VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

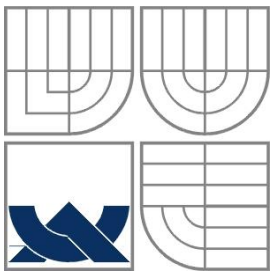
WEBOVÁ APLIKACE REDAKČNÍHO SYSTÉMU PRO SPRÁVU DOKUMENTŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

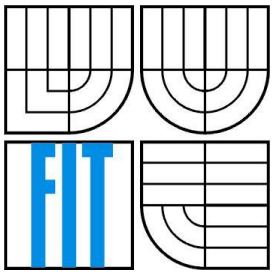
AUTOR PRÁCE
AUTHOR

DAVID SCHIMMEL

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

WEBOVÁ APLIKACE REDAKČNÍHO SYSTÉMU PRO SPRÁVU DOKUMENTŮ

WEB APPLICATION FOR DOCUMENT MANAGEMENT SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — David Schimmel
AUTHOR

VEDOUCÍ PRÁCE — Ing. Michal Drozd
SUPERVISOR

BRNO 2010

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací webové aplikace redakčního systému pro správu dokumentů. Tato aplikace musí být schopna načíst, zpracovat a uložit obsah dokumentů typu Microsoft Office Word. Uložený obsah pak musí být schopna prohledávat, filtrovat a editovat.

Abstract

Aim of this thesis is to design and implement web application for document management system. This application must be able to read, process and save the content of Microsoft Office Word documents. The application must be also able to search, filter and edit the saved content.

Klíčová slova

Redakční systém, publikační systém, systém pro správu obsahu, CMS, webová aplikace, OpenCms, Microsoft Office Word dokumenty.

Keywords

Content Management System, CMS, Document Management System, Web Application, OpenCms, Microsoft Office Word documents.

Citace

Schimmel David: Webová aplikace redakčního systému pro správu dokumentů, bakalářská práce, Brno, FIT VUT v Brně, 2010

Webová aplikace redakčního systému pro správu dokumentů

Prohlášení

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

……
David Schimmel
15. května 2010

Poděkování

Rád bych poděkoval vedoucímu mojí práce Ing. Michalovi Drozdovi za jeho pomoc a rady.

Obsah

1 Úvod

Cílem této práce je navrhnout a implementovat webovou aplikaci redakčního systému pro správu dokumentů. Tato aplikace musí být schopna načíst, zpracovat a uložit obsah dokumentů typu Microsoft Office Word. Uložený obsah pak musí být schopna prohledávat, filtrovat a editovat. Jednotlivé etapy vývoje, postřehy, ale i problémy budou zmíněny v následujících kapitolách.

Tato kapitola tvoří úvod do problematiky redakčních systémů. Vysvětluje samotný pojem redakční systém. Jsou zde rozebrány důvody pro nasazení těchto systémů, výhody a nedostatky. Tento rozbor je však pouze letmým úvodem do této rozsáhlé problematiky a neklade si za cíl ji detailně prozkoumat.

Druhá kapitola se zabývá analýzou a specifikací požadavků. Jsou zde vysloveny požadavky vedoucího práce na webovou aplikaci. Tyto požadavky byly analyzovány a na jejich základě byl vytvořen diagram případů užití. Jednotlivé případy užití jsou poté specifikovány pomocí tzv. detailů případů užití.

Třetí kapitola se věnuje výběru vhodného redakčního systému, především na základě zjištění z minulé kapitoly. Výběr je omezen pouze na dva redakční systémy (OpenCms a Drupal), které jsou v této kapitole analyzovány a srovnávány.

Čtvrtá kapitola se pak zabývá návrhem webové aplikace. A její součástí je i návrh databáze ve formě ER diagramu.

Pátá kapitola je cílena na přípravu všeho potřebného pro zprovoznění serveru, na kterém bude běžet zvolený redakční systém. V úvodu jsou zmíněny operační systémy, které byly použity jako testovací platforma. Dále je vysvětlena instalace a konfigurace softwaru, který vyžaduje redakční systém ke svému běhu. Následuje vytvoření databáze pro webovou aplikaci, instalace a konfigurace redakčního systému. Nakonec dojde na import webové aplikace do redakčního sytému.

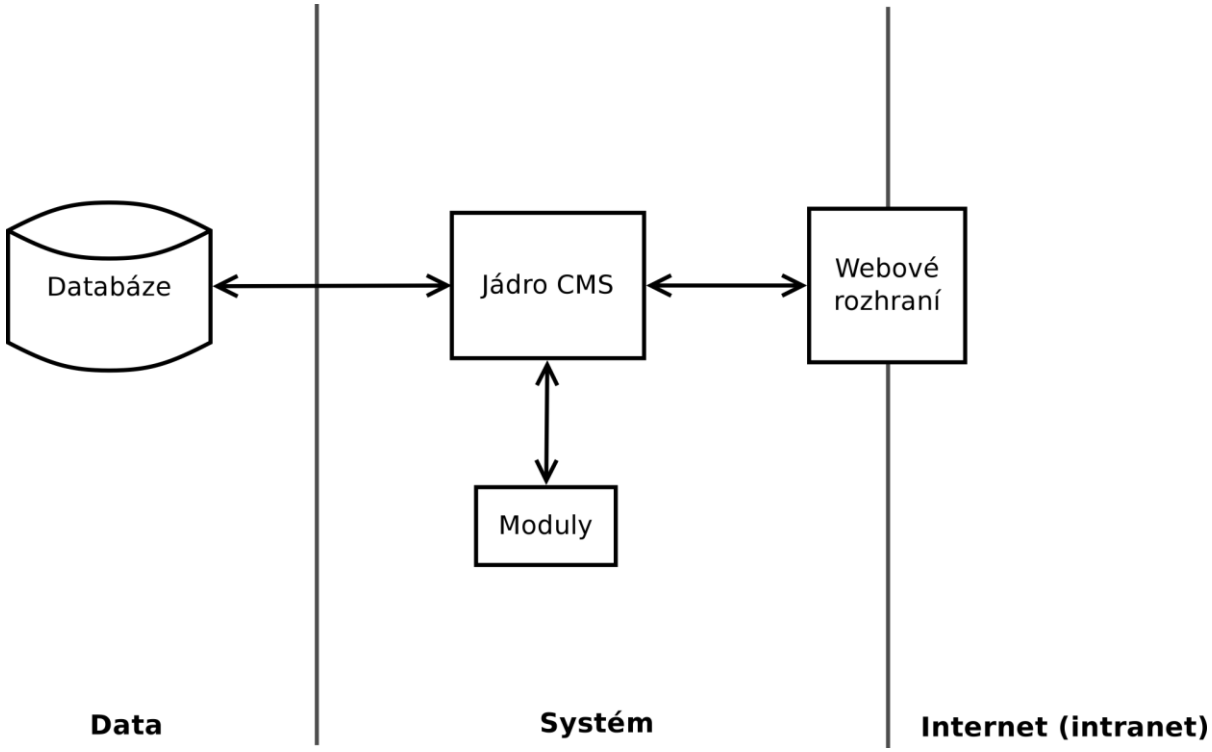
Šestá kapitola se věnuje implementaci webové aplikace pro správu dokumentů. V této kapitole se dočtete, jak bylo řešeno nahrávání a zpracování dokumentů. Také se dozvíte, jak bylo implementováno rozšíření ve formě importu dokumentů ve formátu PDF a HTML, a jaké změny to vyžadovalo.

V sedmé kapitole jsou informace o průběhu testování aplikace a podobě testovacích dokumentů. Dále pak výsledky průzkumu především na ovládání a funkčnost aplikace.

Na závěr tohoto dokumentu jsou zmíněny dosažené výsledky a možná rozšíření aplikace.

1.1 Redakční systém

Nebo také publikační systém či systém pro správu obsahu (CMS - Content Management System) je počítačový program, který slouží ke správě a publikování obsahu. Pod pojmem obsah se skrývají v podstatě veškeré informace, které lze uložit do počítače. Příkladem takového obsahu mohou být dokumenty, audio a video soubory, fotky, obrázky, atd. Takový systém pak zpravidla běží na nějakém serveru, pracuje s daty v databázi a uživatelé se k němu připojují přes webový prohlížeč. Jedná se tedy o architekturu typu klient-server. Velmi zobecněné schéma si můžete prohlédnout na obrázku 1.1.



Obrázek 1.1 Zobecněné schéma redakčního systému.

V současné době je k dispozici nepřeberné množství těchto systémů, a to jak placených, tak volně dostupných. Pro účely této práce jsou zohledňovány jen neplacené systémy, které navíc většinou bývají k dispozici se zdrojovými kódy. Kvalita těchto systémů je i přesto na vysoké úrovni a existuje pro ně velké množství modulů, které rozšiřují funkcionalitu.

Redakční systém uživatelům umožňuje snadno vytvářet, editovat, spravovat a publikovat obsah, bez nutnosti vytváření a editace zdrojových kódů, znalostí databází apod. Administrátorovi pak umožňuje kontrolu a řízení přístupů, jednodušší provádění změn ve vzhledu a obsahu, a celkové nastavení a konfiguraci systému.

Nad použitím redakčního systému pravděpodobně nebudeme uvažovat u malých statických webových stránek. Tedy u webů, kde je pouze malý počet stránek a jejich obsah se příliš nemění. Naopak správa rozsáhlého webu, který má více redaktorů a musí tedy řešit různá oprávnění, a u

kterého se na tvorbě obsahu podílejí i uživatelé (diskuse, ankety, nahrávání souborů, atd.) je bez redakčního systému náročná a téměř nemožná.

Nevýhoda těchto systémů, stejně jako u většiny komplexních nástrojů, je v menší kontrole toho, co systém obsahuje. Může mít vlastnosti, které nepotřebujeme a nevyužijeme. Naopak mohou chybět ty, které bychom potřebovali, ale nejsou k dispozici v samotném systému ani ve formě doplňků. I tento problém je ve většině případů řešitelný vytvořením vlastního doplňku, který nám zajistí potřebnou funkčnost, a propojit ho pak se systémem pomocí API.

2 Analýza a specifikace požadavků

Cílem této kapitoly je analyzovat, sestavit a specifikovat požadavky vedoucího práce na výslednou webovou aplikaci. Neformální požadavky vedoucího jsou volně popsány v následujícím odstavci. Z těchto požadavků byl sestaven níže uvedený diagramu případů užití (obrázek 2.1). Specifikace těchto požadavků najdete v podkapitole 2.3 Detaily případů užití. Jelikož je dobrá analýza a specifikace, spolu s dobrým návrhem, základním předpokladem pro úspěšný softwarový projekt, je této kapitole věnována poměrně rozsáhlá část tohoto dokumentu.

Webová aplikace má sloužit pro pohodlnější práci s dokumenty Microsoft Office Word (formát Word 97-2003). Tyto dokumenty obsahují výsledky analýzy rizik informačních systémů a vhodná bezpečnostní opatření. Aplikace musí být schopna přečíst dokument a získat z něj popis těchto bezpečnostních rizik, které tvoří jen část dokumentu. Následně rozpoznat jednotlivé sekce a oddíly, a vhodně je uložit do databáze. Dokumenty mohou být napsány v různých jazycích, proto musí být umožněn výběr a přiřazení jazykové varianty dokumentu. Aplikace dále musí řešit vyhledávání, a filtrování uložených dokumentů a sekcí. V závislosti na hledaném výrazu pak musí umět vypsat dokumenty, které obsahují hledaný výraz, nebo přímo jednotlivé sekce. V neposlední řadě musí umět i upravovat a mazat dokumenty.

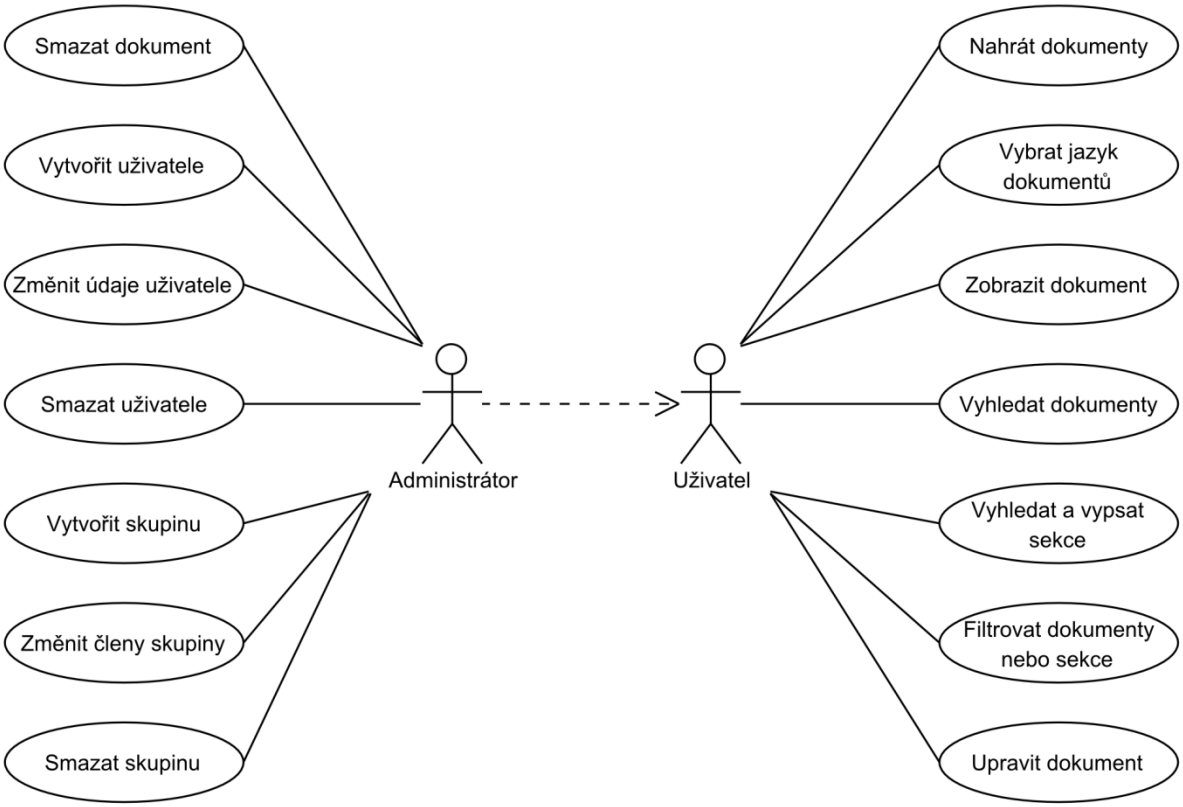
Do aplikace bude uživateli povolen přístup pouze po přihlášení. Aplikace by měla rozlišovat dva typy uživatelských účtů. Běžného uživatele, který bude moci nahrávat, prohlížet a editovat dokumenty. A administrátora, který navíc bude schopný mazat dokumenty, vytvářet a mazat uživatelské účty.

2.1 Uživatelské role

Na základě požadavků na systém od vedoucího práce byly pro tuto aplikaci navrženy dvě uživatelské role. První rolí je uživatel. Uživatel může provádět základní operace v aplikaci jako je nahrávání, vyhledávaní a úpravu dokumentů. Druhou rolí je pak administrátor, který navíc může mazat dokumenty a provádět správu uživatelských účtů a skupin.

2.2 Diagram případů užití

Případy užití představují pohled uživatele na systém z hlediska jeho funkcionality. Slouží k identifikaci hranic systému a definici jeho chování.



Obrázek 2.1 Diagram případů užití

2.3 Detaily případů užití

Následuje specifikace případů užití z obrázku 2.1. Požadavku zahrnující práci s uživatelskými účty a skupinami nebudou specifikovány, protože jsou ve většině případů součástí redakčních systémů.

UC01 - Nahrát dokumenty
Účastníci: Uživatel
Vstupní podmínky: 1. Uživatel je přihlášený do systému.
Posloupnost událostí: 1. Případ užití začíná volbou "Upload documents". 2. Uživatel klikne na odkaz, pomocí kterého se spustí aplikace pro nahrání dokumentů do systému. 3. Systém zobrazí dialogové okno pro výběr dokumentů k nahrání do systému. 4. Uživatel vybere dokument nebo dokumenty, a klikne na tlačítko OK. 5. Systém uloží a zpracuje vybrané dokumenty, a vyzve uživatele ke zvolení jazyka jednotlivých dokumentů. 6. Uživatel vybere jazyk pro jednotlivé dokumenty.
Výstupní podmínky: 1. Dokumenty jsou uloženy v systému.
Alternativní posloupnost: 1. V bodě 4 může uživatel kliknout na tlačítko STORNO a zrušit tak nahrávání dokumentů. 2. Uživatel nemusí vybírat jazyk dokumentů a může se k tomuto kroku vrátit později pomocí položky "Assign language" v menu (viz UC02).

UC02 - Vybrat jazyk dokumentů
Účastníci: Uživatel
Vstupní podmínky: 1. Uživatel je přihlášený do systému.
Posloupnost událostí: 1. Případ užití začíná na záložce "Assign language". 2. KDYŽ všechny dokumenty nahrané uživatelem mají přiřazený jazyk: a. Systém vypíše "Language for all your documents have been assigned." b. Případ užití končí. 3. Systém vypíše dokumenty nahrané uživatelem, které nemají přiřazený jazyk. 4. Uživatel vybere jazyk pro některý dokument. 5. Dokumentu je přiřazen jazyk a je odebrán ze seznamu.
Výstupní podmínky: 1. Všechny dokumenty mají přiřazený jazyk.
Alternativní posloupnost: 1. Uživatel nemusí vybrat jazyk pro všechny dokumenty.

UC03 – Zobrazit dokument
Účastníci: Uživatel
Vstupní podmínky: 1. Uživatel je přihlášený do systému. 2. Dokument je uložený v systému.
Posloupnost událostí: 1. Případ užití začíná na záložce "Browse documents". 2. KDYŽ dokument není v tabulce a. Uživatel vyhledá nebo vyfiltruje dokumenty. 3. Uživatel klikne na požadovaný dokument.
Výstupní podmínky: 1. Systém zobrazí požadovaný dokument a jeho detaily.

UC04 – Vyhledat dokumenty
Účastníci: Uživatel
Vstupní podmínky: 1. Uživatel je přihlášený do systému.
Posloupnost událostí: 1. Případ užití začíná na záložce "Browse documents". 2. Uživatel vybere, v jaké části dokumentu chce hledat. 3. Uživatel zadá hledaný výraz a klikne na tlačítko "Search".
Výstupní podmínky: 1. Systém zobrazí všechny dokumenty, které obsahují hledaný výraz v zadané části dokumentu.

UC05 – Vyhledat a vypsat sekce
Účastníci: Uživatel
Vstupní podmínky: 1. Uživatel je přihlášený do systému.
Posloupnost událostí: 1. Případ užití začíná na záložce "Search documents". 4. Uživatel vybere, v jaké části dokumentu chce hledat. 2. Uživatel zadá hledaný výraz a klikne na tlačítko "Search".
Výstupní podmínky: 1. Systém zobrazí sekce, které obsahují hledaný výraz v zadané části sekce.

UC06 – Filtrovat dokumenty nebo sekce
Účastníci: Uživatel
Vstupní podmínky: 1. Uživatel je přihlášený do systému.
Posloupnost událostí: 1. Případ užití začíná na záložce "Search documents". 2. Uživatel klikne na menu language a vybere jeden ze zobrazených jazyků.
Výstupní podmínky: 1. Systém zobrazí pouze dokumenty, které mají přiřazený vybraný jazyk.
Alternativní posloupnost: 1. Při výpisu sekcí dokumentů, je rovněž možné filtrovat zobrazené sekce podle úrovně zranitelnosti (Risk).

UC07 – Upravit dokument
Účastníci: Uživatel
Vstupní podmínky: 1. Uživatel je přihlášený do systému.
Posloupnost událostí: 1. Případ užití začíná na záložce "Browse documents". 2. KDYŽ dokument není v tabulce a. Uživatel vyhledá nebo vyfiltruje dokumenty. 3. Uživatel klikne na požadovaný dokument. 4. Uživatel klikne na část sekce, kterou chce upravit. 5. Systém zobrazí editovací pole. 6. Uživatel upraví oddíl sekce a klikne na tlačítko Save.
Výstupní podmínky: 1. Systém uloží změnu a zobrazí upravený dokument.

UC08 – Smazat dokument
Účastníci: Administrátor
Vstupní podmínky: 1. Administrátor je přihlášený do systému.
Posloupnost událostí: 1. Případ užití začíná na záložce "Delete documents" 2. Administrátor vybere nebo vyhledá dokument, který chce smazat, a klikne na jeho název. 3. Systém zobrazí výzvu k potvrzení akce. 4. Administrátor potvrdí smazání dokumentu kliknutím na tlačítko YES. 5. Systém zobrazí sekci „Delete documents“.
Výstupní podmínky: 1. Dokument a všechny jeho části jsou odstraněny ze systému.
Alternativní posloupnost: 1. V bodě 3 může administrátor kliknout na tlačítko NO. Dokument pak zůstane beze změny v systému.

3 Výběr redakčního systému

Jak již bylo řečeno v úvodu, existuje mnoho redakčních systémů. Pro implementaci aplikace jsem však vybíral jen ze dvou. Oba spadají do kategorie tzv. free softwaru (jsou zdarma) a u obou jsou veřejně dostupné zdrojové kódy (open source). Prvním je OpenCms 7, vyvíjený firmou Alkacon Software GmbH. Tento systém je založený na technologiích Java a XML. Systém je aktivně vyvíjen od roku 1999 a je distribuován pod licencí GNU LGPL.

Druhým systémem je Drupal 6, jehož autorem je Dries Buytaert. Systém je naprogramovaný v jazyce PHP a šířen pod licencí GNU GPL.

Oba tyto systémy jsou multiplatformní, hojně rozšířené a mají početnou uživatelskou základnu. Hlavní požadavek na výslednou aplikaci je schopnost pracovat s MS Word dokumenty. Konkrétně se jedná o import dokumentů, jejich zpracování, uchování a prohledávání. Z tohoto předpokladu je nutné vycházet a klást na něj největší důraz.

3.1 Zvolený systém

Jako systém pro implementaci webové aplikace byl zvolen redakční systém OpenCms. Jedná se o léty prověřený a stabilní systém. Jeho hlavní výhodou je především knihovna pro práci s textem z různých dokumentů, hlavně pak s dokumenty MS Word.

3.2 Práce s MS Word dokumenty

To, že OpenCms dokáže získat text z různých dokumentů (hlavně Word), je zajisté výhodné. Přesto by nebylo špatné získat i informace o dalších objektech. Zejména pak o vložených obrázcích. Snahou tedy bylo najít bezplatnou externí Java knihovnu, která by dokázala získat i další informace v dokumentu.

Byl nalezen pouze jediný projekt, který se zabývá zpracováním MS Office dokumentů (Word, Excel, PowerPoint a další). Projekt se nazývá Apache POI http://poi.apache.org/ a je podporován organizací Apache Software Foundation. Načítání Word dokumentů je v rané fázi vývoje a neumožňuje získání informací o vložených obrázcích. Navíc knihovny tohoto projektu využívá samotný OpenCms pro prohledávání dokumentů a jsou dostupné pomocí API.

3.3 Analýza binární podoby souborů .doc

Jelikož nebyla nalezena Java knihovna, která by byla schopná získat informace o obrazcích v dokumentech aplikace Word, byla provedena základní analýza binární podoby těchto souborů (v hexadecimálním zobrazení). Každé bezpečnostní riziko v dokumentu (dále je použit výraz sekce), obsahuje položku *skill*, jejíž hodnota je reprezentována pomocí obrázku. Tento obrázek má podobu pěti hvězd, které jsou podle stupně závažnosti vybarveny žlutě (ukázka na obrázku 3.1). Z tohoto důvodu byla provedena analýza se snahou jednoznačně identifikovat obrázek v rámci sekce. Tato informace by byla uložena do databáze a následně použita při výpisu dokumentů. Analýza byla provedena v bezplatném textovém editoru PSPad.

Na obrázcích 3.2 a 3.3 je vidět binární výstup části prvního dokumentu. Ten dokument obsahuje slova PRVNI a DOKUMENT. Přesně mezi slovy PRVNI a DOKUMENT (bez mezer) je v textu vložený ukázkový obrázek se jménem *SKILL_2_TEST.png* viz obrázek 3.1. Obrázek má nastavený alternativní text na hodnotu *SKILL_DVA.png*. Při pohledu na binární podobu souboru v místě, kde se nachází samotný text (adresa 0A00), ovšem informace o obrázku chybí (obrázek 3.2). Název a alternativní text obrázku byly nalezeny až v jiné části souboru (obrázek 3.3).

Obrázek 3.1 Dokument č.1

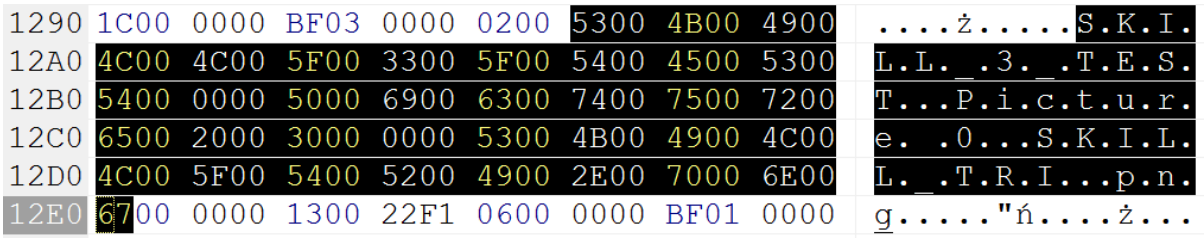
```
09F0 0000 0000 0000 0000 0000 0000 0000 0000  ................
0A00 5052 564E 4901 444F 4B55 4D45 4E54 0D00  PRVNI.DOKUMENT..
0A10 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Obrázek 3.2 Dokument č.1

```
1290 1C00 0000 BF03 0000 0200 5300 4B00 4900  ....ż.....S.K.I.
12A0 4C00 4C00 5F00 3200 5F00 5400 4500 5300  L.L._.2._.T.E.S.
12B0 5400 0000 5000 6900 6300 7400 7500 7200  T...P.i.c.t.u.r.
12C0 6500 2000 3000 0000 5300 4B00 4900 4C00  e. .0...S.K.I.L.
12D0 4C00 5F00 4400 5600 4100 2E00 7000 6E00  L._.D.V.A...p.n.
12E0 6700 0000 1300 22F1 0600 0000 BF01 0000  g.....".ń....ż...
```

Obrázek 3.3 Dokument č.1

Podobná situace je u druhého dokumentu. Ten obsahuje odlišný obrázek se jménem *SKILL_3_TEST.png*. Alternativní text je v tomto případě roven hodnotě *SKILL_TRI.png*. Část výstupu v binární podobě je na obrázku 3.4.



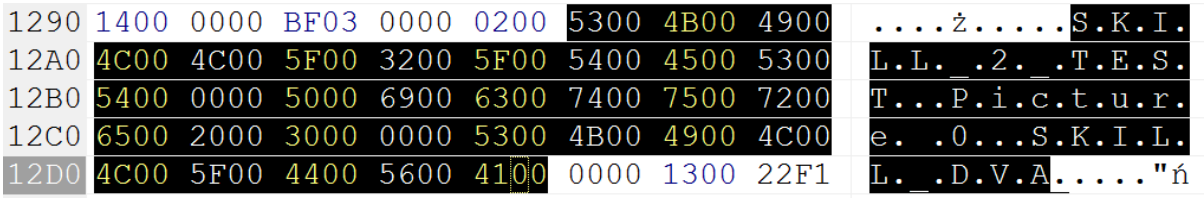
Obrázek 3.4 Dokument č.2

Pokud porovnáme výstupy na obrázcích 3.3 a 3.4, zjistíme, že informace o obrázku a přiřazeném alternativním textu jsou poměrně snadno čitelné. Jak již bylo zmíněno problém je v tom, že se vyskytují v jiné části souboru než samotný text "PRVNI DOKUMENT". Bylo by tedy nutné zjistit, která hodnota určuje polohu obrázku v rámci dokumentu.

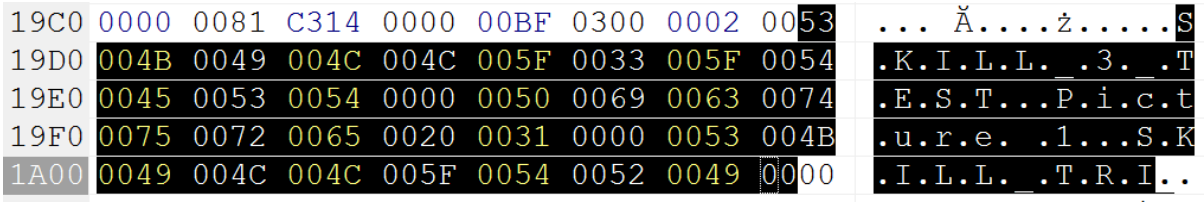
Mohlo by se zdát, že číslo za slovem Picture (umístěné mezi názvem souboru a alternativním textem), určuje pořadí obrázku v dokumentu. A to by mohlo pomoci k určení polohy obrázku. Bohužel tomu tak není. Důkazem jsou následující tři obrázky. Všechny tři pochází z jednoho dokumentu, který obsahoval tři obrázky umístěné za sebou.

Vytvoření dokumentu proběhlo následovně. Nejprve byl vložen první obrázek se jménem *SKILL_2_TEST.png*. Poté následovalo vložení druhého obrázku *SKILL_3_TEST.png*. Nakonec proběhlo označení, zkopírování a umístěním prvního obrázku v rámci dokumentu na jiné místo.

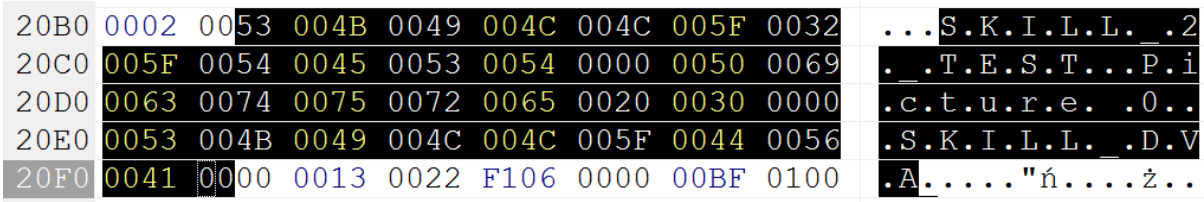
Podoba souboru je pak vidět na obrázcích 3.5, 3.6 a 3.7. Zajímavé jsou hlavně obrázky 3.5 a 3.6, kde za slovem Picture vidíme stejné číslo.



Obrázek 3.5 Dokument č.3

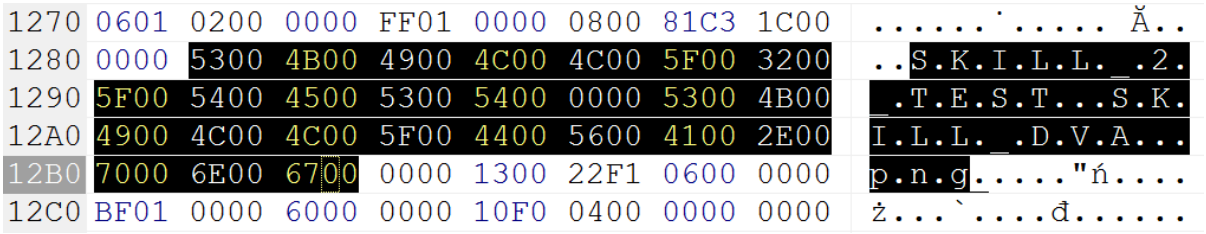


Obrázek 3.6 Dokument č.3



Obrázek 3.7 Dokument č.3

Pokud navíc vložíme obrázek až po uložení a vytvoření souboru, slovo Picture v binární podobě není vůbec přítomno. Viz obrázek 3.8.



Obrázek 3.8 Dokument č.4

Uvedený přístup by zajisté vyžadoval hlubší zamyšlení a zkoumání tohoto problému. A především zvážení, jestli se tento přístup vyplatí. Zvláště když se větší míře začíná používat nový formát Office Open XML (docx atd.).

4 Návrh aplikace

Cílem této kapitoly je vytvořit návrh aplikace. První částí je návrh jednotlivých položek menu a členění aplikace. V druhé části naleznete návrh dráze, ve které budou ukládány všechny podstatné informace o dokumentech a jejich sekcích.

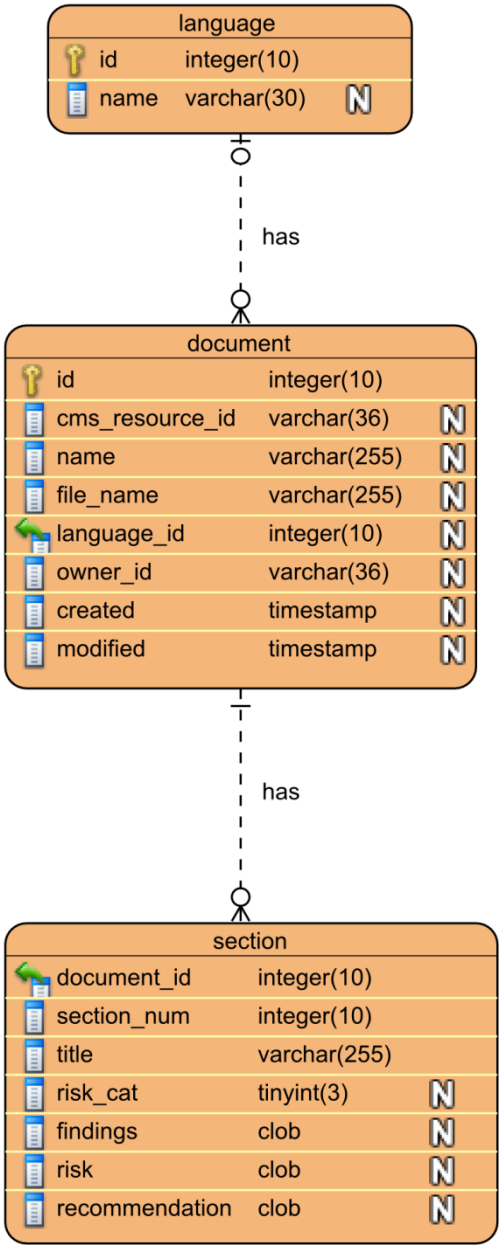
4.1 Členění aplikace

1. Browse document
   - slouží k prohlížení a vyhledávání dokumentů jako celku
2. Search sections
   - Slouží k prohledávání jednotlivých sekcí a jejich filtrování podle, závažnosti a jazyka.
3. Search archive
   - Prohledávání kompletního obsahu původních souborů.
4. Save documents
   - V této části se budou nahrávat dokumenty do systému.
5. Assign language
   - Umožňuje přiřadit jednotlivým dokumentům jazyk, na základě kterého lze dokumenty filtrovat
6. Delete documents
   - Tato položka bude sloužit pouze administrátorovi. Bude v ní moci mazat dokumenty ze systému.

4.2 Entity Relationship Diagram

Následující ER diagram[1] znázorňuje data, které je nutné v systému uchovávat, a vztahy mezi nimi. Obsahuje tři entity. Entita *document* slouží k uchovávání podstatných informací o dokumentu.
Tento dokument může mít 0 – n oddílů. Pro uchování informací o těchto oddílech slouží entita *section*. Poslední entita je pak *language*, ve které jsou uloženy názvy jazyků. Tyto jazyky pak mohou být přiřazeny jednotlivým dokumentům.

[1] Entity Relationship Diagram



Obrázek 4.1 ER diagram

V neposlední řadě je nutné upozornit, že toto jsou objekty, které je nutné uchovávat pro samotnou aplikaci na správu dokumentů. OpenCms využívá vlastní databázi, které je podstatně složitější a rozsáhlejší. Současní verze obsahuje přesně 31 tabulek. Pokud bychom chtěli získat nějaké informace z této databáze, můžeme tak učinit pomocí atributů *cms_resource_id* a *owner_id* v entitě *document*. První atribut *cms_resource_id* jednoznačně identifikuje originálně nahraný dokument ve virtuálním souborovém systému. Druhý atribut pak identifikuje uživatele, který tento dokument do systému nahrál.

4.3 Popis atributů jednotlivých entit

Entita document

- id – jednoznačný číselný identifikátor dokumentu v rámci webové aplikace
- cms_resource_id – identifikátor souboru dokumentu v rámci OpenCms
- name – jméno dokumentu
- file_name – jméno souboru i s příponou v rámci virtuálního souborového systému
- language_id – cizí klíč do tabulky *language*, určuje jazyk dokumentu
- owner_id – identifikátor uživatele, který nahrál dokument do systému
- created – datum a čas nahráni dokumentu do webové aplikace
- modified – datum a čas poslední změny dokumentu v rámci webové aplikace

Entita section

- document_id – cizí klíč do tabulky *document*, který určuje, k jakému dokumentu patří daná sekce
- section_num – pořadí sekce v rámci dokumentu
- title – nadpis sekce
- risk_cat – číselná reprezentace úrovně zranitelnosti
- findings – slouží k uchování textu v oddíle findings, který popisuje nalezenou zranitelnost
- risk – slouží k uchování textu v oddíle risk, který popisuje možná důsledky této zranitelnosti
- recommendation – slouží k uchování textu v oddíle recommondation, který popisuje doporučenou akci k nápravě dané zranitelnosti

Entita language

- id – jednoznačný číselný identifikátor jazyka dokumentu
- name – název jazyka

5 Příprava serveru

V této kapitole jsou popsány nutné kroky pro zprovoznění redakčního systému OpenCms 7 (dále jen OpenCms). Instalace byla provedena na dvou operačních systémech. Jako zástupce platformy Microsoft Windows byl použit operační systém Windows XP Pro SP3 32bit (dále jen Windows). Druhý použitý operační systém byl Ubuntu Desktop 9.10 32bit (dále jen Ubuntu). Jedná se tedy o zástupce platformy GNU/Linux. Tyto systémy byly vybrány především kvůli popularitě, rozšířenosti, a zejména kvůli tomu, že s nimi má autor zkušenosti. Při reálném nasazení bude pravděpodobně zvolen nějaký serverový operační systém. To ale nevadí, protože postup nasazení nebude příliš odlišný.

Proto, abychom mohli začít OpenCms používat, potřebujeme nainstalovat software, který vyžaduje ke svému běhu. Jako modelový příklad byla vybrána trojice Sun Java SDK, Apache Tomcat a MySQL. Následující podkapitoly vysvětlují jednotlivé kroky instalace a čerpají ze zdrojů (1) a (2).

5.1 Hardwarové požadavky

OpenCms nemá zvláštní požadavky na hardware. Minimální doporučená velikost paměti RAM je 512MB. Výkon systému pak závisí hlavně na velikosti paměti RAM, ale také rychlosti CPU. Složité a rozsáhlé webové aplikace s vysokou návštěvností samozřejmě budou vyžadovat výkonnější hardware (3).

5.2 Java SE SDK

Java SDK (Java Software Development Kit) nebo také JDK (Java Development Kit) je soubor základních nástrojů a knihoven určených pro vývoj na platformě Java. Jednotlivé nástroje se ovládají přes příkazovou řádku.

Požadavek: minimálně ve verzi 5 (1.5).

Windows

Byla provedena instalace poslední stabilní verze 6u20, stažená v binární podobě ze stránek společnosti Sun http://java.sun.com/javase/downloads/widget/jdk6.jsp. Samotná instalace probíhá pomocí průvodce. Všechna nastavení a součásti stačí ponechat ve výchozí konfiguraci.

Ubuntu

Je nutné nainstalovat balíček „sun-java6-jdk“ z repositáře.

5.3 Aplikační server

OpenCms ke svému běhu vyžaduje podporu technologií Java Servlet minimálně ve verzi 2.4 a JavaServer pages (JSP) minimálně ve verzi 2.0. Tuto podmínku splňuje například Apache Tomcat od verze 5.0. Apache Tomcat je open source aplikační server, spravovaný již zmiňovanou organizací Apache Software Foundation (ASF). V případě potřeby je možné použít i jiné řešení např. Sun Glassfish, JBoss, Oracle WebLogic, a další. Na těchto serverech může být nutná dodatečná konfigurace (4).

Windows

Byla nainstalována poslední stabilní verze 6.0.26, stažená v binární podobě ze stránek projektu http://tomcat.apache.org/download-60.cgi. Samotná instalace probíhá pomocí průvodce. Všechna nastavení a součásti stačí ponechat ve výchozí konfiguraci.

Ubuntu

Je nutné nainstalovat balíček „tomcat6“ z repositáře.

5.4 Databázový systém

OpenCms ukládá většinu dat do databáze a dokáže pracovat s několika databázovými systémy. Jsou mezi nimi jak komerční, tak neplacené varianty. Mezi podporované systémy patří MySQL, MS SQL Server, Oracle, PostgreSQL. DB2, AS400 a HSQLDB. Pro účely této práce byl vybrán systém MySQL. Ten je podporován alespoň ve verzi 4.0, ale z výkonnostních důvodů je doporučována alespoň verze 5.0.

Windows

Proběhlo stažení a instalace MySQL Community Server v poslední stabilní verzi 5.1.46. Instalace opět probíhá pomocí průvodce, u kterého stačí ponechat vše ve výchozím nastavení. Po instalaci se spustí konfigurační průvodce. V průběhu nastavení je nutné věnovat pozornost zadávání hesla pro uživatele root v části security options. Toto heslo je potřebné při konfiguraci OpenCms.

Ubuntu

Je nutné nainstalovat balíček „mysql-server“ z repositáře. V průběhu instalace se zobrazí výzva k zadání hesla pro uživatele root. Toto heslo je potřebné při konfiguraci OpenCms.

5.5 Shrnutí softwarových požadavků

Tabulka 5.1 shrnuje informace s předchozích podkapitol. Uvádí minimální verze nutné pro běh OpenCms a také verze, které byly použity při vývoji a testování webové aplikace.

Software	Minimální podporovaná verze	Použitá verze	
		Windows XP	Ubuntu 9.10
Java SDK	5 (1.5)	6u20	6.20
Apache Tomcat	5.0	6.0.26	6.0.20
MySQL	4.0	5.1.46	5.1.37

Tabulka 5.1 Přehled použitých technologií a jejich verzí

5.6 Vytvoření databáze a tabulek web aplikace

Pro vytvoření databáze webové aplikace byl vytvořen skript *db_init.sql*. Tento skript zároveň vytvoří dva uživatele. Dva uživatelé jsou vytvořeni kvůli bezpečnosti. Uživatel se jménem *webuser* má nastaveny všechny práva k databázi webové aplikace. Ta má název *ibp*. Druhý uživatel má jméno *cmsuser* a má práva k databázi redakčního systému OpenCms.

Podle návrhu databáze ve třetí kapitole, byl vytvořen druhý skript *db_tables.sql* pro vytvoření jednotlivých tabulek s potřebnými atributy. Skripty lze spustit pomocí utility MySQL pro příkazový řádek, která se jmenuje MySQL Command Line Client. Případně lze doinstalovat jiné, uživatelsky přívětivější nástroje, např. MySQL Workbench nebo phpMyAdmin.

5.7 OpenCms

Postup uvedený v této podkapitole je platný při instalaci výše uvedeného softwaru s výchozím nastavením a do výchozího umístění. Rovněž se předpokládá nasazení na lokálním počítači, proto uváděné adresy směřují na *localhost*.

Při vytváření aplikace byla použita binární verze 7.5.2., kterou je možné stáhnout ze stránek http://www.opencms.org/en/download/. K dispozici je i verze Po stažení je nutné rozbalit archiv. Mezi rozbalenými soubory je soubor opencms.war, který je nutné nakopírovat do adresáře *"C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps"* (ve Windows) a *"/var/lib/tomcat6/webapps"* (v Ubuntu).

V OS Ubuntu je navíc pro spuštění instalačního průvodce nezbytné do souboru *50local.policy* v adresáři *"/etc/tomcat6/policy.d/"* přidat následující příkaz v tabulce 5.2 převzatý z (5):

```
grant codeBase "file:/var/lib/tomcat6/webapps/opencms/-" {
permission java.security.AllPermission;
};
```

Tabulka 5.2 Příkaz pro změnu oprávnění v systému Ubuntu

Po provedení těchto kroků je nutné provést restart serveru Apache Tomcat a po spuštění přejít ve webovém prohlížeči na adresu http://localhost:8080/opencms/setup/, kde bude provedena prvotní konfigurace systému.

5.7.1 Instalační průvodce

Slouží k počátečnímu nastavení a instalaci redakčního systému. Dále jsou uvedeny jednotlivé kroky instalátoru:

1. Odsouhlasení licenčních podmínek.
2. Automatická kontrola nainstalovaných komponent a jejich verzí
3. Výběr databázového systému a zadání údajů pro připojení.
   - v našem případě zvolíme první možnost – MySQL 4.1.x & 5.0.x
   - na řádku Setup Connection je nutné zadat heslo k databázi uživatele root, viz kapitoly 5.4
   - na řádku OpenCms Connection je doporučeno zadat jméno a heslo uživatele, který nemá administrátorská práva k databázi, v našem případě zde zadáme uživatele *cmsuser* a heslo *cmspass*
   - v případě, že databáze běží na jiném počítači nebo portu je nutné adekvátně upravit řádek Connection String, v našem případě ponecháme výchozí hodnotu
   - na řádku database je jméno databáze, výchozí hodnotu opencms neměníme
4. Pokud instalátor najde databázi se stejným jménem, zeptá se, jestli ji má smazat
   - Zvolíme, že ano
5. Automatické vytvoření databáze a tabulek.
6. Výběr modulů OpenCms.
   - webová aplikace **vyžaduje** tyto moduly: Workplace, TemplateTwo a Demo Content for templateTwo.
7. Nastavení URL a jména serveru.
   - ponecháme ve výchozím stavu
8. Import modulů a aplikace nastavení.
9. Zpráva s výsledky instalace.

Pro přístup do administračního rozhraní OpenCms zvaného Workplace je nutné mít ve webovém prohlížeči zapnutý JavaScript a cookies, a povolit pro tuto stránku popup okna. Dále je pro správnou funkčnost aplikace vyžadováno běhové prostředí Java Runtime Environment a mít povolený

odpovídající doplněk ve webovém prohlížeči, kvůli spouštění appletů. Workplace byl oficiálně otestovaný v prohlížečích Internet Explorer (verze 6.0 a novější) a Mozilla (verze 1.5 a novější) viz (6).

5.8 Konfigurace připojení k databázi

OpenCms pro spojení s databází používá z výkonnostních důvodů tzv. connection pooling. Jedná se o způsob práce s databází, kde jsou otevřená připojení uchovávána v tzv. poolu (můžeme si ho představit jako zásobník v paměti). Tato metoda snižuje počet dotazů na server, protože není nutné znovu sestavovat spojení a provádět autorizaci, a zrychluje tak přístup k datům.

Pro připojení do databáze webové aplikace z OpenCms napřed toto připojení musíme nakonfigurovat (nastavit pro něj databázový pool). Konfigurace obsahuje adresu databázového serveru, přihlašovací údaje, jméno databáze, ale také minimální a maximální počet otevřených spojení.

Soubor obsahující konfiguraci pro náš testovací systém se jmenuje *opencms.properties*. Tento soubor je nutné nakopírovat do adresáře WEB-INF\config. Ve Windows ho naleznete zde:
"C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps\opencms\WEB-INF\config"
V Ubuntu pak: *"/var/lib/tomcat6/webapps/opencms/WEB-INF/config"*.

Do stejného adresáře je nutné nakopírovat i soubor opencms-search.xml. Jeho součástí je konfigurace pro vyhledávací index. Tento index slouží k prohledávání původních dokumentů ve složce archivu.

5.9 Import webové aplikace

Pro korektní funkčnost webové aplikace je třeba importovat dva archivy. Archiv *styles.zip* obsahuje definici dodatečných stylů. Tento archiv musí být importován do kořenového adresáře OpenCms (site /). Druhým archivem je *web_application.zip*, ve kterém se nachází samotná webová aplikace. Tento archiv musí být importován do části */sites/default/*.

Import webové aplikace probíhá v administračním rozhraní redakčního systému – Workplace (Administration View), které je na adrese http://localhost:8080/opencms/opencms/system/login/. Defaultní přihlašovací jméno je *Admin* a heslo *admin*. Poté je nutné přejít do sekce Database Management a zde kliknout na položku Import File with HTTP. Pak už stačí naimportovat výše zmíněné soubory do uvedených adresářů. Jakmile je import hotový, je nutné publikovat tyto soubory. To se provede tlačítkem *Publish* v aplikaci Workplace.

6 Implementace webové aplikace

Tato kapitola popisuje jednotlivé nejdůležitější kroky implementace webové aplikace. Funkční stránka aplikace je napsaná v jazyce Java. Grafické rozhranní pak tvoří převážně jazyk HTML a kaskádové styly. Pro některé funkce aplikace je použit Javascript. Při implementaci aplikace byla využívána online API dokumentace OpenCms (7), jako hlavní zdroj informací o dostupných třídách a modulech. Dále pak API specifikace jazyka Java (8)

Samotný proces programování probíhal v administračním rozhranní OpenCms. Integrovaný textový editor podporuje i zvýrazňování syntaxe. Určitě se ale nejedná o nejpohodlnější a nejlepší způsob vývoje. Tento přístup byl zvolen především pro zjištění toho, jak systém funguje. Rovněž kvůli tomu, že byly okamžitě vidět reakce systému na naprogramovanou operaci. Součástí redakčního systému je i jednoduchý WYSIWYG editor, který může být použit při tvorbě jednoduchých webových stránek.

6.1 Nahrávání dokumentů

Pro implementaci nahrávání dokumentů byly zvažovány dva možné přístupy. Použití formuláře a tagu *input* s atributem *type="file"*, anebo integrovaného appletu pro nahrávání souborů. Nakonec byl vybrán integrovaný applet a to hlavně kvůli možnosti uploadu více souborů naráz a možnosti filtrovaní office dokumentů.

Soubory se po nahrání uloženy do složky /IBP/documents, kde čekají na zpracování skriptem processDocs.jsp. Aplet je na tento skript po dokončení uploadu automaticky přesměrován.

6.2 Zpracování Word dokumentů

Po úspěšném přenesení dokumentů do systému musí být dokumenty načteny a zpracovány. Zpracovávají se MS Word dokumenty s příponou doc ve složce *"/IBP/documents/"*, které vytvořil přihlášený uživatel. Načtení a získání textu probíhá pomocí třídy *CmsExtractorMsWord* a metody *extractText()*. Text a případné metainformace jsou pak dostupné pomocí rozhraní *I_CmsExtractionResult* a metody *getContentItems()*, která vrací rozhraní *Map*. Samotný text dokumentu pak lze získat pomocí klíče *ITEM_RAW*.

Mějme tedy ukázkový dokument s jednou sekcí, zobrazený na obrázku 6.1. Zeleně vyznačené části chceme jednotlivě uchovávat v aplikaci.

1.1.1 Strong password is not required L

Risk: LOW Skill: ★★☆☆☆

Findings

The allows the user to change the password. The only requirement for password complexity is its length – 8 characters at minimum, nothing else. The user then can set the password for 8 identical characters – e.g. "00000000".

Risk

Weak passwords can represent the risk for accessing the web application. Even in case of limited attempts, an attacker can find the accounts with weak passwords.

Note: The fact increasing the risk is that authentication process is not two-phased. The user enters only the credentials, there is no second phase such as authorization SMS, GRID or other authorization devices. This second phase of authentication is used usually for internet banking application.

Recommendation

We recommend requiring the strong password by system:

- minimal password length – 8 characters (already required);
- password complexity (numbers, capitals, alphanumeric characters).

Attachments

Example.doc

Obrázek 6.1 Ukázka dokumentu s jedinou sekcí.

Jednotlivé analýzy rizik v dokumentu jsou úspěšně nahrány do systému, pokud splňuje určitá pravidla. Jednotlivá klíčová slova (Risk, Skill, Findings, Recommendation, Attachments), podle kterých se aplikace orientuje, musí být zapsány přesně ve tvaru, jako na obrázku 6.1 (včetně velikosti písmen a dvojteček). Na prvním řádku je titulek rizika, ukončený novým řádkem. Následuje slovo "Risk:", které je rovněž ukončeno novým řádkem. Na obrázku ani v dokumentu to není na první pohled vidět, ale řádek se slovem Risk a Skill je jednořádková tabulka (bez ohraničení) s pěti buňkami. Konec buňky se pak po načtení chová jako konec řádku. Další postup je pak analogický. Výsledek načtení celého dokumentu je vidět na obrázku 6.1.

Po načtení takového dokument do systému, získáme následující výstup:

```
Strong password is not required
Risk:
LOW
Skill:
Findings
The allows the user to change the password. The only requirement for password complexity is its length – 8 characters at minimum, nothing else. The user then can set the password for 8 identical characters – e.g. “00000000”.
Risk
Weak passwords can represent the risk for accessing the web application. Even in case of limited attempts, an attacker can find the accounts with weak passwords.
Note: The fact increasing the risk is that authentication process is not two-phased. The user enters only the credentials, there is no second phase such as authorization SMS, GRID or other authorization devices. This second phase of authentication is used usually for internet banking application.
Recommendation
We recommend requiring the strong password by system:
minimal password length – 8 characters (already required);
password complexity (numbers, capitals, alphanumeric characters).
Attachments
Example.doc
```

Tabulka 6.1 Výstup po načtení dokumentu do systému

Při porovnání originálu a výstupu můžeme vidět, že je ignorováno veškeré formátování, číslování, obrázky, apod. Načteny jsou pouze textové informace.

Jakmile je k dispozici text dokumentu, aplikuje se na něj níže uvedený regulární výraz (tabulka 6.2), který byl vytvořený na základě předchozích znalostí a tutoriálu (9). V aplikaci je použitý regulární výraz s tzv. sytým (reluctant) opakováním. Tento typ opakování je použitý kvůli tomu, aby regulární výraz vyhovoval jen jedné sekci, tedy vyhledává nejkratší možnou shodu. V opačném případě by mohl pokračovat v textu a skončit načtením všech sekcí dokumentu, respektive se zastavit až na posledním z nich. Dále je využito parametru DOTALL. Ten upravuje chování metaznaku '.' (tečka) tak, že vyhovuje i novému řádku.

Aplikace regulárního výrazu na text probíhá pomocí třídy *Matcher* a metody *find ()*. Pokud dojde ke shodě, je text jednotlivých úseků (na obrázku 6.1 jsou vyznačeny zeleně) získán pomocí

metody *group()* a následně uložen do databáze. To se opakuje tak dlouho, dokud se nedojde na konec dokumentu.

```
Pattern.compile("(.*)\\n" +     // matcher.group(1) - nadpis sekce
"Risk:\\n(.*?)\\n" +            // matcher.group(2) - uroven (risk)
"Skill:.*?\\n" +                // Skill
"(?s)" +     // Pattern.DOTALL - metaznak '.' odpovida i konci radku
"Findings\\n(.*?)\\n" +         // matcher.group(3) - findings
"Risk\\n(.*?)\\n" +             // matcher.group(4) - risk
"Recommendation\\n(.*?)\\n" +   // matcher.group(5) - recommendation
"Attachments");
```

Tabulka 6.2 Regulární výraz použitý v aplikaci

Poté co je dokument zpracován, je přesunut do archivu respektive do adresáře */IBP/doc_archive/*.

6.3 Zpracování dokumentů jiných typů

Jako rozšíření bylo implementováno načítání dokumentů jiných typů. Navíc je tedy možné načítat i dokumenty ve formátu Portable Document Format (PDF) a standardní webové stránky s příponou HTML a HTM. Třída pro získání textu z PDF dokumentů se jmenuje *CmsExtractorPdf*. Pro HTML dokumenty pak analogicky *CmsExtractorHtml*. Názvy metod a postup pro zpracování je totožný s popisem v předchozí kapitole. Jediný rozdíl je u zpracování načteného textu u PDF dokumentů. Kvůli rozdílnému výstupu, bylo potřebné mírně modifikovat regulární výraz z tabulky 6.2.

Kvůli chybě v redakčním systému pak nebylo možné implementovat načítání dokumentů programu OpenOffice ve formátu OpenDocument Text (ODT). První problém byl s načítáním nadpisů (styl heading). Veškerý text napsaný v tomto stylu byl ignorován. Dále pak byly ignorovány konce řádků, což by ztížilo, ne-li znemožnilo správné zpracování textu. V případě opravy těchto chyb je ve zdrojovém kódu připravený a okomentovaný úsek, který řeší načítání těchto souborů. Nicméně je nutné počítat s možnými úpravami kódu, zejména pak regulárního výrazu, a následným otestováním výsledku.

6.4 Editace dokumentů

Editace jednotlivých částí dokumentu byla implementována pomocí Javascriptu s použitím tzv. DOM (Document Object Model). Původní verze skriptu je převzata z (10). Pro získání parametru z URL slouží funkce gup získaná z (11). Tato funkce očekává jeden parametr, který specifikuje jméno parametru URL. Pomocí regulárního výrazu je pak z URL získán požadovaný parametr.

Skript funguje na následujícím principu. Po kliknutí myší na webové stránce je zjištěna poloha tohoto kliku. Následně je provedena série několika kontrol (např. nejedná-li se o odkaz, editovací pole atd.). V momentě kdy zjistí, že se jedná o odstavec, který má povolenou editaci (třída edit), provede vytvoření jednotlivých komponent nutných pro editace (textové pole, tlačítko OK a Cancel). Textové pole pak naplní hodnotou, která se nacházela v tomto odstavci. A na pozici kde byl dříve tento odstavec, je umístěno textové pole.

Poté je možné provádět editaci. V případě, že změny nechceme uložit, klikneme na tlačítko Cancel, v opačném případě na OK. Poté dojde k odeslání formuláře a uložení změn do databáze. Editaci dokumentů je záměrně umožněna pouze v části "Browse documents".

7 Testování

Tato kapitola se zabývá testováním výsledné webové aplikace. Účelem testování bylo přijít na chyby, které nebyly zachyceny v rámci implementace. Především se pak se testovala spolehlivost práce s dokumenty. Tedy jestli správně probíhá nahrání dokumentů na server, jejich zpracování a uložení do databáze. Dále pak samotná práce s dokumentu a co se děje v případě zadání chybného vstupu.

Při testování aplikace nebyla použita žádná automatická metoda. Otestování funkčnosti proběhlo vytvořením a nahráním několika testovacích dokumentů v různých formátech na server. Dokumenty byly ve dvou jazykových verzích (v českém a anglickém jazyce).

Pro další ověření funkčnosti aplikace a získání názorů na ovládání a celkovou přívětivost aplikace byl vytvořen dotazník v aplikaci Google dokumenty. Následně jsem oslovil své kamarády, známé a rodinné příslušníky, aby si vyzkoušeli práci s aplikací. Jejich hodnocení a názory pak vyplnili do zmíněného dotazníku.

7.1 Ověření funkčnosti na vzorku dokumentů

V průběhu implementace byly do systému nahrávány různě formátované dokumenty. Na těchto dokumentech byla ověřena správnost importu dokumentů. Ve finální verzi testování byla vytvořena sada různých dokumentů, kdy každý byl uložen do jednoho ze tří podporovaných formátů. Tyto dokumenty byly nahrány do systému. Import proběhl úspěšně a nebyla pozorována žádna chyba.

7.2 Neaktuální data v prohlížeči IE8[2]

Při testování aplikace v prohlížeči Internet Explorer verze 8, docházelo v určitých případech k zobrazování neaktuálních informací. Obsah byl uložen v dočasné paměti webového prohlížeče. Aby se předešlo těmto problémům, pomocí příkazů v tabulce 7.1 bylo zakázáno dočasná paměť u stránek, kde docházelo k problémům.

```
response.setHeader("Pragma", "No-cache");
response.setHeader("Cache-Control", "no-cache");
response.setDateHeader("Expires", 0);
```

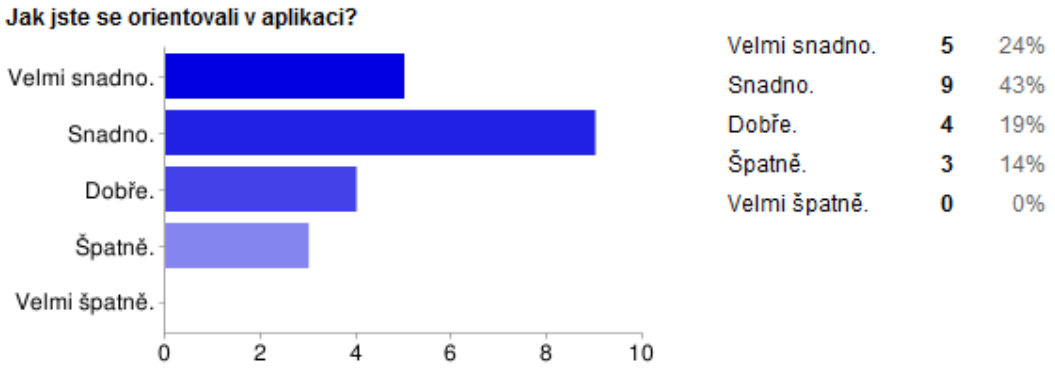
Tabulka 7.1 Příkazy pro vypnutí dočasné (cache) paměti

[2] Microsoft Internet Explorer verze 8

7.3 Názory na ovládání aplikace

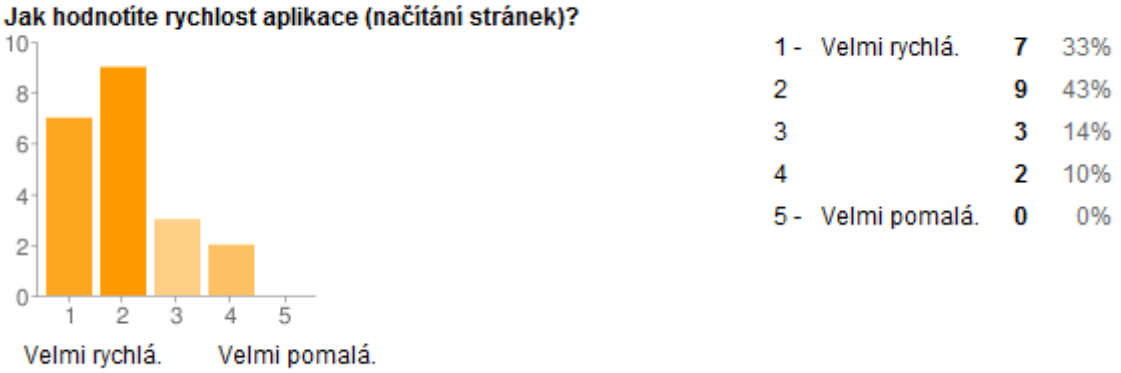
Pro účely otestování aplikace a získání názorů na ovládání aplikace, jednotlivé funkce a uživatelskou přívětivost byl vytvořen dotazník. Do tohoto dotazníku odpovědělo celkem dvacet jedna osob. Jejich odpovědi jsou zobrazeny ve formě grafů níže. Dotazník měl celkem 10 otázek. V závislosti na odpovědi byly zobrazeny jen některé z nich. Tři otázky měly čistě textovou odpověď. Zbytek otázek měl buď jednu možnou odpověď, nebo více možných odpovědí. Výsledky vyvozené z jednotlivých odpovědí budou vždy uvedeny u příslušného grafu.

Na obrázku 7.1 je vidět, že téměř 70% uživatelů se v aplikaci orientovali snadno. Pro tři uživatele pak nebyla orientace v aplikaci příliš jednoduchá. To může být způsobeno tím, že neměli dostatek času na to, aby se s ní naučili pracovat. Dalším důvodem může být uživatelské rozhraní, které je pouze v angličtině.



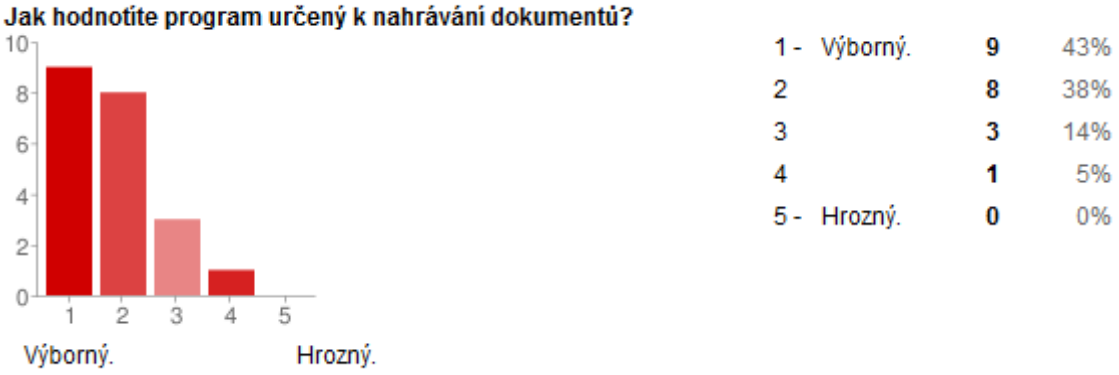
Obrázek 7.1 Obtížnost orientace v aplikaci.

Z následujícího obrázku 7.2 vyčteme, že aplikace působí na uživatele svižně. Jen dvěma uživatelům přišlo načítání stránek pomalejší. Zde mohl být důvod pro horší hodnocení v rychlosti internetového připojení (a to jak na straně serveru, tak na straně uživatele).

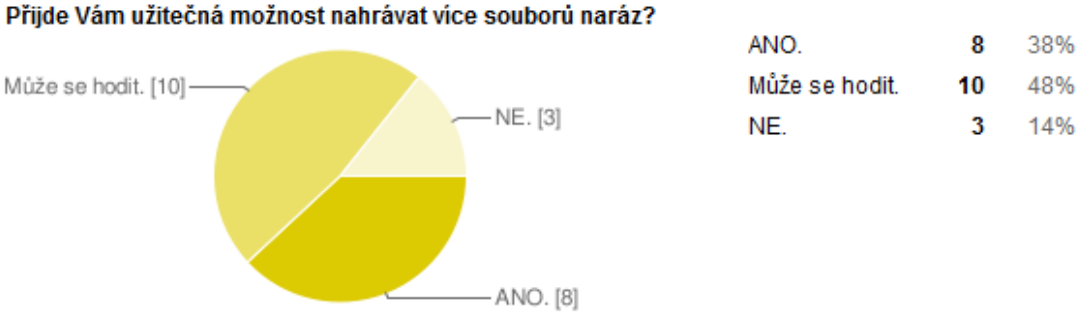


Obrázek 7.2 Rychlost aplikace.

Z grafů na obrázcích 7.3, 7.4 a 7.5 vyplívá, že volba appletu pro nahrávání souborů byla správná. Ani jeden z respondentů neměl problém při nahrávání souboru. Většina pak hodnotila tento applet kladně.



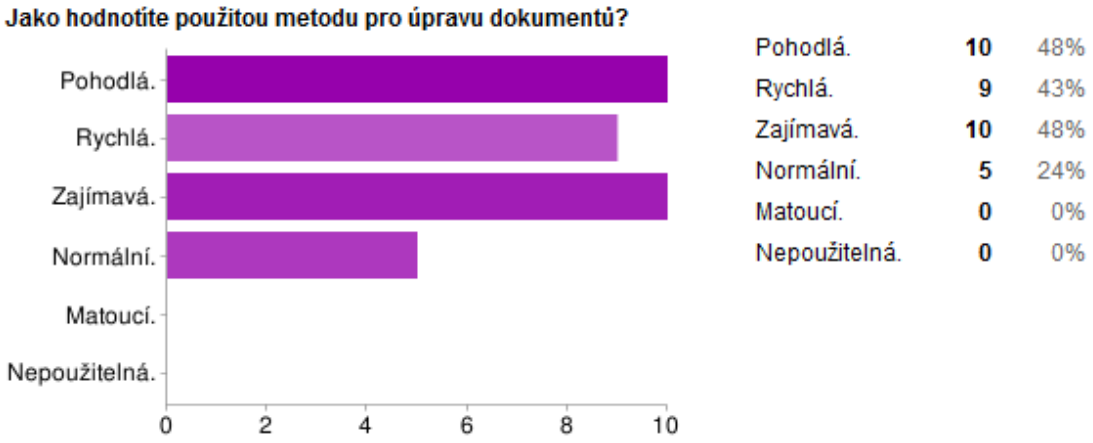
Obrázek 7.3 Applet pro nahrávání dokumentů.



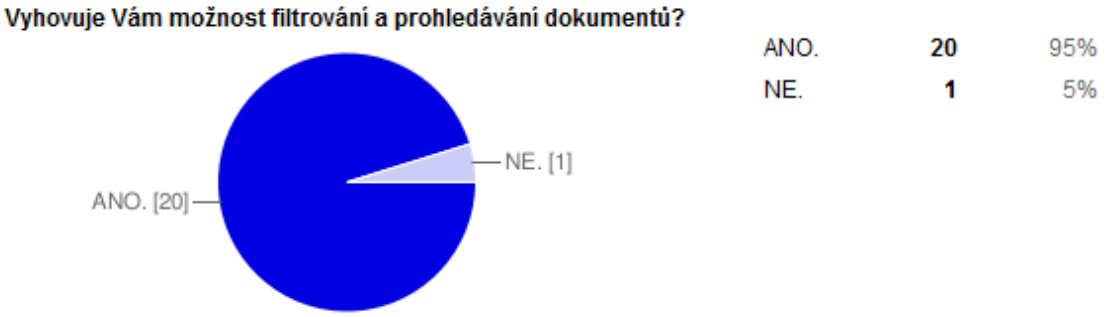
Obrázek 7.4 Nahrávání více dokumentů naráz.

Obrázek 7.5 Problémy při nahrávání dokumentů.

Hodnocení úpravy dokumentů lze vidět na obrázku 7.6. V této otázce šlo zvolit více odpovědí. Pět lidí považovalo použitou metodu jako nijak výjimečnou.



Obrázek 7.6 Funkce pro úpravu dokumentů.



Obrázek 7.7 Filtrování a prohledávání dokumentů.

Na výše uvedeném obrázku 7.7 je poslední netextová otázka. Naprosté většině vyhovovala možnost filtrování a vyhledávání v dokumentech.

Pouze jeden uživatel odpověděl na otázku, kde se odpovídalo slovně. Tento respondent by uvítal možnost zjistit, více o autorovi dokumentu, který byl v systému reprezentován pouze přihlašovacím jménem. Na základě tohoto požadavku pak byla implementována funkce v části pro prohlížení dokumentů ("Browse document"), kde po kliknutí na login autora dokumentu budou zobrazeny další informace. Konkrétně se jedná o jméno, příjmení, email a čas posledního přihlášení do systému.

8 Závěr

Tato kapitola shrnuje dosažené výsledky, dává je do souvislostí s vyslovenými požadavky, zmiňuje praktické využití této aplikace a možný budoucí vývoj.

Cílem této práce bylo vybrat vhodný publikační systém, který by co nejlépe vyhovoval požadavkům na webovou aplikaci, která by ve zvoleném systému byla implementována. Webová aplikace měla umět zpracovávat dokumenty MS Office. Konkrétně tedy uchování, vyhledávání a editaci těchto dokumentů.

Na základě analýzy požadavků ve druhé kapitole, byly definovány a specifikovány další požadavky na aplikaci. Na základě těchto požadavků byl vybrán redakční systém OpenCms.

V zadání práce se můžete dočíst, že měla být implementována i podpora pro dokumenty programu OpenOffice. Od tohoto záměru bylo nakonec upuštěno ze dvou důvodů. Prvním důvodem byla informace od vedoucího práce, že tuto funkci není nezbytně nutné implementovat. Druhým důvodem byla chyba v redakčním systému, která zamezovala korektnímu zpracování dokumentů v tomto formátu. V případě opravení této chyby by bylo možné přidat podporu i pro tento formát.

Jako rozšíření aplikace byla implementována podpora pro zpracování dokumentů ve formátu PDF a HTML. Oba formáty jsou snadno přenositelné a rozhodně představují alternativu k dokumentům ve formátu OpenDocument Text (*.odt) aplikace OpenOffice. Navíc jak MS Word, tak OpenOffice Writer, umí ukládat dokumenty do těchto formátů.

Aplikaci je možné rozšířit o načítání i dalších typů dokumentů. Musela by u nich proběhnout analýza načtených informací a případně i úprava regulárního výrazu, který se stará o zpracování textového vstupu.

Výsledná webová aplikace poměrně přesně splňuje požadavky dané zadáním a také dodatečnou specifikací. Aplikace byla otestována na rozsáhlém vzorku dokumentů, a fungovala bezchybně, pokud měl dokument předepsaný formát.

Budoucí rozšíření mimo již zmíněný import dalších druhů dokumentů, vidím v podrobnějších možnostech vyhledávání a filtrování. Dalším možným rozšířením by mohla být kategorizace podle druhu zranitelností. Systém by také mohl umět export upravených dokumentů do nějakého vhodného formátu (např. pdf nebo HTML).

Redakční systém by také mohl sloužit pro ukládání jiného druhu dat a tvořit tak centrální úložiště. V tomto případě by bylo vhodné, provést analýzu bezpečnosti, aby zde uložená data byla opravdu v bezpečí.

Literatura

1. **Alkacon Software GmbH.** OpenCms 7 server installation. *OpenCms.* [Online] [Citace: 8. května 2010.] http://www.opencms.org/en/development/installation/server.html.

2. —. Finding and installing the prerequisites. *OpenCms.* [Online] 15. října 2008. [Citace: 25. dubna 2010.] http://opencms-wiki.org/Finding_and_installing_the_prerequisites.

3. —. Hardware requirements. *OpenCms Wiki.* [Online] 1. února 2010. [Citace: 25. března 2010.] http://opencms-wiki.org/Hardware_requirements.

4. —. App server related topics. *OpenCms wiki.* [Online] 9. září 2009. [Citace: 2. dubna 2010.] http://opencms-wiki.org/App_server_related_topics.

5. **Nowiasz, Mark.** OpenCMS won't start: Could not initialize class org.opencms.i18n.CmsLocaleManager. *OpenCms Forum.* [Online] 16. listopadu 2009. [Citace: 3. února 2010.] http://www.opencms-forum.de/opencms-forum/viewthread;jsessionid=E4BF74C9E54F3ECC3AD4B2056BAD06B2?thread=3763#9139.

6. **Alkacon Software GmbH.** OpenCms 7 client setup. *OpenCms.* [Online] [Citace: 11. května 2010.] http://www.opencms.org/en/development/installation/client.html.

7. —. OpenCms Core API, version 7.5.2. *OpenCms.* [Online] [Citace: 27. února 2010.] http://files.opencms.org/javadoc/core/index.html.

8. **Oracle Corporation.** Java 2 Platform Standard Edition 5.0 API Specification. *Developer Resources for Java Technology.* [Online] [Citace: 11. března 2010.] http://java.sun.com/j2se/1.5.0/docs/api/.

9. —. Lesson: Regular Expressions. *The Java™ Tutorials.* [Online] 12. ledna 2010. [Citace: 7. února 2010.] http://java.sun.com/docs/books/tutorial/essential/regex/.

10. **Koch, Peter-Paul.** JavaScript - Edit text. *QuirksMode.* [Online] [Citace: 3. května 2010.] http://www.quirksmode.org/dom/cms.html.

11. **lobo235.** Get URL Parameters Using Javascript. *Netlobo.com.* [Online] 17. srpna 2006. [Citace: 3. května 2010.] http://www.netlobo.com/url_query_string_javascript.html.

Seznam příloh

Příloha 1. CD se zdrojovými texty, konfiguračními soubory a tímto dokumentem