnosti vlastně PDF formát nabízí, jaké jsou jeho jednotlivé komponenty, detailněji popíšu možnosti AcroForms a jejich variant a pokusím se srovnat existující řešení pro práci s interaktivními formuláři v PDF dokumentech, a to jak v jazyce PHP, tak pro lepší představu se budu zaobírat i knihovnami pro jazyk Java, který, jak jsem zjistil, má oproti PHP v této oblasti značný náskok.

2.1 Historie PDF

Formát PDF vytvořila společnost Adobe Systems na počátku roku 1990 za účelem sdílení dokumentů, včetně obrázků a formátovaného textu, nezávisle na platformě, kterou daný uživatel používá. Cílem bylo odstranění potřeby kompatibilního software na mnoha různých platformách a do budoucna snad i zavedení standardu pro sdílení dokumentů. V té době vznikalo i několik podobných konkurenčních projektů, včetně druhého formátu společnosti Adobe Systems - PostScript (dále jen PS). Dalšími podobnými projekty byl například DjVu, který se vyvíjí do dnes, nebo Envoy, který však konkurenční střet s PDF nepřežil, a dnes se již téměř nepoužívá.

Formát vznikal v době, kdy ještě internet nebyl tak běžnou záležitostí, jako je tomu dnes, takže se potýkal s nedostatečným výkonem horších či starších zařízení nebo pomalejší konektivitou, oboje způsobovalo výkonnostní problémy při stahování nebo následném vykreslování dokumentů.

V roce 1993 Adobe zveřejnilo první specifikaci PDF a současně vydalo zcela zdarma Adobe Acrobat Reader 2.0, sloužící k prohlížení těchto dokumentů a Adobe Acrobat 2, ve kterém lze PDF dokumenty vytvářet - ten je však již placený. Původně bylo zapotřebí si Acrobat zakoupit i k pouhému prohlížení dokumentů. V průběhu let Adobe formát dále vylepšovalo a vydávalo další verze, až se v lednu roku 2008 stává Adobe PDF 1.7 standardem ISO 32000:1:2008. Rovněž došlo k uvolnění téměř všech patentů souvisejících s implementací, použitím, prodejem a distribucí formátu PDF. [9]

Tabulka 2.1: Vývoj verzi PDF

Verze	**Rok vydání**	**Změny ve formátu**	**Verze Acrobat Readeru**
1.0	1993		1.0
1.1	1996	Heslo dokumentu, šifrování dokumentu (RC4 40 bit), barvy nezávisle na zařízení	2.0
1.2	1996	AcroForms, Forms Data Format pro import, export, přenos a příjem dat z webu, eventy myši, externí přehrávání filmů, externí nebo vestavěné přehrávání hudby, zlib/deflate komprese textu a binárních dat, podpora Unicode, pokročilé funkce práce s barvami a proxy pro obrázky.	3.0
1.3	2000	Digitální podpisy, ICC a DeviceN barvy, JavaScriptové akce, vestavěné streamy všech typů, nové typy anotací, PostScript level 3 obrazový model, masky obrázků, alternativní reprezentace obrázků, vyhlazování hran, vylepšení číslování stránek, schopnost ukládat webovou stránku jako PDF včetně informací o stránce, reprezentace logické struktury nezávisle na grafické, podpora pro CIDFonts, Datové struktury pro mapování řetězců a čísel na PDF objekty, podpora informací pro předtiskovou produkci,nové funkce pro několik funkčních objektový typů.	4.0
1.4	2001	JBIG2, průhlednost, RC4 40-128 bitů, vylepšení formulářů a FDF, XML odesílání formulářů, vestavěné FDF soubory, Unicode specifikace exportovaných hodnot z polí, vzdálená spolupráce a digitální podpisy u FDF souborů, Přístupnost pro postižené uživatele, Metadata prostřednictvím Extensible Metadata Platform (XMP), Tagování PDF, tiskové značky, zobrazování a náhled tiskových hranic, předdefinované CMapy, Alternativní prezentace, importování obsahu z jiného dokumentu PDF, EmbendedFiles ve slovníku jmen dokumentu - standartní lokace pro vestavěná data, OCR textová vrstva.	5.0

Pokračuje na další straně

Tabulka 2.1 – *Pokračuje z předchozí strany*

Verze	**Rok vydání**	**Změny ve formátu**	**Verze Acrobat Readeru**
1.5	2003	JPEG 2000, vylepšená podpora pro vestavěná multimédia, objektové streamy, XML Forms Data Format (XFDF, náhrada formátu z verze 1.4), rich text prvky a atributy založené na Adobe XML Forms Architektuře (XFA) 2.02, bezpečnostní handlery pro veřejné klíče PKCS#7 s SHA-1, RSA o až 4096bitech, vlastní šifrovací algoritmy pro bezpečnostní handler, Sekce dokumentu mohou být skryty autorem nebo čtenářem pro některé prvky, Alternativní prezentace jediným typem je slideshow vyvolaná JavaScriptovou akcí, přerušena podpora Windows 98.	6.0
1.6	2004	podpora 3D(např. Universal 3D), podpora OpenType fontů, podpora XFA 2.2 rich text elementů a atributů, AES šifrování, PKCS#7 se SHA256, DSA až 4096bitů, NChannel barevné prostory, vylepšení podpory vestavěných příloh - včetně linkování mezi dokumenty, vylepšení digitálního podepisování souborů.	7.0
1.7	2006	vylepšená podpora 3D objektů, XFA 2.4, více přiložených souborů (portable colections), požadavky dokumentu na PDF prohlížeč, nové stringové typy, PKCS#7 se SHA384, SHA512, RIPEMD160.	8.0
1.7 Extension Level 3	2008	256 bit AES, XFA Datové sady, vylepšená podpora Flash aplikací, XFA 2.5 a 2.6 rich text konvence.	9.0
1.7 Extension Level 5	2009	XFA 3.0	9.1
1.7 Extension Level 8	2011		10

[9]

2.1.1 Základní charakteristiky PDF Formátu

Jelikož celá definice PDF souboru je poměrně komplexní a většina jejích částí není potřebná pro tuto práci, budu se zaobírat pouze těmi částmi, které souvisí s tématem této práce.

Základním kamenem pro PDF byl jazyk PostScript, některé jeho části jsou implementovány odlišně a jiné nejsou implementovány vůbec. PDF však nabízí i nové vlastnosti, které v původním PostScriptu nebyly. Přidána byla schopnost vložit do dokumentu omezenou podmnožinu znakové sady a podpora uložení jednotlivých částí dokumentu do jediného souboru s využitím komprese.

Struktura souboru [6]

- Head - obsahuje pouze jeden řádek s verzí PDF
- Body - obsahuje všechny objekty v souboru - fonty, formuláře, záložky atd.
- xref table - obsahuje ukazatele na všechny objekty v PDF souboru
- trailer - obsahuje ukazatele na xref table a klíčové objekty obsažené ve slovníku traileru

Uvedená struktura platí u původního, nezměněného, souboru. Jestliže je soubor změněn a uložen, tyto změny se ukládají inkrementálně jako nové části Body, xref table a trailer s každým uložením.

Pokud se ale používá volba "Uložit jako", uložení se chová stejně jako vytvoření nového dokumentu - všechny změny spojí do jediného bloku Body, xref table a trailer. Adobe Acrobat se proto dokonce každé desáté uložení ptá, zda chceme soubor "Uložit jako", aby se zredukovala jeho velikost.

Objekty v PDF Tělo dokumentu je tvořeno z jednotlivých objektů, standard PDF je rozlišuje na následujících 8 základních typů:

- **Boolean (Booleovská hodnota)** - Hodnoty true nebo false, objevují se v polích, slovnících nebo jako výsledky logických funkcí PostScriptu
- **Numbers (Čísla)** - PDF pracuje s číselnými typy integer a real (celá a desetinná čísla)
- **Strings (Řetězce)** - Série bajtů hodnot unsigned integer 0 až 255, k zápisu lze použít dvě varianty. První je se sekvence literálů uzavřená do kulatých závorek (), druhou možností jsou hexadecimální data uzavřená do lomených závorek <>
- **Names (Jména)** - Jedná se o symbol, který je definovaný unikátní sekvencí znaků. Tento symbol nemá vnitřní strukturu a pokud jsou 2 objekty definované stejnou sekvencí znaků, jedná se o stejné objekty. Jméno začíná znakem "/", který do něj ovšem nepatří - pouze určuje, že následující sekvence znaků je jméno.
- **Arrays (Pole)** - Jedná se o seřazenou kolekci objektů. Je to jednorozměrná množina objektů, která může být heterogenní. Zapisuje se jako sekvence objektů uzavřená do hranatých závorek [].

- **Dictionaries (Slovníky)** - Kolekce objektů indexovaná pomocí jmen. Asociativní tabulka obsahující páry jméno - objekt. Zapisuje se pomocí dvojitých lomených závorek «».
- **Streams (Streamy)** - Sekvence bajtů podobně jako řetězce, nemusí však být přečtena celá najednou. Proto se využívá k uložení většího množství dat - obrázky, videa, hudba, popisky stránek...
- **Null (Prázdný objekt)** - Označuje se pomocí klíčového slova null, jeho typ a hodnota se nerovná žádnému jinému objektu.

[7]

Fonty v PDF [5]
Existuje 14 základních typů písma pro PDF dokumenty. Ty podle specifikací PDF musí být vždy v prohlížeči PDF dokumentů a není třeba je přikládat do PDF souboru. Jak je zmíněno výše, PDF formát podporuje přiložení fontů typu TrueType [11] a od verze 1.6 i fonty typu OpenType [11].

Standardní písma:

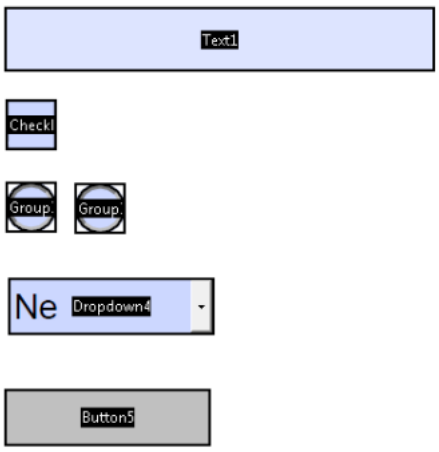
- **Times (v3)** (normální, kurzíva, tučné a tučná kurzíva)
- **Courier** (normální, kurzíva, tučné a tučná kurzíva)
- **Helvetica** (v3) (normální, kurzíva, tučné a tučná kurzíva)
- **Symbol**
- **Zapf Dingbats**

Kódování znaků v PDF [7]
Jedná se o způsob, jakým je textový řetězec uložen, a mapování hodnot typu integer na konkrétní znak z aktuálního fontu. Je třeba brát na vědomí že ne každý font má definovaný znak, který chceme použít - například velká část nestandardních fontů nepodporuje znaky ze sady Latin Extended-A. To je velmi důležité, protože právě tato část obsahuje české znaky.

Existuje mnoho různých typů kódování včetně WinAnsi a MacRoman, které jsou odvozené z historických vlastností systémů Windows a Macintosh. PDF umožňuje specifikovat předdefinované kódování, použít vestavěné kódování fontu a nebo poskytnout vyhledávací tabulku. Principy kódování v PDF byly navrženy pro standardní písma (2.1.1) a jejich použití s TrueType fonty je složité.

2.1.2 AcroForms

Komponenta AcroForms, také známá jako Acrobat forms, byla do standardu PDF přidána roku 1996 (do verze 1.2) a slouží k vytváření dynamický formulářů. Tyto formuláře jsou velice podobné těm, které známe z webových prezentací, což není náhoda - formuláře mají



Obrázek 2.1: Ukázka typů formulářových polí v PDF

kompatibilní specifikaci s formuláři v HTML, konkrétně od verze PDF 1.5 s HTML 4.01, do té doby s historickým HTML 2.0.

Při vyplňování formuláře je nutné ověřit platnost vkládaných dat, tedy zdali jsou data v požadovaném formátu (například platný tvar emailové adresy). K tomu lze využít kód ve skriptovacím jazyce JavaScript.

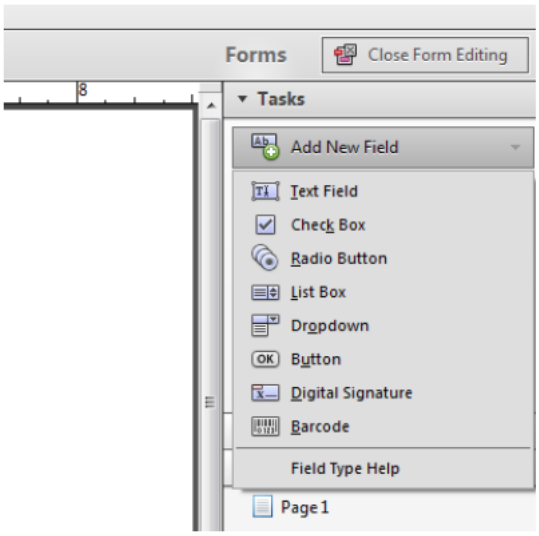
PDF umožňuje data vyplněná ve formuláři odeslat na zvolenou URL adresu v síti internet, kde je pak lze dále zpracovávat. K odeslání dat lze využít jednoho z podporovaných formátů:

- **HTML Form Format** (HTML 2.0 od verze PDF 1.2, HTML 4.01 od verze 1.5)
- **FDF** (Forms Data Format - sada párů klíč - hodnota)
- **XFDF** (XML Forms Data Format, od verze 1.5, nahrazuje XML formát z verze 1.4)
- **PDF** (Odešle se celý PDF soubor)

V případě potřeby lze využít formáty FDF nebo XFDF a data formuláře uchovat v externím souboru. Existují nástroje, které poté umí tyto soubory spojit do PDF nebo naopak data z dokumentu extrahovat do těchto formátů. Příkladem takového nástroje je pdftk, který je dostupný ve většině linuxových distribucí.

Adobe Acrobat je oficiálním nástrojem pro tvorbu a editaci PDF dokumentů, umožňuje tedy i vkládání formulářových polí a jejich editaci.

Využití této technologie je poměrně široké, osobně se pravidelně setkávám s předvyplňováním různých druhů smluv. [7] [13]



Obrázek 2.2: Ukázka nabídky pro práci s formulář v Adobe Acrobat XI

2.1.3 Digitální podpis v PDF

Jedná se o formu ověření pravosti dokumentu za pomocí unikátního otisku dokumentu a šifrovacích algoritmů typu RSA nebo DSA. Princip elektronického podpisu lze zjednodušeně popsat následovně: Nejprve se za pomoci hashovací funkce vytvoří otisk dokumentu (hash). Hashovací funkce by měla být jednosměrná, tedy z otisku nelze získat zpět původní dokument, a bezkolizní - neměla by nastat situace, kdy pro dva různé dokumenty vznikne shodný otisk. Otisk se dále zašifruje soukromým klíčem, výsledek pak společně s veřejným klíčem tvoří digitální podpis. K ověření pravosti dokumentu poté stačí zašifrovaný otisk dešifrovat pomocí přiloženého veřejného klíče a porovnat s aktuálním otiskem dokumentu.

Pravděpodobně každý z nás se již setkal s využitím digitálního podpisu - v emailu, PDF dokumentu nebo jiném souboru. Jedná se o uznávanou formu ověření pravosti dokumentu. U většiny formátů se k digitálnímu podpisu využívají takzvané SMIME přílohy, což jsou přílohy, které obsahuje digitální podpis. PDF je v tomto specifické, podobně jako formáty z rodiny produktů Microsoft Office nebo LibreOffice. Podpis není přiložený k souboru, je obsažený přímo v něm, navíc nikoliv jako příloha na konci souboru, ale rovnou v těle dokumentu. Tato skutečnost celý proces podepsání dokumentu značně komplikuje.

K uložení digitálního podpisu v dokumentu se využívá speciální formulářové pole typu Sig, jehož hodnotou je digitální podpis v hexadecimálním tvaru. Aby bylo možné s podpisem pracovat, je navíc opatřen polem ByteRange, které obsahuje čtyři číselné hodnoty. Ty vymezují obsah dokumentu, který je digitálně podepsán a nesmí se měnit. První z nich je vždy 0 (označení začátku dokumentu), druhá označuje pořadí posledního bajtu před digitálním podpisem, třetí obsahuje pořadí prvního bajtu za digitálním podpisem a čtvrtá pak pořadí posledního bajtu dokumentu. [7] [13] [1] Aby bylo dokument možné podepsat, je třeba rozdělit proces do tří fází:

- **Generování struktury dokumentu** (s mock daty podpisu a ByteRange v jejich nejhorší možné podobě)
- **Vypočítání a nahrazení ByteRange**
- **Vytvoření a vložení digitálního podpisu**

[1]

K tomu je navíc potřeba se vypořádat s verzemi dokumentu, které se podepisují zvlášť a každý nový podpis vytváří vlastní verzi.

Obecně se jedná o poměrně komplikovaný proces, který přidává do dokumentu relativně velký objem informací. Celkově je funkcionalita velmi náročná na implementaci.

2.2 PHP

Jazyk PHP vznikl z jazyku PHP/FI, který v roce 1994 vytvořil Rasmus Lerdorf. Tehdy se ještě jednalo o sadu CGI binárních souborů sepsaných v jazyce C. Původním účelem těchto souborů bylo sledování návštěvnosti na webové stránce s životopisem autora. Tato sada pak dostala název "Personal Home Page Tools". Postupem času narůstala potřeba komplexnější funkcionality, až nakonec došlo k přepisu celé sady. Výsledná nová verze pak obsahovala mnohem bohatější výbavu a mimo jiné byla schopna i interakce s databází. Právě díky tomu se již PHP dalo využit i k vývoji jednoduchých aplikací, jakými jsou například knihy návštěv. [16]

Ukázkový příklad kódu napsaného v PHP/FI:

```
<!--include /text/header.html-->

<!--getenv HTTP_USER_AGENT-->
<!--ifsubstr $exec_result Mozilla-->
    Hey, you are using Netscape!<p>
<!--endif-->

<!--sql database select * from table where user='$username'-->
<!--ifless $numentries 1-->
Sorry, that record does not exist<p>
<!--endif exit-->
Welcome <!--$user-->!<p>
You have <!--$index:0--> credits left in your account.<p>
<!--include /text/footer.html-->
```

Již na první pohled je patrný obrovský rozdíl oproti současnému PHP. Pravděpodobně nejodlišnějším je zápis do HTML komentářů namísto stávajících <?php ?> znaků.

V roce 1995 došlo k uvolnění zdrojového kódu pro veřejnost, a tak vznikla možnost používat PHP Tools pro vlastní potřeby uživatelů. Rovněž se začaly objevovat opravy chyb ve zdrojových kódech. Syntaxe se tehdy podobala spíše jazyku Perl, než dnešnímu PHP. [16]

Postupem času došlo ještě k několika přepisům celého kódu a různým změnám názvů, až došlo v červnu 1998 k představení PHP 3 novým vývojovým týmem. Tato nová verze se během necelých 9 měsíců testování objevila na více než 70000 internetových doménách po celém světě. Navíc PHP 3 již nebylo limitované potřebou POSIX operačního systému a fungovalo i na systémech Windows 95/98/NT a Macintosh.

V zimě 1998 byla zahájena práce na dalším přepisu jádra systému a v polovině roku 1999 byl představen Zend Engine. Jeho autory byli Andi Gutmans a Zeev Suraski. Název Zend vychází ze zkratek jejich křestních jmen. Finálnív verze PHP 4 byla uvedena o necelý rok později. [16]

Nakonec se v červenci roku 2004 objevuje PHP 5, se zcela přepracovaným objektovým modelem a poháněné Zend Engine 2.0. Tato verze je již velmi podobná dnešnímu PHP. Podle serveru w3techs je dnes přes 80% internetových prezentací napsaných v jazyce PHP[19].

Ukázkový příklad kódu napsaného v dnešní verzi PHP:

```
<?php
$name = 'Pepo';
if (!empty($_GET['name'])) {
   $name = htmlspecialchars($_GET['name']);
}
echo "Ahoj {$name}, já jsem PHP";
?>
```

2.3 Rozšíření pro PHP

PHP nabízí široké možnosti rozšíření o různé druhy nové funkcionality, případně o alternativy k funkcionalitě stávající. Tato rozšíření mohou být buď nativní, nebo ve formě PHP knihovny. K instalaci těchto rozšíření se obvykle využívají nástroje jako je PEAR nebo Composer u knihoven a PECL u nativních rozšíření.

PEAR je oficiální služba, která se využívá k instalaci sdílených knihoven na serveru. Je doplněná službou PECL, která je jejím protějškem pro práci s nativními rozšířeními. Ta se obvykle stahují ve formě zdrojového kódu a při instalaci projdou kompilací.

Composer je novějším programem, který dnes PEAR téměř nahradil. Dříve totiž programátor musel ručně postahovat knihovny, které jeho projekt využíval a pokud je chtěl aktualizovat, musel znovu projít jednotlivé knihovny a stahovat jejich aktualizace. Composer tuto činnost automatizuje pomocí souborů composer.json, do kterého vepíšeme závislosti našeho projektu a composer.lock, který obsahuje informace o aktuálních verzích obsažených knihoven. Postará se rovnou i autoloading souborů z těchto knihoven. Jedná se o nástroj podobný například nástroji Maven z jazyka Java.

Další možností rozšíření jazyka PHP je JavaBridge, který umožňuje pomocí protokolu RPC využívat knihovny z jazyka Java. [17]

2.4 Nástroje pro práci s AcroForms

2.4.1 Apache PDFBox

Jedná se o knihovnu pro jazyk Java obsahující vše potřebné pro práci s PDF. Její API je přehledné a práce s knihovnou je dle mého názoru přívětivá. Celému projektu napomáhá také dobrá dokumentace a podpora Apache Software Foundation.

Mezi hlavní funkce knihovny patří:

- Extrahování textu pro použití v aplikace
- Spojování a rozdělování dokumentů
- Vyplňování formulářů
- PDF/A Validace proti ISO standardu
- Tisk PDF dokumentů
- Konverze PDF dokumentů do obrázků
- Tvorba PDF dokumentů
- Podepisování PDF Dokumentů

Většina funkcionality je pro tuto práci zbytečná, nicméně na vyplňování formulářů a podepisování dokumentů jsem se rozhodl blíže podívat a práci s knihovnou vyzkoušet. [12] Vše začíná objektem PDDocument, veškeré ukázky kódu jsou v jazyce Java.

```
PDDocument doc;
try {
   doc = PDDocument.load(f);
} catch (IOException e) {
    System.out.print("PDF file was not found");
return;
}
```

Třída dokumentu je obecné rozhraní pro práci s celým souborem. Obsahuje metody pro tisk, podepisování, šifrování, odemykání souboru a pro přístup k dalším částem, které jsou již více zaměřené na jednotlivé části dokumentu. Přístup k API formulářů je zajištěn třídou PDAcroForm.

```
PDDocumentCatalog catalog = doc.getDocumentCatalog(); //získá katalog objektů dokumentu
PDAcroForm form = catalog.getAcroForm(); //získá formulář z katalogu
```

Pro jednotlivá pole pak existuje abstrakce PDField, která umožňuje jejich vyplnění a čtení aktuální hodnoty.

```
pdField.getValue(); //získá aktuální hodnotu
pdField.setValue("hodnota"); //nastaví novou hodnotu
```

Práci s formuláři hodnotím kladně. Dle mého názoru je uživatelsky velmi přívětivá a jednoduchá. Práce s podepisováním dokumentů je však poněkud komplikovanější. Stará se o ni objekt PDSigner a metoda addSignature na objektu PDDocument.

Zjednodušená ukázka (kód není funkční a vyžaduje doplnění):

```
PDSigner signer = new PDSigner();
... //nastavení
document.addSignature(signer, new SignatureInterface {
byte[] signature;
... //proces generovani digitalniho podpisu

return signature;
})
```

Celkově knihovnu hodnotím kladně, poskytuje veškerou potřebnou funkcionalitu zabalenou v kvalitním rozhraní a to vše pod Apache License, Version 2.0. (A) [12]

2.4.2 iText

iText je další knihovnou pro jazyk Java. Jedná se však o komerční produkt, zdarma dostupný pouze pro OpenSource projekty. Je licencovaná pod AGPL. Tato knihovna nabízí navíc i port iTextSharp, který je klonem pro Microsoft .NET Framework (jazyky C# a VisualBasic).

Hlavní funkce knihovny:

- Generování PDF dokumentů
- Spojování a rozdělování PDF dokumentů
- Úpravy PDF dokumentů
- Vyplňování PDF formulářů
- Digitální podepisování PDF dokumentů
- Funkce pro práci s XML

Vyplňování formulářů probíhá přes jednoduché API pomocí třídy PdfReader pro načtení PDF dokumentu a PdfStamper k modifikaci a uložení dokumentu.

```
String inputPdfFilePath = "/cesta/k/vstupnimu/pdf/souboru.pdf";
String outputPdfFilePath = "/cesta/k/vystupnimu/pdf/souboru.pdf";
PdfReader reader = new PdfReader(inputPdfFilePath); //načtení souboru
PdfStamper stamper = new PdfStamper(reader, new FileOutputStream(outputPdfFilePath)); //vyt
AcroFields form = stamper.getAcroFields();
form.setField("field_id", "field value"); //vyplnění pole
stamper.close(); //ukončení práce s PDF a uložení.
reader.close(); //ukončení čtení PDF
```

Metoda setField() se používá k vyplnění všech typů polí. Pokud pole není typu "text", ale "choice", je třeba nastavit hodnoty klíčů pro jednotlivé volby.

Stejně jako u předchozí knihovny je podepisování dokumentu složitější a v případě zájmu poslouží příklady v dokumentaci na internetu.

Tato knihovna rovněž nabízí rozsáhlé možnosti s velmi přístupným API, bohužel její licence není volná jako Apache Licence. V případě použití knihovny iText v komerčních projektech je třeba zakoupit licenci. Její cena je však vysoká, v závislosti na počtu stanic, na kterých se knihovna využívá. To z ní dělá dobrý nástroj pro open source projekty nebo velké korporace. U malých firem je dle mého názoru výhodnější použít Apache PDFBox popsaný výše, který nabízí velmi podobné vlastnosti, jen má o něco složitější API. [10]

2.4.3 Seta PDF Form Filler

Knihovna Seta PDF Form Filler je vytvořena přímo pro PHP, a je i v PHP napsaná. Jedná se o placený produkt německé společnosti Setasign. Je k dispoici ve dvou verzích a nabízí pouze funkce spojené s prací s formuláři. Jeho největší slabinou je dle mého názoru horší dokumentace ve srovnání s produkty pro jazyk Java. To je částečně zapříčiněno i tím, že se jedná o projekt s neveřejným zdrojovým kódem. Maskování kódu je řešeno pomocí Ioncube nebo Zend Guard, obě řešení kód udělají nečitelným a nepoužitelným bez patřičného rozšíření na serveru - to ale například na sdíleném hostingu často nenajdeme.

Ukázka vyplnění formuláře přes Seta PDF Form Filler

```
$document = SetaPDF_Core_Document::loadByFilename($filename, $writer);
$filler = new SetaPDF_FormFiller($document);

$fields = $filler->getFields();

$fields['jmeno_pole']->getValue(); //získání hodnoty pole
$fields['jmeno_pole']->setValue('nová hodnota'); //vepsání nové hodnoty pole
```

Vyplnění formuláře je intuitivní a není složité. Knihovnu však nelze využít k jiným účelům. Společnost Setasign totiž vydává pro jednotlivé úkony individuální produkty - pro potřeby spojené s touto prací se jedná o SetaPDF-FormFiller a SetaPDF-Signer (2.4.4).

Produkt je licencován ve dvou verzích - Lite a Full. Verze Lite umí pracovat jen s poli typu "Text", pro většinu projektů bude tedy nedostačující. Verze Full zvládá práci se všemi typy polí včetně "Choice"a "Button". Cena knihovny není malá (od 180 USD za verzi Lite a od 380 USD za verzi Full, ceny k 27. 12. 2013).

Zakoupením však získám pouze jednoúčelovou knihovnu bez zdrojového kódu, navíc za poplatek pro každý projekt nebo pro celý server. Největší nevýhodou je podle mého názoru absence našeptávání v IDE. I přes jednoduchost práce s knihovnou může našeptávání při častém používání ušetřit mnoho času, navíc zrychluje i přístup k dokumentaci.

S touto knihovnou mám i praktické zkušenosti. Narazil jsem na problém s tehdy nezdokumentovanou konverzí kódování znaků. Knihovna totiž ukládá data do dokumentu v stejném kódování, v jakém je dostala, což je v pořádku. Co však neumožňuje, je získat kódování dokumentu, popřípadě funkcionalitu, která by obstarala konverzi do korektního kódování znaků.

To může způsobit celou řadu nepříjemností při vývoji, obzvláště pokud uživatel nemá s knihovnou zkušenosti. [2]

2.4.4 Seta PDF Signer

Jedná se o další produkt společnosti Setasign, tentokrát určený k digitálnímu podepisování dokumentů. Produkt má opět uzavřený zdrojový kód, distribuovaný ve formátech Ioncube nebo Zend Guard a licencovaný na projekt nebo server. Cena se pohybuje od 200 USD za licenci na projekt (cena k 27. 12. 2013), zakoupením získáme právo provozovat Seta PDF Signer na jedné doméně. API je podobné projektům z jazyka Java. [3]

Ukázka podepsání dokumentu přes Seta PDF Signer

```
$document = SetaPDF_Core_Document::loadByFilename("Boombastic-Box.pdf", $writer);
SetaPDF_Core_Writer_TempFile::setTempDir("_tmp/");

$signer = new SetaPDF_Signer($document);

// Nastaví vlastnosti podpisu
$signer->setReason('Testování');
$signer->setLocation('cvut.cz');
$signer->setContactInfo('+420 123 456 789');

// Vytvoří instanci OpenSSL modulu
$module = new SetaPDF_Signer_Signature_Module_OpenSsl();
// Nastaví podpisový certifikát
$module->setCertificate(file_get_contents("certificate.pem"));
// Nastaví privátní klíč pro podpis
$module->setPrivateKey(array(file_get_contents("private-key.pem"), "password"));

// Samotné podepsání dokumentu
$signer->sign($module);
```

[4]

Výhody a nevýhody jsou stejné jako u knihovny FormFiller (2.4.3) tedy absence našeptávání v IDE, častá absence Ioncube nebo Zend Guard na sdíleném hostingu a pro menší projekty relativně vysoká cena. [3]

2.4.5 pdftk

Jedná se o třídu nad pdftk serverem - konsolovým nástrojem, který je dostupný přes správce balíků na většině linuxových distribucí.

Mezi jeho funkce patří

- Spojování a rozdělování PDF dokumentů
- Otáčení stránek nebo celého PDF dokumentu

- Šifrování/Dešifrování dokumentů
- Vyplňování PDF formulářů
- Generování FDF šablon pro formuláře
- Vkládání pozadí nebo vodoznaků
- Reportování informací o dokumentu - metadata, záložky, rozměry
- Přidávání a upravování záložek
- Přidání/Rozbalení příloh u PDF dokumentu
- Komprese/dekomprese stránek v PDF dokumentu
- Oprava PDF dokumentu, pokud je poškozený (pouze pokud je to možné)

Tento nástroj bohužel neumí přidat digitální podpis, a k vyplnění formuláře je třeba FDF soubor. Pokud je však zapotřebí formuláře vyplňovat častěji, existuje rozšíření PHP: FDF, které umí s FDF soubory pracovat. [14]:

2.4.6 ZendPdf

Zend Pdf je komponenta Zend Frameworku, která umožňuje práci s PDF dokumenty. Od většiny knihoven pro práci s PDF se tato knihovna liší tím, že umí pracovat i s existujícími dokumenty, nikoliv pouze vytvářet nové. To dává potenciál k rozšíření knihovny o nové funkce, jako je právě vyplňování PDF dokumentů nebo jejich digitální podepisování. Ve výchozím stavu ani jednu ze zmíněných funkcí nemá, ale pro vyplňování formulářů existuje rozšíření, které umožňuje vyplňovat textová pole.

Aktuální verze knihovny je 2.0.2 a zatím pro ni ještě není k dispozici hotová dokumentace. V dřívějších verzích knihovny, tedy 1.9, však byly mimo jiné i následující funkce:

- Tvorba nových a úprava existujících PDF dokumentů
- Získání jednotlivých revizí dokumentu
- Manipulace se stránkami dokumentu (přehazování pořadí, přidávání nových, smazání existujících)
- Kreslení křivek a geometrických primitiv
- Vpisování textů za pomoci vestavěných fontů nebo vlastních TrueType fontů
- Rotace
- Inkrementální updaty PDF souboru

Knihovna již obsahuje většinu tříd potřebných pro přidání funkcionality, jako je právě vyplňování formulářů. Díky tomu je k dispozici rozšíření na vyplňování textových polí. Knihovna je distribuována pod licencí BSD-3-Clause, která umožňuje znovupoužití kódu pro vlastní knihovny/produkty a je tedy i vhodná jako základ pro další rozšiřování. [20]

2.5 Výsledek

PHP je dnes jedním z nejpoužívanějších jazyků pro tvorbu internetových prezentací. Server w3techs, který se zabývá právě statistikou použití různých jazyků a nástrojů na webu, uvádí, že k 27. 12. 2013 je podíl PHP na webu 81,6%. [19] Většina webových dynamických internetových prezentací je tedy napsaná v PHP. Dlouhou dobu nebyl na PHP postaven žádný opravdu velký projekt, lze tedy říci, že éra velkých a robustních aplikací v PHP teprve začíná.

Výstupem uvedeného srovnání knihoven pro PHP je zjištění, že v současné době není k dispozici žádná komplexní knihovna pro práci s PDF v PHP, jako je například PDFBox pro jazyk Java. Je zde tedy určitě prostor pro nové projekty, nápady a vylepšení.

Kapitola 3

Analýza a návrh řešení

3.1 Analýza

Cílem je vytvořit sadu tříd schopnou pracovat s PDF formuláři (formou zápisu a čtení hodnot) a přidat k dokumentu digitální podpis. Tato sada tříd by zároveň neměla mít komplikované rozhraní a tím by mělo být umožněno snadné používání.

Pro práci s PDF je potřeba parser dokumentů schopný načíst dokument a poskytnout jej dále v podobě, se kterou již lze pracovat v rámci této knihovny. Volba parseru má zásadní vliv na použité technologie.

3.1.1 Nativní rozšíření

Ideálním řešením z hlediska výkonu je nativní rozšíření pro PHP založené na C/C++ knihovně, která již veškerou potřebnou funkcionalitu má a stačí jí tedy obalit třídami pro PHP a poskytnout jako rozšíření. Všechny knihovny, které to umožňují, jsou však komerční a jejich licence se často vztahují až na koncový stroj, na kterém se využívají. Taková knihovna by tak měla omezení v podobě relativně vysoké ceny, což by značně limitovalo její využitelnost. Navíc rozšíření vyžaduje správcovský přístup k systému, na kterém poběží a většina internetových prezentací v PHP využívá sdílené hostingové služby. Na nich by musel instalaci umožnit poskytovatel služby. Z vlastní zkušenosti s hostingovými službami vím, že je problém i s instalací knihovny z balíčků operačního systému, vlastní rozšíření tedy nepřipadají v úvahu. Naopak velkou výhodou takového řešení je výrazně vyšší výkonnost, ve srovnání s kódem napsaným v PHP. [15]

Nejbližší možnou knihovnou pro jazyk C/C++ je knihovna PoDoFo, které je aktuálně ve verzi 0.9.2. Nejedná se však zatím o stabilní verzi a není tudíž ani příliš dobře dokumentovaná. Komunita kolem celého projektu není početná, a v případě potíží se tedy nelze spolehnout na jakoukoliv podporu. Celkově na mě knihovna nedělá dobrý dojem, částečně i proto, že podle oficiálního postupu k instalaci se instalace nezdařila a musel jsem přistoupit k alternativním metodám a ne zcela aktuální verzi knihovny. [15] [18] Další nevýhodou takové knihovny je, že ne každý programátor PHP umí programovat v jazyce C/C++, a jakékoliv rozšíření z hlediska programátora uživatele knihovny by bylo komplikované. Lze tedy říci, že jediným přínosem opravdu zůstává vyšší výkon. [8]

3.1.2 Knihovna

Další možností je PHP knihovna, sada tříd napsaných v PHP, která obstarává potřebnou funkcionalitu. Oproti nativnímu rozšíření poskytuje menší výkon, ale nabízí snadnější vývoj a údržbu bez potřeby znalosti dalšího programovacího jazyka. Další výhodou je snadná oprava chyb v kódu, možnost přidání nové funkcionality a nakonec i nápověda (našeptávání) ve vývojovém prostředí (IDE). Rovněž existují knihovny, které poskytují přijatelný základ, ze kterého lze vycházet a dále ho rozšiřovat.

Zend PDF je jednou z těchto knihoven, poskytuje příjemné rozhraní pro práci s dokumentem, potřebné zejména pro práci s formuláři. Bohužel však nenabízí žádné nástroje pro práci s digitálními podpisy. V PHP ovšem neexistuje žádná volná knihovna, která by toto umožňovala a zároveň uměla pracovat s načteným dokumentem. Existují pouze knihovny pro tvorbu nových PDF a jejich následné podepsání, nebo placené produkty, obvykle bez zdrojových kódů - využívají totiž maskování, například za pomocí IonCube. Rozšiřitelnost takové knihovny je komplikovaná a její vývoj je pomalejší, protože v IDE nelze využít našeptávání.

3.2 Návrh řešení

Po prozkoumání jednotlivých možností jsem dospěl k názoru, že nejzajímavější variantou je napsat rozšíření knihovny Zend PDF o funkcionalitu práce s formuláři. Je to také z toho důvodu, že tuto knihovnu již v projektech, kde tuto funkcionalitu potřebuji využívám k jiným účelům a není tedy zapotřebí zbytečně přidávat novou funkcionalitu. Pro správu závislostí projektu využiji composer, který je podle mého názoru nejlepším dostupným nástrojem. Práci s knihovnou bych si pak představoval následovně:

```
$document = PdfDocument::load('moje.pdf');
$form = $document->getAcroform();
$form['nazev_pole']->setValue('hodnota');
$hodnota = $form['nazev_pole']->getValue();

$document->save('moje2.pdf');
```

Takové rozhraní podle mého názoru usnadní vývoj aplikací, kde bude knihovna použita. Zároveň využívá magických metod php __get(), __set() pro přístup k objektům jednotlivých formulářových polí. K podepisování PDF jsem se rozhodl zvolit jiný způsob. V rešerši jsem zmínil knihovnu iText 2.4.2, na které staví mnoho různých projektů. Zaujal mě ovšem jeden konkrétní - PortableSigner, který poskytuje rozhraní k digitálnímu podepsání PDF dokumentů. Tento nástroj umožňuje jak grafické, tak i terminálové rozhraní a proto jsem se ho rozhodl použít k vyhotovení součásti pro digitální podpis dokumentů.

```
$certificate = new Certificate('verejny_klic', 'soukromi_klic', 'heslo');
$signer = new Signer($certificate);
$signer->sign('in.pdf', 'out.pdf');
```

Opět je cílem co nejjednodušší rozhraní, které umožňuje snadné a rychlé použití.

Kapitola 4

Realizace

Celou knihovnu jsem uzavřel do namespace `JanKubat\Pdf`.

4.1 Formulářová součást

4.1.1 PdfDocument

Třída PdfDocument obstarává základní manipulaci s dokumentem a propojení mezi Zend PDF a mou knihovnou. Obsahuje statickou metodu load, která přetěžuje původní metodu load a vytváří instanci sebe sama. Je to proto, aby při jejím zavolání nedošlo k vytvoření instance původní třídy, která by neumožňovala práci s formuláři.

Další metodou je metoda getAcroForm, která slouží k získání zinicializované třídy AcroForm.

```
/**
 * Retrieves a list with the names of the AcroForm textfields in the PDF
 *
 * @return AcroForm
 */
public function getAcroForm(StringEncoder $encoder = null) {
    if (empty($this->_acroFrom)) {
        if (!$encoder) {
            $encoder = new StringEncoder();
        }
        $this->_acroForm = new AcroForm($this->_trailer, $encoder);
    }
    return $this->_acroForm;
}
```

Třída bere jako parametr StringEncoder 4.1.3 na data, ten obstarává konverzi mezi znakovou sadou použitou ve skriptu (u PHP obvykle UTF-8) a UTF-16BE s BOM, které se využívá v dokumentech PDF. Zároveň také využívá tzv. lazy-loading pomocné struktury, to znamená, že inicializace této struktury proběhne až při prvním přístupu k ní.

4.1.2 AcroForm

Tato třída se stará o načtení dat do pomocné struktury tříd k obsluze formulářových polí. Pro snadný přístup k jednotlivým polím obsahuje metody magické metody __get a __set, které v PHP umožňují specifikovat, jak se přistupuje k jednotlivým atributům objektu, pokud nejsou nalezeny a tak je přesměrovat někam jinam, například do vnořeného objektu nebo pole. Dále jsem implementoval rozhraní arrayaccess, to umožňuje přistupovat k polím formuláře stejně jako kdyby byla v poli ($promena['klic']).

```
    function __get($name) {
    if (array_key_exists($name, $this->fields)) {
        return $this->fields[$name];
    }
    throw new UnknownFieldException(...);
}

function __set($name, $value) {
    if ($value instanceof AbstractField) {
        $this->fields[$name] = $value;
    }
    $class = get_class($value);
    throw new \InvalidArgumentException(...);
}

$form->pole->getValue();

$form['pole']->getValue();
$pole = $form['pole'];
```

Metody jsem implementoval úmyslně tak, aby jejich použití bylo transparentní a umožnilo přístup k celému objektu. Někdo jiný by mohl metody implementovat například tak, aby rovnou nastavovaly hodnoty na polích formuláře.

4.1.3 StringEncoder

Tato třída je pouze zapouzdření funkce iconv a automatické práce s Byte Order Mark, což je úvozovací znak řetězce, který určuje, v jakém pořadí jsou uloženy bajty řetězce. Oba způsoby jsou rovněž známé pod pojmy Little-endian a Big-endian.

4.1.4 AbstractField

Tato třída realizuje abstrakci a sdílenou funkcionalitu mezi jednotlivými typy polí. Rovněž také všem polím dává povinnost implementovat metody getValue a setValue. Dále obsahuje metodu create, která implementuje návrhový vzor Factory a slouží k inicializaci jednotlivých polí formuláře z textových názvů a dat uložených v PDF Dokumentu.

4.1.5 TextField

Reprezentuje textové pole. Automaticky konvertuje kódování hodnot pomocí nastaveného StringEncoderu.

4.1.6 ChoiceField

Pole volby - Selectbox, navíc obsahuje metodu getAllowedValues, která slouží k získání seznamu dostupných hodnot.

4.1.7 ButtonField

Pole reprezentující tlačíka, zejména checkboxy, nastavuje se na něm hodnota true/false a ta se následně překládá na interní hodnoty PDF dokumentu, toto pole může vždy nabývat 2 stavů, přičemž první hodnota je pro zaškrnutý a druhá pro nezaškrtnutý stav. Druhá hodnota je obvykle Off, první je ve výchozím stavu On.

4.1.8 ButtonGroupField

Reprezentuje skupinu tlačítek, tedy RadioButton. Opět má metodu getAllowedValues pro získání dostupných hodnot.

4.2 Digitální podpisy

Podepisování dokumentů obstarávají pouze 2 třídy. Jedná se o zapouzdření nad funkcionalitou poskytnutou externím programem s jediným účelem, vytvoření digitálního podpisu na dokumentu.

- Certificate
- PortableSigner

4.2.1 Certificate

Je třída reprezentující uživatelský certifikát použitý k podpisu dokumentu.

4.2.2 PortableSigner

Je třída, která na základě certifikátu vytvoří podpis k dokumentu, podepíše jej a podepsaný dokument uloží.

4.3 Testování knihovny

K odzkoušení správné funkce knihovny jsem využil FoxitPDF reader a PDF dokument s několika formulářovými prvky všech typů, do kterých jsem zkoušel vyplňovat různé hodnoty a následně ověřoval, zda jsou skutečně vyplněna.

Tento dokument jsem poté i podepsal a ověřil, že se v něm skutečně nachází validní podpis.

Kapitola 5

Závěr

Cílem práce bylo vytvoření knihovny, která by umožnila vyplnění PDF interaktivních formulářů, načtení hodnot, které v nich již jsou vyplněné a digitálně podepsat dokument. V průběhu rešerše bylo zmíněno několik již existujících řešení i pro jiné programovací jazyky a došlo i k jejich porovnání. Prostudoval jsem si vlastnosti PDF dokumentů a blíže se seznámil s jejich specifikacemi. Jako hlavní přínos celé práce vidím možnost využití nástroje s otevřeným zdrojovým kódem, který je v jazyce PHP ve své kategorii ojedinělý.

Asi největším omezením knihovny je parser PDF dokumentů knihovny Zend PDF, který je udržovaný, ale zatím nepodporuje novější verze dokumentů.

V průběhu práce jsem zjistil, že v jazyce PHP zatím chybí kvalitní open souce udržované knihovny pro práci s PDF, které by byly srovnatelné s těmi v jazyce Java. Ten se ukazuje v uvedené oblasti jako mnohem rozvinutější a určitě nabízí možnost inspirace. Právě na jejím základě jsem přišel na několik nápadů na další rozvoj této knihovny. Nejvíce mne zaujala knihovna Apache PDF Box, jejíž přítomnost v PHP by pro mnohé programátory měla velký přínos. Proto si myslím, že budoucím rozšířením knihovny by mohlo být postupné nahrazování funkcionality Zend PDF, přepsanou funkcionalitou do PHP právě z této knihovny.

Dalším námětem na pokračování této práce může být například převod HTML do PDF. Během vyhledávání informací k problematice formulářů jsem se často setkával s požadavky právě na tuto funkci a s absencí kvalitního software, který by tento převod umožnil.

Knihovna jako taková jistě nabízí prostor pro další rozšiřování a vylepšování.

Literatura

[1] Adobe. Digital signatures in a pdf. http://www.adobe.com/devnet-docs/acrobatetk/tools/DigSig/Acrobat_DigitalSignatures_in_PDF.pdf, stav z 2. 5. 2014.

[2] Setasign. Setapdf-formfiller - fill in existing pdf forms with php. http://www.setasign.com/products/setapdf-formfiller/details/, stav z 2. 5. 2014.

[3] Setasign. Setapdf-signer - digital sign pdf documents with php. http://www.setasign.com/products/setapdf-signer/details/, stav z 2. 5. 2014.

[4] Setasign. Setapdf-signer - simple usage demo. http://www.setasign.com/products/setapdf-signer/demos/simple-usage-demo/, stav z 2. 5. 2014.

[5] Base 14 fonts. http://desktoppub.about.com/od/glossary/g/base14fonts.htm, stav z 28. 12. 2013.

[6] Adobe pdf 101 — quick overview of pdf file format. http://partners.adobe.com/public/developer/tips/topic_tip31.html, stav z 27. 12. 2013.

[7] Adobe pdf reference 1.7. http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf, stav z 27. 12. 2013.

[8] Extending php. http://docstore.mik.ua/orelly/webprog/php/ch14_01.htm, stav z 2. 5. 2014.

[9] The history of pdf | how the file format and acrobat evolved. http://www.prepressure.com/pdf/basics/history, stav z 27. 12. 2013.

[10] itext core. http://itextpdf.com/product/itext, stav z 7. 5. 2014.

[11] Jaký je rozdíl mezi písmy truetype, postscript a opentype? http://windows.microsoft.com/cs-cz/windows/difference-truetype-postscript-opentype-fonts#1T, stav z 28. 12. 2013.

[12] Apache pdfbox - a java pdf library.
`http://pdfbox.apache.org/`, stav z 29. 12. 2013.

[13] The application/pdf media type.
`http://tools.ietf.org/html/rfc3778`, stav z 27. 12. 2013.

[14] Pdftk server.
`http://www.pdflabs.com/tools/pdftk-server/`, stav z 29. 12. 2013.

[15] Ten reasons for using php-cpp.
`http://www.php-cpp.com/documentation/ten-reasons-for-using-php-cpp`, stav z 2. 5. 2014.

[16] Php internals structures.
`http://us1.php.net/manual/en/history.php.php`, stav z 2. 5. 2014.

[17] Php internals structures.
`http://www.php.net/manual/en/internals2.structure.php`, stav z 2. 5. 2014.

[18] Podofo.
`http://podofo.sourceforge.net/`, stav z 2. 5. 2014.

[19] Usage statistics and market share of php for websites.
`http://w3techs.com/technologies/details/pl-php/all/all`, stav z 28. 12. 2013.

[20] Zend pdf - zend framework reference.
`http://framework.zend.com/manual/1.12/en/zend.pdf.html`, stav z 2. 5. 2014.

Příloha A

Apache License, Version 2.0

Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

A.1 Definitions.

"License"shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor"shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity"shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control"means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You"(or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source"form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object"form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work"shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works"shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution"shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted"means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor"shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

A.2 Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

A.3 Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

A.4 Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this License; and You must cause any modified files to carry prominent notices stating that You changed the files; and You must retain, in the Source form of any Derivative Works that You distribute,

all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and If the Work includes a "NOTICE"text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

A.5 Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

A.6 Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

A.7 Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS"BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

A.8 Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

A.9 Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

A.10 APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]"replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page"as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
```

```
limitations under the License.
```

Příloha B

Návod k použití

Postup instalace:

- Nakopírujte src/JanKubat do vašeho adresáře knihoven
- Do vašeho projektu stáhněte knihovnu Zend PDF ve verzi 2.0.0 nebo novější
- Nastavte si autoloading tříd, příklad najdete v souboru require.php
- Knihovna vyžaduje pro správnou funkci podepisování PDF dokumentů Java 1.7
- Knihovnu začněte používat zavoláním `JanKubat\Pdf\PdfDocument::load($cestaKSouboru)`
- Digitální podpis vytvoříte pomocí třídy `JanKubat\Pdf\PortableSigner`

Příloha C

Seznam použitých zkratek

API Application Programming Interface

CGI Common Gateway Interface

DSA Digital Signature Algorithm

FDF Forms Data Format

IDE Integrated Development Environment

PDF Portable Document Format

PHP PHP: Hypertext Preprocessor

PHP/FI Personal Home Page, Form Interpreter

PS PostScript

RSA Rivest, Shamir, Adleman - šifrovací algoritmus

URL Uniform Resource Locator

XFDF XML Forms Data Format

Příloha D

Obsah přiloženého CD

```
.
 docs //zdrojový kód textu práce v latexu
  text
      cd_content.txt
      cd_content.txt.aux
      figures
       LogoCVUT.eps
       LogoCVUT.pdf
       seznamcd.eps
       seznamcd.pdf
      Hron-thesis-2009.aux
      Hron-thesis-2009.bbl
      Hron-thesis-2009.blg
      Hron-thesis-2009.dvi
      Hron-thesis-2009.lof
      Hron-thesis-2009.log
      Hron-thesis-2009.lot
      Hron-thesis-2009.out
      Hron-thesis-2009.pdf
      Hron-thesis-2009.ps
      Hron-thesis-2009.synctex.gz
      Hron-thesis-2009.tex
      Hron-thesis-2009.toc
      hyphen.tex
      img
       adobe_acrobat_form_creation_menu.png
       pdf_form_elements.png
      k336_thesis_macros.sty
      Makefile
      missfont.log
      reference.bib
      texput.log
```

```
examples //pokusy a experimenty s různými knihovnami
 apachepdfbox
  apachepdfbox.iml
  lib
   pdfbox-app-1.8.3.jar
  pom.xml
  src
   main
   test
  target
      classes
      generated-sources
 Fillable_PDF_Sample_from_TheWebJockeys_vC5.pdf
 fpdf
  fpdf
   doc
   FAQ.htm
   font
   fpdf.css
   fpdf.php
   changelog.htm
   install.txt
   license.txt
   makefont
   tutorial
  test.php
 seta
 zend_pdf
src //zdrojové kódy samotné knihovny
 bin
  PortableSigner
      lib
      linux
      linux-install.sh
      PortableSigner.jar
 cert
  357.crt
  357.key
  357.pem
  357.pfx
 composer.json
 composer.lock
 composer.phar
 experimets
  info.php
  signtest.php
```

```
  test.php
 JanKubat
  Pdf
      AcroForm
      PdfDocument.php
      PdfSigner.php
      PortableSigner.php
      Signature
 portablesigner.php
 require.php
 tests
 test7.pdf
 vendor
  autoload.php
  composer
   autoload_classmap.php
   autoload_namespaces.php
   autoload_psr4.php
   autoload_real.php
   ClassLoader.php
   installed.json
  contao
   tcpdf
  github
  zendframework
      zend-memory
      zendpdf
      zend-stdlib
 zend.php
text //text práce v PDF
 jankubat-bp-2014.pdf
tools //nástroje které jsem vyzkoušel ale nebyli přímo použitelné pro tuto práci.
    ACROSDK7.0
     docs.html
     Documentation
      Custom_Configurations
      index
      index.pdx
      InterApplication_Communication
      Introduction_To_SDK
      JavaScript
      PDF_Creation_APIs_And_Specs
      Plugins
     index.html
     JavaScriptSupport
      Samples
```

```
 LicenseAgreements
  License_Enu.pdf
 PluginSupport
  Headers
  Samples
 ReleaseNotes.pdf
 Tools
     Reader-enabling Tools
Extension
 Makefile
 Makefile~
PDF
 main.cpp
 main_1.cpp
 Makefile
 nbproject
  configurations.xml
  Makefile-Debug.mk
  Makefile-impl.mk
  Makefile-Release.mk
  Makefile-variables.mk
  Package-Debug.bash
  Package-Release.bash
  private
  project.xml
 PDF.cpp
 pdf_extension.ini
 PdfForm.cpp
 PdfForm.h
 PDF.h
 Pokus.cpp
 Pokus.h
 yourextension.so
podofo-build
 CMakeCache.txt
 CMakeFiles
  CMakeError.log
  cmake.check_cache
  CMakeOutput.log
  CMakeTmp
  CheckTypeSize
  TestEndianess.bin
  2.8.12.2
 cmake_uninstall.cmake
 debian
  CMakeFiles
```

```
 man
 examples
  CMakeFiles
  helloworld
  helloworld-base14
 PoDoFoConfig.cmake
 podofo_config.h
 src
  CMakeFiles
 test
  CMakeFiles
  ContentParser
  CreationTest
  DeviceTest
  FilterTest
  FormTest
  LargeTest
  ObjectParserTest
  ParserTest
  SignatureTest
  TokenizerTest
  unit
  VariantTest
  WatermarkTest
 tools
     CMakeFiles
     podofobox
     podofocolor
     podofocountpages
     podofocrop
     podofoencrypt
     podofogc
     podofoimgextract
     podofoimg2pdf
     podofoimpose
     podofoincrementalupdates
     podofomerge
     podofopages
     podofopdfinfo
     podofotxtextract
     podofotxt2pdf
     podofouncompress
     podofoxmp
podofo-0.9.2
    AUTHORS
    cmake
```

```
 modules
CMakeLists.txt
cmake_uninstall.cmake.in
CODINGSTYLE.txt
CONTRIBUTIONS.txt
COPYING
COPYING.LIB
debian
 CMakeLists.txt
 man
doc
 html
 latex
 podofo_architecture.png
Doxyfile
examples
 CMakeLists.txt
 helloworld
 helloworld-base14
 pdfcontentsgraph
FAQ.html
ChangeLog
INSTALL
NEWS
podofo
 base
 compilercompat
 podofo.h
podofo_config.h.in
README.html
src
 base
 CMakeLists.txt
 doc
 podofo-base.h
 podofo.h
test
 CMakeLists.txt
 ContentParser
 CreationTest
 DeviceTest
 FilterTest
 FormTest
 LargeTest
 ObjectParserTest
 ParserTest
```

```
 pdfs
 PdfTest.h
 SignatureTest
 SignTest
 system
 TokenizerTest
 unit
 valgrind.suppressions
 VariantTest
 WatermarkTest
TODO
tools
 CMakeLists.txt
 podofobox
 podofocolor
 podofocountpages
 podofocrop
 podofoencrypt
 podofogc
 podofoimgextract
 podofoimg2pdf
 podofoimpose
 podofoincrementalupdates
 podofomerge
 podofopages
 podofopdfinfo
 podofotxtextract
 podofotxt2pdf
 podofouncompress
 podofoxmp
vcincludes
    unistd.h
```