Masarykova univerzita
Fakulta informatiky

Portál digitálního kulturního dědictví

Bakalárska práca

Peter Koza

Brno, jar 2016

Namiesto tejto stránky vložte kópiu oficiálneho podpísaného zadania práce a prehlásenie autora školského diela.

Prehlásenie

Prehlasujem, že táto bakalárska práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Peter Koza

Vedúci práce: RNDr. Jaroslav Ráček, Ph.D.

Poďakovanie

Ďakujem RNDr. Jaroslavovi Ráčkovi, PhD. za vedenie mojej bakalárskej práce.
Ďakujem mame, kolegom a priateľom za podporu počas tvorby praktickej časti a písania práce.

Zhrnutie

Cieľom tejto bakalárskej práce je popísať princípy digitalizácie a publikovania objektov kultúrneho dedičstva. Je potrebné previesť analýzu požiadaviek na základe stretnutí so zákazníkom, navrhnúť konečnú podobu systému a implementovať celý systém. Dôraz je kladený na portálové technológie.

Kľúčové slová

digitalizácia dokumentov, sprístupňovanie dokumentov, webový portál, kultúrne dedičstvo, indexácia popisných dát

Obsah

Zoznam obrázkov

1 Úvod

Táto práca sa zaoberá elektronickým spracovaním a zverejnením kultúrneho dedičstva. V súčasnej dobe je to stále väčší trend v zverejňovaní obsahov múzejných zbierok a zbierok ďalších kultúrnych inštitúcií. Kvôli povahe dát sa ako najjednoduchšie javí použitie portálu na zverejňovanie knižných záznamov, ale je tiež možné evidovať obrazy, sochy, zbrane alebo hudobné nástroje. V tomto prípade sa zaoberáme portálom, ktorý sprístupňuje viac zbierok na jednom mieste a rieši nie len sprístupňovanie dokumentov, ale aj integráciu s ďalšími portálmi. Typickým vlastníkom môžu byť mestá, kraje alebo celoštátne inštitúcie ako NPÚ [1]. V našom prípade bol vyvíjaný pre koncového zákazníka, ktorým je jeden z krajov Českej republiky. Vieme si ale predstaviť, že podobný portál by mohol byť vhodný pre cirkev, archívy alebo univerzity.

Práca je rozdelená do piatich kapitol. Prvá opisuje tvorbu portálov kultúrneho dedičstva. Druhá sa venuje analýze požiadavkov zákazníka. Tretia približuje technológie využité pri implementácii, ktorá je popísaná poslednou kapitolou.

1. NPÚ - Národní památkový ústav

2 Portály digitálneho kultúrneho dedičstva

Zámerom tejto iniciatívy je vytvoriť webový portál, ktorého prostredníctvom bude zaisťovaná príprava a samotné sprístupnenie digitálneho obsahu vybraných fondov pamäťových inštitúcií pôsobiacich na území Českej republiky širokej verejnosti a odborným bádateľom. Ďalej bude umožňovať on-line úpravu metadát a možnosť priloženia ďalších materiálov ktoré s dokumentom súvisia. Web by mal užívateľa zaujať a vytvoriť dojem jednoducho používateľnej interaktívnej stránky aj u používateľa, ktorého daná tématika nezaujíma. Digitálny obsah sa primárne skladá z kníh. Problémom existujúcich nástrojov na prehliadanie tohto obsahu je nepríťažlivosť pre užívateľa.

Tabuľkové zobrazenie knižných záznamov nespĺňa požiadavky modernej webovej aplikácie. Príkladom je katalóg Slezského zemského múzea[1]. Vyhľadávanie je realizované jedným formulárom. Výsledky sú zobrazované jednoduchou tabuľkou, ktorá poskytuje minimum informácií o dokumente a jeho uložení. V tabuľke sú ako prvé uvedené informácie Dok a Sign, ktorých názvy sú neintuitívne a teda pre koncového užívateľa pri prvom prístupe na web nepodstatné. Ďalej sú zobrazené údaje autor, názov, časť, rok a počet. Zvyšok údajov je zobrazený až po kliknutí na názov dokumentu. Ostatné atribúty sú neaktívne. V detaile dokumentu chýbajú užívateľsky príťažlivé prvky, ako napríklad mapa uloženia, diskusia k dielu, možnosť rezervácie alebo zobrazenie skenovaných obrázkov. Aj napriek tomu, že tento web podáva hodnoverné informácie o dokumentoch, nie je možné ho označiť za užívateľsky príťažlivý a po prvom použití nenavádza užívateľa k ďalšej návšteve.

2.1 Realizácia vyhľadávania

Na stránke systému Kramerius[2] je sprístupnené vyhľadávanie dokumentov Národnej knižnice Českej republiky. Užívateľské rozhranie je oproti predošlému webu obohatené o funkciu hľadania v celom texte, ktorá umožňuje jednoduchšie zoznámenie sa s funkciami aplikácie. Pokročilé vyhľadávanie je možné zobraziť a skryť pridaným tlačidlom.

1. Viď `http://knihovna.szmo.cz/katalog/`.
2. Viď `http://kramerius.nkp.cz/kramerius/Welcome.do`.

Priamo na úvodnej stránke sú umiestnené príklady skenovaných dokumentov, ktoré odkazom vedú na detail diela. Zobrazenie výsledkov vyhľadávania je riešené jednoduchým zoznamom. Tento prístup prináša vysokú mieru neprehľadnosti, pretože užívateľ nemôže porovnať hodnoty, či zoradiť výsledky na základe jednotlivých atribútov. Detail obsahuje stručné informácie o diele a náhľad strán, ktorý ale nie je dostupný bez inštalácie zásuvného modulu.

2.2 Smerovanie portálov kultúrneho dedičstva

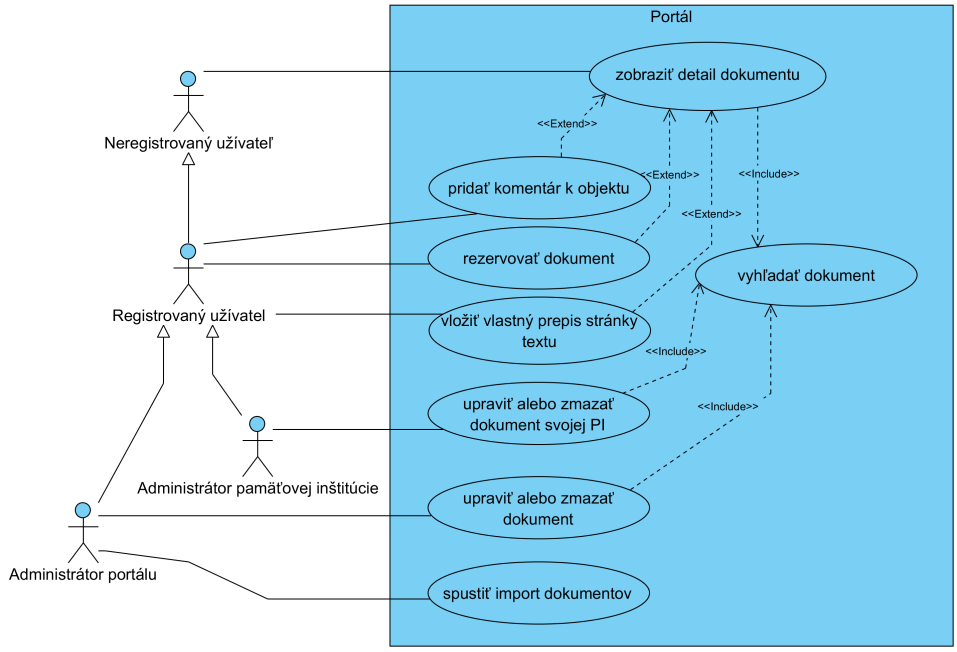
Z analýzy existujúcich nástrojov na prehliadanie kultúrneho dedičstva bolo zistené, že pri návrhu aplikácie by sme sa mali zamerať na nasledujúce oblasti:

- vyhľadávanie
  - umožniť hľadanie v celom texte
  - pridať prvky pre oživenie ako napr. mapa alebo rezy
- zobrazenie výsledkov
  - vytvoriť prehľadnú tabuľku obsahujúcu relevantné atribúty
  - v prípade použitia mapy poskytnúť možnosť jednoduchého prepnutia
- detail diela
  - pridať užívateľsky príťažlivé prvky
    - mapa umiestnenia
    - diskusia k dielu
    - zobrazenie náhľadu dokumentov
    - rezervácia knihy

3 Analýza

3.1 Požiadavky zákazníka

Po komunikácii so zákazníkom na analytických stretnutiach boli zistené požiadavky zobrazené diagramom na obrázku *3.1*.[1],[2]



Obr. 3.1: *Prípady použitia PD*

3.1.1 Kategórie dokumentov

Dokumenty, ktoré sú predmetom prípravy a zverejnenia prostredníctvom portálu digitálneho kultúrneho dedičstva (ďalej už len PD) sú rozdelené do dvoch základných kategórií:

- dokumenty archívnych fondov a múzejných zbierok
- dokumenty knižničných fondov podľa štandardov NDK[1]

Pre vyššie uvedené jednotlivé kategórie je nutné použiť v PD rozdielne pracovné postupy a to hlavne v prípravnej fáze pred zverejnením.

1. NDK - Národní digitální knihovna

Dôvodom je samotná podstata dokumentu z pohľadu toho, či ide o dokumenty, ktoré sú v nemennej podobe (monografie, periodiká) alebo dokumenty, u ktorých sa v čase môžu meniť predovšetkým ich popisné informácie. Pre každú kategóriu sú taktiež definované odlišné dátové štruktúry metadát. Predpokladaný objem dokumentov spravovaných PD je 100 000 jednotlivých dokumentov, 1 500 000 strán prepočítaných na formát A4 a približne 20 terabajtov obrazových dát.

3.1.2 Dokumenty archívnych fondov a múzejných zbierok

Druhy dokumentov:

- fotografie, negatívy
- zväzky - kroniky, zápisy, katalógy
- voľné archy - dokumenty, plagáty, plány
- staré výtlačky, spevníky
- kartografický materiál
- filmy, zvukové nahrávky
- zbierkové predmety

Jedná sa o dokumenty, ktoré vznikli digitalizáciou vybraných archívnych fondov a múzejných zbierok. Z hľadiska nemennosti týchto dokumentov sa jedná o dokumenty premenlivej povahy. Popisné metadáta z KDJ[2] pre PD u tejto kategórie preto nie je možné použiť. Jedná se teda o dokumenty, u ktorých je nutné ručné doplnenie metadát.

3.1.3 Dokumenty knižničných fondov podľa štandardov NDK

Druhy dokumentov:

- monografie - jednodielne alebo viacdielne knižné dokumenty
- periodiká - pravidelne vychádzajúce výtlačky

2. KDJ - Krajská digitalizační jednotka

Jedná sa o dokumenty prevažne nemennej povahy. Predpokladá se teda, že ako uložené digitálne obrazy, tak metadáta majú konečný a nemenný stav. Všetky tieto dokumenty sú uložené v úložisku KDJ vo formáte SIP. KDJ bude pre tento typ dokumentov zdrojovým poskytovateľom informácií pre PD.

3.1.4 Typy dokumentov

Z hľadiska typu súborov, v ktorých sú digitálne dokumenty uložené, se jedná o dokumenty:

- obrazové
- videá
- zvukové záznamy
- vo formáte XML (popisné metadáta)
- vo formáte PDF

3.1.5 Vstup dokumentov

Táto časť PD bude zabezpečovať riadený import dát pre prípravu sprístupnenia dokumentov. Vstupné mechanizmy budú rešpektovať potreby jednotlivých kategórií dokumentu. Po importe budú dáta pripravených dokumentov uložené do zverejňovacej databázy. Princíp predávania dokumentov, detailný popis rozhrania a jeho dátová štruktúra budú súčasťou implementačnej analýzy. Zákazník špecifikoval nasledujúce podmienky:

- PD bude umožňovať import viacerých dokumentov súčasne.
- Pri zahájení prípravy vstupu dokumentu do PD bude prevedená kontrola duplicity s predchádzajúcimi importovanými dokumentmi.
- Behom importu dokumentu do PD systém priradí jeho vlastníka – osobu kompletne zodpovednú za celé sprístupnenie a zverejnenie. Vlastníkom se stáva užívateľ PD, ktorý import úspešne dokončil. Zdrojom dát pre import bude mimo iných

webové rozhranie PD a zdieľané diskové zložky unikátne pre každého administrátora.

- PD bude disponovať funkciou pre import a aktualizáciu popisných metadát a prípadných obrazových dát zo systému JANUS2000 (evidenčný systém okresných archívov) a systému VISMO (dáta projektu *MG on-line*).

3.1.6 Príprava dokumentov k sprístupneniu

Táto časť PD bude zahŕňať proces sprístupnenia dokumentov a možnosť ich úpravy pred zverejním. Zákazník špecifikoval nasledujúce funkcie:

- PD po prihlásení do časti pre prípravu dokumentu ponúkne oprávnenému užívateľovi zoznam všetkých jeho dokumentov s informáciou o ich stave a možnosťou zverejnenia dokumentu pre všetkých užívateľov.

- PD umožní oprávnenému užívateľovi zobraziť náhľad upraveného dokumentu, znázorňujúci stav, v ktorom bude dokument zverejnený.

- PD umožňuje oprávnenému užívateľovi kedykoľvek zrušiť zverejnenie dokumentu.

- Oprávnený užívateľ, vlastník dokumentu, má na karte otvoreného dokumentu k dispozícii históriu úprav.

- Portál PD umožní oprávnenému užívateľovi možnosť úpravy popisných dát podľa úrovne jeho oprávnení.

- Štruktúra dokumentu bude navrhnutá podľa štandardu Dublin Core.

- Všetky dôležité akcie PD ako sprístupnenie, úprava alebo zmazanie dokumentu budú sprevádzané notifikačným e-mailom pre všetkých dotknutých užívateľov.

3.1.7 Operácie nad zverejnenými dokumentmi

Táto časť portálu obsahuje funkcie dostupné pre širokú verejnosť, tzn. registrovaných a neregistrovaných užívateľov. Zákazník špecifikoval nasledujúce funkcie, dostupné všetkým užívateľom:

- PD bude poskytovať možnosť vyhľadávania v celom texte.
- PD bude poskytovať možnosť vyhľadávania podľa konkrétnych atribútov.
- PD umožní export obsahu príslušného dokumentu do PDF súboru.

Nasledujúce funkcie budú dostupné všetkým registrovaným užívateľom:

- Možnosti pre užívateľský popis príslušného dokumentu. Ide o diskusiu k dielu a prepis nerozpoznaného textu dokumentu.
- PD umožní rezerváciu diela pre vypožičanie.

3.2 Návrh riešenia

Na základe analýzy existujúcich nástrojov na prehliadanie digitálnych fondov a komunikácie so zákazníkom bol vytvorený konečný návrh projektu. Pre implementáciu PD bude využité open source portálové riešenie *Liferay* vo verzii *6.2.2 community edition GA3*. Prípadný prechod na vyššiu verziu je možný prostredníctvom štandardných postupov pre portál Liferay a nieje zahrnutý v rozsahu tohto projektu. Zaznamenávanie operácií v PD bude zabezpečené pomocou frameworku pre Java aplikácie *Log4j*. Výstupné hlášky sú štandardne zaznamenávané do súboru na aplikačnom serveri, na ktorom aplikácia beží. Jednotlivé hlášky budú rozdelené do úrovní. Nastavovanie týchto úrovní bude možné prostredníctvom administračného rozhrania portálu Liferay. Administrácia zverejňovania digitálneho obsahu bude dostupná všetkým oprávneným užívateľom. Vyhľadávanie bude dostupné všetkým užívateľom bez ohľadu na ich identitu.[3]

3.2.1 Grafický vzhľad

Grafický vzhľad aplikácie bude meniteľný pomocou špeciálnych zásuvných modulov podporovaných portálom Liferay (tzv. témy). Téma bude využitá naprieč všetkými časťami portálu. Tento prístup zabezpečuje jednotný dizajn aplikácie. Zmena vzhľadu vyžaduje vývoj alebo úpravu zásuvného modulu.

3.2.2 Komunikácia s externými systémami

Pre komunikáciu s externými systémami budú použité následujúce technológie:

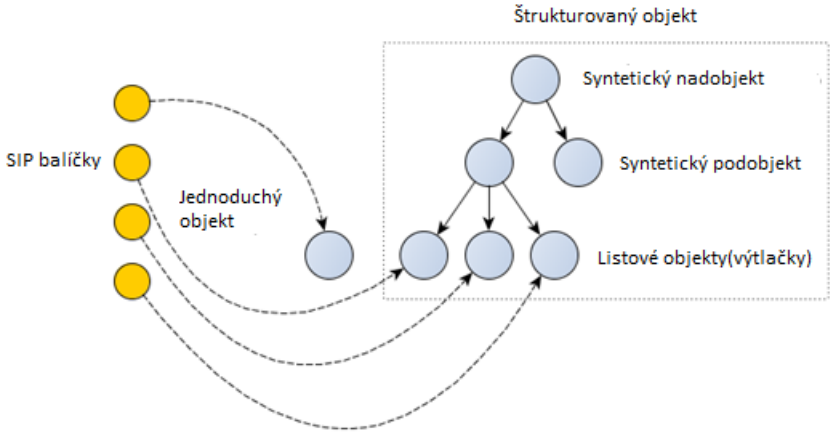
- *Protokol Z39.50* – pre komunikáciu s pamäťovými inštitúciami kvôli aktualizácii metadát dokumentov
- *Protokol OAI-PMH* – pre poskytnutie rozhrania, na zbieranie metadát v databáze externými aplikáciami

3.2.3 Digitálny objekt

Digitálny objekt je štruktúra, ktorá vstupuje na vonkajšom rozhraní PD do importovacej dávky. Táto štruktúra bude v priebehu importovacieho procesu spracovaná a transformovaná do vnútornej štruktúry PD. Načítané digitálne objekty budú na základe svojich vlastností a zdrojových systémov delené do niekoľkých podskupín. Na delenie typov je možné nahliadať z dvoch rôznych perspektív.

- Z pohľadu štruktúry dát
- Z pohľadu vlastného obsahu digitálneho objektu

Digitálne objekty importované zo systému SIRIUS sú fyzicky reprezentované pomocou SIP balíčkov. A práve tieto SIP balíčky odpovedajú v prípade systému SIRIUS jednotlivým typom objektov vnútornej štruktúry dát PD.



Obr. 3.2: *Príklad štrukturovaného dokumentu s tromi úrovňami*

3.2.4 Typy objektov z hľadiska štruktúry

Jeden SIP balíček bude transformovaný do práve jedného listového objektu (v prípade štrukturovaného objektu) alebo do jedného jednoduchého objektu. Napríklad v prípade monografie je to priamo vlastný digitálny objekt. V prípade periodika je SIP balíček listový objekt, ktorý sa stane súčasťou štrukturovaného objektu a bude zaradený ako digitálny objekt na jeho najnižšej úrovni. Z dát v listových objektoch bude možné zložiť celú štruktúru hierarchického digitálneho objektu pretože každý SIP balíček nesie informácie o celom štrukturovanom objekte a je možné príslušný objekt z týchto informácií vytvoriť. Dáta štrukturovaného objektu sú teda uložené redundantne v niekoľkých listových objektoch. Či sa jedná o jednoduchý objekt alebo sa jedná o štrukturovaný objekt a typ dokumentu z pohľadu obsahu bude určovať algoritmus rozoznania typu objektu.

3.2.4.1 Jednoduchý objekt

Jednoduchý objekt znamená, že v portáli budú uložené dáta o danom digitálnom objekte bez možnosti jeho prechádzania po úrovniach. V tomto prípade môžeme jeden digitálny objekt označiť za ekvivalent jedného SIP balíčku. Z pohľadu obsahu sa jedná o nasledujúce typy:

- Monografia
- Zbierkový predmet
- Rukopis
- Veduta

3.2.4.2 Hierarchický objekt

Hierarchický alebo štrukturovaný objekt je na najnižšej úrovni zložený z listových objektov a ďalších syntetických objektov. Jednotlivé úrovne ako aj vlastný koreňový objekt bude vždy možné identifikovať zo všetkých listových objektov. Z pohľadu obsahu se jedná o nasledujúce typy:

- Dvojúrovňové objekty
  - Viaczväzková monografia
  - Kronika
  - Fotografia
  - Mapa
  - Listina
- Trojúrovňové objekty
  - Periodikum
  - Úradný zápis

3.2.4.3 Listový objekt

Listový objekt je základným stavebným prvkom štrukturovaného objektu. Na rozdiel od jednoduchého objektu je vždy sprevádzaný svojim syntetickým nadobjektom. Tento syntetický nadobjekt je vždy čitateľný zo všetkých SIP balíčkov, ktoré sú uložené do objektu na najnižšej úrovni. Pokiaľ bude v štrukturovanom objekte iba jeden listový objekt, bude ho portál stále zobrazovať ako viacúrovňový objekt.

3.2.5 Typy objektov z hľadiska obsahu

Z hľadiska obsahu objektov rozoznávame niekoľko typov. U jednotlivých typov obsahu je vždy potrebné rozhodnúť aké atribúty budú načítané zo SIP balíčkov do vnútornej štruktúry PD. Všetky relevantné atribúty budú zanesené do vyhľadávacieho indexu.

Pre jednoznačné odlíšenie digitálnych objektov budú využité atribúty generované systémom SIRIUS, ktoré budú zaisťovať ich jedinečnosť. Tieto atribúty môžu byť v závislosti od jednotlivých typov objektov rôzne. Podľa identifikátoru môžeme dokumenty rozdeliť do nasledujúcich skupín:

- Parameter *ČČNB* - monografie, viaczväzkové monografie, periodiká
- Parameter *DIGIKUJIF* - fotografie
- Parameter *DIGIKUJIM* - mapy
- Parameter *DIGIKUJIK* - kroniky
- Parameter *DIGIKUJIL* - listiny
- Parameter *DIGIKUJIR* - rukopisy
- Parameter *DIGIKUJIV* - veduty
- Parameter *DIGIKUJIZ* - úradné zápisy

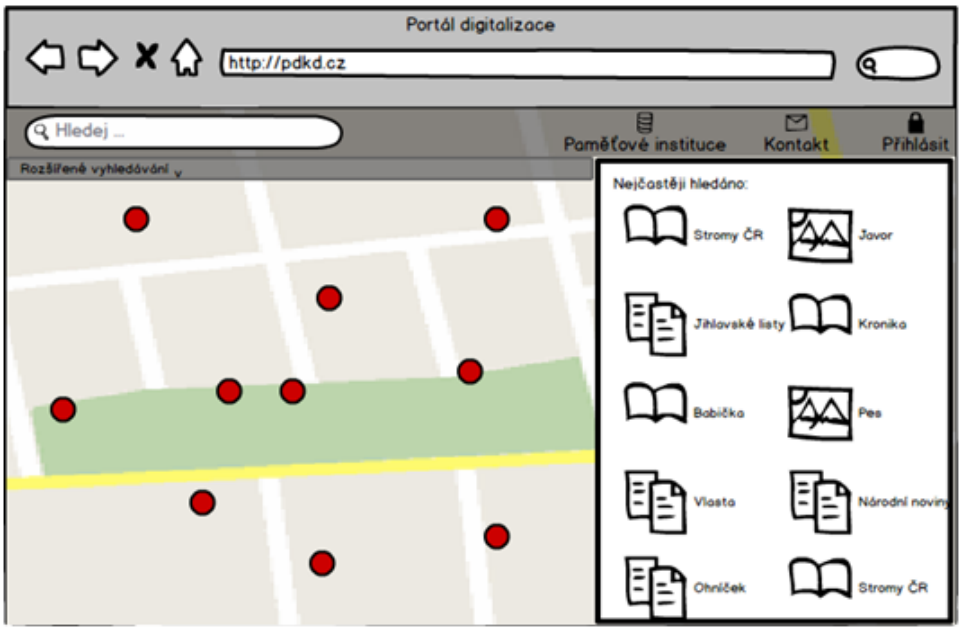
3.2.6 Návrh dizajnu

Táto kapitola popisuje dizajn aplikácie a funkcie dostupné užívateľovi v jednotlivých sekciách. Pre lepšiu predstavu, ako bude popisovaná funkcionalita vyzerať vo výslednom systéme, je pri prípadoch použitia uvedený nákres užívateľského rozhrania (tzv. mockup).

3.2.6.1 Úvodná stránka portálu

Úvodná stránka PD bude obsahovať čo najmenej informácií v záujme čo najväčšej prehľadnosti a jednoduchosti pre užívateľa. Konkrétne bude obsahovať:

- vyhľadávací formulár
- najčastejšie hľadané objekty
- odkaz na kontaktné údaje
- odkaz na informácie o jednotlivých pamäťových inštitúciách
- odkaz na autorizované operácie pre prihláseného užívateľa
- mapu umiestnenia dokumentov



Obr. 3.3: *Nákres úvodnej stránky*

3.2.6.2 Stručný detail objektu

Stručný detail objektu sa zobrazí pri kliknutí na konkrétny vyhľadaný objekt. Stručný detail nebude zobrazený cez celú plochu okna prehliadača, ale iba na polovici okna. Druhá polovica bude obsahovať mapu s bodmi predstavujúcimi miesta vydania, uloženia, či miesta, ku ktorým sa objekt obsahovo vzťahuje.

Obr. 3.4: *Nákres stručného detailu objektu*

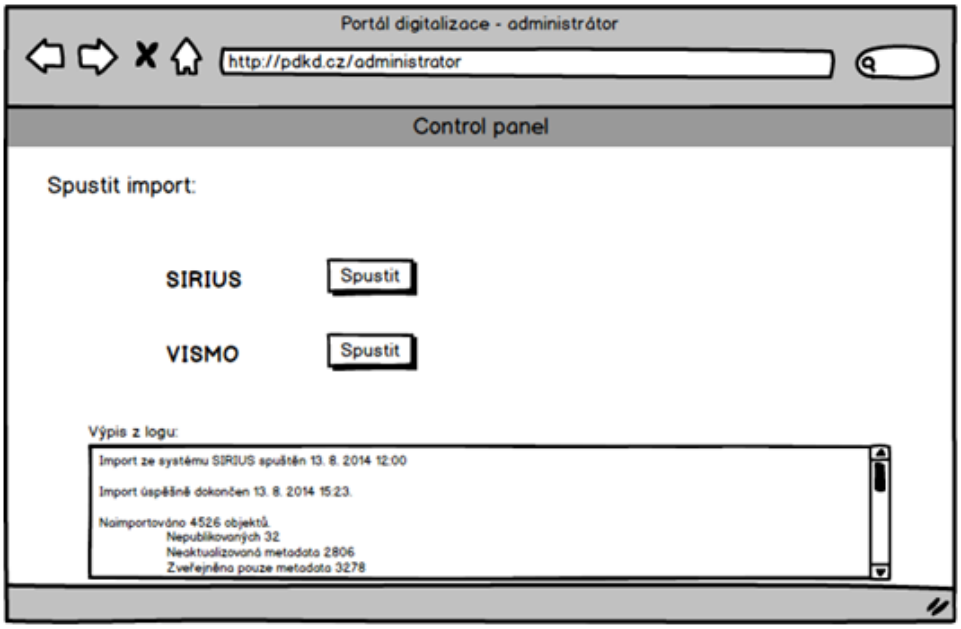
3.2.6.3 Plný detail objektu

Úplný detail objektu sa zobrazí po kliknutí na tlačidlo *Zobraziť plný detail* v stručnom detaile objektu. Plný detail bude zobrazený cez celú plochu okna prehliadača a bude zobrazovať všetky informácie o danom objekte. Z plného detailu bude možné tento detail vytlačiť a exportovať do PDF. Rovnako bude možné vytlačiť a exportovať do PDF samotný objekt. Plný detail bude zobrazený na adrese, ktorá bude vo formáte permalinku. To znamená, že pri skopírovaní adresy z adresného riadku prehliadača do nového okna, bude zobrazený ten istý detail pôvodne zobrazeného objektu. Plný detail objektu bude tiež obsahovať napojenie na štandardné Facebook API, umožňujúce zdieľanie stránky.

Obr. 3.5: *Nákres plného detailu objektu*

3.2.6.4 Rozhranie pre import dát zo systému Sirius a Vismo

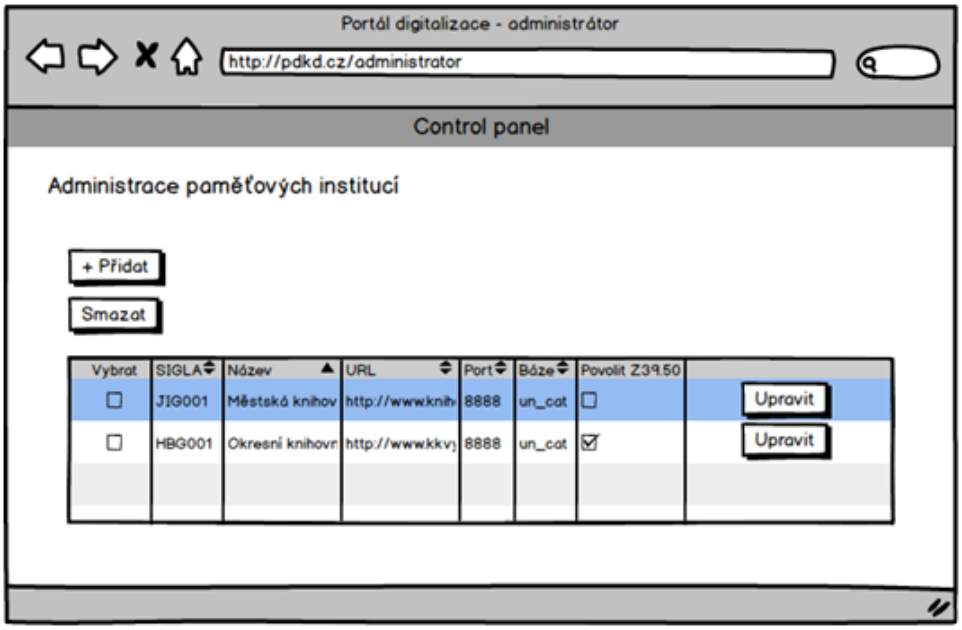
V príslušnej aplikácii v administračnom rozhraní užívateľ klikne na tlačidlo *Spustit* pri položke *SIRIUS* alebo *VISMO*. Import je zahájený a do výpisu v spodnej časti obrazovky sa vypisujú informácie o prebiehajúcom importe.



Obr. 3.6: *Nákres administračného rozhrania pre import dokumentov*

3.2.6.5 Správa pamäťových inštitúcií

Oprávnený užívateľ bude mať možnosť v administračnom rozhraní vytvárať, upravovať a mazať pamäťové inštitúcie so všetkými ich údajmi.



Obr. 3.7: *Nákres rozhrania pre správu inštitúcií*

4 Technológie

4.1 Indexácia

4.1.1 Výber vyhľadávacieho nástroja

Pre potreby indexácie boli v analytickej fáze zvažované tieto možnosti:

- *Elastic* – vyhľadávací nástroj postavený na Java knižnici *Apache Lucene*
- *Sphinx* – vyhľadávací nástroj vytvorený v jazyku C++

Po konzultácii s vývojovým tímom sa nakoniec rozhodlo pre *Elastic* z dôvodu vyššej spoľahlivosti a jednoduchého používania.

4.1.2 Elastic

Elastic je nástroj na indexáciu dát. Jeho primárnym účelom je vyhľadávanie nad relatívne veľkým množstvom textu. Oproti databáze ponúka oveľa rýchlejšiu odpoveď a rozvinutejšie možnosti štrukturovaného ukladania dát.

Aplikácie využívajúce Elastic s ním komunikujú pomocou HTTP protokolu a dotazy ako aj odpovede sú vo formáte JSON. Príklad dotazu na Elastic:

```
1  GET
2  {
3  "query" : {
4          "term" : { "gender": "female" },
5          "range" : {
6                  "age" : {
7                  "from" : 20,
8                  "to" : 30
9                  }
10         }
11 }}
```

Tento dotaz vyhľadá všetky záznamy s uvedenými parametrami. *Term* je označenie výrazu, ktorý vyžaduje presný obsah určitého atribútu.

Range vyhľadá všetky záznamy ktoré sa nachádzajú v určenom rozmedzí. V tomto prípade hľadáme ženu od 20 do 30 rokov. Odpoveďou na takýto dotaz by mohlo byť napríklad:

```
"hits":{
  "total" : 1,
        "hits" : [
                  {"_index" : "pdkd",
                   "_type" : "person",
                   "_id" : "1",
                   "_source" : {
                         "name" : "Kim",
                         "age" : 22,
                         "gender" : "female"
                  }
        }
        ]
}
```

Výsledkom dotazu je teda 1 záznam.[4]

4.1.3 Ďalšie možnosti vyhľadávania

4.1.3.1 Radenie

Poradie vrátených záznamov môže byť vytvorené na základe určitého atribútu, relevancie výsledku alebo kombinácie týchto faktorov.

4.1.3.2 Score

Ďalšou zaujímavou funkciou elasticu je *relevance scoring*. Po tom čo získame zoznam vyhovujúcich záznamov, je potrebné ich zoradiť podľa relevancie. V prípade kombinovaného dotazu s logickými spojkami neobsahuje každý záznam všetky podmienky. Skóre relevancie sa počíta na základe troch faktorov:

1. **Frekvencia výskytu hľadaného výrazu v zázname** - čím viac je v zázname zmienená, tým je záznam podstatnejší.
2. **Inverzná frekvencia hľadaného výrazu** – relevancia výrazu sa znižuje v závislosti od množstva záznamov obsahujúcich tento výraz.

3. **Rozsah atribútu , v ktorom bol výraz nájdený** – čím je obsah atribútu dlhší, tým je relevancia tohto výskytu nižšia

4.1.3.3 Aggregations

Výsledky vyhľadávania je niekedy potrebné zlučovať do skupín na základe určitého parametru. *Elastic* ponúka funkciu *Aggregations*. Pokiaľ je teda v dotaze špecifikovaný atribút, na základe ktorého sa majú záznamy zoskupovať, je výsledkom objekt obsahujúci niekoľko skupín ďalších objektov, ktoré môžu predstavovať samotné záznamy alebo ďalšie skupiny zjednotení. Vďaka formátu JSON je možné tieto objekty serializovať. V prípade objektových jazykov ako Java je potom jednoduché poskladať štruktúry, s ktorými jazyk pracuje, pomocou knižnice na deserializáciu.

4.2 OAI-PMH

OAI-PMH[1] je protokol, ktorý sa používa na získavanie metadat elektronicky evidovaných dokumentov. Vznikol ako iniciatíva na zjednodušenie zbierania informácií vo viacerých separátnych repozitároch. Aj keď presný formát dát nie je určený, pre lepšiu interoperabilitu sa odporúča použitie štandardu Dublin Core.[5] Pri použití protokolu sa na jednej strane komunikácie nachádza poskytovateľ, ktorý dáta vystavuje. Typicky ide o inštitúciu spravujúcu dané dokumenty. Záznamy sú do databázy poskytovateľa vkladané ručne alebo v prípade dokumentu vystaveného inou inštitúciou je možné proces automatizovať a použiť informácie získané z tohto zdroja. Záznamy môžu byť upravované a vykonané zmeny sa prejavia vo vystavených dátach. Na druhej strane je zberateľ, ktorý o dáta žiada. Pomocou protokolu môže zozbierať metadáta dokumentov z rôznych inštitúcií a vytvoriť rozhranie pre užívateľov, ktorí takto môžu vyhľadávať nad zjednotenými informáciami na jednom mieste.[6],[7]

1. OAI-PMH - The Open Archives Initiative Protocol for Metadata Harvesting

4.3 Z39.50

Protokol Z39.50 je protokol na vyhľadávanie nad textovými databázami. Typicky sa využíva v knižničných systémoch. Na rozdiel od OAI-PMH nie je nutné dáta zozbierať a vytvoriť vyhľadávacie rozhranie, ale je možné vyhľadávať priamo nad databázou konkrétnej inštitúcie. Je tiež možné vytvoriť rozhranie pre jednoduchšie používanie, zjednotenie informácií z viacerých zdrojov alebo kombináciu s vyhľadávaním nad miestnou databázou. [8],[9]

4.4 Portál

4.4.1 Použitie portálu

Z užívateľského pohľadu je portál webová stránka, ktorá slúži ako jednotný bod prístupu k informáciám. Zámerom portálu je vytvoriť vstupnú bránu pre užívateľa po pripojení na internet. Môže zoskupovať informácie z rôznych zdrojov a zobrazovať ich užívateľovi na jednom mieste. Portál sa automaticky prispôsobuje potrebám konkrétneho užívateľa, ktorý ho používa. Užívateľ by tiež mal mať možnosť prispôsobiť zobrazený obsah podľa svojich potrieb.[10]

4.4.2 Portálové technológie

K zobrazovaniu zjednotených informácií používa portál portlety. Portlet je webová aplikácia, ktorá je zodpovedná za jednu alebo viac funkcií. Portlety bývajú štandardne súčasťou portálu po inštalácii, pretože niektoré požiadavky na funkcionalitu sa opakujú. Týmto spôsobom je možné zabezpečiť prihlasovanie užívateľov, systém oprávnení alebo zobrazovanie jednoduchého obsahu bez potreby ďalšej implementácie.[11]

Do portálu je tiež možné nasadiť vlastné portlety. Najnovšia špecifikácia popisujúca vývoj týchto portletov je JSR[2] 286, ktorá popisuje interakciu aplikácií s portálom. Vďaka tomu je možné vytvárať portlety, ktoré sú ľahko rozšíriteľné a prenositeľné bez ohľadu na typ použitej portálovej technológie.

2. JSR - Java Specification Request

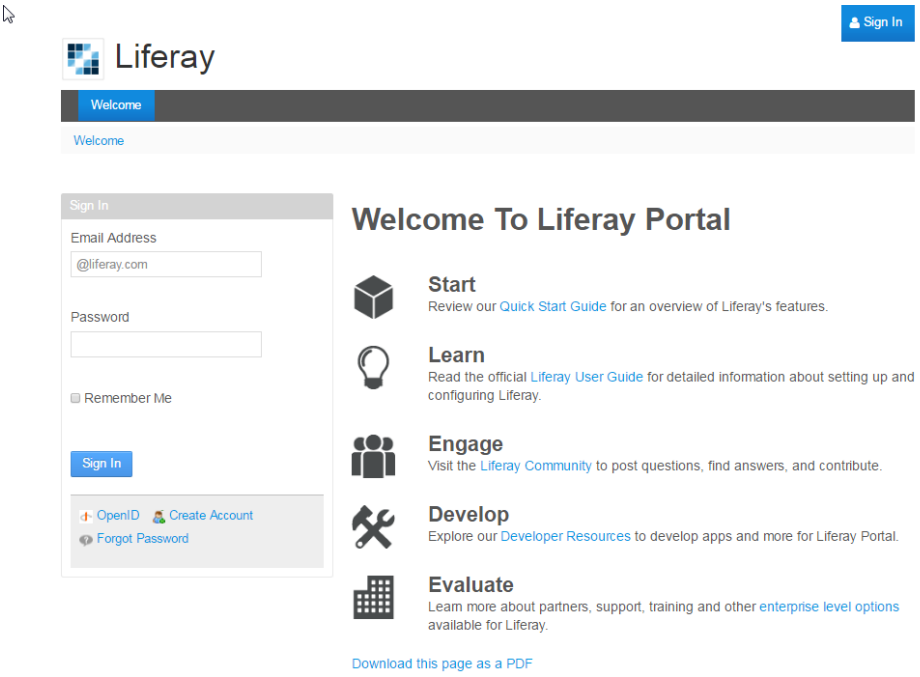
4.5 Databázový systém

Z analýzy požiadavkov vyplýva, že aplikácia potrebuje pre svoj beh databázu, v ktorej budú uložené všetky dáta vytvorené samotnou aplikáciou. Ostatné informácie ako užívateľské údaje, rozloženie portletov na stránke alebo akékoľvek dáta generované existujúcimi aplikáciami portálu Liferay sú ukladané do oddelenej databázy. Obidve databázy sú v správe KDJ. Pre tento prípad sú vystavené dve inštancie databázovej technológie MSSQL.

4.6 Liferay Portal

V analytickej fáze projektu boli zvažované viaceré portálové technológie ako napr. GateIn, Jetspeed a WebSphere. Nakoniec bola zvolená technológia Liferay Portal. Táto technológia vyniká výbornou dokumentáciou, celosvetovým počtom užívateľov a dostupnosťou. Liferay Portal je open source riešenie, ktoré je možné zakúpiť v troch úrovniach podpory alebo ho využívať bezplatne vo verzii CE[3]. Od verzie 6 je liferay otvorený pod licenciou LGPL, ktorá garantuje prístup ku zdrojovým kódom počas implementácie. Táto vlastnosť v kombinácií s možnosťou nahradenia existujúcich modulov z neho robí vysoko prispôsobiteľnú vývojovú platformu[12],[13].

3. CE - Community edition



Obr. 4.1: *Práve nainštalovaný Liferay Portal*

Na obrázku *3.1* je zobrazená úvodná stránka portálu bez akejkoľvek modifikácie. Obsahuje odkazy na prihlásenie, manuály na používanie portálu alebo integráciu s OpenID.

5 Implementácia

5.1 Využité portlety Liferay Portal

Pred implementáciou samotnej aplikácie bolo potrebné určiť, akú časť funkcionality poskytovanej technológiou Liferay Portal je možné použiť v PD. Rozhodli sme sa pre nasledujúce moduly:

- *Web Content Management Portlet* - obsahuje funkcie CMS[1]
- *Message Boards Portlet* - umožňuje pridávanie diskusie k objektu na základe primárneho kľúča
- *Asset Publisher Portlet* - umožňuje označovanie dokumentov pomocou tzv. štítkov, v PD je zodpovedný za žiadosti o rezervácie dokumentov
- *Asset Publisher Portlet* - stará sa o evidenciu kľúčových operácií v portáli, administrátor ju môže zobraziť priamo zo správcovského rozhrania

5.2 Štruktúra aplikácie

Obslužné triedy aplikácie sa delia do troch balíkov. Každý z nich je z pohľadu programovacieho jazyka Java samostatná knižnica. Ich funkcie sú nasledovné:

- *core* - balík zodpovedný za funkcionalitu na pozadí aplikácie, obsahuje entity predstavujúce záznamy v databáze, stará sa o všetky operácie nad databázou aplikácie, implementuje rozhrania z balíku *interface* ale nie je na ňom závislá
- *interface* - obsahuje objekty typu DTO[2], predstavuje komunikáciou medzi frontend a backend, *portlet* využíva operácie, kotré tento balík poskytuje, *core* ich implementuje
- *portlet* - balík zodpovedný za frontend funkcionalitu, obsluhuje požiadavky užívateľa, stará sa o zobrazovanie obsahu[14]

1. CMS - content management system
2. DTO - data transfer object

5.3 Digitálny objekt

Na začiatku implementácie bola vytvorená štruktúra digitálnych objektov. Na vrstve *interface* je reprezentovaná týmito objektmi:

- *DigitalObjectDto* - abstraktná trieda, predstavuje spoločný základ pre všetky digitálne objekty, obsahuje základné atribúty ako napríklad *id, miesto uloženia, názov*
- *SiriusObjectDto* - abstraktná trieda, predstavuje základ pre všetky objekty okrem zbierkových predmetov, rozširuje triedu *DigitalObjectDto* o nasledujúce atribúty:
  - *sigla* - identifikátor inštitúcie, ktorá vlastní objekt
  - *sipPackageName* - názov SIP balíčku, z ktorého bol záznam poskladaný
  - *PhysicalDescriptionDto* - samostatný objekt, v ktorom je uložený popis dokumentu a informácie o jeho vytvorení
- *MonographSingleDto* - reprezentuje monografiu, pridáva atribút *ccnb* (číslo České národní bibliografie)
- *MonographMultivolumeDto* - reprezentuje jeden výtlačok viaczväzkovej monografie, pridáva atribút *partNumber*
- *PeriodicalIssueDto* - reprezentuje jeden výtlačok periodika, pridáva atribút *issueNumber* a *subTitle*
- *MapDto* - reprezentuje mapu, pridáva atribút *scale* (mierka mapy)
- *PhotographDto* - reprezentuje fotografiu, pridáva atribút *imageNumber*
- *OfficialRecordIssueDto* - reprezentuje úradný dokument
- *VedutteDto*

Pre prácu s digitálnymi objektmi a pomocnými triedami boli v balíku *interface* vytvorené nasledujúce objekty:

- *DigitalObjectService* - vykonáva CRUD[3] operácie nad digitálnymi objektmi
- *DigitalObjectConverter* - konvertuje entity na DTO a naopak
- *PhysicalDescriptionService* - CRUD a konverzia popisov objektov
- *LanguageService* - CRUD a konverzia nad *LanguageDto*
- *LocationService* - CRUD a konverzia nad *LocationDto*

Tieto rozhrania sú implementované v balíku *core*. Názvy tried sú tvorené z názvov rozhraní pridaním prípony *Impl*.

5.4 Controller

Controller je komponenta systému, ktorá obsluhuje prichádzajúce požiadavky užívateľa. Je súčasťou návrhového vzoru MVC[4]. Z hľadiska funkcionality aplikácie PD predstavuje jeden logický celok. Nasledujúce triedy reprezentujú príslušné controllery:

- *DigitalObjectDetailViewController* - obsluhuje funkcie plného detailu digitálneho objektu
- *DocumentReaderViewController* - obsluhuje funkcie vyhľadávania a mapy
- *ImportAdminViewController* - obsluhuje funkcie rozhrania pre import
- *InstitutionViewController* - obsluhuje funkcie rozhrania pre správu a výpis pamäťových inštitúcií

5.5 Príklad fungovania aplikácie

Logiku aplikácie si môžeme najlepšie ukázať na jednoduchom príklade. Zvolíme si akcie a vstupné informácie. Užívateľ sa nachádza v

3. CRUD - create, read, update, delete
4. MVC - model-view-controller

administrácii a má otvorený formulár na úpravu dokumentu v ktorom zmenil názov diela. Po kliknutí na tlačidlo *Uložiť* vykoná aplikácia nasledujúce kroky:

1. Na základe parametra *ACTION_DIGITAL_OBJECT_NAME* rozozná o aký typ akcie sa jedná a priradí hodnotu atribútu *ADMINISTRATED_DTO* v tele requestu do parametra *digitalObjectDto* v metóde *updateMonographSingleDocument()*.
2. Skontroluje formulár. V prípade chyby zapíše výnimku do logu aplikácie a upovedomí užívateľa.
3. V prípade akceptovateľného formulára zavolá metódu *updateDigitalObject(digitalObjectDto)* v triede *DigitalObjectManager*, ktorá vykoná nasledujúce kroky:
   (a) Pomocou konvertora digitálnych objektov získa z objektu typu *Dto* entitu digitálneho objektu.
   (b) Uloží zmenenú entitu do databázy na základe parametra *id*.
   (c) Upraví záznam vo vyhľadávacom indexe tak aby odpovedal databázovému záznamu.
4. Nastaví hodnotu parametra *PARAM_PAGE* na *PAGE_EDIT_DIGITAL_OBJECT* a hodnotu parametra *PARAM_ID* na id upravovaného dokumentu, čím zabezpečí návrat na stránku úpravy dokumentu.

Nasledujúca ukážka kódu vykonáva funkcionalitu popísanú vyššie.

```
1  @ActionMapping(ACTION_DIGITAL_OBJECT_SAVE)
2  public void updateMonographSingleDocument(Model model,
3     ActionRequest request, ActionResponse response,
4  @Valid @ModelAttribute(ADMINISTRATED_DTO) DigitalObjectDto
5     digitalObjectDto, BindingResult result) {
6
7     if (!result.hasErrors()) {
8        sendMetadataEditedEmails(digitalObjectDto.getId());
9        digitalObjectManager.updateDigitalObject(digitalObjectDto);
10       SessionMessages.add(request, "digital-object-form-saved");
11       LogMF.info(LOG, "Digital object ID: {0} updated successfully",
12          new Object[]{digitalObjectDto.getId()});
13    } else {
14       LogMF.error(LOG, "Error updating Monograph Single Document:
             {0}",
15          new Object[]{result.getAllErrors()});
16       SessionErrors.add(request, "monograph-form-contain-error");
17    }
```

```
18
19    response.setRenderParameter(PARAM_ID,
20       digitalObjectDto.getId());
21    response.setRenderParameter(PARAM_PAGE,
22       PAGE_EDIT_DIGITAL_OBJECT);
23  }
```

6 Záver

Ciele zvolené na začiatku práce sa podarilo naplniť. Vytvorili sme portál, ktorý spĺňa požiadavky popísané v analýze. Aktuálna verzia prešla testovaním a je nasadená na produkčnom serveri, ktorý je sprístupnený verejnosti.

Zvolený spôsob digitalizácie dát sa ukázal ako vhodný. Ako dobrá voľba sa tiež ukázala mapa, ktorá dodala portálu vysokú mieru prehľadnosti a robustná platforma *Liferay Portal*, vďaka ktorej bolo možné implementovať dodatočné požiadavky zákazníka spoľahlivo a rýchlo.

Za zmienku stojí aj fakt, že celé riešenie aplikácie využíva open source technológie, za ktorými ale stojí široká vývojárska komunita. Touto cestou sme boli schopní postaviť robustné, spoľahlivé a zároveň lacné riešenie bez použitia platenej verzie *Liferay Portal* alebo zvažovanej technológie *SharePoint*.

Po nasadení portálu do ostrej prevádzky sa ponúkajú ďalšie možnosti rozvoja portálov kultúrneho dedičstva. Pre toto riešenie sa zatiaľ rozhodol len jeden kraj Českej republiky. Ak by ich postupne pribudlo viac, bolo by oveľa jednoduchšie zdieľať dáta medzi týmito inštitúciami, keďže jednou z najväčších prekážok pre digitalizáciu je nejednotný formát dát a zverejňovacích rozhraní. Týmto spôsobom by tiež bolo možné evidovať nehnuteľnosti alebo historicky významné miesta. Portálové riešenie ale nie je obmedzené len na kultúrne dedičstvo a časom by sa mohlo pristúpiť k evidencii prírodopisných zbierok, minerálov, živočíchov ale aj prírodných útvarov a pamiatok. Vzhľadom k tomu, že mapová nadstavba je už súčasťou riešenia, nie je ťažké si v budúcnosti predstaviť využitie portálu na plánovanie náučnej trasy po okresoch Českej republiky.

Literatúra

[1] Chris Marshall. „*Enterprise Modeling with UML: Designing Successful Software Through Business Analysis*“. In: illustrated edition. Addison-Wesley Professional, 2000, 2000. ISBN: 0201433133.

[2] Pascal Roques. „*UML in Practice: The Art of Modeling Software Systems Demonstrated through Worked Examples and Solutions*“. In: John Wiley & Sons, 2006. ISBN: 0470092793.

[3] R.E. Giachetti. „*Design of Enterprise Systems: Theory, Architecture, and Methods*“. In: CRC Press, 2016. ISBN: 9781439882894.

[4] Elastic Inc. *Elasticsearch Reference Guide*. `https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html`. 2016.

[5] Javier Nogueras-Iso, Francisco Javier Zarazaga-Soria, Pedro R. Muro-Medrano. „*Geographic Information Metadata for Spatial Data Infrastructures: Resources, Interoperability and Information Retrieval*“. In: illustrated edition, 263 p. Springer Science & Business Media, 2005. ISBN: 3540244646.

[6] Jonas X. Yuan. „*Object reuse and exchange (OAI-ORE)*“. In: Library technology reports, v. 46, no. 4. Chicago, IL : ALA TechSource, ©2010, 2010. ISBN: 0838958109.

[7] Rachel Heery, Liz Lyon. „*Research and Advanced Technology for Digital Libraries: 8th European Conference, ECDL 2004, Bath, UK, September 12-17, 2004, Proceedings*“. In: Springer, 2005. ISBN: 3540302301.

[8] Marcia J. Bates. „*Understanding Information Retrieval Systems: Management, Types, and Standards*“. In: 1st ed. Auerbach Publications, 2011. ISBN: 1466551356.

[9] M Pandian, C R Karisiddappa. „*Emerging Technologies for Knowledge Resource Management*“. In: Elsevier, 2007. ISBN: 1780631200.

[10] Arthur Tatnall. „*Web Portals: The New Gateways to Internet Information and Services*“. In: August, 2004. Idea group publishing, 2004. ISBN: 159140438X.

[11] Richard N. Katz. „*Web Portals and Higher Education: Technologies to Make IT Personal*“. In: August, 2004. Wiley, 2004. ISBN: 078796171X.

[12] Jonas X. Yuan. „*Liferay user interface development: develop a powerful and rich user interface with Liferay Portal 6*“. In: Birmingham, U.K. xi, 359 p. 2010. Packt Pub, 2010. ISBN: 1782162356.

[13] Jonas X. Yuan. „*Liferay portal systems development: build dynamic, content-rich, and social systems on top of Liferay*“. In: 1st ed. Birmingham, U.K. xiv, 515 p, 2012. Packt Open Source, 2012. ISBN: 1849515999.

[14] Dhrubojyoti Kayal. „*Pro Java EE Spring Patterns: Best Practices and Design Strategies Implementing Java EE Patterns with the Spring Framework*“. In: Apress, 2008. ISBN: 1430210109.

Prílohy

Súčasťou práce sú aj nasledujúce prílohy:

- Diagram tried PD