VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

TVORBA TISKOVÝCH SESTAV NA PLATFORMĚ JAVA

PRINT REPORT CREATION ON THE JAVA PLATFORM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — ZUZANA SYNAKOVÁ
AUTHOR

VEDOUCÍ PRÁCE — Ing. RADEK BURGET RADEK, Ph.D.
SUPERVISOR

BRNO 2019

Ústav informačních systémů (UIFS) Akademický rok 2018/2019

Zadání bakalářské práce

22026

Studentka: **Synaková Zuzana**
Program: Informační technologie
Název: **Tvorba tiskových sestav na platformě Java**
Print Report Creation on the Java Platform
Kategorie: Informační systémy
Zadání:

1. Seznamte se s platformou Java EE pro tvorbu informačních systémů. Zaměřte se na nástroje a knihovny pro vytváření tiskových sestav ve formátu PDF.
2. Prostudujte existující nástroje pro zpracování HTML a CSS předloh na platformě Java.
3. Navrhněte architekturu řešení pro vytváření tiskových PDF sestav na základě šablony ve formátu HTML a CSS s možností snadné integrace do informačních systémů na platformě Java EE.
4. Implementujte navržené řešení a jednoduchou demonstrační aplikaci umožňující vytvoření tiskové sestavy z existující databáze.
5. Proveďte testování vytvořeného řešení.
6. Zhodnoťte dosažené výsledky.

Literatura:

- Červinka, Z.: Generování PDF dokumentů z webových stránek, bakalářská práce, Brno, FIT VUT v Brně, 2015
- Juneau, J.: Java EE 7 Recipes, Apress, 2013
- Bien, A.: Real World Java EE Patterns: Rethinking Best Practices, lulu.com, 2009

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Burget Radek, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 30. října 2018

Abstrakt

Hlavným cieľom tejto bakalárskej práce je tvorba tlačových zostáv na platforme Java s využitím HTML šablón.Táto práca sa obsahuje opis implementácie, spracovania HMTL stránok a generovania PDF súborov. Výsledkom je demonštračná webová aplikácia, ktorej vstupom je HTML stránka a výstupom vygenerovaná tlačová zostava.

Abstract

The main goal of this bachelor's thesis is to create print reports on Java platform with HTML templates. The product is a demonstration web application whom input is the HTML page and the output is generated print report.

Kľúčové slová

Tvorba tlačových zostáv, Java EE, PDF, HTML, CSS, XTHML

Keywords

Print Report Creation, Java EE, PDF, HTML, CSS, XTHML

Citácia

SYNAKOVÁ, Zuzana. *Tvorba tiskových sestav na platformě Java.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Radek Burget Radek, Ph.D.

Tvorba tiskových sestav na platformě Java

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracovala samostne pod vedením pána Ing. Radka Burgeta, Ph.D. Uviedla som všetky literárne pramene a publikácie, z ktorých som čerpala.

.....
Zuzana Synaková
16. mája 2019

Obsah

Kapitola 1

Úvod

V súčasnej dobe existuje veľa nástrojov, v ktorých môžeme pracovať s informáciami či štatistikami. Ak máme však napríklad iba súbor čísel bez komentáru, pre nového čitateľa je nepoužiteľný. Každý z nás sa už asi stretol iba s výpisom niečoho, čo bez uvedenia komentáru a popisu nemalo zmysel. Tlačové zostavy je pojem, s ktorým sa človek každodenne nestretáva. Tlačová zostava sa dá vysvetliť ako šablóna pre údaje, ktorá im pridáva čitateľnosť a význam. Ako príklad nutnosti využitia tlačových zostáv sa dá použiť neprehľadná a zahustená tabuľka(excel), pri ktorej by nezainteresovaná osoba iba ťažko pátrala po jej význame.

Cieľom tejto práce je tvorba tlačovej zostavy z HTML formátu do PDF výstupu a teda uľahčenie tohto generovania programátorovi, ktorý sa rozhodne túto prácu využiť. V dnešnej dobe existujú už rôzne knižnice, ktoré podporujú generovanie PDF súborov v čistej jave a mojou motiváciou ich bolo optimalizovať, aby na ich využitie nemusel budúci užívateľ študovať zdĺhavú dokumentáciu.

HTML ako vstupný formát má svoje výhody, ale aj nevýhody. HTML je jazyk, ktorý je celosvetovo používaný a pre tvorbu webových aplikácií nevyhnutný. Nakoľko programátor musí sám vytvoriť šablónu, je ideálne použitie práve jazyka, s ktorým je už oboznámený. Nevýhodou použitie HTML stránky je ich analýza a vygenerovanie PDF. Väčšinu elementov je ľahké spracovať, ale aj napriek tomu nebu výsledné PDF na sto percent totožné. Toto poznanie však môžeme do istej mieri zanedbať, nakoľko tlačové zostavy napríklad vo firmách majú zvyčajne jednoduchší formát.

PDF formát ako výstup je veľmi rozšírený a populárny. Jeho výhodou je napríklad možnosť ľahkého prenosu medzi platformami, ale taktiež aj zabezpečenie. PDF súbor nie je tak ľahko manipulovateľný ako textové, čo môže byť pri jeho šírení výhodou. Popularizáciou tohto súborového formátu sa stal zaujímavým aj pre vývojové prostredie webových aplikácií už pred mnohými rokmi. V súčasnosti na internete vieme nájsť mnoho knižníc, ktoré sa zaoberajú generovaním PDF súborov, ich transformáciou na iné súbory, čítanie a iné.

Generovanie tlačovej zostavy priamo vo webovej aplikácii na servery so sebou nesie mnohé výhody. Automatizácia a šetrenie času je jedným s hlavných motívov vývoja ďalšej knižnice, ktorá prepája generovanie a spracovávanie HTML stránky. Generovanie PDF súborov nie je vo web aplikáciách žiadnou novinkou a preto som sa snažila zamerať svoju prácu na čo najefektívnejšie uchopenie tejto témy.

Táto práca je dokumentáciou a teoretickým podkladom do problematiky tvorby tlačových zostáv na platforme Java. Nachádzajú sa tu logicky členené kapitoly, odkazy a prílohy pre lepšie uchopenie tejto témy a postupu riešenia použitého pri implementácii.

V teoretickej kapitole 2 sú uvedené vysvetlenia základných pojmov pre webové aplikácie, komunikáciu po internete, krátky prehľad nástrojov použitých pri vývoji, ale aj opis knižníc, ktoré generujú PDF súbory a spracovávajú HTML súbory, súčasný prehľad a aj historický vývoj generovania a reportovania.

V ďalšej kapitole 3 je opísaný všeobecný návrh riešenia a spôsob komunikácie medzi časťami návrhu, očakávane vstupy a výstupy, špecifikácia požiadavkov na generovanie PDF súborov a aj štruktúra vstupy a výstupu. Tento návrh by mal byť použiteľný aj pre iné platformy ako je Java.

Nasledujúca kapitola 4 opisuje konkrétnu implementáciu tejto práce, jej rozsah, použité triedy a štruktúry, prepojenie logických celkov návrhu, prípadné chyby nedostatky a vylepšenia, či ďalšie smerovanie vývoju.

Po implementovaní nasleduje kapitola 5 o testovaní a dosiahnutých výsledkoch. Táto kapitola opisuje do podrobnosti testy, ich vstupy, výstupy, zhodnotenie výsledku a návrhy na rozšírenie knižnice a iné.

V záverečnej kapitole 6 je zhodnotenie práce, rekapitulácia výsledkov a dôležitých častí.

Kapitola 2

Teoretická časť

Táto kapitola sa zaoberá teóriou a informáciami potrebnými pre lepšie porozumenie problematiky vytvárania tlačových zostáv na platforme java. Nachádzajú sa tu vysvetlenia základných pojmov, použitých knižníc, či opis problému pri vytváraní tlačových zostáv.Súčasné vyvinuté knižnice a ich predchodcovia sú taktiež uvedené v tejto kapitole.

2.1 Základné pojmy

V tejto časti sa nachádzajú vysvetlenia základných pojmov potrebných pri neskoršom opise návrhu a implementácie.

2.1.1 SGML

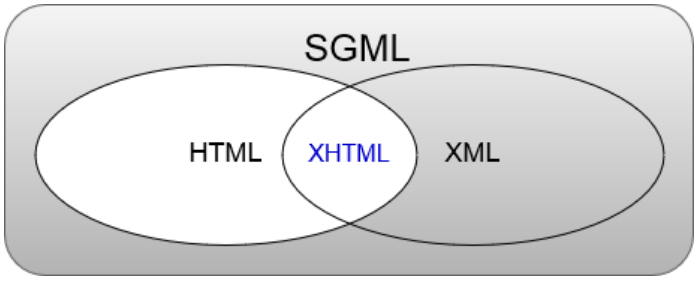
SGML(Standard Generalized Markup Language) je štandardizovaný jazyk pod správou ISO (Interternational Organization for Standardization). Na základe tohto štandardu bol napísaný aj hlavný jazyk, ktorý sem spadá a to je XML. Jedným s ďalších príkladov môže byť aj HTML a z nich skombinované XHTML. V rámci objektového programovania, by sa dalo SGML porovnať s nadradenou triedou a jazyky HTML a XML odvodené z nej.

2.1.2 XML

XML (eXtensible Markup Language) je značkovací jazyk, ktorý má pomerne prísne syntaxné pravidlá. Vývojom týchto značkovacích jazykov a stanovením presných pravidiel programátori získali čitateľný súborový formát, ktorý je navyše kompatibilný naprieč rôznymi rozhraniami a aplikáciami. Vďaka prehľadnému značkovaniu je preto jednoduchšie súbor aj čítať, ale to podstatnejšie, aj analyzovať spracovávať.

2.1.3 HTML a XHTML

HTML(Hypertext Markup Language) je značkovací jazyk, ktorý slúži na značkovanie bežného textu do webových stránok pomocou sémantických pravidiel a štruktúry. XHTML (Extensible Hypertext Markup Language) je úpravou tohto jazyku. Na rozdiel od klasického HTML má striktnejšie pravidlá značkovania prvkov. Tak ako aj v XML, z ktorého tento jazyk prebral syntax, každý prvok musí mať aj uzatváraciu značku a všetky atribúty elementov musia byť uvedené malými písmenami[9]. XHTML, vďaka týmto vlastnostiam, je teda viac kompatibilný naprieč rôznorodými editormi. Grafické znázornenie vzťahov medzi týmito značkovacími jazykmi je uvedené na obrázku 2.1



Obr. 2.1: Grafické znázornenie vzťahov

2.1.4 CSS

CSS (Cascading Style Sheets) dodáva vzhľad HTML stránke. HTML je často označované za kostru, alebo základ a CSS za vzhľad webovej stránky. Napriek tomu, že kaskádové štýly môžu byť umiestnené aj priamo v HTML, vo väčších projektoch je to iba veľmi zriedkavé a neprehľadné.

2.1.5 HTTP

HTTP (Hypertext Transfer Protocol) je protokol, ktorý zabezpečuje štandardizovanú komunikáciu cez WWW (World Wide Web)[10] Klient zašle pomocou prehliadača požiadavok (HTTP request),zvyčajne buď kliknutím na hypertextový odkaz, alebo zaslaním formuláru a server požiadavok spracuje a následne zašle odpoveď. Názorná ukážka je na obrázku 2.2.

2.1.6 HTTP request

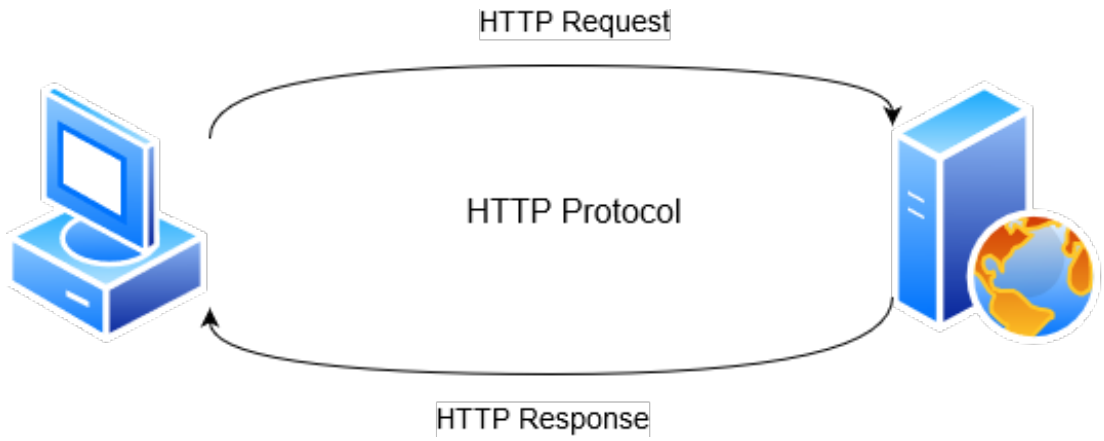
HTTP request, alebo Slovenky HTTP požiadavok, je správa ktorú posiela klient serveru a jej štruktúra má svoj štandard. Obsahuje *Request-line, Request-URI a HTTP verziu.*
Príklady HTTP request metód:

- GET
- POST
- PUT
- DELETE
- CONNECT

V Java aplikáciách poskytuje rozhranie k požiadavku ServletRequest. HttpServletRequest je potom trieda vytvorená servlet kontajnerom, ktorý tento vytvorený objekt predá servlet funkciám ako argument.

2.1.7 HTTP response

HTTP response , alebo Slovensky HTTP odpoveď, je správa, ktorú vytvára a následne odosiela server späť klientovi. Táto odpoveď obsahuje *status line, HTTP headers a message body.* V prípade Java webových aplikácií je odpoveď generovaná servletom, najčastejšie funkciami doGet, alebo *doPost.*



Obr. 2.2: HTTP protokol

2.1.8 URI

URI (Uniform Resource Identifier) je reťazec, ktorý v sebe nesie umiestnenie a meno súboru, pričom tak nemusí byť zároveň, a tým jednoznačne identifikuje prostriedky[1]. Často sa mýli a zamieňa s URL (Uniform Resource Locator). URL odkazuje, na konkrétnu web stránku, alebo súbor a teda na jeho lokáciu. Relatívna URI však môže obsahovať napríklad iba cestu k súboru ale nie jeho názov.

2.1.9 Tlačová zostava

Tlačová zostava (česky tisková sestava), je dôležitou súčasťou spracovania dát. Tlačová zostava slúži ako šablóna a formátuje vstupné dáta, ktoré sú do nej vložené. Správnou formou tlačovej zostavy môže byť uľahčená interpretácia informácií, ktoré by inak neboli ľahko pozorovateľné. Tlačové zostavy sa využívajú napríklad v informačných systémoch firiem na faktúry, výplatné pásky a iné štatistické údaje o napríklad produktoch.

2.1.10 PDF

PDF (Portable Document Format), ako aj samotný názov nesie, je dokumentový (súborový) formát, ktorý bol vynájdený pre lepšiu kompatibilitu medzi rôznymi aplikačnými rozhraniami a na spracovanie z nedigitálnej do digitálnej formy. PDF formát sa od svojho vzniku stal už overenou súčasťou komunikácie po internete a je v súčasnosti štandardizovaný organizáciou ISO (Interternational Organization for Standardization) [6].

2.1.11 Webová aplikácia

Webová aplikácia využíva model klient server a spracováva požiadavky užívateľa. Klient požiada cez svoj prehliadač a následne tento požiadavok je odoslaný buď lokálne na sieti, alebo po internete až ku serveru na ktorom je spustená aplikácia.

Model klient - server priniesol so sebou množstvo výhod hlavne pre podnikové prostredie. Znížila sa tak potreba aktualizácie desktopových aplikácii, ktoré boli distribuované lokálne na mnohých počítačoch. Webové aplikácie taktiež zabraňujú aj neoprávnenému používaniu kódu a jeho kopírovania, ktoré porušuje licenčné podmienky a to vďaka tomu, že koncový užívateľ nemá prístup k aplikačnej logike[4].

Webové aplikácie v dnešnej dobe implementujú online nástroje ako emailový klient, aplikácie na zdieľanie dát v reálno čase (online úprava dokumentov) a podnikové informačné systémy.

2.2 Pojmy z prostredia Java

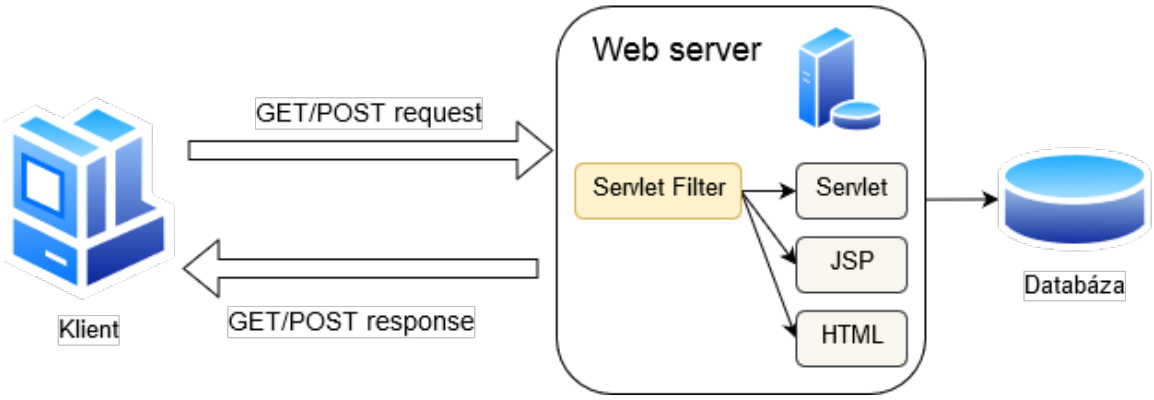
2.2.1 JSP

JSP je skratka JavaServer Page. JSP doplňujú servlety a zároveň poskytujú nové možnosti v kombinácii s nimi. JSP stránky sa na rozdiel od servletov nekompilujú. Pri prvom zavolaní JSP kontajner preloží stránku na triedu servlet. Po zmene v JSP súbore je nutné ho znova skompilovať.[3] Na zmenšenie odozvy odpovede je možná aj pred-kompilácia. JSP trieda pomáha jednoduchšie vytvárať dynamické webové stránky, pričom jej statické prvky sú reprezentované HTML.

2.2.2 Servlet

Java Servlet API definuje HTTP triedy. Vďaka servletom boli výrazne zlepšené možnosti vývoju webových aplikácií, ktoré sú založené na dotazoch a odpovediach zo strany klienta a serveru[5].

Servlet je trieda Javy, ktorá beží v servlet kontajnery. Servlet kontajner je špeciálny druh web servera [2], a bez neho servlet samotný nemôže fungovať. Spracováva http dotazy z prehliadača užívateľa a odosiela ich konkrétnym servletom. Tie môžu byť načítané pri prvom požiadavku načítané, alebo až po konkrétnom dotaze na servlet. To, ktorú z týchto dvoch možností kontajner volí, nie je na programátorovi [7]. V našom prípade testovania bol použitý Tomcat, ktorý využíva načítanie servletov čím skôr o pustení.



Obr. 2.3: Webová aplikácia

2.2.3 JAR

JAR (Java ARchive) je súborový formát, ktorý slúži na prenos a distribúciu Java knižníc. Používajú sa na zoskupenie určitých skupín tried a metadát potrebných k ich správnemu fungovaniu. Ak sa v JAR súbore nachádza aj *main* funkcia, je tento balíček spustiteľným. V tejto práci boli využité knižnice v tejto podobe napríklad ako 2.4.4.

2.2.4 WAR

WAR(Web Application Resource) je súborový formát, ktorý v sebe nesie všetky potrebné knižnice(JAR), triedy a statické HTML, alebo JSP stránky pre webovú aplikáciu. Tento druh súboru má svoje výhody, ale aj nevýhody. Ak požadovaná web aplikácia nepodlieha častým zmenám je ideálny na prenos a spustenie napríklad na Tomcat servery. Časté zmeny v zdrojových kódoch však môžu takéto distribuovanie, alebo prenos znepríjemniť neprehľadne veľkým množstvom verzií súboru.

2.2.5 JDBC API

JDBC (Java Database Connectivity) API (application programming interface) je rozhranie cez ktoré môžu servlety komunikovať s databázami. Rozdeľuje sa do dvoch častí. Prvá je na aplikačnej úrovni kedy aplikácia pristupuje k databáze a druhá je pripojenie databázy k Java EE platforme[5].

2.3 Apache Software

ASF(Apache Software foundation) je nezisková organizácia, ktorá podporuje open source a voľno dostupné softvéry. Stojí za vývojom a sponzorovaním mnohých programov od napríklad širšie známeho Apache http serveru až po softvéry, ktoré boli využité aj pri tejto práci.

2.3.1 Apache Tomcat

Tomcat je open source web server a implementácia Java EE tried ako Java Servletov, JSP stránok. Servlet kontajner je implementovaný a nazvaný ako Catalina. Je to jeden z najrozšírenejších webových serverov a poskytuje čisto Java prostredie, v ktorom môže kód bežať. Tomcat bol spolu s vývojovým prostredím Eclipse využitý pri testovaní tejto webovej aplikácii.

2.3.2 Apache Maven

Maven je software, ktorý umožňuje ľahšie a prehľadnejšie ovládanie a tvorbu projektov. Slovo maven znamená akumulátor vedomostí[8]. Jedným z dôležitých súborov, na ktorom je tento software postavený je POM (Project Object Model). V tomto súbore sa nachádzajú závislosti(knižnice) potrebné pre zhotovenie príslušného projektu. Maven následne pri vytváraní projektu stiahne uvedené knižnice (ak chýbajú) a tým vytvára jednotné a jednoduché prostredie.

2.3.3 Apache PDFBox

PDFBox je knižnica, ktorá je podporovaná organizáciou Apache. PDFBox je open source nástroj a umožňuje rôzne spracovanie a vytváranie PDF súborov. Táto knižnica umožňuje taktiež napríklad zmenu PDF súboru na jpg. Táto knižnica však sama o sebe nepodporuje ľahké spracovanie HTML súboru do PDF. Obsahuje základných 14 fondových rodín a ďalšie môžu byť aj manuálne pridané. Je možné rozdeľovanie, načítanie a spracovanie existujúceho PDF a mnoho funkcií. Podporuje aj PDF/A formát. Vo všeobecnosti je problémové vytváranie PDF súborov so špeciálnymi znakmi. PDFBox po vytvorení vlastného fondu túto funkciu splňuje.

2.4 Súčasné možnosti generovania PDF

Na generovanie PDF súborov existuje v súčasnosti viacero nástrojov, niektoré z nich čisto komerčné prípadne pod rôznymi licenciami GPL (General Public License).

2.4.1 OPEN HTML TO PDF

OPEN HTML TO PDF je knižnica napísaná v čistej Jave. Táto knižnica podporuje syntaktickú analýzu a generovanie PDF zo správne formátovaných XHTML stránok. Novšie HTML5 však nepodporuje. Tento projekt bol pôvodne inšpirovaný knižnicou *Flying Saucer* a spadá pod LGPL licenciu.

2.4.2 Flying Saucer

Flying Saucer je taktiež knižnica napísaná v čistej Jave a s použitím iText knižnice. Táto knižnica je stále aktualizovaná a spadá pod licenciu LGPL. Jedným z nedostatkov je napríklad využívanie knižnice iText 5, ktorá už v dnešnej dobe nie je podporovaná.

2.4.3 iText

IText 7 je knižnica, ktorá je stále pod vývojom a je dostupná pod licenciou AGPL, ale taktiež poskytujú aj komerčnú licenciu. Nakoľko táto knižnica existuje už vyše 15rokov prešla si aj viacerými licenciami a v tejto práci bol využitý itext.v. 5.5.10. a itext.v.7.1.6. Na internete existujú aj mnohé vlastné implementácie iTextu pod názvami ako napríklad OpenPDF[1], Lowagie[2]. Názov Lowagie nesie po hlavnom vývojárovi iTextu v jeho začiatkoch. Tieto knižnice už nieje doporučené používať z bezpečnostných dôvodov v skorších verziách iTextu, ale taktiež aj kvôli znehodnoteným a zastaralým triedam.

IText bol v roku 2016 naprogramovaný úplne od znova a z verzie 5 sa už v súčastnosti stal iba predchodca bez podpory. Nová verzia 7 je vytvorená na Java 7, ktorá bola kompromisom, aby bola knižnica v prípade potreby stále kompatibilná aj so staršími, ale aj novšími verziami Javy. IText sa skladá s viacerých nástrojov.

IText 7 *pdfHTML* je nástroj na vytváranie pdf súborov z HTML stránok. *PdfHTML* je nástupcom *XML Worker*, ktorý bol vo verzii 5. Tento nástroj podporuje mnoho HTML prvkov a ich vsadenie do PDF. V kapitole 5 sú porovnané výsledky výstupov.

Open source nástroje sa oproti verzii 5 tiež rozrástli. Medzi platené nástroje patrí pdf2Data, ktorý slúži na spracovanie dát z PDF súborov. Ďalšími nástrojmi sú iText DITO, pdfXFA a pdfCalligraph, ktorý podporuje zložitejšiu typografiu, PdfSweep, pdfDebug je vhodný na korekciu a nachádzanie chýb pri práci s iTextom. Prehľad tried, vysvetlenie a dôležité vlastnosti:

- com.itextpdf.text.Document – *trieda Document* je základom pre vkladanie rozparsovaných HTML štruktúr. Za predpokladu, že chceme do PDF uložiť aj metadáta (napr kto ho vytvoril, v akom jazyku je PDF napísané) musia byť tieto informácie nastavené pred otvorením dokumentu.
  Použité funkcie na pridanie metadát:
  - document.addAuthor()

[1] https://github.com/LibrePDF/OpenPDF
[2] https://www.lowagie.com/iText

  - document.addCreator()
  - document.addLanguage()
  - document.addKeywords()

- com.itextpdf.text.PageSize - použitý na formát strany dokumentu
- com.itextpdf.text.DocumentException - zachytávanie chýb pri práci s dokumentom
- com.itextpdf.text.pdf.PdfWriter - zapisovač, potrebný pre spracovanie HTML vstupu. Každý element pridaný do *PdfWriter*, ktorý je spárovaný s PDF dokumentom bude pridaný aj to výstupného streamu.
- com.itextpdf.tool.xml.XMLWorkerHelper - nástroj použitý na spracovanie HTML vstupu, ktorý pomocou dokumentu a zapisovača spracoval HTML stránku do PDF podoby. Použité funkcie:
  - XMLWorkerHelper.getInstance().parseXHtml()

2.4.4 jPDFWriter

Je voľne dostupná knižnica pre tvorbu PDF. Spadá pod svoju vlastnú licenciu a táto knižnica sa dá stiahnuť na stránke vývojára. Knižnica je dostupná voľne aj na komerčné účely, avšak na ne by som ju asi podľa môjho názoru neodporúčala.
Táto knižnica je veľmi jednoduchá na použitie, avšak to aj obmedzuje možnosť spracovania dát, ktorú poskytuje. V kapitole 5 budete môcť porovnať výsledky aj tejto knižnice. Prehľad tried,vysvetlenie a dôležité vlastnosti:

- com.qoppa.pdfWriter.PDFDocument - ktorá slúži ako základný dokument pre vytváranie PDF
  Použité funkcie:
  - PDFDocument.loadHTML() - funkcia na vstupných parametroch vyžaduje URL na vstupné HTML, ktoré chceme generovať, veľkosť strany PDF a parameter na zmenšenie obsahu vstupu do výstupného PDF.
  - PDFDocument.saveDocument() - funkcia na uloženie dokumentu

Zaujímavosťou tejto knižnice je, že na prácu s ňou a vytvorenie PDF súboru vám stačí naozaj zopár riadkov. Túto knižnicu, by som doporučila na prvé pokusy generovania tlačovej zostavy, pretože okrem týchto funkcií nie je možné implementáciu generovania vylepšiť, iba priamo zásahom do načítacej funkcie HTML vstupu.

2.4.5 JasperReports

JasperReports je open source knižnica pod liceniou LGPL a je to jedným z najpopulárnejších Java nástrojov používaných pri reportovaní. Pre túto prácu však daný nástroj nebol použitý, nakoľko bolo za potreby skôr knižnicu ako nástroj s UI (User Interface). Reporty sú uložené vo vlastnom súborovom formáte JRXML, ktorý spĺňa pravidlá DTD SGML.

2.4.6 Problematika vytvárania PDF súboru

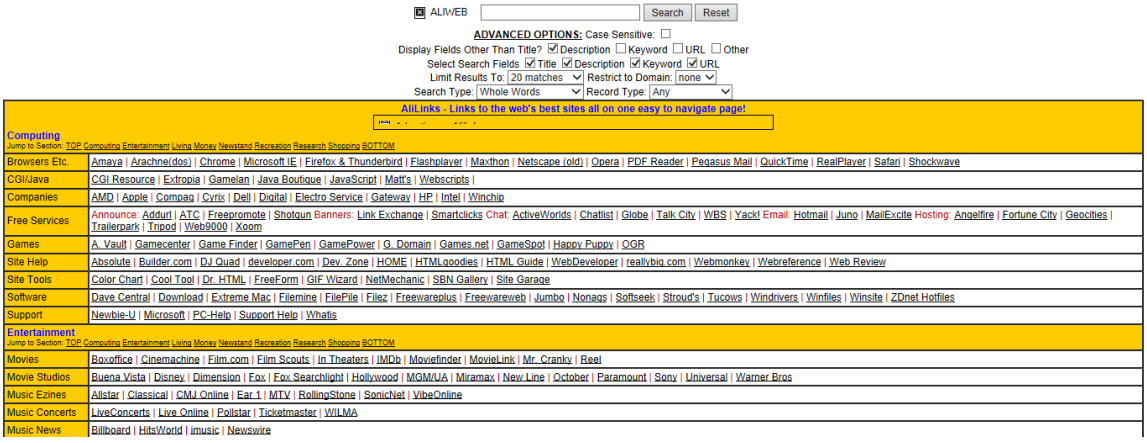
Vstupným formátom zadania je HTML súbor, ktorý slúži ako šablóna pre výslednú tlačovú zostavu. Tu však narážame na problém prenosu vizuálnej podoby HTML do súboru PDF. Po preskúmaní rôznych knižníc a implementácii testovania s nimi, som dospela k názoru, že výsledné PDF sa nebude na HTML stránku nikdy na sto percent podobať. Tu však nachádzame široké pole optimalizácií a vylepšení, s ktorými sa môžeme pokúsiť o čo najvierohodnejší výsledok.

2.4.7 Problematika dostatočnej podobnosti

HTML jazyk a jeho syntax je veľmi široký a väčšina knižníc na jeho spracovanie a následné generovanie PDF súboru nepodporuje všetky prvky. Snahou tejto práce bolo získať dostatočný prehľad v knižniciach na toto spracovanie a nakoľko pri každej tlačovej zostave, môže niektorá knižnica optimálnejšie vyhodnotiť dané HTML prvky bolo implementované generovanie pomocou viacerých knižníc.

2.4.8 Pokrok v analýze HTML súborov

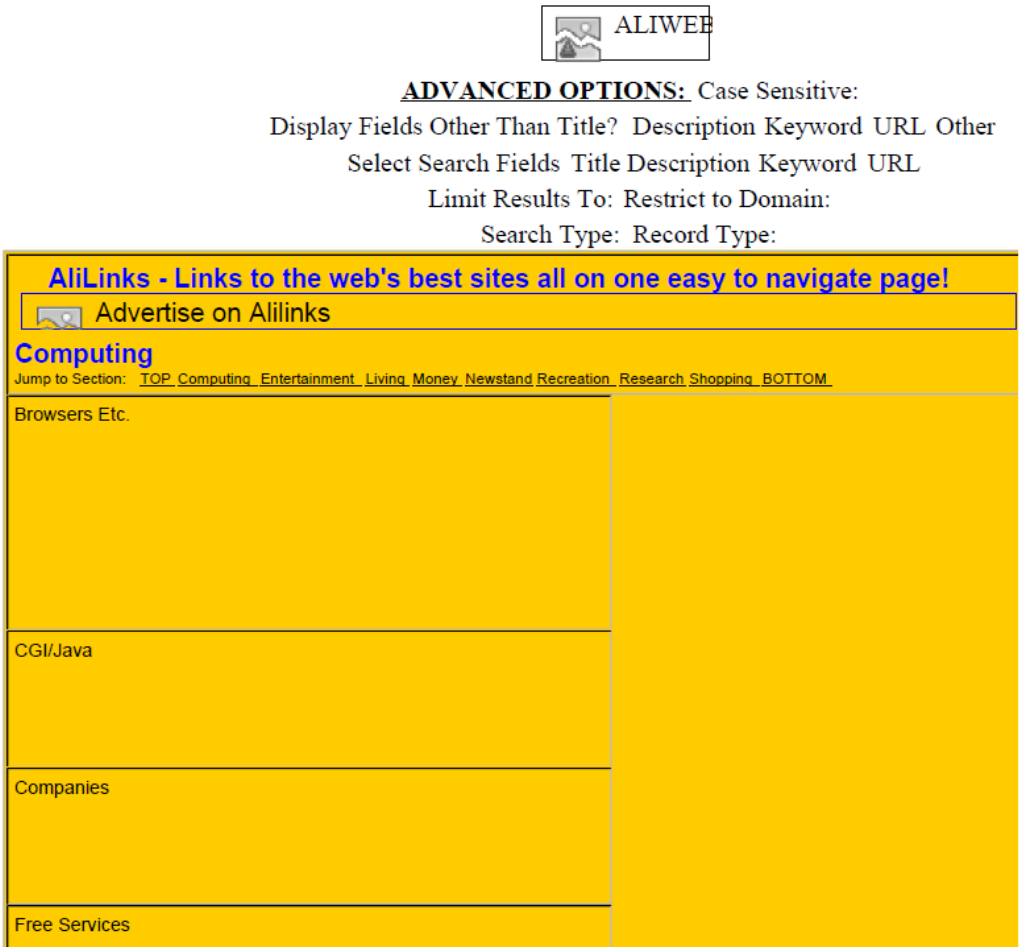
Za posledných 10 rokov analýza a spracovanie HTML súborov postúpili výrazne dopredu. Ak by sme porovnali výstup spracovania verzie iTextu päť a sedem, ktorý je opísaný bližšie v kapitole 5, zmeny sú badateľné už na prvý pohľad. Za výrazným pokrokom v tejto oblasti môžeme vďačiť súčasným populárnym open source licenciám, ktoré majú množstvo zanietených programátorov, ktorý prispievajú do vývoja. Otvorenie knižníc pod týmito licenciami znamenalo aj inšpiráciu pre vývoj ďalších funkcionalít, ľahšie odstraňovanie a hľadanie chýb pri tvorbe PDF súborov a analýzy HTML stránok (nakoľko má používateľ prístup k zdrojovým kódom a nie sú pre neho knižnice iba čiernou skrinkou).Porovnanie generovania klasickej webovej stránky a nie tlačovej sústavy je na obrázkoch 2.4 a 2.5.



Obr. 2.4: Jeden z prvých webových vyhľadávačov

2.4.9 Knižnice pre spracovanie HTML vstupov

Na platforme Java EE existujú mnohé prepracované knižnice na spracovanie HTML vstupu na PDF, avšak tu sa často stretávame s komerčnou licenciou. To pre malé, rozbiehajúce sa projekty, môže byť veľká finančná záťaž. Preto som sa rozhodla vyhľadať , čo najlepšie



Obr. 2.5: pdf vygenerovaní pdfWriterom od quoppa

open source nástroje na ich spracovanie. Licencie v tejto práci spadajú pod voľne dostupné kódy, alebo AGPL. Nakoľko bola táto práca robená s motívom verejného publikovania, tiež spadá pod túto licenciu a jej kód je voľne prístupný a verejný.

Kapitola 3

Návrh

Táto kapitola obsahuje všeobecný návrh tvorby tlačových zostáv a požiadavky na vstupné a výstupné dáta, či ich formát. Tento návrh a požiadavky boli vyhodnotené na základe môjho prehľadu v danej téme a pohľadu na nedostatky.

3.1 Zhrnutie súčasného stavu

Súčasný stav generovania PDF súborov, ako bolo spomenuté v podkapitole 2.4, nesmierne za posledné roky pokročil. Knižnice ako iText poskytujú veľké množstvo nástrojov na generovanie PDF súborov, ktoré až prekračujú požiadavky pre tvorbu tlačových zostáv. Táto knižnica však sama o sebe je zložitá a preto tento návrh obsahuje jej zjednodušenie a vybranie najdôležitejších prvkov pre generovanie tlačových zostáv. Itext poskytuje nástroje aj na syntaktickú analýzu a spracovanie HTML súborov.

Nakoľko open source licencovanie verzií iTextu je rovnaké(AGPL alebo LGPL) pre hlavnú časť práce bol využitý iText 7. Na porovnanie výstupných súborov bolo implementované aj riešenie (bez plnej funkcionality) s verziou 5 a inými open source knižnicami.

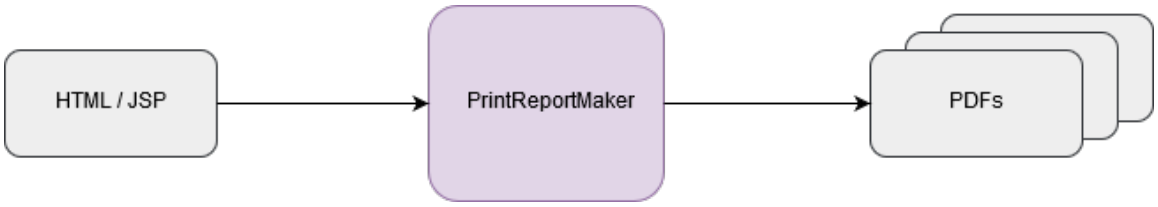
Ak by ste chceli využiť výsledky tejto práce v prostrední inom ako pod licenciou LGPL, je potrebné si túto knižnicu zakúpiť. Pri získavaní informácií o téme tvorby tlačových zostáv som sa na začiatku snažila nachádzať open source knižnice, ktoré nepodliehali potrebe publikačne kódy zverejniť na internete. Tieto knižnice sa však neosvedčili. Často bolo hlavným problémom nedostatočný počet implementovaných elementov HTML jazyka a s ním spojených kaskádových štýlov a špeciálnych znakov. V súčasnej dobe má knižnica iText stále tiež malé nedostatky generovania špeciálnych znakov do PDF, avšak tie sa zväčša týkali čínskeho jazyka a nie slovenského a českého ako v milosti.

Na záver zhrnutia vyhodnocujem, že generovanie PDF súborov je výrazne zjednodušené a prepracovanejšie ako v milosti. Vývoj v tejto oblasti veľmi napomohol aj vývoju reportovacích nástrojov, kde je často očakávaný výstup do formátu ako PDF a teda ľahko čitateľného naprieč rôznymi zariadeniami a operačnými systémami.

3.2 Zjednodušená schéma návrhu

Vstupným formátom sú HTML stránky, ktoré sú spracované PrintReportMaker-om a následne výstupným formátom je PDF. Tento návrh je však príliš zjednodušený (viď 3.1). Analýza a pretransformovanie HTML elementov, je zložitý proces, na ktorý v dnešnej dobe už existujú mnohé knižnice. Jedným z hlavných problémov vytváranie PDF stránok je

rozmanitosť atribútov a elementov jazyku HTML. Väčšina knižníc na spracovanie HTML stránok sa snaží pridávať, čo najväčšie množstvo rozpoznaných elementov, avšak stále je potrebné rátať s odlišnosťami od originálnej šablóny. Po analýze HTML elementov sú vhodným spôsobom vložené do PDF súboru a uložené.



Obr. 3.1: Návrh implementácie

3.3 Požiadavky na návrh

Požiadavky na návrh som sa snažila klásť najme na prostredie, v ktorom má byť táto práca využívaná. Reportovanie a tlačové šablóny sú najčastejšie využívané vo firemnom prostredí. Tlačové zostavy slúžia zvyčajne na interpretáciu dát, poprípade zrýchlenie práce zamestnanca pri vypĺňaní faktúr a tak ďalej. Pracuje sa tu často s dôvernými údajmi (napríklad výplatné pásky a iné) a preto jedným z požiadavkov na výslednú knižnicu je bezpečnosť spracovávania dát.
Jedným z ďalších praktických požiadavkov je spoločné, prípadne separované generovanie PDF súborov. Táto funkcia môže byť vhodná, napríklad pre odosielanie mailov zo štatistikami konkrétnemu zamestnancovi, alebo výber zamestnanca vedúcemu. Zabezpečenie vygenerovaných PDF súborov, je potrebné, ak nechceme, aby sa k vygenerovaným súborom nedostali neoprávnené osoby. Preto je vhodné aj zamykanie PDF súborov heslom. Zoznam požiadavkov:

- Spracovanie HTML/JSP stránky
- Vytvorenie PDF súboru, čo najpodobnejšiemu tlačovej zostave
- Pridanie metadát o autorovi, jazyku a kľúčových slovách
- Podpora kaskádových štýlov
- Podpora enkrypcie a zabezpečenia PDF
- Výstupný PDF formát
- Výber veľkosti formátu
- Možnosť separátneho, alebo spoločného generovania PDF súborov

3.4 Schéma návrhu

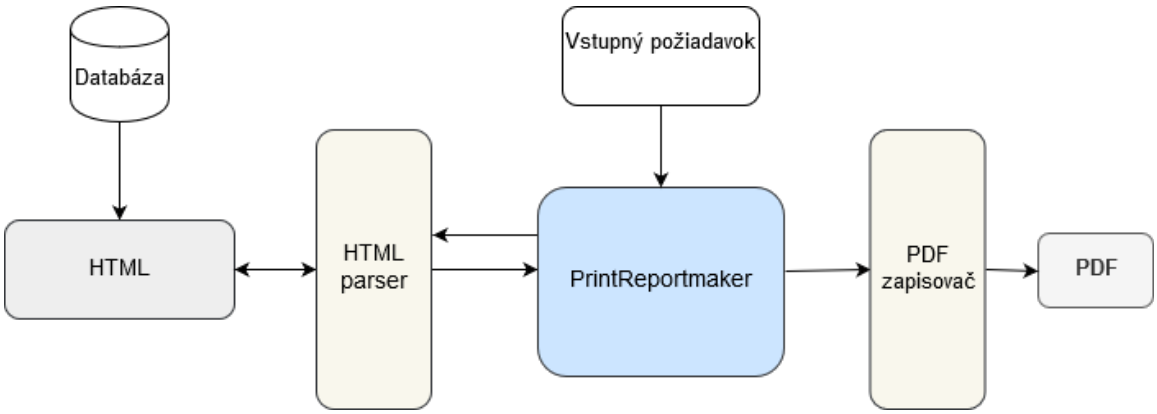
V tejto časti sa nachádza opis konkrétnych krokov pri tvorbe tlačových zostáv, požiadavky na ich vstupy a výstupy.

3.4.1 Rozdelenie častí schémy návrhu

PrintReportMaker je hlavným zameraním tejto práce. Ako je vidieť na obrázku 3.2 PrintReportMaker je vstupným bodom návrhu. Jeho hlavným zameraním je spracovanie požiadavkov a ich sprostredkovanie ďalším knižniciam, ktoré vykonajú najskôr prepojenie s HTML stránkou, na ktorej sa nachádza tlačová zostava už s dátami z databázy. Po syntaktickej analýze HTML stránky je potrebné predať jej výsledok vo vhodnej forme na vytvorenie PDF súboru.

Z pohľadu tohto vstupného bodu sú možné 2 postupy. Jedným je priame využitie *PrintReportMakeru* v kóde a druhým je predspracovanie dát vyformovanie požiadavku a ten zaslať *PrintReportMakeru*.

Súčasťou návrhu je aj predpoklad, že používateľ získa dáta z databázy pred ich využitím v PrintReportMaker. Táto časť nie je priamo v zadaní, avšak je potrebná aj na testovanie a preto je uvedená na konkrétnom príklade aj v implementácii. Tento návrh sa však dá zovšeobecniť. Nakoľko v zadaní je uvedená práca s databázou, iným analyzovaním dát napríklad z XML súboru sa tento návrh nezaoberá.



Obr. 3.2: Podrobnejší návrh

Nakoľko súčasťou zadania bolo oboznámenie sa s knižnicami na syntaktickú analýzu boli v implementácii použité viaceré knižnice. Túto voľbu viacerých riešení (aspoň základného generovania) som sa rozhodla zvoliť na základne rôznych implementácií analyzátorov. Pri hľadaní informácií o nedostatkoch knižníc boli často výsledky neurčité. Vizuálna prezentácia dát je vždy vhodnejšia na porovnanie pri takomto generovaní z predlohy a preto návrh zahŕňa aspoň skúšobné verzie generovania. Rôznorodosť knižníc v jednom celku môže byť zaujímavá aj čo sa týka porovnania licencovania. Ak by bol programátor požiadavku iba na minimalistické tlačové zostavy v podobe základného formátovania textu, tabuliek a obrázkov mu môžu postačovať aj nie úplne komplexné knižnice. Táto časť je bližšie opísaná v implementácii. Nakoľko analyzátory sú zvyčajne čiernou skriňou(poskytnuté sú iba triedy a ich funkcie bez možnosti nahliadnutia do zdrojového kódu) je ich optimalizácia pre konkrétnu situáciu zložitá.V prípadne open source knižníc by bolo možné napríklad ďalšie elementy či kaskádové atribúty a ich rozpoznanie implementovať.

Kapitola 4

Implementácia

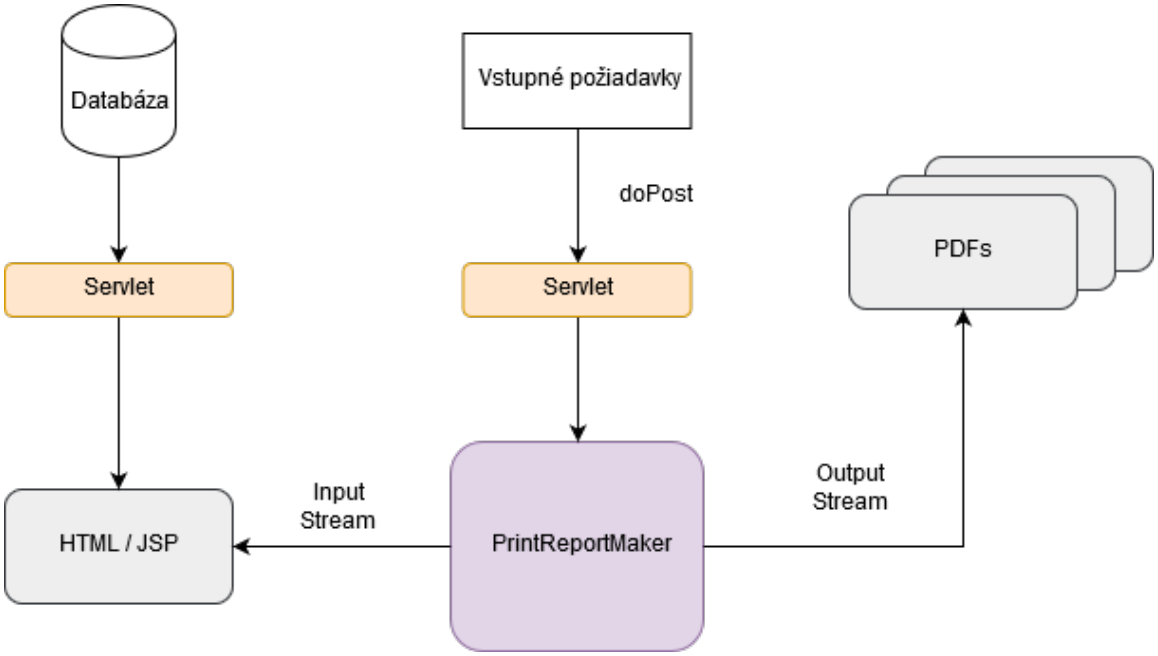
Táto kapitola opisuje konkrétnu a podrobnú implementáciu návrhu vzhľadom na prostredie využitia, ktorým sú webové aplikácie na platforme Java. Krátky úvod k implementačnému prostrediu a zdôvodnenie použitia.

4.1 Vývojové prostredie Java EE

V úvodnej teoretickej časti 2 boli uvedené základné pojmy a vysvetlenia prostredia Java EE. V tejto je bližšie zasadenie vzhľadom na implementáciu práce. Java je výborným jazykom na tvorbu informačných systémov z podnikového hľadiska, ale aj všeobecne pre webové aplikácie. Keďže už samotný názov nesie slovo web, s HTML jazykom sa nie je možné nestretnúť. Môžeme teda implikovať názor, že aj začínajúci vývojár Java EE musí byť oboznámený s jazykom HTML. Vstupná šablóna má formát HTML a je požadovaná od programátora, ktorý chce PrintReportMaker použiť.Ďalej budeme teda preto aj v 5 testovaní predpokladať dobre formátované XHTML alebo HTML.

Nakoľko Java EE poskytuje HTTP orientované triedy ako *request* a *response*, je ideálnym prostredím na vývoj webovej aplikácie. Tu však treba podotknúť, že toto poskytuje veľa ďalších objektovo orientovaných jazykov. Java má širokú komunitu aj pre open source knižnice, čo bolo aj ďalšou motiváciou pre tento výber. Pri webových aplikáciách Java EE je dobrou praktikou písať aplikačnú časť do tried Servlet a v JSP stránkach generovať dynamický obsah. Ak máme logicky zložitejšie časti na spracovanie aj v JSP stránkach aj Servletoch, je oveľa ťažšie a neprehľadnejšie opravovať a meniť kód aplikácie. Ako príklad nevhodného programovania môžeme uviesť dotazy do databázy aj na JSP stránke aj Servletoch, kde potom zbytočne programátor stráca čas hľadaním miesta, kde dotaz môže pôsobiť nechceným a nezámerným spôsobom.

V implementácii projektu som sa rozhodla použiť Apache Maven a to kvôli jeho prehľadnosti pri pracovaní s viacero knižnicami. Ak programátor zabudne pridať knižnice do *build path* sa vyskytnú problémy pri využití takejto knižnice, aj v jeho domnení, že bola pridaná. Maven rieši bezpečné a praktické vytváranie projektov pomocou súboru POM.xml. V tomto súbore sú aj v mojej práci uvedené závislosti potrebné pre správne fungovanie projektu. Treba podotknúť, že neboli pridané všetky knižnice iTextu a preto sa pri ďalšom vývoju tohto projektu prípadne budú musieť pridať.Prejdime však už k samotnej implementácii.



Obr. 4.1: Návrh implementácie

4.2 Popis implementácie tried

4.2.1 servlets.ReportMakerSrvlt

Hlavnou vyhodnocovacou triedou tejto práce, ktorá komunikuje s inými webovými aplikáciami, alebo užívateľmi tejto práce je trieda **servlets.ReportMakerSrvlt**, ktorá rozširuje triedu *HttpServlet*. Táto trieda pri invokovaní funkcie *doPost*, spracuje dáta z formuláru, ktorý užívateľ ručne vytvoril a s použitím triedy PrintReportMaker vytvorí z HTML stránky uvedenej vo formulári PDF súbor. Formulár sa nachádza na obrázku 4.2

Použitie formuláru na vyplnenie dotazu však nie je povinné. Pri invokovaní metódy *doGet* sa *request* taktiež spracuje a nasleduje vytváranie PDF tlačovej zostavy. Táto forma je viac orientovaná na prepojenie dotazov z ďalšej webovej aplikácie, nakoľko tu rozhranie pre vstupné dáta od užívateľa nemáme.

4.2.2 DBUtilities.Student

Trieda *Student* je potrebná na spoluprácu Servletu, databázy a JSP stránky. Trieda opisuje svojimi členmi jednotlivé stĺpce z tabuľky v databáze. Vytvorenie tabuliek a vloženie vzorových dát sa nachádza v prílohe. Ďalej sa k tejto triede nachádza funkcia *DBConnect.queryStudent()* ktorej argumentom je *java.sql.Connection* a výstupom List s objektami triedy Student. Tu je všeobecne možnosť rozšírenia o konkrétne požiadavky na dotaz na databázu a jej filtrovanie. Členovia triedy sú *id, firstname, lastname, email, birthday, prog.* Člen *prog* je typu integer a hodnota $prog = 1$ znamená, že študent ešte neukončil bakalárske štúdium a číslo dva zas pokračovanie na magisterskom študijnom programe.

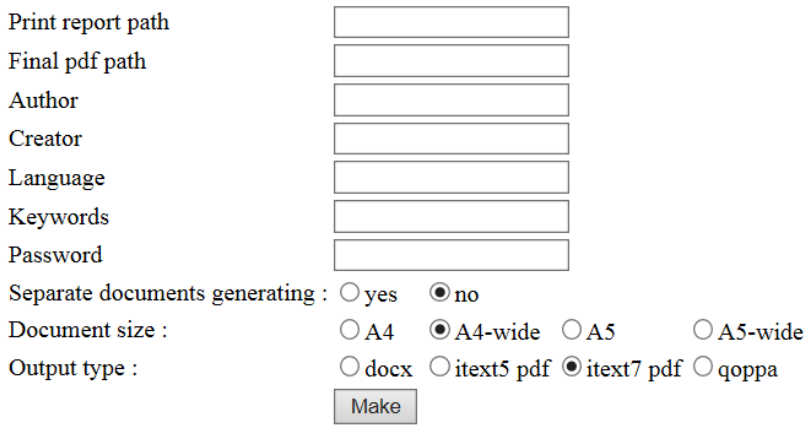
4.2.3 DBUtilities.Employee

Obdobnou triedou ako 4.2.2 je aj trieda *Employee*. Táto trieda reprezentuje dáta uložené v MySQL tabuľke priprojenej cez JDBC ku webovej aplikácii. Obsahuje členov *id, firstname, lastname, email, birthday, adress, idnumber*. Pomenovania som sa snažila vyberať, aby bolo už z názvu jasný ich charakter a využitie. *Idnumber* je identifikačné číslo priradené zamestnávateľom.

4.2.4 DBUtilities.DBConnect

DBConnect je trieda, ktorá zabezpečuje komunikáciu s databázovým serverom, ktorý bol v prípade tejto práci MySQL. Funkcia *getDB* má argumenty *dbName, user, pass* a jej výstupom je *java.sql.Connection*. V tejto triede sa nachádzajú aj implementácie dotazujúcich sa funkcií do databázy.

Pripájanie do databázy zabezpečujú servlety, ktoré presmerovávajú na JSP stránky šablón. Servlet sa pri požiadavku pripojí do databázy, získa aktuálne dáta spracuje ich, následne pridá k *request* a prepošle ďalej pomocou *javax.servlet.RequestDispatcher*.



Obr. 4.2: Formulár pre vygenerovanie PDF

4.2.5 PrintReportMaker

PrintReportMaker je trieda, ktorá obsahuje všetky podstatné informácie a funkcie k syntaktickej analýze HTML súborov a tvorbe PDF. Jej členov sa dá rozdeliť do 3 logických častí. Členovia *pdfPath, reportPath* sú cesty k vstupu a výstupu *String*. *pdfPath* obsahuje absolútnu cestu a meno súboru, ktorý chceme vytvoriť. Ak nie je súčasťou tohto *String* aj ".pdf"tak je táto prípona pridaná.

Člen *reportPath* obsahuje cestu k JSP súboru v kontexte serveru. Nakoľko vychádzame z návrhu a z prehľadnosti pri väčších aplikáciach si myslím, že je vhodné mať tieto Servlety pre JSP stránky zvlášť. Druhou možnosťou implementácie by mohla byť absolútna cesta a potom presmerovanie na JSP stránku. Tu však musíme zabezpečiť aby bola v kontexte serveru a teda načítaná JSP kontajnerom a nie iba lokálne uloženie súboru. Vytvorenú statickú tlačovú zostavu v pohode vygenerovaného HTML súboru podporujú funkcie iText 7 a 5.

Člen *pSize* má predurčenú hodnotu *PageSize.A4* a je typu *com.itextpdf.text.Rectangle* a ako je už aj s názvu balíka vidieť jedná sa o knižnicu iText. Veľkosti strany, ktoré boli

implementované ako možnosti generovania sú A4 portrét aj na šírku a A5 tak isto. Tu je ľahká možnosť pridania podpory aj iných formátov ich pridaní do konštrukcie switch. Nastavovanie veľkosti stránky pre iText knižnicu je implementované vo funkcii *setSize()*.

Ďalšími členmi sú *marginLeft, marginRight, marginTop, marginBottom* typu *float* potrebnými pre nastavenie okrajov v generovanom PDF. Ich predvolenou hodnotou je 36f.

Nasledujúcou skupinou členov sú metadata. Metadáta sú podstatnou časťou PDF súboru, nakoľko vďaka nim sa môžeme dozvedieť spôsob vytvorenia PDF, jeho auta, jaky a kľúčové slová. Správny výber metadát PDF súbru môže taktiež zvýšiť jeho šancu nájdenia napríklad na internetových vyhľadávačoch. SEO (Search Engine Optimization) síce slúži primárne na webové stránky, ale podobné pravidlá napríklad dobrého pomenovania môžeme aplikovať aj na PDF.*author, creator, keyWords, language, password* sú členovia typu *String* a súžia teda na nastavenie týchto metadát v PDF súbore, ktorý vytvárame.

Špeciálnym prípadom je ešte heslo *password* na enkrypciu súboru. Táto vlastnosť môže byť vhodná pre generovanie tlačových zostáv z citlivých dát prípadne na bezpečné preposielanie cez email bez úniku informácii. Heslá v kontexte PDF súborov sa dajú rozdeliť na heslo vlastníka a heslo užívateľa.

Poslednými dvomi členmi sú *metaData* a *separateGenerating*. *metaData* má predvolenú hodnotu false, čo znamená že nebudeme pridávať metadáta k súboru a *Integer separateGenerating* je číslo, na separátne generovanie PDF súborov. Toto číslo udáva počet opakovaní generovania a jeho záporná hodnota znamená, že výsledok generovania bude v jednom PDF súbore.

4.3 Opis funkcií triedy PrintReportMaker

V nasledujúcej časti sa nachádza potrebnejší opis jednotlivých funkcií generovania tlačových zostáv, ich fungovanie a prípadné nedostatky.

4.3.1 Funkcia setParameters()

Funkcia *setParameters()* slúži na spracovanie argumentov dotazu. Celá deklarácia funkcie je *public void setParameters(HttpServletRequest request)*. Očakávané parametre dotazu sú : *printreport, pdfpath, size, author, keywords, language, creator, pass*. Pričom pri nastavení členov metadáta sa zmení aj člen *metaData* typu Boolean na hodnotu *true*.

4.3.2 Funkcia setSize()

Funkcia *setSize()* slúži na vyhodnotenie reťazca do štruktúry vhodnej na použitie pre konštruktor triedy *com.itextpdf.text.Document*.

4.3.3 Funkcia generateq()

Táto funkcia používa na syntaktickú analýzu HTML a generovanie PDF súboru knižnicu *com.qoppa.pdfWriter*. Táto knižnica je pomerne jednoduchšia v porovnaní s ostatnými, avšak má vlastné licencovanie. Pre začiatok bolo potrebné vytvoriť *java.net.URL.URL* objekt, ktorý slúži ako input pre funkciu na analýzu HTML. Ďalej bola použitá trieda *java.awt.print.PageFormat* a teda veľkosť výsledného PDF súboru. PDF dokument je vytvorený funkciou *saveDocument()*, ktorý vkladá do zanalyzované HTML ako TIFF. Toto riešenie sa osvedčilo ako vhodné iba pri veľmi ľahkých tlačových zostavách.

4.3.4 Funkcia generatePDFFromHTMLv7()

Táto funkcia vytvára tlačové zostavy z pomedzi všetkých najefektívnejšie. *PdfWriter* je trieda použitá na zapisovanie do PDF súboru. Tento zapisovač sa pripojí k *pdfdokument*u, nakoľko sám o sebe neukladá. Pri vytváraní tohto zapisovača môžeme nastaviť vlastnosti dokumentu, ako tlačenie, heslo, pripojenie certifikátov a iné. Pre separované generovanie súborov, je za menom súboru iterované číslo súboru. V tejto časti programu môže nastať pomerne veľa behových chýb programu. Väčšinu som sa snažila zachytiť formou *try=cathc*, ale je možné, že pri testovaní nebolo dosiahnutých všetkých chybových stavov.

Po otvorení spojenia s URL stránkou a nastavení metadát pomocou funkcie *getCatalog()* prichádza ku konvertovaniu HTML stránky. Za predpokladu, že celá funkcia prebehla bez chýb je výsledné PDF úspešne uložené.

Na syntaktickú analýzu HTML bola použitá funkcia *convertToPdf()* triedy *HtmlConverter*. Spolu s argumentami *Inputstream* a *PdfDocument* tak bolo vytvorené výsledné pdf.

4.3.5 Funkcia generatePDFFromHTMLv5()

Funkcia generatePDFFromHTMLv5() generuje výsledné PDF pomocou staršej verzie iText 5. S touto verziou som pracovala na začiatku, kvôli možnosti využitia aj na starších platformách Java, avšak obsahuje veľa obmedzení a chýb. Nakoľko spadá pod rovnakú licenciu ako iText 7, by som využitie tejto knižnice v novo vytváraných webových aplikáciách nedoporučila.

Funkcia podporuje vkladanie metadát do PDF súboru a aj analýzu XHTML. Pri zle formátovanom XHTML, zabudnutí uzatváracieho elementu a iných chýb funkcia nie je schopná vytvoriť PDF súbor.

použitým nástrojom na analýzu HTML je *XMLWorkerHelper*, ktorý je predchodcom *HtmlConverter*. Document je trieda, do ktorej sa pridávajú textové elementy. PdfWriter zapisuje prvky z dokumentu do OutputStreamu, ktorým je výsledný PDF súbor.

Pred zápisom musí byť dokument otvorený. Následne funkcia *parseXHtml()* spolu s dokumentom *pdfWrite*rom a input streamom, ktorým je URL na ktorej sa nachádza tlačová zostava aj s dátami vygeneruje triedy vhodné pre *pdfWriter*. Po zatvorení a ukončení funkcie by malo byť úspešne vygenerované PDF. Ak počas funkcie nastala chyba, jej výpis je pre reportovacie účely uložený vo výstupnom String.

4.3.6 Funkcia gener()

Funkcia *gener()* je nad rámec tejto práce a vytvára word dokument. Túto funkciu som sa rozhodla pridať iba na porovnanie výsledkov s PDF súbormi. Syntaktická analýza HTML súboru je podporovaná triedou *XHTMLImporterImpl*. *WordprocessingMLPackage* do tejto triedy sa vkladajú prvky reprezentujúce nadpisy paragrafy a tak ďalej. Táto funkcia môže byť zaujímavo pre vytváranie súborov za účelom ľahkej manipulácie a úpravy po otvorení, nakoľko PDF súborom sa tak jednoducho manipulovať s obsahom nedá.

4.3.7 Funkcia czechSlovakCorrection()

Popri testovaní generovania PDF súborov častokrát staršie PDF zapisovače nevedeli rozpoznať špeciálne diakritické znaky slovenského a českého jazyka. Funkcia mení tieto znaky na HTML entity reprezentované HTML kódom &#xxx , kde x sú čísla.

4.4 Implementácia demonštračnej aplikácie

V nasledujúcej časti sa nachádza krátky opis implementácie a návod použitia webovej aplikácie.

4.4.1 Návod na obsluhu aplikácie

Výsledkom tejto práce je webová aplikácie na generovanie tlačových zostáv. Aplikáciu je možné skompilovať pomocou nástroju Maven a to konkrétne príkazom *maven install*, ktorý zabalí aplikáciu do súboru war. Tento súbor je následne možný spustiť napríklad na servery Tomcat.

Po spustení aplikácie je zobrazená úvodná stránka, ktorá obsahuje formulár viď 4.2. Pole *Print report path* očakáva cestu k tlačovej zostave, ktorú chceme generovať, *Final pdf path* cestu, kde chceme súbor uložiť, *Author, Creator, Language, Keywords, Password* sú polia pre metadáta a sú nepovinné. Pri zvolení *Separate documents generating* sa vytvorí toľko súborov, koľko bude databáza obsahovať položiek. Následne je možnosť si zvoliť veľkosť stránky a typ použitého generátoru na výsledné tlačové zostavy.

Treba poznamenať, že vytvorené tlačové zostavy nebudú fungovať ak im nebude umožnené pristupovať k databáze. Implicitné nastavenie mena databázy je *jsp*, užívateľ je *root* a heslo 2255. Bližšie informácie k spojazdneniu aplikácie sa nachádzajú v prílohe.

4.5 Zhodnotenie implementácie

Generovanie Tlačových zostáv bolo implementované a miera úspešnosti závisí pri niektorých spracovaniach od vstupnej šablóny. Výsledok tejto práce môže byť použitý v rámci webovej aplikácie priamo, napríklad vložením triedy PrintReportMaker, alebo spustením WAR súboru už skompilovanej webovej aplikácii.

Kapitola 5

Testovanie

V tejto kapitole je opísaný postup a spôsob testovania implementácie tejto práce. Počas testovania boli zavedené aj zmeny v implementácii.

5.1 Príprava na testovanie

K príprave na testovanie implementovaného návrhu v jednoduchej webovej aplikácii bolo potrebné splniť niekoľko podmienok. Prvou z nich bolo vytvorenie databáze, ktorá by slúžila ako zdroj informácií do tlačových šablón. Druhou podmienkou bolo vytvorenie JSP stránky a teda šablóny potrebnej na spracovanie vstupných dát z databázy. JSP stránka by mala byť pre správne fungovanie všetkých použitých knižníc správne formátovaná, ideálne podľa syntaxu XHTML. Ďalšou podmienkou, aby sa toto testovanie aspoň z časti podobalo prostrediu, v ktorom by mala byť webová aplikácia využívaná bolo vytvorenie servletu na preposlanie dát do šablóny. Všetky vyššie spomenuté súbory sa nachádzajú v prílohe.

5.2 Priebeh testovania

Testovanie prebiehalo na Apache Tomcat Servery v9.0. Domovská stránka tejto webovej aplikácie slúži na jednoduché grafické vyplnenie argumentov pre vytvorenie PDF súboru a metódou post predáva ďalej tieto argumenty servletu, ktorý ich spracuje. Na testovanie boli použité vytvorené servlety tlačových zostáv. Predpokladaný tvar kontextu webovej aplikácie "http://localhost:8080/PrintReports/".

- (kontext webovej aplikácie)/students
- (kontext webovej aplikácie)/employeesrv

5.3 Testovanie tlačovej zostavy potvrdenia

Ako jeden z príkladov využitia tlačových zostáv v praxi máme potvrdenie o štúdiu. V tejto tlačovej zostave bolo automaticky z databázy doplnené meno študenta, dátum narodenia a program, v ktorom študuje. Na obrázku 5.2 môžeme vidieť výstupný PDF súbor. Pre porovnanie na obrázku je ukážka tejto vstupnej HTML stránky v prehliadači 5.1. Táto stránka mala v sebe špeciálnu slovenskú diakritiku, ktorú tento analyzátor nedokázal správne do PDF súboru vložiť.

Vysoká škola života v Bratislave

Fakulta informačných technológií a mágie

U dunaja 447, Bratislava

V Bratislave dňa 10.5.2009

POTVRDENIE

Vedenie fakulty informačných technológií a mágie Vysokej školy života v Bratislave týmto dokumentom potvrdzuje, že študent/študentka John Doe narodný/á dňa 07.11.1987 je v akad. roku 2019 denným/ou študentom/ študentkou(ou) bakalářského študijného programu. Toto potvrdenie je platné po dobu 3 mesiacov od jeho vydania

.....

Z. S.

podpis

Vysoká škola života v Bratislave

Fakulta informačných technológií a mágie

U dunaja 447, Bratislava

V Bratislave dňa 10.5.2009

POTVRDENIE

Vedenie fakulty informačných technológií a mágie Vysokej školy života v Bratislave týmto dokumentom potvrdzuje, že študent/študentka Zuz Syni narodný/á dňa 01.02.2001 je v akad. roku 2019 denným/ou študentom/ študentkou(ou) bakalářského študijného programu. Toto potvrdenie je platné po dobu 3 mesiacov od jeho vydania

Obr. 5.1: Tlačová zostava potvrdenia zobrazená prehliadačom

Z. S.

podpis

VysokĂˇ Ĺˇkola Ĺľivota v Bratislave

Fakulta informaÄŤnĂ˝ch technolĂłgii a mĂˇgie

U dunaja 447, Bratislava

V Bratislave dĹ□a 10.5.2009

POTVRDENIE

Vedenie fakulty informaÄŤnĂ˝ch technolĂłgiĂ a mĂˇgie Vysokej Ĺˇkoly Ĺľivota v Bratislave tĂ˝mto dokumentom potvrdzuje, Ĺľe Ĺˇtudent/Ĺˇtudentka Zuz Syni narodnĂ˝/Ăˇ dĹ□a 01.02.2001 je v akad. roku 2019 dennĂ˝m/ou Ĺˇtudentom/ Ĺˇtudentkou(ou)
bakalářského ĹˇtudijnĂ©ho programu. Toto potvrdenie je platnĂ© po dobu 3 mesiacov od jeho vydania

.....

Obr. 5.2: Tlačová zostava potvrdenia vytvorená PdfWriterom

V Bratislave dĹa 10.5.2009

POTVRDENIE

Vedenie fakulty informaĂŤnĂ˝ch technolĂłgiĂ a mĂ˝gie Vysokej Ĺ˝koly ĹľivotaĺV Bratislave tĂ˝mto dokumentom potvrdzuje, ĹľeĺĹ˝tudent/Ĺ˝tudentka Zuz Syni narodnĂ˝/Ă˝ dĹa 01.02.2001 je v akad. roku 2019 dennĂ˝m/ou Ĺ˝tudentom/ Ĺ˝tudentkou(ou)

bakalářského Ĺ˝tudijnĂ©ho programu. Toto potvrdenie je platnĂ© po dobu 3 mesiacov od jeho vydania

.....

Z. S.

podpis

VysokĂ˝ Ĺ˝kola Ĺľivota v Bratislave

Fakulta informaĂŤnĂ˝ch technolĂłgii a mĂ˝gie

U dunaja 447, Bratislava

Obr. 5.3: Tlačová zostava potvrdenia vytvorená pomocou iText 5

Po viacerých testoch som zistila, že ak je táto diakritika nahradená HTML entitami, tento problém je v niektorých knižniciach redukovaný. Na obrázku môžeme taktiež vidieť, že kaskádový štýl je v porovnaní s inými výsledkami väčší a zalamovanie strán nebolo tiež dodržané. Táto knižnica pravdepodobne nevedela spracovať *style="page-break-inside: avoid;"* a tak bol plne ignorovaným. IText 5 viď na obrázku 5.3 dokázal správne zalomiť stránky, aj napriek tomu, že diakriticky boli znaky vložené nesprávne.

IText 7 dosiahol najlepších výsledkov. Stránka bola správne zalomená a všetko formátovanie ako aj diakritika sa vierohodne podobajú originálu. K celkovému zhodnoteniu generovania potvrdenia je možné skonštatovať veľké rozdiely na základe použitých knižníc. Nakoľko na stránke nebola nastavená šírka, pozitívnym zistením bolo, že každá knižnica si poradila so vsadením textu do okrajov. Asi za najväčší problém považujem ignorovanie kaskádových štýlov pre tlač stránky, na základe ktorých by malo byť upresnené zalamovanie.

POTVRDENIE

Vedenie fakulty informačných technológií a mágie Vysokej školy života v Bratislave týmto dokumentom potvrdzuje, že študent/študentka Zuz Syni narodný/á dňa 01.02.2001 je v akad. roku 2019 denným/ou študentom/ študentkou(ou) bakalářského študijného programu. Toto potvrdenie je platné po dobu 3 mesiacov od jeho vydania

.....

Z. S.

podpis

Vysoká škola života v Bratislave

Fakulta informačných technológii a mágie

U dunaja 447, Bratislava

V Bratislave dňa 10.5.2009.

POTVRDENIE

Vedenie fakulty informačných technológií a mágie Vysokej školy života v Bratislave týmto dokumentom potvrdzuje, že študent/študentka Kate Doe narodný/á dňa 11.12.2065 je v akad. roku 2019 denným/ou

Obr. 5.4: Tlačová zostava potvrdenia vytvorená pomocou iText 7

5.4 Testovanie tlačovej zostavy zamestnancov

Ako druhú sadu testovania generovania tlačových zostáv som sa rozhodla použiť tabuľku zamestnancov s mierne zložitejším využitím kaskádových štýlov, ako je napríklad pozadie. Na obrázku 5.5 je zobrazená pôvodná tlačová zostava v prehliadači.

Ako prvé generovanie prebehlo pomocou PdfWriteru a je zobrazené na obrázku 5.6. Ako je už na prvý pohľad vidno, niektoré štýlové prvky boli zachované, avšak napríklad pozadie a výška stĺpcov sa stratila. Na druhú stranu môžeme opäť pozorovať, že tabuľka bola prispôsobená veľkosti výsledného PDF súboru. Aj napriek tomu, že sa výsledok nepodobá šablóne výsledok by mohol byť použitý, pretože mimo formát obsahuje rovnaké dáta ako vstup.

Ďalšou testovacou knižnicou bol iText 5 zobrazené na obrázku 5.7. Výsledok sa podobá na predošlý generátor s miernou odlišnosťou kraju tabuľky. Opäť však nebolo vygenerované pozadie tabuľky.

Posledným generovaním bolo za pomoci knižnice iText 7 a toto zobrazenie sa najviac podobá. Ako môžeme vidieť na obrázku 5.8 do PDF súboru bolo vložené aj pozadie tabuľky. Na prvý pohľad sa môže zdať byť výsledné PDF z časti iné, avšak to je spôsobené zarovnaním do šírky okrajov súboru, ktoré v prehliadači nebolo.

Zoznam zamestnancov

Meno	Priezvysko	Rok narodenia	Email	Id zamestnanca
Mark	Lesky	Mark@example.com	Sturova 22, Bratislava	112564
Colton	Leslie	pajas@hotmail.com	PO udeym 6, Bratislava	145894
Manie	Troxell	ahuillet@optonline.net	Masten 72, Bratislava	114784
Gabriella	Commons	fhirsch@gmail.com	Ústí nad Labem 2	85661
Wilfred	Wene	fiesh@gmail.com	Ústí nad Labem 452	95661
Valentina	Whisenant	fhirsch@gmail.com	Ústí nad Labem 2	4755841
Pedro	Bise	arnold@att.net	Ústí nad Labem 28	45661
Nikita	Melchor	rnewman@sbcglobal.net	Ústí nad Labem 27	75451
Miyoko	Skeens	thurston@optonline.net	Ústí nad Labem 23	54561
Freda	Mcdavid	rnewman@sbcglobal.net	Ústí nad Labem 24	45757
Mark	Lesky	Mark@example.com	Sturova 22, Bratislava	112564
Colton	Leslie	pajas@hotmail.com	PO udesdm 6, Bratislava	145894
Manie	Troxell	ahuillet@optonline.net	Masten 72, Bratislava	114784
Gabriella	Commons	fhirsch@gmail.com	Ústí nad Labem 2	85661
Wilfred	Wene	fiesh@gmail.com	Ústí nad Labem 452	95661
Valentina	Whisenant	fhirsch@gmail.com	Ústí nad Labem 2	4755841
Pedro	Bise	arnold@att.net	Ústí nad Labem 28	45661

Obr. 5.5: Tlačová zostava zoznamu zamestnancov zobrazená v prehliadači

Zoznam zamestnancov

Meno	Priezvysko	Rok narodenia	Email	Id zamestnanca
Mark	Lesky	Mark@example.com	Sturova 22, Bratislava	112564
Colton	Leslie	pajas@hotmail.com	PO udeym 6, Bratislava	145894
Manie	Troxell	ahuillet@optonline.net	Masten 72, Bratislava	114784
Gabriella	Commons	fhirsch@gmail.com	ĂšstĂ nad Labem 2	85661
Wilfred	Wene	fiesh@gmail.com	ĂšstĂ nad Labem 452	95661

Obr. 5.6: Tlačová zostava zoznamu zamestnancov vytvorená PdfWriterom

5.5 Zhodnotenie testovania

Testovanie prebehlo úspešne na dvoch tlačových zostavách, ktoré slúžili ako príklad využitia knižnice. Pri tomto testovaní sa opäť potvrdil problém spojený s generovaním PDF súborov. V tejto práci som sa predovšetkým snažila využiť knižnice aj na podporu špeciálnym slovenských a českých znakov, ktoré bývajú problematické pri vkladaní do PDF súboru. V celku považujem toto testovanie za úspešné, nakoľko odhalilo aj pozitíva aj negatíva rôznych knižníc. Mimo tieto vizuálne testy bolo taktiež otestovanie použitia hesla na verzii iText 7 a taktiež aj separované generovanie súborov položka po položke do oddelených PDF súborov.

Zoznam zamestnancov

Meno	Priezvysko	Rok narodenia	Email	Id zamestnanca
Mark	Lesky	Mark@example.com	Sturova 22, Bratislava	112564
Colton	Leslie	pajas@hotmail.com	PO udeym 6, Bratislava	145894
Manie	Troxell	ahuillet@optonline.net	Masten 72, Bratislava	114784
Gabriella	Commons	fhirsch@gmail.com	ĂšstĂ nad Labem 2	85661
Wilfred	Wene	fiesh@gmail.com	ĂšstĂ nad Labem 452	95661
Valentina	Whisenant	fhirsch@gmail.com	ĂšstĂ nad Labem 2	4755841
Pedro	Bise	arnold@att.net	ĂšstĂ nad Labem 28	45661

Obr. 5.7: Tlačová zostava zoznamu zamestnancov vytvorená pomocou iText 5

Zoznam zamestnancov

Meno	Priezvysko	Rok narodenia	Email	Id zamestnanca
Mark	Lesky	Mark@example.com	Sturova 22, Bratislava	112564
Colton	Leslie	pajas@hotmail.com	PO udeym 6, Bratislava	145894
Manie	Troxell	ahuillet@optonline.net	Masten 72, Bratislava	114784
Gabriella	Commons	fhirsch@gmail.com	Ústí nad Labem 2	85661
Wilfred	Wene	fiesh@gmail.com	Ústí nad Labem 452	95661
Valentina	Whisenant	fhirsch@gmail.com	Ústí nad Labem 2	4755841
Pedro	Bise	arnold@att.net	Ústí nad Labem 28	45661
Nikita	Melchor	rnewman@sbcglobal.net	Ústí nad Labem 27	75451

Obr. 5.8: Tlačová zostava zoznamu zamestnancov vytvorená pomocou iText 7

Kapitola 6

Záver

Cieľom tejto práce bolo vytváranie tlačových zostáv vo webových aplikáciách Java EE. Pri riešení tejto práce bolo podstatné si spraviť celkový prehľad riešenia tvorby súborov PDF v Java EE a taktiež aj nástroje na syntaktickú analýzu a spracovanie HTML súborov. V súčasnosti existuje mnoho knižníc, ktoré podporujú vytváranie PDF súborov z HTML vstupu. Mnohé knižnice sa teda prezentujú touto vlastnosťou avšak miera podobnosti s pôvodnou stránkou je bez vyskúšania na konkrétnych tlačových zostavách niekedy ťažko porovnateľná. V tejto práci sa podarilo implementovanie vytvárania tlačových zostáva je na výbere programátora, ktorú z nich si vyberie. Do budúcna by sa dalo v tejto práci pokračovať doplnením PDF formátov ako PDF/UA a s tým aj značkované PDF. Keďže hlavným zámerom sú webové aplikácie, ktoré často využívajú podniky, by do budúcna mohla pribudnúť aj podpora certifikácie a podpisovania PDF súborov. Pri vypracovávaní tejto práce som sa dozvedela veľa nových teoretických poznatkov o pojmoch, ktoré sa často vo svete IT vyskytujú, ale nevieme ich uchopiť pri položení otázky slovne. Taktiež som preštudovala životnosť Servletov a možnosti vývoju webových aplikácií. V porovnaní webovej aplikácie, ktorú som vyvíjala v jazyku Python sa mi vývojové prostredie Java EE páčilo viac.Všeobecným výhľadom do budúcna tejto by bola webová aplikácia, ktorá by na požiadavok zamestnanca komplexne pristupovala do databázy reportovacích šablón a generovala formou stiahnuteľnej PDF tlačovej zostavy ako odpovede. Jedným z ďalších rozšírení by mohla byť podpora zasielania tlačových zostáv generovaných v časových intervaloch, alebo po vyvolaní spúšťača generovania a ich odoslanie na email používateľom v databáze.

Literatúra

[1] , R. F. L. M., T. Berners-Lee: *RFC 3986 Uniform Resource Identifier (URI): Generic Syntax*. 2006, [Online; navštívené 9.5.2019].
URL https://tools.ietf.org/html/rfc3986

[2] Budi Kurniawan, P. D.: *Servlet, JSP and Spring MVC*. Brainy Software Inc, 2015, 1-771-97002-2.

[3] Kurniawan, B.: *Java for the Web with Servlets, JSP, and EJB*. New Riders Publishing, 2002, 0-735-71195-X.

[4] Ndegwa, A.: *What is a Web Application?* 2016, [Online; navštívené 7.5.2019].
URL https://www.maxcdn.com/one/visual-glossary/web-application/

[5] Nourie, D.: *Java Technologies for Web Applications*. 2006, [Online; navštívené 9.5.2019].
URL
https://www.oracle.com/technetwork/articles/java/webapps-1-138794.html

[6] Padova, T.: *Adobe Acrobat 9 PDF Bible*. Wiley Publishing, 2009, iSBN 0-470-37919-6.

[7] Pilgrim, P. A.: *Java EE 7 Developer Handbook*. Packt Publishing Ltd, 2013, 1-849-68795-1.

[8] Project, A. M.: *What is Maven?* [Online; navštívené 7.5.2019].
URL https://maven.apache.org/what-is-maven.html

[9] Robbins, J. N.: *HTML and XHTML Pocket Reference*. O'Reilly and Associates, 2006, iSBN 0-596-52727-6.

[10] Wong, C.: *HTTP pocket reference*. O'Reilly and Associates, 2000, iSBN 15-659-2862-8.

Príloha A

Obsah pribaleného CD

Obsahom CD je:

- **source** zložka obsahujúca zdrojový kód práce
- **bpsource** zložka obsahujúca zdrojový kód LaTeX
- **bpxsynak02.pdf** súbor obsahujúci text tejto práce
- **README** súbor obsahujúci informácie k spustení demonštračnej aplikácie