

## **Dokumentácia k inžinierskemu dielu**

*Tvorba "ľahko" sémantického obsahu pre adaptívny webový  
(výučbový) portál*

Tímový projekt

tím č. 13 – Thirteam

*Thirteam*

Bc. Martin Franta  
Bc. Martin Gajdoš  
Bc. Martin Habdák  
Bc. Róbert Kocian  
Bc. Petra Vrablecová  
Bc. Zuzana Zimová

Vedúci pedagóg: Ing. Marián Šimko  
e-mail: tim13@googlegroups.com  
Ak. rok : 2011/2011



# Obsah

---

<b>1</b>	<b>Úvod.....</b>	<b>1-1</b>
1.1	Použité pojmy .....	1-2
<b>2</b>	<b>Šprint č. 1 – Agricola.....</b>	<b>2-1</b>
2.1	CRUD – dokument a verzie.....	2-1
2.2	CRUD – repozitár .....	2-2
2.3	CRUD – RDT a vzťahy medzi RDT.....	2-6
<b>3</b>	<b>Šprint č. 2 – Beethoven.....</b>	<b>3-1</b>
3.1	Revert dokumentu.....	3-1
3.2	CRUD – anotácie.....	3-5
3.3	CRUD – varianty .....	3-8
3.4	CRUD – typy vzťahov .....	3-11
3.5	CRUD – vzťahy medzi RDT a dokumentmi .....	3-15
3.6	CRUD – vzťahy medzi dokumentmi .....	3-17
3.7	Validácia dokumentu voči XML schéme (XSD, RELAX NG).....	3-19
3.8	Prihlásenie používateľov .....	3-21
3.9	CRUD – obrázky .....	3-21
<b>4</b>	<b>Šprint č. 3 – Chopin.....</b>	<b>4-1</b>
4.1	Presun dokumentov medzi repozitármi .....	4-1
4.2	Zobrazenie grafu variantu metadát .....	4-2
4.3	Zobrazenie variantu metadát ako zoznamu RDT a vzťahov.....	4-3
4.4	Zobrazenie View dokumentu.....	4-4
4.5	Editovanie dokumentu a import súboru, informácie o zmenách .....	4-6
4.6	Možnosť preview aktuálneho dokumentu (Editovanie dokumentu).....	4-9
4.7	Zobrazenie detailu repozitára ako zoznamu dokumentov a variantov metadát .....	4-11
4.8	Zoznam repozitárov po prihlásení .....	4-12
<b>5</b>	<b>Šprint č. 4 – Dvořák .....</b>	<b>5-1</b>
5.1	Zobrazenie anotačného pásika .....	5-1
5.2	Zobrazovanie a editovanie anotácie po výbere z pásika .....	5-2
5.3	Pridávanie RDT a vzťahov v grafe .....	5-3
5.4	Import repozitára.....	5-4
5.5	Export repozitára.....	5-7

5.6	Prepojenie tabuľky a grafu .....	5-9
<b>6</b>	<b>Opis prototypu .....</b>	<b>6-1</b>
6.1	Architektúra systému .....	6-1
6.2	Dátový model systému .....	6-2
<b>Príloha A: Obrazovky .....</b>	<b>.....</b>	<b>A-1</b>
A.1	Obrazovka H1 .....	A-1
A.2	Obrazovka RD1 .....	A-2
A.3	Obrazovka DE1 .....	A-3
A.4	Obrazovka DP1 .....	A-5
A.5	Obrazovka V1 .....	A-6
A.6	Obrazovka VG1 .....	A-7
A.7	Tabuľky v systéme .....	A-8

# 1 Úvod

---

Témou nášho tímového projektu je *Tvorba "lahko" sémantického obsahu pre adaptívny webový (výučbový) portál*. Cieľom projektu je vytvoriť systém správy obsahu pre adaptívny webový portál a využiť tento systém pre správu výučbového obsahu systému ALEF. Výsledný projekt bude môcť byť používaný pedagógmi, ktorí prostredníctvom neho budú môcť vytvárať učebné texty s využitím metadát.

Vo všeobecnosti by mal byť projekt využiteľný pre rôzne typy adaptívnych webových portálov, ktoré využívajú sémantické metadáta, vzťahy medzi dokumentmi a na nich založené odporúčania.

Implementácia projektu je vytvorená v jazyku Ruby a frameworku Ruby on Rails, použitom aj pri vývoji systému ALEF. Zároveň použitie Ruby on Rails napomáha organizovaniu zdrojových kódov a jednoduchšej údržbe, čo bude výhodou pre ďalšie tímy, ktoré tento systém budú spravovať a vylepšovať.

Tím vyvíja projekt podľa metodiky SCRUM.

Dokumentácia obsahuje viacero častí, na ktorých sa autorsky podieľali jednotliví členovia tímu nasledovne:

**Úvod** Bc. Martin Gajdoš

## Šprint č. 1 – Agricola

CRUD – dokument a verzie	Bc. Martin Franta
CRUD – repozitár	Bc. Martin Gajdoš
CRUD – RDT a vzťahy medzi RDT	Bc. Petra Vrabecová

## Šprint č. 2 – Beethoven

Revert dokumentu	Bc. Martin Franta
CRUD – anotácie	Bc. Martin Gajdoš
CRUD – varianty	Bc. Martin Gajdoš
CRUD – typy vzťahov	Bc. Petra Vrabecová
CRUD – vzťahy medzi RDT a dokumentmi	Bc. Petra Vrabecová
CRUD – vzťahy medzi dokumentmi	Bc. Petra Vrabecová
Validácia dokumentu voči XML schéme (XSD, RELAX NG)	Bc. Martin Habdák
Prihlásenie používateľov	Bc. Martin Habdák
CRUD – obrázky	Bc. Róbert Kocian

## Šprint č. 3 – Chopin

Presun dokumentov medzi repozitármi	Bc. Zuzana Zimová
Zobrazenie grafu metadát	Bc. Martin Gajdoš
Zobrazenie variantu metadát ako zoznamu RDT a vzťahov	Bc. Petra Vrabecová
Zobrazenie View dokumentu	Bc. Martin Habdák
Editovanie dokumentu a import súboru,	Bc. Martin Franta

informácie o zmenách

Možnosť preview aktuálneho dokumentu  
(Editovanie dokumentu) Bc. Martin Habdák

Zobrazenie detailu repozitára ako zoznamu dokumentov  
a variantov metadát Bc. Zuzana Zimová

Zoznam repozitárov po prihlásení Bc. Zuzana Zimová

### Šprint č. 4 – Dvořák

Zobrazenie anotačného pásika Bc. Zuzana Zimová

Zobrazovanie a editovanie anotácie po výbere z pásika Bc. Zuzana Zimová

Pridávanie RDT a vzťahov v grafe Bc. Martin Gajdoš

Import repozitára Bc. Martin Habdák

Export repozitára Bc. Martin Franta

Prepojenie tabuľky a grafu Bc. Petra Vrablecová

### Opis prototypu

Architektúra systému Bc. Martin Franta

Dátový model Bc. Martin Habdák

### Prílohy

Obrazovky Bc. Petra Vrablecová

## 1.1 Použité pojmy

*Product backlog* zoznam požiadaviek zákazníka vytvorený na začiatku šprintu – obsahuje vlastnosti, ktoré by mal systém obsahovať po ukončení šprintu

*Dokument* obsahová entita, v prípade systému ALEF predstavuje XML dokument s výučbovým obsahom (tzv. *learning object*)

*Repozitár* úložisko dokumentov – priečinok s dokumentmi rovnakej témy, v prípade systému ALEF predstavuje jeden *kurz*

*RDT* *relevant domain term* – kľúčové slovo alebo viacslovný výraz vzťahujúci sa na tému resp. zameranie dokumentu

## 2 Šprint č. 1 – Agricola

---

V prvom šprinte sme mali za úlohu vytvoriť základnú kostru systému podľa doménového modelu.

Obsahom *product backlogu* bolo vytvorenie entít repozitára, dokumentu a RDT, vrátane základných CRUD operácii s nimi.

### 2.1 CRUD – dokument a verzie

Ako autor chcem vytvárať a upravovať dokumenty a spravovať ich verzie.

#### 2.1.1 Analýza

Vytvorená bude entita *Document*, s atribútmi obsah a názov.

Dokumenty majú byť verziované, čiže používateľ po uložení zmien v dokumente bude mať možnosť vrátiť sa k predchádzajúcej verzii dokumentu. V prvom šprinte je požadované vytvorenie verzie pri vytvorení a úprave dokumentu.

Verziovanie je možné poňať dvomi spôsobmi:

- ukladá sa celý obsah dokumentu,
- ukladajú sa iba prevedené zmeny.

Rozhodli sme sa ukladať celý obsah dokumentu. Pri návrate k mnohokrát zmeneným verziám je to rýchlejší spôsob, ako získať požadovaný obsah. História dokumentu, zaznamenanú vo verziách, má používateľ možnosť vidieť.

Verzie sa v prípade vymazania dokumentu nestratia, to znamená, že dokument zostáva v databáze, ale je ukrytý. Zvolenou akciou môže byť opäť vrátený do pôvodného – viditeľného stavu.

Správu verzií je možné v Ruby on Rails vyriešiť použitím tzv. gemu, čo je zásuvný modul resp. knižnica rozširujúca funkcionality frameworku.

#### 2.1.2 Návrh

Používateľ môže vytvárať, upravovať a odstraňovať dokumenty. Pri vytvorení dokumentu vzniká nová verzia s obsahom nového dokumentu. Po úprave dokumentu sa zachová predchádzajúca verzia, zároveň sa vytvorí verzia s novým obsahom a nastaví sa ako aktuálna. Vymazaním dokumentu sa dokument zdanlivo odstráni, zostane však zachovaný pre prípadné neskoršie obnovenie, spolu so všetkými svojimi verziami.

#### 2.1.3 Implementácia

Rozdelenie implementácie medzi dátovú, aplikačnú a prezentačnú vrstvu zodpovedá využitiu vzoru MVC, danému frameworkom Ruby on Rails.

Model Document je rozšírený gemom `Acts_as_versioned`, ktorý zabezpečuje automatické vytváranie verzií pri akciách vytvorenia a úpravy dokumentu.

Do aplikačnej vrstvy boli pridané akcie na získanie aktuálnej verzie dokumentu v JSON formáte a obnovenie odstráneného dokumentu. Akcia odstránenia dokumentu, bola upravená tak, aby sa dokumentu zmenil príznak zmazania a podľa neho sa dal identifikovať ako odstránený.

### 2.1.4 Testovanie

Funkcionalitu overujú nasledovné akceptačné testy:

<b>Názov testu</b>	Vytvorenie verzie pri úprave dokumentu	<b>ID</b>	2-1-1
<b>Rozhranie</b>	Formulár editácie dokumentu	<b>autor</b>	Martin Franta
<b>Účel</b>	Uchovanie pôvodnej verzie dokumentu		
<b>Vstupné podmienky</b>	Dokument je vytvorený		
<b>Výstupné podmienky</b>	Existuje nová verzia, ktorá je nastavená ako aktuálna		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Upravíme pole Content tak, aby bol obsah validný podľa schémy.	Text je upravený.	Text bol upravený.
2	Potvrdenie úpravy formuláru stlačením tlačidla Update.	Dôjde k presmerovaniu na upravený dokument s novou verziou.	Presmerované na stránku s upraveným dokumentom. Dokument je označený novou verziou.

## 2.2 CRUD – repozitár

Práca s dokumentmi je zabezpečená v rámci repozitára.

### 2.2.1 Analýza

Entita repozitára by sa dala prirovnať k objektu priečinka, v rámci súborového systému. Ten má obsahovať jednotlivé dokumenty. V tejto fáze vývoja sa jedná o pomerne jednoduchú entitu, ktorej jediné vlastnosti sú zatiaľ názov a kľúč API. Kľúč API bude v neskorších štádiách vývoja použitý pre napojenie sa a kooperáciu s externými systémami.

### 2.2.2 Návrh

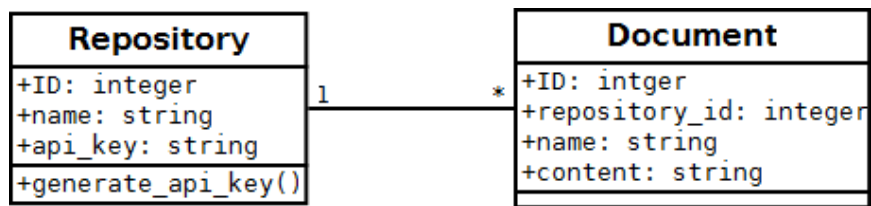
Atribúty repozitára sú nasledovných typov (názvy a typy sú uvedené v anglickom jazyku):

- name: string
- api\_key: string

Používateľ pri tvorbe a úprave repozitára bude pracovať iba s atribútom názov. ApiKey je zatiaľ náhodne generovaný reťazec 32 znakov.

Prepojenie s dokumentmi sa realizuje na strane dokumentov, ktoré si uchovávajú číslo repozitára, v rámci ktorého boli vytvorené.





Obr. 1. Vzťah dokumentu a repositára.

### 2.2.3 Implementácia

Úvodná implementácia prebiehala za pomoci funkcie *scaffold*, poskytnutej frameworkom *Rails*. Boli vygenerované nasledujúce súbory:

```

db/migrate/20100207214725_create_repositories.rb
app/models/repository.rb
test/fixtures/repositories.yml
app/controllers/repositories_controller.rb
app/views/repositories/index.html.erb
app/views/repositories/edit.html.erb
app/views/repositories/show.html.erb
app/views/repositories/new.html.erb
app/views/repositories/_form.html
app/helpers/repositories_helper.rb Helper
app/assets/stylesheets/scaffolds.css.scss
app/assets/stylesheets/repositories.css.scss
app/assets/javascripts/repositories.js.coffee
test/unit/repository_test.rb Unit
test/functional/repositories_controller_test.rb
test/unit/helpers/repositories_helper_test.rb
  
```

V nich bolo nutné upraviť nasledujúce:

- **model**
  - » pridať mu funkciu generovania API kľúča,
  - » pridať overovanie názvu,
  - » vytvoriť vzťah *repository – documents*.
- **controller**
  - » spustiť generovanie API kľúča pred tým, ako sa uloží nový repositár do databázy.
- **views**
  - » bolo potrebné odstrániť načítavanie API kľúča od používateľa,

- » upraviť obrazovky dokumentov, tak aby sa im pri ich tvorbe dal zvoliť repozitár, ku ktorému patria.

#### 2.2.4 Testovanie

<b>Názov testu</b>		List all repositories	<b>ID</b>	2-2-1
<b>Rozhranie</b>	Zobrazenie repozitárov		<b>autor</b>	Martin Gajdoš
<b>Účel</b>	Správne načítanie repozitárov			
<b>Vstupné podmienky</b>		Vytvorené repozitáre		
<b>Výstupné podmienky</b>		Zobrazené repozitáre		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme nové repozitáre.	Nové repozitáre v systéme.	Nové repozitáre v systéme.	
2	Zobrazenie stránky repozitárov.	Zobrazené repozitáre.	Zobrazené repozitáre.	

<b>Názov testu</b>		View a repository	<b>ID</b>	2-2-2
<b>Rozhranie</b>	Zobrazenie repozitáru		<b>autor</b>	Martin Gajdoš
<b>Účel</b>	Správne načítanie repozitáru			
<b>Vstupné podmienky</b>		Vytvorené repozitáre		
<b>Výstupné podmienky</b>		Zobrazený repozitár		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme nové repozitáre.	Nové repozitáre v systéme.	Nové repozitáre v systéme.	
2	Zobrazenie stránky druhého repozitáru v systéme.	Zobrazený repozitár.	Zobrazený repozitár.	

<b>Názov testu</b>		Edit a repository	<b>ID</b>	2-2-3
<b>Rozhranie</b>	Úprava repozitárov		<b>autor</b>	Martin Gajdoš
<b>Účel</b>	Správne upravený repozitár			
<b>Vstupné podmienky</b>		Vytvorený repozitár		
<b>Výstupné podmienky</b>		Upravený repozitár		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme nový repozitár.	Nový repozitár v systéme.	Nový repozitár v systéme.	
2	Úprava repozitáru.	Upravený repozitár.	Upravený repozitár.	
3	Zobrazenie upraveného repozitáru.	Zobrazený upravený repozitár.	Zobrazený upravený repozitár.	

<b>Názov testu</b>		Delete a repository via the index page	<b>ID</b>	2-2-4
<b>Rozhranie</b>	Zmazanie repozitáru		<b>autor</b>	Martin Gajdoš
<b>Účel</b>	Správne zmazanie repozitáru			
<b>Vstupné podmienky</b>		Vytvorený repozitár		
<b>Výstupné podmienky</b>		Zmazaný repozitár		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme nový repozitáre.	Nové repozitáre v systéme.	Nové repozitáre v systéme.	
2	Zmazanie repozitáru.	Zmazaný repozitár.	Zmazaný repozitár.	
3	Zobrazenie repozitárov.	Zmazaný repozitár nie je zobrazený.	Zmazaný repozitár nie je zobrazený.	

## 2.3 CRUD – RDT a vzťahy medzi RDT

Ako autor chcem vytvárať a upravovať termy – *RDT* a vytvárať medzi nimi vzťahy.

### 2.3.1 Analýza

Používateľ má mať možnosť vytvoriť si vlastné termy, upravovať ich a vymazávať ich. Medzi termami je možné vytvárať orientované vzťahy rôznych typov. Každý vzťah má svoju váhu (číslo z intervalu  $<0,1>$ ). V budúcnosti sa termy budú priradovať k dokumentom.

### 2.3.2 Návrh

Používateľ môže vytvárať, upravovať a odstraňovať RDT. Vytvorená bude entita *RDT*, s atribútom názov, ktorú bude neskôr možné prepojiť na ďalšie entity. Po vytvorení termu sa term uloží v databáze a používateľ bude mať k nemu prístup cez používateľské rozhranie.

Používateľ môže vytvárať, upravovať a odstraňovať vzťahy medzi termami. Vytvorená bude entita vzťahu s atribútmi typ, váha a dvomi ID termov, medzi ktorými je vzťah (*from, to*). Po vytvorení vzťahu sa vzťah uloží v databáze a používateľ bude mať k nemu prístup cez používateľské rozhranie.

### 2.3.3 Implementácia

Rozdelenie implementácie medzi dátovú, aplikačnú a prezentačnú vrstvu zodpovedá využitiu vzoru MVC, danému frameworkom Ruby on Rails.

Implementovaný bol model s názvom *RDT* a povinným atribútom *name*. V aplikačnej časti sú implementované funkcie na vytvorenie, zmazanie, upravenie termu. Ďalej bol implementovaný model *RDTsRelationship* s povinnými atribútmi typ, váha a dve ID termov, medzi ktorými je vzťah. V aplikačnej časti sú implementované funkcie na vytvorenie, zmazanie, upravenie vzťahu medzi dvomi termami.

### 2.3.4 2.3.4 Testovanie

Funkcionalitu overujú nasledovné akceptačné testy:

<b>Názov testu</b>		Vytvorenie RDT	<b>ID</b>	2-3-1
<b>Rozhranie</b>	Formulár vytvárania RDT		<b>autor</b>	Petra Vrablecová
<b>Účel</b>	Overenie funkcionality vytvárania RDT			
<b>Vstupné podmienky</b>	Používateľ chce vytvoriť nový term.			
<b>Výstupné podmienky</b>	Je vytvorený nový term a používateľ ho vidí v použ. rozhraní.			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí, že chce vytvoriť nový term.	Dôjde k presmerovaniu na formulár pre vytváranie termu.	Presmerované na stránku pre vytváranie termu.	

2	Používateľ vyplní názov termu.	V poli formulára „Name“ bude zobrazený zadaný názov termu.	Vo formulári bol zobrazený používateľom zadaný názov termu.
3	Používateľ potvrdí, že chce vytvoriť term so zadaným názvom.	Dôjde k presmerovaniu na stránku so zobrazeným vytvoreným termom.	Presmerované na stránku so zobrazeným vytvoreným termom.

<b>Názov testu</b>		Upravenie RDT	<b>ID</b>	2-3-2
<b>Rozhranie</b>	Formulár úpravy RDT		<b>autor</b>	Petra Vrablecová
<b>Účel</b>	Overenie funkcionality úpravy RDT			
<b>Vstupné podmienky</b>	Je vytvorený term. Používateľ chce upraviť existujúci term.			
<b>Výstupné podmienky</b>	Term je upravený podľa používateľa a používateľ ho vidí v používateľskom rozhraní.			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí, že chce upraviť jeden term zo zoznamu existujúcich termov.	Dôjde k presmerovaniu na formulár pre úpravu termu.	Presmerované na stránku pre úpravu termu.	
2	Používateľ zmení názov termu.	V poli formulára „Name“ bude zobrazený zadaný názov termu.	Vo formulári bol zobrazený používateľom zadaný názov termu.	
3	Používateľ potvrdí, že chce vytvoriť term so zadaným názvom.	Dôjde k presmerovaniu na stránku so zobrazeným vytvoreným termom.	Presmerované na stránku so zobrazeným vytvoreným termom.	

<b>Názov testu</b>	Vymazanie RDT		<b>ID</b>	2-3-3
<b>Rozhranie</b>	Formulár zmazanie RDT		<b>autor</b>	Petra Vrablecová
<b>Účel</b>	Overenie funkcionality zmazania RDT			
<b>Vstupné podmienky</b>	Je vytvorený term. Používateľ chce zmazať existujúci term.			
<b>Výstupné podmienky</b>	Term je zmazaný a používateľ ho už nevidí v použ. rozhraní.			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí, že chce zmazať jeden term zo zoznamu existujúcich termov.	Dôjde k zobrazeniu potvrdzujúceho dialógu.	Zobrazil sa potvrdzujúci dialóg.	
2	Používateľ zvolí, že chce zvolený term zmazať.	Dôjde k presmerovaniu na stránku so zobrazeným zoznamom existujúcich termov, ktorý neobsahuje zmazaný term.	Presmerované na stránku so zobrazeným zoznamom existujúcich termov, ktorý neobsahoval zmazaný term.	

<b>Názov testu</b>		Vytvorenie vzťahu medzi dvomi RDT	<b>ID</b>	2-3-4
<b>Rozhranie</b>		Formulár vytvárania vzťahu medzi dvomi RDT	<b>autor</b>	Petra Vrablecová
<b>Účel</b>		Overenie funkcionality vytvárania neorientovaného vzťahu medzi dvomi RDT		
<b>Vstupné podmienky</b>		Existujú aspoň dva termy. Používateľ chce vytvoriť neorientovaný vzťah medzi dvomi termami.		
<b>Výstupné podmienky</b>		Sú vytvorené dva nové orientované vzťahy a používateľ ich vidí v použ. rozhraní.		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí, že chce vytvoriť nový vzťah.	Dôjde k presmerovaniu na formulár pre vytváranie vzťahu.	Presmerované na stránku pre vytváranie vzťahu.	
2	Používateľ vyplní typ, váhu vzťahu a zvolí dva termy, medzi ktorými má byť vytvorený vzťah a zvolí, či má byť vzťah orientovaný.	V poli formulára „Type“ bude zobrazený zadaný typ vzťahu, v poli „Weight“ bude zobrazená zadaná váha vzťahu, v poliach „From“ a „To“ bude v každom vybraný jeden term, pole „Oriented“ nebude označené.	Vo formulári boli zobrazené používateľom zadané atribúty vzťahu.	
3	Používateľ potvrdí, že chce vytvoriť vzťah so zadanými atribútmi.	Dôjde k presmerovaniu na stránku so zobrazeným vytvorenými vzťahmi.	Presmerované na stránku so zobrazenými dvomi vytvorenými orientovanými vzťahmi.	





## 3 Šprint č. 2 – Beethoven

---

V druhom šprinte vznikla nová entita – Používateľ. Bolo potrebné vyriešiť jej prepojenia na dokumenty a repozitáre. Taktiež pribudli anotácie dokumentov a vzťahy medzi RDT a dokumentmi.

### 3.1 Revert dokumentu

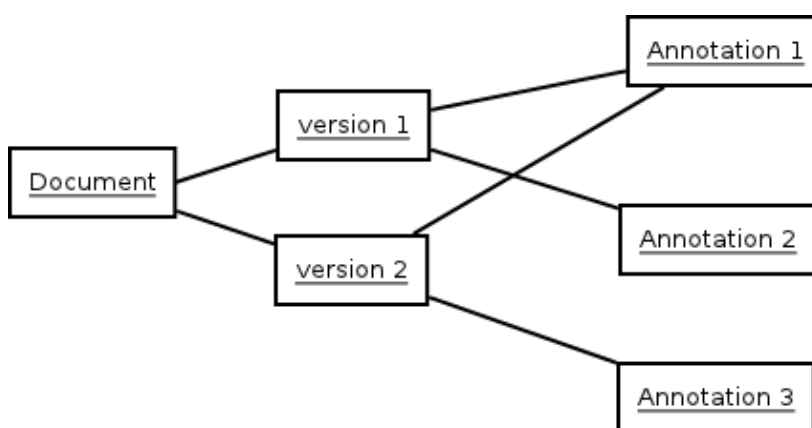
Ako autor chcem vrátiť stav dokumentu do štádia uloženého v danej verzii a zároveň spravovať anotácie priradené k tomuto dokumentu.

#### 3.1.1 Analýza

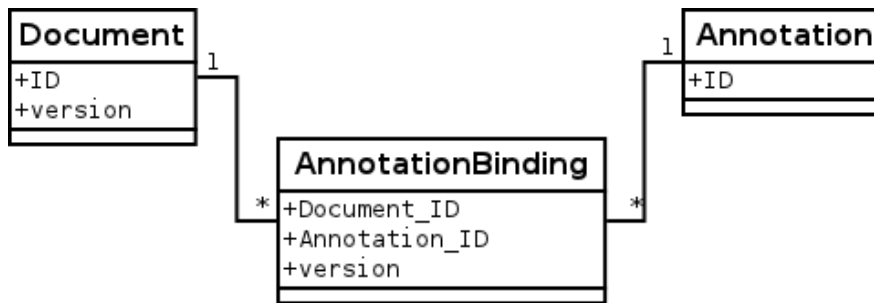
V systéme je po prvom šprinte implementované verziovanie dokumentov. Revert je akcia, ktorá vráti dokument do stavu daného istou verziiou. Vykonaním tejto akcie zostávajú v systéme všetky verzie dokumentu a nevzniknú žiadne nové.

Aby sme mohli určiť, kto novú verziu vytvoril, do verziovania pribudne používateľ. Zmena samotného používateľa nesmie vyvolať uloženie novej verzie, to sa stále vykonáva len pri zmene nastavených atribútov (názov, obsah). Pri vytváraní verzie si tiež treba zapamätať, z ktorej verzie aktuálna verzia vznikla.

K dokumentu môžu byť priradené anotácie, ktoré sa vzťahujú na určitý úsek textu. Úpravami textu (odstránením anotovaného úseku, opravou chyby) môžu stratiť anotácie význam. Sú preto prepojené s verziami dokumentu. Pri vytvorení novej verzie sa anotácie prenesú a používateľ ich bude môcť upravovať a vymazávať. Kým úprava anotácie zmení jej obsah pre všetky verzie dokumentu, vymazanie sa udeje iba v kontexte danej verzie.



Obr. 2. Možné prepojenie anotácií a verzií dokumentu.



Obr. 3. Diagram prepojenia anotácií a dokumentov.

Používateľ by mal vidieť spolu s dokumentom taktiež zoznam verzií s autormi, ktorí ich vytvorili. Každú z týchto verzií si môže zvlášť zobraziť. Úpravou verzie vzniká nová verzia, ktorá sa nastaví ako aktuálna a zobrazením dokumentu získame jej obsah. Kvôli zaznamenaniu vývoja dokumentu bude potrebné ukladať informáciu o tom, z ktorej verzie vznikla pri úprave nová verzia. Tento spôsob ukladania predchádzajúcich verzií nám umožní sledovať vetvy vývoja dokumentu.

### 3.1.2 Návrh

Používateľ so svojou identitou v rámci systému môže upravovať a nastavovať existujúce verzie dokumentu s ohľadom na existujúce anotácie a ich význam v upravenom dokumente. Ovládanie týchto akcií bude zabezpečené novou obrazovkou, na ktorej sa bude pracovať s vybranou verziou dokumentu. Do obrazovky pre zobrazenie dokumentu pribudne zoznam anotácií s možnosťou vymazávať, upravovať a pridávať anotácie k danej verzii.

### 3.1.3 Implementácia

Základný návrat verzie je riešený funkciou `Acts_as_versioned`, ktorá do inštancie triedy dokument načíta atribúty z vybranej verzie. Až po explicitnom volaní funkcie uloženia do databázy, je dokument skutočne zmenený na vybranú verziu. Túto vlastnosť sme využili pri zobrazení obsahu verzie, kedy nechceme aby sa menila aktuálna verzia vybraného dokumentu.

Zoznam anotácií pre danú verziu dokumentu získavame dopytom na tabuľku anotácií s využitím väzobnej entity `AnnotationBinding`. Tá obsahuje väzbu na dokument (ID dokumentu), jeho verziu a na druhej strane väzbu na anotáciu (ID anotácie). Kombinácia ID dokumentu, verzia a ID anotácie je pre väzby unikátna. Odstránením väzby zaniká prepojenie verzie dokumentu na anotáciu, ale anotácia a jej ďalšie prepojenia na iné verzie zostávajú.

Kopírovanie prepojení anotácií a verzií prebieha pri vytvorení novej verzie a je zabezpečené spätnými volaniami pred a po uložení modelu dokumentu.

### 3.1.4 Testovanie

Akceptačné testy:

<b>Názov testu</b>		Vytvorenie verzie pri úprave dokumentu	<b>ID</b>	3-1-1
<b>Rozhranie</b>	Formulár editácie dokumentu		<b>autor</b>	Martin Franta
<b>Účel</b>	Uchovanie pôvodnej verzie dokumentu spolu so zaznamenaným autorom zmien			
<b>Vstupné podmienky</b>	Dokument je vytvorený, používateľ je prihlásený			
<b>Výstupné podmienky</b>	Existuje nová verzia, ktorá je nastavená ako aktuálna			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Upravíme pole Content tak, aby bol obsah validný podľa schémy.	Text je upravený.	Text bol upravený.	
2	Potvrdenie úpravy formuláru stlačením tlačidla Update.	Dôjde k presmerovaniu na upravený dokument s novou verziou, ako autor verzie je označený prihlásený používateľ.	Presmerované na stránku s upraveným dokumentom. Dokument je označený novou verziou s autorstvom prihláseného používateľa.	

<b>Názov testu</b>		Zobrazenie verzie dokumentu	<b>ID</b>	3-1-2
<b>Rozhranie</b>	Obrazovka dokumentov		<b>autor</b>	Martin Franta
<b>Účel</b>	Zobrazenie obsahu inej než aktuálnej verzie			
<b>Vstupné podmienky</b>	Dokument má viac 1 verziu, používateľ je prihlásený			
<b>Výstupné podmienky</b>	Zobrazený je obsah vybranej verzie			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Vyberieme dokument, ktorý má viac verzií.	Zobrazenie obsahu dokumentu a prehľadu verzií.	Zobrazený bol obsah dokumentu a prehľad verzií.	
2	Klikneme na prvú položku v prehľade verzií.	Zobrazí sa obsah vybranej verzie.	Zobrazený je obsah vybranej verzie.	

## 3 Šprint č. 2 – Beethoven

<b>Názov testu</b>		Obnova vybranej verzie	<b>ID</b>	3-1-3
<b>Rozhranie</b>	Obrazovka dokumentov		<b>autor</b>	Martin Franta
<b>Účel</b>	Zmena obsahu dokumentu na inú než aktuálnu verziu			
<b>Vstupné podmienky</b>		Dokument má viac 1 verziu, používateľ je prihlásený		
<b>Výstupné podmienky</b>		Dokument má obsah vybranej verzie		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Vyberieme dokument, ktorý má viac verzií.	Zobrazenie obsahu dokumentu a prehľadu verzií.	Zobrazený bol obsah dokumentu a prehľad verzií.	
2	Klikneme na prvú položku v prehľade verzií.	Zobrazí sa obsah vybranej verzie.	Zobrazený je obsah vybranej verzie.	
3	Klikneme na odkaz Revert.	Zobrazí sa potvrdzovacia otázka.	Zobrazila sa potvrdzovacia otázka.	
4	Klikneme na súhlas (OK).	Presmerovanie na obrazovku dokumentu s oznámením o návrate verzie a zobrazeným obsahom vybranej verzie.	Presmerované na obrazovku dokumentu, zobrazené oznámenie a obsah verzie.	

<b>Názov testu</b>		Zobrazenie novej anotácie	<b>ID</b>	3-1-4
<b>Rozhranie</b>	Obrazovka dokumentu		<b>autor</b>	Martin Franta
<b>Účel</b>	Prehľad pripojených anotácií			
<b>Vstupné podmienky</b>		Používateľ je prihlásený, v systéme existuje typ anotácie		
<b>Výstupné podmienky</b>		Na obrazovke dokumentu je zobrazená nová anotácia		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Klikneme na odkaz New Annotation.	Zobrazenie formuláru novej anotácie.	Zobrazený formulár novej anotácie.	
2	Vyplníme text a vyberieme typ anotácie a odošleme formulár.	Presmerovanie na obrazovku dokumentu s novou anotáciou.	Presmerované na obrazovku dokumentu, v zozname anotácií je pridaná anotácia.	

<b>Názov testu</b>		Zobrazenie anotácií v novej verzii dokumentu	<b>ID</b>	3-1-5
<b>Rozhranie</b>	Obrazovka dokumentov		<b>autor</b>	Martin Franta
<b>Účel</b>	Prepojenie anotácií na novú verziu dokumentu			
<b>Vstupné podmienky</b>	Používateľ je prihlásený, existuje dokument s anotáciami			
<b>Výstupné podmienky</b>	Na obrazovke dokumentu sú zobrazené anotácie			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Vyberieme dokument s 1 alebo viacerými anotáciami.	Zobrazenie obsahu dokumentu a zoznam anotácií.	Zobrazený obsah dokumentu a zoznam anotácií.	
2	Klikneme na odkaz Edit.	Zobrazenie formuláru úpravy dokumentu.	Zobrazený formulár úpravy dokumentu.	
3	Upravíme obsah dokumentu aby bol validný a odošleme.	Presmerovanie na obrazovku dokumentu s upraveným obsahom a nezmeneným zoznamom anotácií.	Presmerované na obrazovku dokumentu s upraveným obsahom. Zoznam anotácií je nezmenený.	

## 3.2 CRUD – anotácie

Systém má uchovávať spolu s jednotlivými dokumentmi aj anotácie týchto dokumentov.

### 3.2.1 Analýza

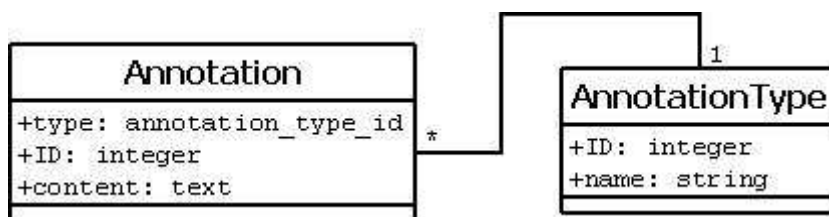
V našom prípade je anotácia tvorená určitou časťou dokumentu. V tejto fáze vývoja sa jedná o jednoduchú entitu, ktorej jediné vlastnosti sú zatiaľ typ anotácie a samotný obsah. Je potrebné ešte upraviť model a controller anotácie takým spôsobom, aby sa obsah anotácie načítaval priamo z obsahu priradeného dokumentu na základe určeného počiatočného a koncového znaku.

### 3.2.2 Návrh

Atribúty anotácie sú nasledovných typov (názvy a typy sú uvedené v anglickom jazyku):

- content: text
- type: annotation\_type\_id

Každá anotácia je určitého typu. Pre rôzne typy anotácií sme vytvorili samostatnú tabuľku, ktorá bude uchovávať typy anotácií.



Obr. 4. Prepojenie anotácie a typu anotácie.

### 3.2.3 Implementácia

Úvodná implementácia prebiehala za pomoci funkcie *scaffold*, poskytnutej frameworkom *Rails*. Boli vygenerované nasledujúce súbory:

```

db/migrate/20111107175037_create_annotations.rb
app/models/annotation.rb
test/fixtures/annotations.yml
app/controllers/annotations_controller.rb
app/views/annotations/index.html.erb
app/views/annotations/edit.html.erb
app/views/annotations/show.html.erb
app/views/annotations/new.html.erb
app/views/annotations/_form.html.
  app/helpers/annotations_helper.rb Helper
app/assets/stylesheets/scaffolds.css.scss
app/assets/stylesheets/annotations.css.scss
app/assets/javascripts/annotations.js.coffee
test/unit/annotation_test.rb Unit
test/functional/annotations_controller_test.rb
test/unit/helpers/annotations_helper_test.rb
  
```

V nich bolo nutné upraviť nasledujúce:

- **model**
  - » definovať prístupové atribúty,
  - » pridať overovanie prítomnosti obsahu,
  - » vytvoriť vzťah *annotation* – *annotation\_type*.
- **views**
  - » upraviť obrazovku pridania resp. editácie anotácie tak, aby bolo možné vybrať typ anotácie z modelu *annotation\_type*.

## 3.2.4 Testovanie

<b>Názov testu</b>		List all annotations	<b>ID</b>	3-2-1
<b>Rozhranie</b>	Zobrazenie anotácií		<b>autor</b>	Gabriel Tekel'
<b>Účel</b>	Správne načítanie anotácií			
<b>Vstupné podmienky</b>		Vytvorené anotácie		
<b>Výstupné podmienky</b>		Zobrazené anotácie		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme nové anotácie.	Nové anotácie v systéme.	Nové anotácie v systéme.	
2	Zobrazenie stránky anotácií.	Zobrazené anotácie.	Zobrazené anotácie.	

<b>Názov testu</b>		Edit a annotation	<b>ID</b>	3-2-2
<b>Rozhranie</b>	Úprava anotácií		<b>autor</b>	Gabriel Tekel'
<b>Účel</b>	Správne upravená anotácia			
<b>Vstupné podmienky</b>		Vytvorená anotácia		
<b>Výstupné podmienky</b>		Upravená anotácia		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme novú anotáciu.	Nová anotácia v systéme.	Nová anotácia v systéme.	
2	Úprava anotácie.	Upravená anotácia.	Upravená anotácia.	
3	Zobrazenie upravenej anotácie.	Zobrazená upravená anotácia.	Zobrazená upravená anotácia.	

<b>Názov testu</b>		Delete a annotation via the index page	<b>ID</b>	3-2-3
<b>Rozhranie</b>	Zmazanie anotácie		<b>autor</b>	Gabriel Tekel'
<b>Účel</b>	Správne zmazanie anotácie			
<b>Vstupné podmienky</b>	Vytvorená anotácia			
<b>Výstupné podmienky</b>	Zmazaná anotácia			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme novú anotáciu.	Nové anotácie v systéme.	Nové anotácie v systéme.	
2	Zmazanie anotácie.	Zmazaná anotácia.	Zmazaná anotácia.	
3	Zobrazenie anotácií.	Zmazaná anotácia nie je zobrazená.	Zmazaná anotácia nie je zobrazená.	

### 3.3 CRUD – varianty

Systém má pracovať s kľúčovými slovami – RDT, priradených dokumentu. RDT a ich vzťahy tvoria metadáta, ktoré bude spracovávať systém. Rozdielne súbory týchto metadát sú reprezentované entitou *variant*.

#### 3.3.1 Analýza

Pre *variant* v tomto štádiu vývoja je dôležitý atribút, podľa ktorého si ho bude môcť používateľ vybrať – meno (name). Aby spĺňal svoju úlohu treba vytvoriť asociácie

- Variant – RDT
- Variant – Vzťah medzi RDT

Používateľ si tak bude mať možnosť v rámci repozitára zvoliť, s ktorou skupinou metadát bude pracovať na úrovni dokumentu.

#### 3.3.2 Návrh

Prepojenie *variant* – *vzťah RDT* bolo sprostredkované na strane *vzťahu RDT*, ktorý si bude pamätať, ku ktorému variantu patrí.

Nakoľko asociácia RDT môže patriť viacerým variantom a variant môže obsahovať viacero RDT, pre asociáciu *variant* – *RDT* bolo nutné použiť väzobnú entitu. Tá slúži iba na udržiavanie vzťahov medzi týmito dvoma entitami.

Pre jednoduchšie pridávanie viacerých RDT a ich vzťahov ku variantu naraz, bude obrazovka úprav variantu poskytovať ich zoznam. Ten bude zabezpečený možnosťou výberu viacerých položiek naraz.



### 3.3.3 Implementácia

Úvodná implementácia prebiehala za pomoci funkcie *scaffold*, poskytnutej frameworkom *Rails*. Boli vygenerované nasledujúce súbory:

```
db/migrate/20100207214725_create_repositories.rb
app/models/variant.rb
test/fixtures/variants.yml
app/controllers/variants_controller.rb
app/views/variants/index.html.erb
app/views/variants/edit.html.erb
app/views/variants/show.html.erb
app/views/variants/new.html.erb
app/views/variants/_form.html.
app/helpers/variants_helper.rb Helper
app/assets/stylesheets/scaffolds.css.scss
app/assets/stylesheets/variants.css.scss
app/assets/javascripts/variants.js.coffee
test/unit/variant_test.rb Unit
test/functional/variants_controller_test.rb
test/unit/helpers/variants_helper_test.rb
```

V nich bolo nutné upraviť nasledujúce:

- **model**
  - » pridať overovanie mena,
  - » pridať asociácie ku RDT a ich vzťahom.
- **controller**
  - » pridať funkcie pre pridanie viacerých RDT a ich vzťahov naraz.
- **views**
  - » upraviť formulár zobrazenia, tak aby sa dalo pridať viacero RDT a ich vzťahov naraz.

## 3.3.4 Testovanie

<b>Názov testu</b>		List all variants	<b>ID</b>	3-3-1
<b>Rozhranie</b>	Zobrazenie variantov		<b>autor</b>	Martin Gajdoš
<b>Účel</b>	Správne načítanie variantov			
<b>Vstupné podmienky</b>		Vytvorené varianty		
<b>Výstupné podmienky</b>		Zobrazené varianty		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme nové varianty.	Nové variant v systéme.	Nové variant v systéme.	
2	Zobrazenie stránky variantov.	Zobrazené varianty.	Zobrazené varianty.	

<b>Názov testu</b>		View a variant	<b>ID</b>	3-3-2
<b>Rozhranie</b>	Zobrazenie variantu		<b>autor</b>	Martin Gajdoš
<b>Účel</b>	Správne načítanie variantu			
<b>Vstupné podmienky</b>		Vytvorené variant		
<b>Výstupné podmienky</b>		Zobrazený variant		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme nové varianty.	Nové varianty v systéme.	Nové varianty v systéme.	
2	Zobrazenie stránky druhého variantu v systéme.	Zobrazený variant.	Zobrazený variant.	

<b>Názov testu</b>		Edit a variant	<b>ID</b>	3-3-3
<b>Rozhranie</b>	Úprava variantov		<b>autor</b>	Martin Gajdoš
<b>Účel</b>	Správne upravený variant			
<b>Vstupné podmienky</b>		Vytvorený variant		
<b>Výstupné podmienky</b>		Upravený variant		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme nový variant.	Nový variant v systéme.	Nový variant v systéme.	
2	Úprava variantu.	Upravený variant.	Upravený variant.	
3	Zobrazenie upraveného variantu.	Zobrazený upravený variant.	Zobrazený upravený variant.	

<b>Názov testu</b>		Delete a variant via the index page	<b>ID</b>	3-3-4
<b>Rozhranie</b>	Zmazanie variantu		<b>autor</b>	Martin Gajdoš
<b>Účel</b>	Správne zmazanie variantu			
<b>Vstupné podmienky</b>		Vytvorený variant		
<b>Výstupné podmienky</b>		Zmazaný variant		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Pridáme nový variant.	Nové variant v systéme.	Nové variant v systéme.	
2	Zmazanie variantu.	Zmazaný variant.	Zmazaný variant.	
3	Zobrazenie variantov.	Zmazaný variant nie je zobrazený.	Zmazaný variant nie je zobrazený.	

### 3.4 CRUD – typy vzťahov

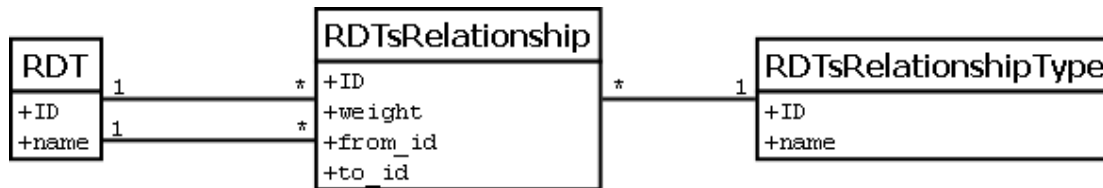
Ako autor chcem vytvoriť typ vzťahu medzi dvomi RDT, aby sa pri vytváraní vzťahu dal vybrať jeden z mnou vytvorených.

#### 3.4.1 Analýza

Po predchádzajúcom šprinte sú implementované reprezentácie termu a vzťahu medzi termami. Každý vzťah má atribút typ. Používateľ bude mať možnosť vytvoriť množinu typov vzťahov, z ktorých bude mať používateľ možnosť vybrať pri vytváraní nového vzťahu medzi termami.

### 3.4.2 Návrh

Používateľ môže vytvárať, upravovať a odstraňovať typy vzťahov medzi dvomi RDT. Vytvorená bude entita *RDTsRelationshipType*, s atribútom názov, ktorá bude prepojená na entitu *RDTsRelationship* (obr. 5). Po vytvorení typu vzťahu sa typ uloží do databázy a používateľ bude mať k nemu prístup cez používateľské rozhranie. Pri vytváraní vzťahu sa vo formulári v poli „Type“ zobrazia na výber existujúce typy vzťahov.



Obr. 5. Začlenenie typu vzťahu do doterajšieho modelu.

### 3.4.3 Implementácia

Rozdelenie implementácie medzi dátovú, aplikačnú a prezentačnú vrstvu zodpovedá využitiu vzoru MVC, danému frameworkom Ruby on Rails.

Implementovaný bol model s názvom *RDTsRelationshipType* a povinným atribútom *name*. V aplikačnej časti sú implementované funkcie na vytvorenie, zmazanie, upravenie typu vzťahu medzi dvomi termami.

### 3.4.4 Testovanie

Funkcionalitu overujú nasledovné akceptačné testy:

<b>Názov testu</b>	Vytvorenie typu vzťahu medzi RDT	<b>ID</b>	3-4-1
<b>Rozhranie</b>	Formulár vytvárania typu vzťahu medzi RDT	<b>autor</b>	Petra Vrablcová
<b>Účel</b>	Overenie funkcionality vytvárania vzťahu medzi RDT		
<b>Vstupné podmienky</b>	Používateľ chce vytvoriť nový typ vzťahu medzi termami.		
<b>Výstupné podmienky</b>	Je vytvorený nový typ vzťahu medzi termami a je vidieť v použ. rozhraní.		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Používateľ zvolí, že chce vytvoriť nový typ vzťahu medzi termami.	Dôjde k presmerovaniu na formulár pre vytváranie typu vzťahu medzi termami.	Presmerované na stránku pre vytváranie typu vzťahu medzi termami.
2	Používateľ vyplní názov typu vzťahu medzi termami.	V poli formulára „Name“ bude zobrazený zadaný názov typu vzťahu medzi termami.	Vo formulári bol zobrazený používateľom zadaný názov typu vzťahu medzi termami.

3	Používateľ potvrdí, že chce vytvoriť typ vzťahu so zadaným názvom.	Dôjde k presmerovaniu na stránku so zobrazeným vytvoreným typom vzťahu.	Presmerované na stránku so zobrazeným vytvoreným typom vzťahu.
---	--	---	--

<b>Názov testu</b>		Upravenie typu vzťahu medzi RDT	<b>ID</b>	3-4-2
<b>Rozhranie</b>	Formulár úpravy typu vzťahu medzi RDT		<b>autor</b>	Petra Vrablecová
<b>Účel</b>	Overenie funkcionality úpravy typu vzťahu medzi RDT			
<b>Vstupné podmienky</b>	Je vytvorený typ vzťahu medzi termami. Používateľ chce upraviť existujúci typ vzťahu.			
<b>Výstupné podmienky</b>	Typ vzťahu je upravený podľa používateľa a používateľ ho vidí v používateľskom rozhraní.			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí, že chce upraviť jeden typ vzťahu medzi termami zo zoznamu existujúcich typov.	Dôjde k presmerovaniu na formulár pre úpravu typu vzťahu.	Presmerované na stránku pre úpravu typu vzťahu.	
2	Používateľ zmení názov typu vzťahu.	V poli formulára „Name“ bude zobrazený zadaný názov typu vzťahu.	Vo formulári bol zobrazený používateľom zadaný názov typu vzťahu.	
3	Používateľ potvrdí, že chce vytvoriť typ vzťahu so zadaným názvom.	Dôjde k presmerovaniu na stránku so zobrazeným vytvoreným typom vzťahu.	Presmerované na stránku so zobrazeným vytvoreným typom vzťahu.	

<b>Názov testu</b>		Vymazanie typu vzťahu medzi RDT	<b>ID</b>	3-4-3
<b>Rozhranie</b>	Formulár zmazanie typu vzťahu medzi RDT		<b>autor</b>	Petra Vrablecová
<b>Účel</b>	Overenie funkcionality zmazania typu vzťahu medzi RDT			
<b>Vstupné podmienky</b>		Je vytvorený typ vzťahu. Používateľ chce zmazať existujúci typ vzťahu.		
<b>Výstupné podmienky</b>		Typ vzťahu je zmazaný a používateľ ho už nevidí v použ. rozhraní.		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí, že chce zmazať jeden typ vzťahu zo zoznamu existujúcich typov vzťahov.	Dôjde k zobrazeniu potvrdzujúceho dialógu.	Zobrazil sa potvrdzujúci dialóg.	
2	Používateľ zvolí, že chce zvolený typ vzťahu zmazať.	Dôjde k presmerovaniu na stránku so zobrazeným zoznamom existujúcich typov vzťahov, ktorý neobsahuje zmazaný typ vzťahu.	Presmerované na stránku so zobrazeným zoznamom existujúcich typov vzťahov, ktorý neobsahoval zmazaný typ vzťahu.	

<b>Názov testu</b>		Vytvorenie vzťahu medzi dvomi RDT	<b>ID</b>	3-4-4
<b>Rozhranie</b>	Formulár vytvárania vzťahu medzi dvomi RDT		<b>autor</b>	Petra Vrablecová
<b>Účel</b>	Overenie funkcionality výberu typu vzťahu v rozhraní pri vytváraní vzťahu.			
<b>Vstupné podmienky</b>		Existujú aspoň dva termy a aspoň jeden typ vzťahu. Používateľ chce vytvoriť neorientovaný vzťah medzi dvomi termami.		
<b>Výstupné podmienky</b>		Sú vytvorené dva nové orientované vzťahy a používateľ ich vidí v použ. rozhraní. Používateľ vybral typ vzťahu zo zoznamu existujúcich.		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí, že chce vytvoriť nový vzťah.	Dôjde k presmerovaniu na formulár pre vytváranie vzťahu.	Presmerované na stránku pre vytváranie vzťahu.	

2	Používateľ zvolí typ vzťahu, vyplní váhu vzťahu a zvolí dva termy, medzi ktorými má byť vytvorený vzťah a zvolí, či má byť vzťah orientovaný.	V poli formulára „Type“ bude zobrazený vybraný typ vzťahu, v poli „Weight“ bude zobrazená zadaná váha vzťahu, v poliach „From“ a „To“ bude v každom vybraný jeden term, pole „Oriented“ nebude označené.	Vo formulári boli zobrazené používateľom zadané atribúty vzťahu.
3	Používateľ potvrdí, že chce vytvoriť vzťah so zadanými atribútmi.	Dôjde k presmerovaniu na stránku so zobrazenými vytvorenými vzťahmi.	Presmerované na stránku so zobrazenými dvomi vytvorenými orientovanými vzťahmi.

### 3.5 CRUD – vzťahy medzi RDT a dokumentmi

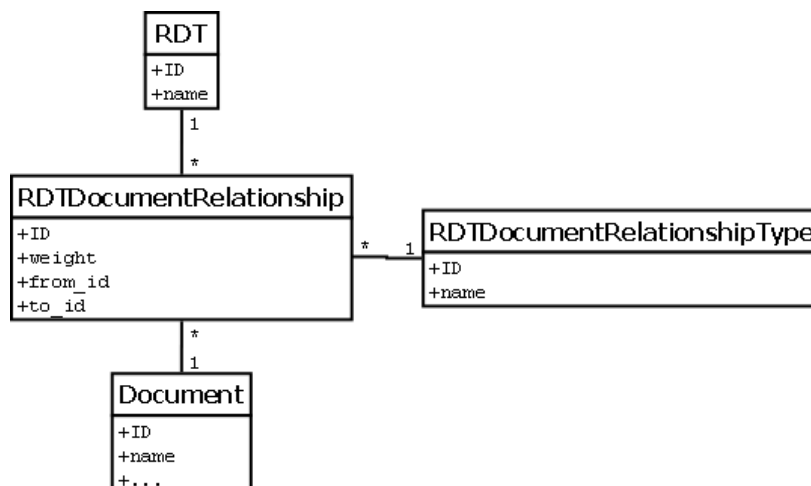
Ako autor chcem priradiť termy k dokumentu, aby bol jeho obsah označenými nejakými kľúčovými slovami, na základe ktorých sa dá vyhľadávať, odporúčať.

#### 3.5.1 Analýza

Po predchádzajúcom šprinte sú implementované reprezentácie termu a dokumentu. Používateľ bude mať možnosť priradiť jeden alebo viacero termov k vybranému dokumentu.

#### 3.5.2 Návrh

Používateľ môže vytvárať, upravovať a odstraňovať vzťahy medzi dokumentmi a termami. Používateľ môže vytvárať, upravovať a odstraňovať typy vzťahov medzi termami a dokumentmi. Vytvorená bude entita *RDTDocumentRelationshipType* s atribútom názov. Ďalej bude vytvorená entita *RDTDocumentRelationship*, s atribútmi váha, ID RDT a ID dokumentu, medzi ktorými je vzťah. Bude prepojená na entity *RDT*, *Document* a *RDTDocumentRelationshipType* (obr. 6). Po vytvorení typu vzťahu alebo vzťahu bude typ vzťahu alebo vzťah uložený v databáze a používateľ bude mať k nemu prístup cez používateľské rozhranie. Pri vytváraní vzťahu sa vo formulári v poli „Type“ zobrazia na výber existujúce typy vzťahov.



Obr. 6. Model vzťahu medzi dokumentom a RDT.

### 3.5.3 Implementácia

Rozdelenie implementácie medzi dátovú, aplikačnú a prezentačnú vrstvu zodpovedá využitiu vzoru MVC, danému frameworkom Ruby on Rails.

Implementované boli modely s názvom *RDTDocumentRelationshipType* a *RDTDocumentRelationship*. Model *RDTDocumentRelationshipType* má povinný atribút *name*. Model *RDTDocumentRelationship* má povinné atribúty *weight* (váha), *from\_id* (ID RDT) a *to\_id* (ID dokumentu). V aplikačnej časti sú implementované funkcie na vytvorenie, zmazanie, upravenie typu vzťahu medzi dokumentom a RDT a obdobné funkcie pre prácu so vzťahom medzi dokumentom a RDT.

### 3.5.4 Testovanie

Akceptačné testy:

Testy pre otestovanie vytvárania, úpravy a umazania typu vzťahu medzi dokumentom a RDT sú obdobné ako testy pre otestovanie práce s typom vzťahu medzi dvomi RDT z predchádzajúceho šprintu.

<b>Názov testu</b>		Priradenie RDT k dokumentu	<b>ID</b>	3-5-1
<b>Rozhranie</b>	Stránka so zobrazeným dokumentom.		<b>autor</b>	Petra Vrablecová
<b>Účel</b>	Overenie funkcionality priradenia RDT k dokumentu.			
<b>Vstupné podmienky</b>	Existuje aspoň jeden term, jeden dokument a aspoň jeden typ vzťahu. Používateľ chce vytvoriť vzťah medzi dokumentom a termom.			
<b>Výstupné podmienky</b>	RDT je priradené k dokumentu.			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí, že chce priradiť RDT k dokumentu.	Dôjde k presmerovaniu na formulár pre priradovanie RDT k dokumentu.	Presmerované na stránku pre priradovanie RDT k dokumentu.	
2	Používateľ zvolí typ vzťahu, vyplní váhu vzťahu a zvolí term, ktorý má byť priradený k dokumentu.	V poli formulára „Type“ bude zobrazený vybraný typ vzťahu, v poli „Weight“ bude zobrazená zadaná váha vzťahu, v poli „RDT“ bude vybraný jeden term.	Vo formulári boli zobrazené používateľom zadané atribúty vzťahu.	



3	Používateľ potvrdí, že chce vytvoriť vzťah so zadanými atribútmi.	Dôjde k presmerovaniu na stránku so zobrazeným dokumentom, kde bude vidno nový vzťah.	Presmerované na stránku so zobrazeným dokumentom, pod ktorým pribudol nový vzťah s priradeným RDT.
---	---	---	--

### 3.6 CRUD – vzťahy medzi dokumentmi

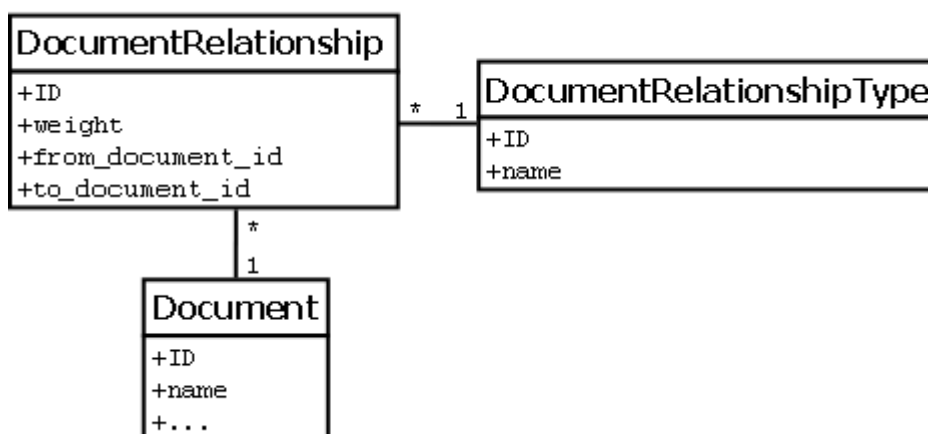
Ako autor chcem manažovať vzťahy medzi dokumentmi, aby bolo možné odkázať z jedného dokumentu na iné dokumenty.

#### 3.6.1 Analýza

V predchádzajúcom šprinte bola implementovaná reprezentácia dokumentu. Používateľ bude mať možnosť priradiť k dokumentu jeden alebo viac dokumentov.

#### 3.6.2 Návrh

Používateľ môže vytvárať, zobrazovať, upravovať a mazať vzťahy medzi dokumentmi. Používateľ môže vytvárať, zobrazovať, upravovať a mazať typy vzťahov medzi dokumentmi. Je vytvorená entita *DocumentRelationship* s atribútmi *weight*, *document\_type\_id*, *from\_document\_id* a *to\_document\_id*, prepojená na entitu *Document* a entita *DocumentRelationshipType* s atribútom *name*, prepojená na entitu *DocumentRelationship* (obr. 7). Po vytvorení typu vzťahu alebo vzťahu bude typ vzťahu alebo vzťah uložený v databáze a používateľ bude mať k nemu prístup cez používateľské rozhranie. Pri vytváraní vzťahu sa vo formulári v poli „Type“ zobrazia na výber existujúce typy vzťahov.



Obr. 7. Model vzťahu medzi dokumentmi.

#### 3.6.3 Implementácia

Rozdelenie implementácie medzi dátovú, aplikačnú a prezentačnú vrstvu zodpovedá využitiu vzoru MVC, danému frameworkom Ruby on Rails.

Implementované boli modely s názvom *DocumentRelationship* a *DocumentRelationshipType*. Model *DocumentRelationship* má povinné atribúty *weight*, *relationship\_type\_id*, *from\_document\_id* a *to\_document\_id*. Model *RDTDocumentRelationshipType* má povinný atribút *name*. V aplikačnej

časti sú implementované funkcie na vytvorenie, zmazanie, zobrazenie a upravenie vzťahu a typu vzťahu medzi dokumentmi.

### 3.6.4 Testovanie

Akceptačné testy:

<b>Názov testu</b>	Priradenie vzťahu k dokumentom	<b>ID</b>	3-6-1
<b>Rozhranie</b>	Stránka so zobrazeným dokumentom.	<b>autor</b>	Zuzana Zimová
<b>Účel</b>	Overenie funkcionality priradenia vzťahu k dokumentom		
<b>Vstupné podmienky</b>	Existujú aspoň dva rôzne dokumenty a aspoň jeden typ vzťahu. Používateľ chce priradiť vzťah k dvom rôznym dokumentom.		
<b>Výstupné podmienky</b>	Je priradený vzťah k dvom rôznym dokumentom.		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Používateľ zvolí, že chce vytvoriť vzťah medzi dvoma dokumentmi.	Dôjde k presmerovaniu na formulár pre vytvorenie vzťahu medzi dokumentmi.	Presmerované na stránku pre vytvorenie vzťahu medzi dokumentmi.
2	Používateľ zvolí typ vzťahu, vyplní váhu vzťahu, zvolí, či má byť vzťah orientovaný a zvolí dva rôzne dokumenty, medzi ktorými má byť vytvorený vzťah.	V poli formulára „Type“ bude zobrazený vybraný typ vzťahu, v poli „Weight“ bude zobrazená zadaná váha vzťahu, v poli „From document“ bude vybraný jeden dokument a v poli „To document“ bude vybraný iný dokument.	Vo formulári boli zobrazené používateľom zadané atribúty vzťahu.
3	Používateľ potvrdí, že chce vytvoriť vzťah so zadanými atribútmi.	Dôjde k presmerovaniu na stránku so zobrazeným dokumentom, kde bude viditeľný nový vzťah.	Presmerované na stránku so zobrazenými vzťahmi medzi dokumentmi.

## 3.7 Validácia dokumentu voči XML schéme (XSD, RELAX NG)

Ako používateľ chcem, aby sa pri uložení dokumentu validoval jeho obsah voči schéme. Zároveň chcem, aby ma systém upozornil na chyby v dokumente.

### 3.7.1 Analýza

Systém dovoľuje vytvárať a meniť obsah dokumentov. Ten bude reprezentovať text vo formáte DocBook. Keďže sa jedná o XML, bude potrebné overovať správnosť jeho štruktúry pomocou schémy, aby sa predišlo chybám. Na popis schémy máme na výber z jazykov XSD alebo RELAX NG.

Spustenie validácie zrejme treba vždy, keď sa používateľ pokúsi uložiť obsah dokumentu. Výnimkou budú prípady, keď je treba zámerne uložiť obsah aj s chybami. Medzi takéto patrí napríklad prerušenie práce používateľa a dopracovanie zmien dokumentu v iný čas. Používateľ by mal mať možnosť vypnúť validáciu.

Ak sa v dokumente nachádzajú chyby, je potrebné vypísať. Aby ich používateľ mohol odstrániť, musia chybové výpisy obsahovať dostatočné informácie.

### 3.7.2 Návrh

Ak chce používateľ uložiť obsah dokumentu pri jeho vytváraní alebo editovaní, systém ho pred predtým validuje voči schéme. Pri nájdení chýb v dokumente sa tieto zobrazia pod poľom, do ktorého sa zadáva samotný obsah dokumentu. Výpis chyby obsahuje popis s miestom jej výskytu. Dokument sa neuloží, ak obsahuje aspoň jednu chybu. Používateľ môže validáciu vypnúť pomocou príslušného checkboxu.

### 3.7.3 Implementácia

Na overovanie obsahu voči schéme je použitý gem Nokogiri. Jedná sa o wrapper pre knižnicu libxml. Samotná validácia sa vykonáva metódou `validate`, ktorá je definovaná v modeli dokumentu. Jej návratová hodnota je pole s chybami v dokumente. Metóda overovania je spustená pri vyvolaní akcie na uloženie dokumentu. Uloženie sa nevykoná, ak pole chýb nie je prázdne. V opačnom prípade sa uložia všetky zmeny a zmení sa verzia dokumentu.

Schéma určená na validáciu obsahu je zapísaná v jazyku RELAX NG. Pred jej použitím je spracovaná na inštanciu triedy `Nokogiri::XML::RelaxNG`.

Výpis chýb sa zabezpečuje pomocou návratovej hodnoty metódy `validate`, ktorá predstavuje zoznam chýb. Vo formulári pre prácu s dokumentom sa pristupuje ku každej štruktúre v zozname a zobrazí sa jej obsah.

Vypisujú sa nasledovné atribúty:

- `code` - kód chyby,
- `level` – level atribútu,
- `line` – riadok, v ktorom sa chyba nachádza,
- `message` – text chyby,
- `str1` – názov elementu.

## 3.7.4 Testovanie

<b>Názov testu</b>		Zobrazenie chýb v dokumente	<b>ID</b>	3-7-1
<b>Rozhranie</b>	Obrazovka editácie alebo vytvorenia dokumentu		<b>autor</b>	Martin Habdák
<b>Účel</b>	Upozornenie na chyby v dokumente			
<b>Vstupné podmienky</b>	Obsah dokumentu má aspoň 1 chybu, používateľ je prihlásený			
<b>Výstupné podmienky</b>	Zoznam chýb v dokumente			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Vyplníme obsah dokumentu a odošleme formulár.	Zobrazenie obrazovky editácie alebo vytvorenia dokumentu, výpis chýb v dokumente.	Zobrazená bola obrazovka editácie alebo vytvorenia dokumentu, boli vypísané chyby v dokumente.	

<b>Názov testu</b>		Uloženie validného dokumentu	<b>ID</b>	3-7-2
<b>Rozhranie</b>	Obrazovka editácie alebo vytvorenia dokumentu		<b>autor</b>	Martin Habdák
<b>Účel</b>	Uloženie dokumentu po validácii			
<b>Vstupné podmienky</b>	Obsah dokumentu neobsahuje ani 1 chybu, používateľ je prihlásený			
<b>Výstupné podmienky</b>	Uložený dokument je validný			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Vyplníme obsah dokumentu a odošleme formulár.	Uloženie dokumentu a presmerovanie na obrazovku dokumentu, výpis chýb v dokumente.	Dokument bol uložený a presmerované na obrazovku dokumentu.	

## 3.8 Prihlásenie používateľov

Ako používateľ sa chcem prihlásiť aby som mohol pristupovať k funkciám systému. Súčasne chcem aby ma systém odlišoval od iných používateľov.

### 3.8.1 Analýza

V systéme by mali pracovať len autorizovaní používatelia. Zabráni sa tak nedovolenému manipulovaniu a poškodeniu entít. Preto sa každý používateľ musí prihlásiť pred tým, ako začne pracovať v systéme. Po prihlásení môže začať s prácou.

V systéme by malo byť viac používateľov. Mali by byť od seba rozlišovaní na základe určitého atribútu. Pri práci s dokumentmi alebo s inými entitami v systéme sa potom dá určiť, ktorý používateľ s nimi manipuloval.

Osobné údaje používateľa použité na jeho identifikáciu musia byť ochránené proti zneužitiu. Používateľ by mal mať možnosť zmeniť tieto údaje.

### 3.8.2 Návrh

Používateľ bude definovaný dvomi atribútmi – email a heslo. Heslo používateľa by nemalo byť uložené v jeho základnej podobe. Treba ho zatajiť. Preto bude zašifrované. Atribút email bude pre používateľa jednoznačným identifikátorom. Pred používaním systému sa používateľ prihlási pomocou svojho emailu a hesla. Odhlásiť sa môže v ľubovoľnom momente. Používateľ zostane prihlásený, pokiaľ sa neodhlási, alebo nevypne prehliadač.

Ak používateľ pracuje so systémom prvýkrát, je mu ponúknutá možnosť vytvoriť si používateľský účet. V prípade, že zabudne svoje prístupové údaje, bude mu umožnené získať ich späť.

Po odhlásení bude používateľ navrátený na úvodnú obrazovku.

### 3.8.3 Implementácia

Používateľ je v systéme vytvorený pomocou gemu „Devise“. Je to rozšírený gem s dostatočnou funkcionalitou. Informácie o používateľovi uchováva v tabuľke „Users“, ktorá obsahuje viacero atribútov. Používateľ je identifikovaný atribútom email. Heslo sa ukladá zašifrované z dôvodu bezpečnosti.

Prihlásenie je realizované pomocou formulára na uvítacej obrazovke. Obsahuje pole pre email a heslo používateľa. Prihlásenie sa vykoná stlačením tlačidla „Sign in“. Používateľ má možnosť zapamätania v systéme. Urobí tak zaškrtnutím checkboxu s názvom „Remember me“. Výsledkom je, že pri ďalšom spustení systému je automaticky autentifikovaný bez zadávania prihlasovacích údajov.

Formulár obsahuje aj možnosť vytvorenia nového používateľského konta. Proces vytvorenia sa vykoná kliknutím na odkaz s názvom „Sign up“. Systém ponúka formulár, vyplnením ktorého sa vytvorí nové používateľské konto.

V prípade zabudnutých prihlasovacích údajov sa poskytuje možnosť ich získania. Túto funkcionalitu tiež zabezpečuje implementácia pomocou gemu „Devise“.

Na odhlásenie používateľa je vytvorené menu. Je vložené do šablóny aplikácie a nachádza sa v pravom hornom rohu. Používateľovi umožňuje kedykoľvek sa odhlásiť. Zobrazuje aj jeho email.

### 3 Šprint č. 2 – Beethoven

Integrovanie používateľa do systému je zabezpečené vložím riadku „before\_filter: :authenticate\_user!“ do application controllera. Všetky ostatné controllery zedia túto vlastnosť. Výsledkom je, že pred vykonaním ľubovoľnej akcie sa v každom z nich musí overiť identita používateľa.

#### 3.8.4 Testovanie

Funkcionalita bola otestovaná pomocou metódy čiernej skrinky.

<b>Názov testu</b>	Odhlásenie	<b>ID</b>	3-8-1
<b>Rozhranie</b>	Všetky obrazovky systému	<b>autor</b>	Martin Habdák
<b>Účel</b>	Odhlásenie používateľa		
<b>Vstupné podmienky</b>	Používateľ je prihlásený		
<b>Výstupné podmienky</b>	Používateľ je odhlásený		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Klikneme na odkaz Sign out.	Odhlásenie používateľa, presmerovanie na privítaciu obrazovku.	Odhlásenie používateľa, presmerované na privítaciu obrazovku.

<b>Názov testu</b>	Prihlásenie	<b>ID</b>	3-8-2
<b>Rozhranie</b>	Úvodná obrazovka systému	<b>autor</b>	Martin Habdák
<b>Účel</b>	Odhlásenie používateľa		
<b>Vstupné podmienky</b>	Používateľ je odhlásený		
<b>Výstupné podmienky</b>	Používateľ je prihlásený		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Vyplníme prihlasovacie údaje a odošleme formulár .	Prihlásenie používateľa, zobrazenie úvodnej obrazovky.	Prihlásenie používateľa, zobrazila sa úvodná obrazovka.

<b>Názov testu</b>		Vytvorenie používateľského konta	<b>ID</b>	3-8-3
<b>Rozhranie</b>	Úvodná obrazovka systému		<b>autor</b>	Martin Habdák
<b>Účel</b>	Pridanie nového používateľského konta do systému			
<b>Vstupné podmienky</b>		Neexistencia používateľského konta s daným emailom		
<b>Výstupné podmienky</b>		Vytvorené nové používateľské konto		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Klikneme na odkaz „Sign up“.	Zobrazenie formuláru na vytvorenie nového používateľského konta.	Zobrazil sa formulár na vytvorenie nového používateľského konta.	
2	Vyplníme email a heslo a odošleme formulár.	Vytvorenie nového používateľského konta, presmerovanie na obrazovku repozitárov.	Vytvorilo sa nové používateľské konto, presmerované na obrazovku repozitárov.	

<b>Názov testu</b>		Vytvorenie existujúceho používateľského konta	<b>ID</b>	3-8-4
<b>Rozhranie</b>	Úvodná obrazovka systému		<b>autor</b>	Martin Habdák
<b>Účel</b>	Upozornenie na existenciu používateľského konta			
<b>Vstupné podmienky</b>		Existencia používateľského konta s daným emailom		
<b>Výstupné podmienky</b>		Zobrazenie chyby		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Klikneme na odkaz „Sign up“.	Zobrazenie formuláru na vytvorenie nového používateľského konta.	Zobrazil sa formulár na vytvorenie nového používateľského konta.	
2	Vyplníme email a heslo existujúceho používateľského konta a odošleme formulár.	Zobrazenie chybovej hlášky o existencii používateľského konta so zadaným emailom.	Zobrazila sa chybová hláška o existencii používateľského konta so zadaným emailom.	

### 3.9 CRUD – obrázky

Ako autor chcem vytvoriť, zobrazovať a pridávať obrázky do dokumentov. Taktiež upravovať a mazať samotné obrázky a informácie o nich.

#### 3.9.1 Analýza

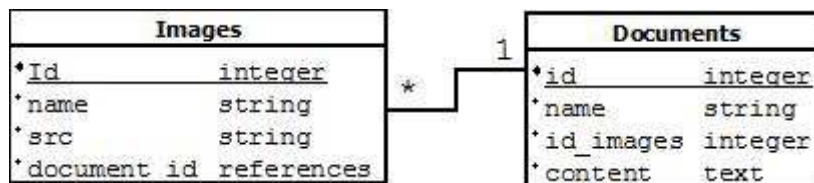
Entita dokument, ktorá bola vytvorená v predchádzajúcom šprinte tvorí základ pre entitu obrázkov. Užívateľ bude mať možnosť pri vytváraní a prezeraní dokumentu vytvoriť a upravovať obrázok pripojený k tomuto dokumentu.

#### 3.9.2 Návrh

Autor dokumentu bude mať možnosť pridávať do dokumentu obrázky. Bude naprogramovaná entita obrázky, ktorá bude obsahovať dve dátové zložky a dva kľúče s toho jeden bude cudzí kľúč, ktorý bude slúžiť pre asociáciu z dokumentmi a druhý kľúč bude primárny. Prvý typ bude dátovej zložky bude názov obrázku a druhý jeho adresa pre uloženie do databáze. Pre používateľa bude vytvorené používateľské rozhranie, kde by mohol zobrazovať dáta z databáze, upravovať, mazať a zobrazovať obrázky prislúchajúce k dokumentu.

#### 3.9.3 Implementácia

Entita obrázky splňuje požiadavky architektúry Ruby on Rails pre model MVC, ktorý ju delí na tri časti a to model, pohľad a controller. Ako prvý je vytvorený controller pre entitu obrázky s názvom *Images*. Ten obsahoval kód pre vkladanie, upravovanie, mazanie a zobrazovanie vložených obrázkov. Následne bol vytvorený model, ktorý obsahoval tabuľku *Image* s dátovými zložkami. V tabuľke sa nachádzajú povinné atribúty *id*, *name*, *src* a atribút *document\_id* slúžiaci na prepojenie obrázkov a dokumentu.



Obr. 8. Vzťah medzi entitou dokument a obrázok.



### 3.9.4 Testovanie

Akceptačné testy:

Testy pre otestovanie vytvárania, úpravy a umazania obrázkov.

<b>Názov testu</b>		Výpis dostupných obrázkov	<b>ID</b>	3-9-1
<b>Rozhranie</b>		Stránka so zobrazenými obrázkami.	<b>autor</b>	Róbert Kocian
<b>Účel</b>		Overenie funkcionality pre výpis obrázkov		
<b>Vstupné podmienky</b>		Používateľ chce zobraziť všetky obrázky v databáze		
<b>Výstupné podmienky</b>		Zobrazenie informácií vytvoreného obrázka a hlásenia o úspešnom vytvorení		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ chce zobraziť všetky obrázky.	Dôjde k presmerovaniu na stránku index.	Výpis všetkých obrázkov.	

<b>Názov testu</b>		Zobrazenie obrázka	<b>ID</b>	3-9-2
<b>Rozhranie</b>		Stránka so zobrazenými obrázkami.	<b>autor</b>	Róbert Kocian
<b>Účel</b>		Overenie funkcionality pre zobrazenie obrázka		
<b>Vstupné podmienky</b>		Používateľ chce zobraziť nový obrázok		
<b>Výstupné podmienky</b>		Zobrazenie informácií vytvoreného obrázka a hlásenia o úspešnom vytvorení		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí, že chce vytvoriť obrázok k dokumentu.	Dôjde k presmerovaniu na formulár pre vytvorenie obrázkov k dokumentu.	Presmerované na stránku pre vytvorenie obrázka.	
2	Používateľ zvolí typ dokumentu, vyplní názov obrázka a jeho src adresu.	Zobrazenie informácií o konkrétnom obrázku.	Presmerované na stránku so zobrazenými informáciami obrázkov.	

<b>Názov testu</b>	Vytvorenie obrázka	<b>ID</b>	3-9-3
<b>Rozhranie</b>	Stránka so zobrazeným dokumentom.	<b>autor</b>	Róbert Kocian
<b>Účel</b>	Overenie funkcionality pre vytvorenie obrázka		
<b>Vstupné podmienky</b>	Používateľ chce vytvoriť nový obrázok		
<b>Výstupné podmienky</b>	Zobrazenie informácií vytvoreného obrázka a hlásenia o úspešnom vytvorení		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Používateľ zvolí, že chce vytvoriť obrázok k dokumentu.	Dôjde k presmerovaniu na formulár pre vytvorenie obrázkov k dokumentu.	Presmerované na stránku pre vytvorenie obrázka.
2	Používateľ zvolí typ dokumentu, vyplní názov obrázka a jeho adresu.	V poli pre dokument bude zobrazený názov dokumentu.	Presmerované na stránku so zobrazeným informáciami obrázkov.

<b>Názov testu</b>	Upravenie obrázka	<b>ID</b>	3-9-4
<b>Rozhranie</b>	Stránka s výpisom všetkých dokumentov.	<b>autor</b>	Róbert Kocian
<b>Účel</b>	Overenie funkcionality pre upravenie obrázka		
<b>Vstupné podmienky</b>	Používateľ chce upraviť príslušný obrázok		
<b>Výstupné podmienky</b>	Zobrazenie informácií upraveného obrázka		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Používateľ zvolí u príslušného obrázka z výpisu obrázkov, tlačidlo edit.	Dôjde k presmerovaniu na formulár pre upravenie obrázkov.	Presmerované na stránku pre upravenie obrázka.
2	Používateľ zvolí typ dokumentu, vyplní názov obrázka a jeho adresu.	V poli pre dokument bude zobrazený názov dokumentu.	Presmerované na stránku so zobrazeným informáciami obrázkov.

<b>Názov testu</b>	Vymazanie obrázka		<b>ID</b>	3-9-5
<b>Rozhranie</b>	Stránka s výpisom všetkých dokumentov.		<b>autor</b>	Róbert Kocian
<b>Účel</b>	Overenie funkcionality pre vymazanie obrázka			
<b>Vstupné podmienky</b>	Používateľ chce vymazať príslušný obrázok			
<b>Výstupné podmienky</b>	Vymazanie obrázka			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí u príslušného obrázka z výpisu obrázkov, tlačidlo destroy.	Dôjde k vymazaniu príslušného obrázka.	Presmerované na stránku pre výpis obrázkov.	



## 4 Šprint č. 3 – Chopin

### 4.1 Presun dokumentov medzi repozitármi

Úloha bola zameraná na presun jedného alebo viacerých dokumentov z aktuálneho repozitára do zvoleného repozitára.

#### 4.1.1 Analýza

Po zobrazení detailu repozitára máme možnosť vidieť zoznam dokumentov, ktoré obsahuje. Pri práci s vybraným repozitárom je vhodné mať možnosť editovať aj jeho obsah, teda pracovať s dokumentmi, ktoré v ňom sú, čo okrem vytvárania, zobrazovania a mazania dokumentov, znamená aj ich presúvanie medzi dvomi repozitármi.

#### 4.1.2 Návrh

Feature je začlenená do obrazovky RD1, čiže detail repozitára. Používateľ si môže označiť jeden alebo viacero dokumentov, zvolí jeden z existujúcich repozitárov a presunúť zvolené dokumenty z aktuálneho repozitára do zvoleného repozitára. Feature pracuje s entitami *Document* a *Repository* a s atribútom *repository\_id*. Po presunutí dokumentov sa zobrazí detail aktuálneho repozitára, z ktorého boli dokumenty presunuté. K presunutým dokumentom môže používateľ pristupovať z jeho nového umiestnenia.

#### 4.1.3 Implementácia

Implementácia bola prevedená v controlleri *documents\_controller* a vo view *repositories\\_documents*. Implementovaná bola metóda *move*, ktorá napĺňala samotnú funkcionálnu feature zmenou ID rodičovského repozitára vybraných dokumentov. Okrem toho bola doplnená možnosť výberu jedného a viacerých dokumentov v detaile repozitára a možnosť výberu cieľového repozitára z existujúcich, takisto v obrazovke RD1.

#### 4.1.4 Testovanie

Akceptačné testy:

<b>Názov testu</b>	Presun dokumentov medzi repozitármi	<b>ID</b>	4-1-1
<b>Rozhranie</b>	Stránka s detailom repozitáru (RD1)	<b>autor</b>	Zuzana Zimová
<b>Účel</b>	Overenie funkcionality presunu dokumentov medzi repozitármi		
<b>Vstupné podmienky</b>	Existujú aspoň dva rôzne repozitáre a aspoň v jednom z nich je umiestnený aspoň jeden dokument.		
<b>Výstupné podmienky</b>	Dokument bol premiestnený z aktuálneho repozitára do cieľového.		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Používateľ zvolí jeden alebo viac dokumentov zo zoznamu.	Vybrané dokumenty budú označené.	Vybrané dokumenty sú označené.

## 4 Šprint č. 3 – Chopin

2	Používateľ zvolí cieľový repozitár zo zoznamu repozitárov.	V poli bude zobrazený názov cieľového repozitára.	V poli je zobrazený názov cieľového repozitára.
3	Používateľ potvrdí, že chce presunúť zvolené dokumenty do zvoleného repozitára.	Dôjde k zmene ID rodičovského repozitára na ID cieľového repozitára u všetkých označených dokumentov.	Dokumenty sa nezobrazujú v aktuálnom repozitári, ale v cieľovom.

### 4.2 Zobrazenie grafu variantu metadát

Úloha sa zameriavala na vizualizáciu metadát prislúchajúcim k danému variantu, prostredníctvom grafu.

#### 4.2.1 Analýza

V časti analýzy sme sa snažili zistiť, ktorá z možností pre zobrazovanie grafu by bola najvhodnejšia. Medzi kandidátov patrili:

- Java Applet,
- Flash,
- Silverlight,
- Flex,
- jQueryUI,
- knižnice JavaScriptu.

K výslednému rozhodnutiu sme sa dopracovali vylučovacou metódou. Java applety sú už zastarané a nie príliš podporované. S technológiami Flash, Silverlight a Flex nikto z tímu zatiaľ nepracoval, preto by bol tento prístup príliš náročný. jQueryUI je rozšírenie knižnice jQuery o grafické prvky, no implementovanie vlastného grafu by s ním bolo taktiež príliš komplikované. Nakoniec sme po odporúčení vedúceho tímu zvolili JavaScript knižnicu Jit, ktorá sa špecializuje na vizualizáciu grafov a diagramov.

Po jej bližšom preskúmaní sme zistili, že poskytuje základné funkcionality pre vizualizáciu a úpravy grafu. Z pomedzi ostatných kandidátov (Dracula, Prefuse, CanViz...) sa teda javila ako najvhodnejšia.

#### 4.2.2 Návrh

Graf mal byť začlenený do obrazovky V1. V nej mala byť poskytnutá možnosť prepnutia medzi tabuľkové zobrazenie variantu, alebo za pomoci nášho grafu. V rámci tejto obrazovky vlastne vznikla nová „podobrazovka“ so skratkou VG1.

Graf bol začlenený do vlastnej sekcie. Pomocou funkcií jQuery sme teda mohli zabezpečiť možnosť prepínania medzi tabuľkovým a grafovým zobrazením.

### 4.2.3 Implementácia

Aj keď sa jednalo o prvotné štádium začlenenia grafu do projektu, rozhodli sme sa ku nemu pridať aj výpis základných informácií. Medzi tie patrili:

- stav načítania grafu,
- výpis susedov zvoleného uzlu.

Pre zobrazovanie prislúchajúcich dát, bolo nutné zabezpečiť načítavanie nami zadaných hodnôt. Proces je nasledovný:

1. Do tela http odpovedi vo formáte html bolo začlenené vytvorenie JavaScriptovej premennej vo formáte JSON pre entity RDT a ich vzťahov.
2. Tieto premenné slúžili ako parametre pre funkciu zobrazovania grafu.
3. V rámci tejto funkcie sa premenné parsovali do vhodného formátu, z ktorého bola knižnica schopná vizualizovať graf.
4. Nastavili sme ostatné parametre pre zobrazovanie grafu (veľkosť uzlov, dĺžka hrán...)
5. Spustenie vizualizácie.

## 4.3 Zobrazenie variantu metadát ako zoznamu RDT a vzťahov

Používateľ chce po zvolení variantu vidieť tabuľku obsahujúcu RDT nachádzajúce sa vo variante a tabuľku obsahujúcu vzťahy medzi RDT z variantu.

### 4.3.1 Analýza

Tabuľka zobrazujúca zoznam RDT nachádzajúcich sa vo variante by mala byť abecedne zoraditeľná podľa názvu RDT, mala by byť filtrovateľná podľa názvu, mal by sa dať v nej označiť viacero prvkov, aby sa dali naraz zmazať a mala by mať pevnú výšku, v ktorej sa bude dať skrolovať. Tabuľka so vzťahmi by mala mať rovnaké vlastnosti až na to, že nebude skrolovateľná, ale stránkovacia.

### 4.3.2 Návrh

Vzhľad obrazovky zodpovedá návrhu obrazovky V1 v prílohe (Príloha A, časť A.5). Požadované vlastnosti tabuľky sú ľahko zabezpečiteľné pomocou gemu *datatables*.

### 4.3.3 Implementácia

View variantu bol implementovaný podľa návrhu. Controller variantu bol doplnený o metódy *destroy\_rdt* a *destroy\_rdtrelationships*, vďaka ktorým je možné vymazávať viacero položiek tabuľky naraz.

#### 4.3.4 Testovanie

Akceptačné testy:

<b>Názov testu</b>	Zobrazenie variantu	<b>ID</b>	4-3-1
<b>Rozhranie</b>	Úvodná stránka (obrazovka H1).	<b>autor</b>	Petra Vrablecová
<b>Účel</b>	Overenie funkcionality zobrazenia variantu vo forme tabuľky RDT a vzťahov.		
<b>Vstupné podmienky</b>	Používateľ chce zobrazit' variant vo forme tabuľky RDT a vzťahov.		
<b>Výstupné podmienky</b>	Variant je zobrazený v požadovanej forme.		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Používateľ zvolí zobrazenie jedného z variantov.	Dôjde k presmerovaniu na stránku variantu.	Presmerované na stránku variantu. Používateľ vidí dve tabuľky – tabuľku s RDT a tabuľku so vzťahmi.

#### 4.4 Zobrazenie View dokumentu

Ako používateľ chcem zobrazit' náhľad na obsah dokumentu.

##### 4.4.1 Analýza

Pri práci s dokumentmi je vhodné vidieť, ako bude dokument vyzerat', keď sa zobrazí vo webovom portáli. Obsah dokumentu je reprezentovaný vo formáte *DocBook* - XML. Na zobrazenie náhľadu ho však potrebujeme pretransformovať do formátu (x)HTML. Obsah dokumentu môže byť však nevalidný. V tomto prípade by sa náhľad nemal zobrazit'.

##### 4.4.2 Návrh

Zobrazenie dokumentu vo formáte (x)HTML bude dostupné z obrazovky editovania dokumentu. Obsah dokumentu sa pretransformuje pomocou XSLT transformácie. Obsah dokumentu vo formáte (x)HTML sa zobrazí len v prípade, ak je dokument validný. Táto skutočnosť sa overí pred zobrazením náhľadu. Ak je dokument nevalidný, namiesto náhľadu sa zobrazí upozornenie pre používateľa. Náhľad sa zobrazí v novej obrazovke.

##### 4.4.3 Implementácia

Na prácu s obsahom dokumentov sa používa gem „Nokogiri“. Pred vytvorením náhľadu sa overí validita uloženého obsahu dokumentu pomocou schémy popisujúcej štruktúru dokumentu typu *DocBook*. Je zapísaná v jazyku RELAX NG a rozšírená o prvky používané v portáli ALEF. Model dokumentu obsahuje metódu, ktorá validáciu uskutoční.



Ak je dokument validný, vykoná sa XSLT transformácia z formátu XML do HTML. Výsledok sa zobrazí v obrazovke náhľadu na dokument. V prípade, že dokument validný nie je, transformácia sa neuskutoční a zobrazí sa nápis, ktorý informuje používateľa o nemožnosti zobrazenia náhľadu.

#### 4.4.4 Testovanie

<b>Názov testu</b>		Zobrazenie náhľadu na dokument	<b>ID</b>	4-4-1
<b>Rozhranie</b>	Obrazovka editovania dokumentu		<b>autor</b>	Martin Habdák
<b>Účel</b>	Zobrazenie uloženého obsahu dokumentu vo formáte HTML			
<b>Vstupné podmienky</b>	Používateľ je prihlásený, obsah dokumentu je validný			
<b>Výstupné podmienky</b>	Náhľad obsahu dokumentu vo formáte HTML			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Klikneme na odkaz „View“.	Presmerovanie na obrazovku náhľadu, zobrazenie náhľadu obsahu dokumentu vo formáte HTML.	Presmerované na obrazovku náhľadu, zobrazený náhľad vo formáte HTML.	

<b>Názov testu</b>		Zobrazenie chybovej hlášky v náhľade dokumentu	<b>ID</b>	4-4-2
<b>Rozhranie</b>	Obrazovka editovania dokumentu		<b>autor</b>	Martin Habdák
<b>Účel</b>	Zobrazenie chybovej hlášky namiesto obsahu dokumentu vo formáte HTML			
<b>Vstupné podmienky</b>	Používateľ je prihlásený, obsah dokumentu je nevalidný			
<b>Výstupné podmienky</b>	Chybová hláška o nevalidite obsahu dokumentu			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Klikneme na odkaz „View“.	Presmerovanie na obrazovku náhľadu, zobrazenie chybovej hlášky o nevalidite obsahu dokumentu.	Presmerované na obrazovku náhľadu, zobrazená chybová hláška o nevalidite obsahu dokumentu.	

## 4.5 Editovanie dokumentu a import súboru, informácie o zmenách

Po výbere „editovať“ v detaile dokumentu môžem vytvoriť/editovať XML dvoma spôsobmi:

- ✦ píšem XML priamo do prehliadača,
- ✦ importujem XML súbor.

Pri ukladaní zmien dokumentu chcem vidieť informácie o zmenách v dokumente:

- ✦ diff obsahu,
- ✦ ktoré anotácie sú v konflikte.

### 4.5.1 Analýza

Súčasťou požiadavky je úprava obrazovky editácie dokumentu podľa návrhu obrazovky DE1 (Príloha A). Návrh obrazovky predpokladá vytvorenie dvoch režimov, ktoré budú obsahovať niektoré spoločné prvky a líšiť sa budú vo forme zobrazenia dokumentu. Editačný režim zobrazí XML reprezentáciu dokumentu, režim „View“ poskytuje transformovaný obsah dokumentu na HTML. Je potrebné vytvoriť ovládacie prvky pre zmenu režimu. Okrem editácie a náhľadu dokumentu chce používateľ znázorňovať históriu zmien dokumentu.

Používateľ by mal mať možnosť vložiť do formulára obsah súboru z disku. Môže to byť napríklad šablóna, alebo záloha dokumentu uložená na lokálnom počítači používateľa. Vkladanie by malo byť prevedené bez znovu načítania formuláru, aby sa neuložené zmeny v ostatných poliach nestratili.

Obsah zobrazený v jednoduchom textovom poli je neprehľadný a zle sa formátuje. Formulár by mal podporovať zvýraznenie syntaxe XML dokumentov, čo by uľahčilo prácu s rozsiahlymi textami. Existuje viacero knižníc, ktoré umožňujú zvýraznenie syntaxe zdrojového kódu, ale len sú len obmedzené možnosti výberu WYSIWYG editora s touto vlastnosťou. Po otestovaní knižnice *CodeMirror* som zvolil jej použitie v projekte. Umožňuje zvýraznenie XML syntaxe, upozorňuje na chybné umiestnenie elementov a automaticky formátuje dokumenty.

Súčasťou formulára by mal byť náhľad upravovaného dokumentu.

Po uložení dokumentu chce byť používateľ informovaný o zmenách, ktoré v obsahu nastali a o prípadných konfliktoch v anotáciách. Konflikty môžu nastať ak sa v anotovanom úseku dokumentu zmenil obsah. Zmeny v obsahu dokumentu by mali byť znázornené ako zoznam zmenených riadkov. Jazyk Ruby poskytuje viacero riešení pre určovanie zmien v textoch. Vybral som ruby gem *Diffy*, ktorý dokáže výsledky zobrazit' vo viacerých formátoch a má jednoduchú aplikáciu.

### 4.5.2 Návrh

Hlavné textové pole formulára, v ktorom sa zobrazuje obsah dokumentu, bude obohatené o zvýraznenie syntaxe XML s použitím externej knižnice napísanej v JavaScripte.

V hornej časti obrazovky bude jednoduché menu – prepínač medzi režimami zobrazenia súboru (zobrazenie, úpravy a história zmien). Do formulára pribudne tlačidlo vyvolávajúce náhľad súboru, pričom náhľad aktuálne upravovaného súboru sa zobrazí v dolnej časti obrazovky.

História zmien bude reprezentovaná jednoduchým stromom s vyznačenou aktuálnou verziou a odkazmi na ostatné verzie. Pri zobrazení inej ako aktuálnej verzie, bude mať používateľ možnosť zmeniť zobrazenú verziu na aktuálnu (funkcia revert z predchádzajúceho šprintu) a prehliadať zmeny, ktoré táto verzia do obsahu dokumentu priniesla.

Po uložení budú zmeny obsahu zobrazené v modálnom dialógu vo farebne odlíšených riadkoch. Konflikty v anotáciách budú prezentované používateľovi vo forme jednoduchého výpisu v tomto dialógu.

### 4.5.3 Implementácia

Zmena vzhľadu formulára bola prevedená zmenou šablón a aplikáciou CSS štýlov. Prepínacie tlačidlá režimov sú podmienené aktuálnou akciou, ktorá zabezpečuje vykreslenie šablóny. Akcia ovplyvňuje znefunkčnenie a zvýraznenie odkazu View, Edit alebo History.

Import súboru je zabezpečený asynchrónnou požiadavkou. Ruby gem *remotipart* umožňuje nahranie súborov na server prostredníctvom technológie AJAX. Nahraný súbor je na serveri prečítaný a jeho obsah je vložený do skriptu, ktorý naplní textové pole vo formulári. O vykonanie skriptu sa postará gem *remotipart*. Pri vkladaní obsahu do textového poľa bolo potrebné vyriešiť aktualizáciu editora so zvýraznenou syntaxou.

Po uložení dokumentu sa zisťuje, či nastali zmeny a po presmerovaní sa vypíšu vo forme farebne odlíšených riadkov, čo je zabezpečené voľbou HTML formátu výpisu diff nástroja. Zobrazenie dialógu so zmenami je podmienené iba prítomnosťou GET parametra „changed“, takže používateľ si tieto zmeny môže prehliadať v ktorejkoľvek verzii dokumentu, nie iba po vytvorení novej.

Pri hľadaní konfliktov v anotáciách sa postupuje nájdením spoločných anotácií dvoch verzií (existencia prepojení z dvoch rôznych verzií na tú istú anotáciu). V spoločných anotáciách sa zisťuje zhoda anotovaného obsahu, a pri odchýlkach je oznámený konflikt.

### 4.5.4 Testovanie

Testované bolo nájdenie spoločných anotácií a nezhôd v anotovaných fragmentoch, ako aj zistenie konfliktov a zobrazenie zmien po úprave dokumentu.

<b>Názov testu</b>		Zobrazenie zmien vo vybranej verzii	<b>ID</b>	4-5-1
<b>Rozhranie</b>	Obrazovka dokumentu		<b>autor</b>	Martin Franta
<b>Účel</b>	Prehľad o zmenách v danej verzii dokumentu			
<b>Vstupné podmienky</b>	Používateľ je prihlásený, v systéme existuje dokument			
<b>Výstupné podmienky</b>	Zobrazené zmeny v dokumente			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Upravíme obsah dokumentu tak, aby bol validný.	V editore bude validný obsah.	V editore je validný obsah.	
2	Potvrdíme uloženie dokumentu.	Presmerovanie na obrazovku editácie dokumentu a zobrazenie dialógu.	Presmerované na obrazovku editácie dokumentu a zobrazený dialóg so zmenami.	

#### 4 Šprint č. 3 – Chopin

3	Zobrazíme históriu dokumentu.	Zobrazí sa strom minimálne dvoch verzií.	Je zobrazený strom minimálne dvoch verzií.
4	Vyberieme aktuálnu verziu.	Zobrazí sa aktuálna verzia dokumentu.	Zobrazená je aktuálna verzia dokumentu.
5	Vyberieme akciu „Show changes“	Zobrazí sa dialóg so zmenami, ktoré sme previedli v 1. kroku.	Sú zobrazené zmeny, ktoré sme previedli v 1. kroku dokumentu.

<b>Názov testu</b>		Nájdenie konfliktu v anotáciách	<b>ID</b>	4-5-2
<b>Rozhranie</b>	Obrazovka dokumentu		<b>autor</b>	Martin Franta
<b>Účel</b>	Prehľad o konfliktoch, ktoré nastali úpravou dokumentu			
<b>Vstupné podmienky</b>	Používateľ je prihlásený, v systéme existuje dokument s anotáciou			
<b>Výstupné podmienky</b>	Zobrazený konflikt v anotácií			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Upravíme obsah dokumentu tak, aby bol validný, na mieste, ktoré je anotované	V editore bude validný obsah	V editore je validný obsah	
2	Potvrdíme uloženie dokumentu.	Presmerovanie na obrazovku editácie dokumentu a zobrazenie dialógu s vypísanou anotáciou v konflikte.	Presmerované na obrazovku editácie dokumentu a zobrazený dialóg s anotáciou v konflikte.	

<b>Názov testu</b>	Importovanie obsahu dokumentu	<b>ID</b>	4-5-3
<b>Rozhranie</b>	Obrazovka dokumentu	<b>autor</b>	Martin Franta
<b>Účel</b>	Načítanie obsahu dokumentu zo súboru		
<b>Vstupné podmienky</b>	Používateľ je prihlásený		
<b>Výstupné podmienky</b>	Načítaný obsah v textovom poli		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Zvolíme vytvorenie nového dokumentu.	Zobrazí sa prázdny formulár nového dokumentu.	Zobrazený je prázdny formulár nového dokumentu.
2	Zvolíme výber súboru zo súborového systému klienta.	Zobrazí sa dialóg výberu súboru.	Zobrazený je dialóg výberu súboru.
3	Vyberieme súbor a v dialógu potvrdíme.	Názov súboru sa zobrazí v textovom poli výberu súboru.	Názov súboru je zobrazený v textovom poli výberu súboru.
4	Klikneme na tlačidlo „Load file“.	Obsah vybraného súboru sa načíta do textového poľa formulára.	Obsah vybraného súboru je načítaný do textového poľa formulára.

## 4.6 Možnosť preview aktuálneho dokumentu (Editovanie dokumentu)

Ako používateľ chcem zobrazíť náhľad na editovaný obsah dokumentu.

### 4.6.1 Analýza

Ak používateľ edituje obsah dokumentu, potrebuje vedieť, ako bude zobrazený vo webovom portáli. Treba mu preto umožniť ukážku dokumentu vo formáte (X)HTML. Podobne ako pri zobrazení náhľadu je obsah dokumentu uložený vo formáte XML. Môže byť tiež nevalidný, preto by mal byť zmenený obsah porovnaný voči schéme. Ukážka by sa mala zobrazíť len v tom prípade, ak je obsah dokumentu validný.

### 4.6.2 Návrh

Na vytvorenie ukážky dokumentu sa spracuje jeho práve editovaný obsah. Proces sa spustí na požiadanie používateľa pomocou tlačidla „Preview“. Pred zobrazením ukážky sa editovaný obsah zvaliduje voči schéme. Ak validácia uspeje, zobrazí sa ukážka dokumentu vo formáte X(HTML). Zobrazenie prebieha na tej istej obrazovke. Ukážku možno vidieť v spodnej časti obrazovky. V prípade, že editovaný obsah nie je validný, používateľ je na to upozornený a ukážka sa nezobrazí.

### 4.6.3 Implementácia

Implementácia vytvorenia ukážky je rovnaká ako v časti „Zobrazenie view dokumentu“. Po spustení procesu zobrazenia ukážky sa pomocou JavaScriptu ukáže informácia o jej vytváraní. Obsah pretransformovaný do formátu HTML sa zobrazí v oblasti označenej elementom „div“, ktorý má atribút „id“ nastavený na „preview\_area“.

### 4.6.4 Testovanie

<b>Názov testu</b>	Zobrazenie ukážky editovaného obsahu dokumentu	<b>ID</b>	4-6-1
<b>Rozhranie</b>	Všetky obrazovky systému	<b>autor</b>	Martin Habdák
<b>Účel</b>	Importovanie obsahu z archívu		
<b>Vstupné podmienky</b>	Používateľ je prihlásený, editovaný obsah dokumentu je validný		
<b>Výstupné podmienky</b>	Ukážka editovaného obsahu dokumentu vo formáte HTML		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Klikneme na tlačidlo „Preview“.	Zobrazenie indikátora spracovávania editovaného obsahu dokumentu, zobrazenie ukážky editovaného obsahu dokumentu vo formáte HTML.	Zobrazený indikátor spracovávania editovaného obsahu dokumentu, zobrazená ukážka editovaného obsahu dokumentu vo formáte HTML.

<b>Názov testu</b>	Zobrazenie chybovej hlášky v ukážke editovaného obsahu dokumentu	<b>ID</b>	4-6-2
<b>Rozhranie</b>	Obrazovka editovania dokumentu	<b>autor</b>	Martin Habdák
<b>Účel</b>	Zobrazenie chybovej hlášky namiesto editovaného obsahu dokumentu vo formáte HTML		
<b>Vstupné podmienky</b>	Používateľ je prihlásený, editovaný obsah dokumentu je nevalidný		
<b>Výstupné podmienky</b>	Chybová hláška o nevalidite editovaného obsahu dokumentu		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Klikneme na tlačidlo „Preview“..	Zobrazenie indikátora spracovávania editovaného obsahu dokumentu, zobrazenie chybovej hlášky o nevalidite editovaného obsahu dokumentu.	Zobrazený indikátor spracovávania editovaného obsahu dokumentu, zobrazená chybová hláška o nevalidite editovaného obsahu dokumentu.

## 4.7 Zobrazenie detailu repozitára ako zoznamu dokumentov a variantov metadát

Úloha bola zameraná na zobrazenie zoznamu dokumentov a variantov v rámci zobrazenia detailu konkrétneho repozitára.

### 4.7.1 Analýza

Každý repozitár môže obsahovať nula alebo viac dokumentov a variantov metadát. V rámci zobrazenia jeho detailu je teda potrebné zobrazit' aj jeho obsah, teda všetky prislúchajúce dokumenty a varianty.

### 4.7.2 Návrh

Feature je v skutočnosti implementácia obrazovky RD1, čiže detailu repozitára. V rámci zobrazenia detailu zvoleného repozitára sú zobrazené aj všetky jeho dokumenty a varianty. Feature pracuje s už implementovanými entitami *Repository*, *Document* a *Variant*.

### 4.7.3 Implementácia

Implementácia bola prevedená vo view *repositories\show*, kde bol, okrem zoznamu dokumentov prislúchajúcich k zobrazenému repozitáru, pridaný aj zoznam variantov. a možnosť pridať nový variant.

#### 4.7.4 Testovanie

Akceptačné testy:

<b>Názov testu</b>	Zobrazenie detailu repozitára	<b>ID</b>	4-7-1
<b>Rozhranie</b>	Stránka so zoznamom repozitárov (H1)	<b>autor</b>	Zuzana Zimová
<b>Účel</b>	Overenie funkcionality zobrazenia detailu repozitáru		
<b>Vstupné podmienky</b>	Existuje aspoň jeden repozitár.		
<b>Výstupné podmienky</b>	Detail repozitára bol zobrazený.		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Používateľ zvolí existujúci repozitár v zozname repozitárov.	Zobrazí sa detail repozitára.	Prepojenie na obrazovku detailu repozitára.

### 4.8 Zoznam repozitárov po prihlásení

Úloha bola zameraná na zobrazenie zoznamu repozitárov ako hlavného menu po prejdení na stránku *Home*.

#### 4.8.1 Analýza

Po prejdení na domovskú stránku, ktorú je možné zobrazit' po prihlásení, je potrebné zobrazit' zoznam existujúcich repozitárov. Takisto musí existovať možnosť pridať nový repozitár alebo editovať, zobrazit', či vymazať už existujúci.

#### 4.8.2 Návrh

Feature je v skutočnosti implementácia obrazovky H1, čiže úvodnej stránky. V rámci zobrazenia úvodnej obrazovky sú zobrazené aj všetky existujúce repozitáre a s nimi spojené akcie. V prípade, že ešte nebol vytvorený repozitár, zoznam repozitárov je prázdny. Feature pracuje s už implementovanou entitou *Repository*.

#### 4.8.3 Implementácia

Implementácia bola prevedená v *routes.rb*, kde ako úvodná stránka bolo nastavené view *repositories\index*.



#### 4.8.4 Testovanie

Akceptačné testy:

<b>Názov testu</b>		Zobrazenie repozitárov po prihlásení	<b>ID</b>	4-8-1
<b>Rozhranie</b>	Stránka so zoznamom repozitárov (H1)		<b>autor</b>	Zuzana Zimová
<b>Účel</b>	Overenie funkcionality zobrazenia zoznamu repozitárov			
<b>Vstupné podmienky</b>	Používateľ sa úspešne prihlásil.			
<b>Výstupné podmienky</b>	Zoznam repozitárov je zobrazený hneď po prihlásení.			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ sa úspešne prihlási.	Zobrazí sa zoznam repozitárov.	Prepojenie na obrazovku so zoznamom repozitárov.	



## 5 Šprint č. 4 – Dvořák

---

### 5.1 Zobrazenie anotačného pásika

Úloha bola zameraná na zobrazenie anotačného pásika pri view dokumentu.

#### 5.1.1 Analýza

Po zobrazení dokumentu je možné vidieť jeho obsah, pokiaľ je validný. Vedľa obsahu je potrebné implementovať pásik, ktorý bude zobrazovať polohu anotácií v texte a bude prepojený s vyznačovaním anotácií v texte.

#### 5.1.2 Návrh

Feature je implementácia v obrazovke view dokumentu, kde je zobrazený obsah dokumentu. Pracuje s entitami *Document* a *Annotation*. V prípade, že žiadne anotácie k dokumentu neexistujú, pásik je prázdny. Po ukázaní na vyznačené miesto v pásiku sa v texte vyznačí príslušná anotácia.

#### 5.1.3 Implementácia

Implementácia bola prevedená vo view *documents\\_annotation\_highlight* použitím JavaScriptu. Bola napojená na už existujúcu implementáciu zvýrazňovania anotácií v texte, ktorá však bola pôvodne implementovaná len staticky, teda nie na existujúce anotácie.

#### 5.1.4 Testovanie

Akceptačné testy:

<b>Názov testu</b>	Zobrazenie existujúcej anotácie v texte	<b>ID</b>	5-1-1
<b>Rozhranie</b>	View dokumentu.	<b>autor</b>	Zuzana Zimová
<b>Účel</b>	Overenie funkcionality zobrazenia anotácie po jej zvolení v anotačnom pásiku.		
<b>Vstupné podmienky</b>	Existuje aspoň jeden dokument s validným obsahom a je k nemu priradená aspoň jedna anotácia.		
<b>Výstupné podmienky</b>	Zvýraznenie anotácie v texte.		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Používateľ ukáže na zvýraznené miesto v anotačnom pásiku.	V texte sa zvýrazní príslúchajúca anotácia.	V texte sa zvýraznila príslúchajúca anotácia.

## 5.2 Zobrazovanie a editovanie anotácie po výbere z pásika

Úloha bola zameraná na zobrazenie anotačného okienka po výbere konkrétnej anotácie z anotačného pásika.

### 5.2.1 Analýza

Každú anotáciu, zobrazenú v anotačnom pásiku, môže používateľ editovať. Je možné zmeniť jej obsah, začiatkový a koncový index.

### 5.2.2 Návrh

Feature je implementácia v obrazovke view dokumentu, kde je zobrazený obsah dokumentu. Pracuje s entitami *Document* a *Annotation*. Po kliknutí na anotáciu, zvýraznenú v pásiku, sa otvorí anotačné okienko, v ktorom je zobrazený text, na ktorý sa anotácia viaže. Okrem toho okienko obsahuje obsah zvolenej anotácie a jej počiatkový a koncový index. Tieto hodnoty je možné editovať.

### 5.2.3 Implementácia

Implementácia bola prevedená vo view *documents\\_annotation\\_highlight* použitím JavaScriptu. Bola napojená na už existujúcu implementáciu anotačného pásika.

### 5.2.4 Testovanie

Akceptačné testy:

<b>Názov testu</b>	Zobrazenie anotačného okienka	<b>ID</b>	5-2-1
<b>Rozhranie</b>	View dokumentu.	<b>autor</b>	Zuzana Zimová
<b>Účel</b>	Overenie funkcionality zobrazenia anotačného okienka k prislúchajúcej anotácii.		
<b>Vstupné podmienky</b>	Existuje aspoň jeden dokument s validným obsahom a je k nemu priradená aspoň jedna anotácia.		
<b>Výstupné podmienky</b>	Zobrazenie anotačného okienka		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Používateľ zvolí zvýraznené miesto v anotačnom pásiku.	Pri anotačnom pásiku sa zobrazí anotačné okienko k prislúchajúcej anotácii.	Pri anotačnom pásiku sa zobrazilo anotačné okienko k prislúchajúcej anotácii.

<b>Názov testu</b>	Editácia anotácie v anotačnom okienku		<b>ID</b>	5-2-2
<b>Rozhranie</b>	View dokumentu.		<b>autor</b>	Zuzana Zimová
<b>Účel</b>	Overenie funkcionality editovania anotácie v anotačnom okienku.			
<b>Vstupné podmienky</b>	Existuje aspoň jeden dokument s validným obsahom a je k nemu priradená aspoň jedna anotácia.			
<b>Výstupné podmienky</b>	Uloženie nových hodnôt anotácie			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí zvýraznené miesto v anotačnom pásiku.	Pri anotačnom pásiku sa zobrazí anotačné okienko k prislúchajúcej anotácii.	Pri anotačnom pásiku sa zobrazilo anotačné okienko k prislúchajúcej anotácii.	
2	Používateľ zmení pôvodné atribúty anotácie na nové a svoje zmeny uloží.	Hodnoty v poliach pre editáciu budú zmenené.	Hodnoty v poliach pre editáciu sú zmenené.	

### 5.3 Pridávanie RDT a vzťahov v grafe

Používateľovi je poskytnutá možnosť pridania nových záznamov priamo do grafu.

#### 5.3.1 Analýza

Z analýzy vyplynulo, že knižnica poskytuje možnosť načítania grafu cez JSON objekt, no export upraveného grafu už nie je možný. Dodatočné zadanie údajov, ktoré nie sú potrebné pre vizualizáciu grafu totižto graf odstráni. Graf taktiež neposkytuje priamu možnosť, ako zobrazovať viacnásobné prepojenie medzi dvoma vrcholmi.

#### 5.3.2 Návrh

Pre prekonanie obmedzení grafu sme sa rozhodli vytvoriť vlastnú dátovú štruktúru, ktorá by pracovala nezávisle od grafu, no pri každej jeho zmene by bola aktualizovaná. Týmto spôsobom by sme teda graf mohli používať čisto iba pre vizualizáciu, zatiaľ, čo by sme používali vlastnú logiku, ktorá je jednoducho modifikovateľná.

#### 5.3.3 Implementácia

Pri načítaní grafu sme teda vytvorili pomocnú štruktúru, ktorú sme upravovali podľa akcií vykonaných v grafe. Ak sme napríklad vymazali uzol, v štruktúre sme mu nastavili hodnotu *changed*: “*removed*”. Po úpravách bolo nutné zabezpečiť schopnosť zaslania upravených dát na server. Toho sme docielili nadviazaním AJAX komunikácie, cez ktorú zasielame štruktúru ako parameter metódou POST. Štruktúra je formátovaná ako JSON. Na serveri sa následne prevedie do Ruby formátu, zistia sa zmeny, ktoré sa následne vykonajú.

## 5.4 Import repozitára

Ako používateľ chcem importovať existujúce repozitáre dokumentov z archívu vo formáte „zip“.

### 5.4.1 Analýza

Ak má používateľ k dispozícii veľké množstvo dokumentov, je zdĺhavé pridávať ich rad za radom. Takýto prípad môže nastať aj pri presúvaní celých repozitárov z jednej inštancie programu do druhej, alebo pri presune dokumentov z existujúceho webového portálu do vyvíjaného nástroja. Nástroj by mal umožňovať jednoduchý spôsob importovania dokumentov z daného miesta. Keďže sa jedná o veľké množstvo dokumentov, uložené sú v archíve.

Pri vývoji je k dispozícii archív vo formáte „zip“, vytvorený z dokumentov portálu ALEF. Obsahuje adresáre pomenované podľa výukového kurzu. Každý kurz má nasledovnú štruktúru:

- Exercises – dokumenty(vzdelávacie objekty) typu „exercise“ vo formáte XML
- Explanations – dokumenty(vzdelávacie objekty) typu „explanations“ vo formáte XML
- Metadata – obsahuje RDT(Related Domain Term), relácie medzi RDT a relácie medzi vzdelávacími objektmi a RDT
- Questions – dokumenty(vzdelávacie objekty) typu „questions“ vo formáte XML
- Resources – obrázky a súbory priložené k jednotlivým dokumentom
- Manifest.mf - informácie o balíku kurzu
- Structure.xml – súbor vizualizujúci štruktúru kurzu vo formáte XML

Importovanie zahŕňa spracovanie všetkých dokumentov, RDT a vytvorenie relácií medzi nimi. Ďalej sa importujú aj súbory priložené k jednotlivým dokumentom. Súčasne treba dbať na integritu celého dátového modelu a ošetriť konfliktné situácie, ktoré by mohli nastať.

### 5.4.2 Návrh

Importovanie bude umožnené pomocou formulára. Obsahuje pole na výber archívu, ktorý chce používateľ importovať a tlačidlo na spustenie procesu importovania. Prístup k formuláru sa nachádza v obrazovke vytvorenia nového repozitára.

Po spustení importovania sa dáta z archívu uložia do systému, kde budú dočasne uložené a následne spracovávané. Na toto miesto sa pri importovaní kopírujú dokumenty každého používateľa. Z tohto dôvodu treba zabrániť možným konfliktom, ktoré by vznikli pri zapisovaní súborov z importovaného archívu.

Ak sú skopírované všetky entity z archívu, spustí sa ich vkladanie do databázy systému. Pre dokumenty sa vytvorí príslušný repozitár. Ak takýto repozitár už existuje, doplní sa jeho obsah súbormi, ktoré sa v ňom ešte nenachádzajú.

RDT sa pridávajú do príslušného variantu. Ak daný variant ešte neexistuje, je vytvorený nový. V prípade, že dané RDT už existujú, nebudú sa vkladať do systému, aby nevznikli duplikáty. Následne sa vytvoria vzťahy medzi RDT a medzi RDT a dokumentmi. Po importovaní obrázkov sa zmažú všetky nepotrebné súbory.

### 5.4.3 Implementácia

Formulár importovania je vytvorený pomocou partial-u. Obsahuje pole pre výber súboru, ktorý bude importovaný. Tlačidlo „Import“ spúšťa proces importovania. Partial taktiež obsahuje aj JavaScript, ktorý zobrazí indikátor činnosti po spustení importovania. Celý formulár je vložený do view-u pre nový repozitár.

Dočasné miesto, kde sa ukladá importovaný archív, je v adresári „./tmp/import“. Aby sa zabránilo kolíziám súborov rôznych používateľov, vytvára sa pre každého z nich vlastný adresár. Jeho názov je odvodený od identifikačného čísla používateľa. Vytvorený je ako podadresár dočasného miesta pre importované archívy (napr. /tmp/import/2). Po skončení procesu importovania sa všetky dočasné súbory zmažú.

Dokumenty sa importujú do požadovaného repozitára. Ak takýto už existuje, dokumenty sa doplnia doňho doplnia. V repozitári sa nesmú nachádzať dva dokumenty s rovnakým názvom. Pri importovaní sa najprv overuje, či daný dokument neexistuje pomocou metódy „find\_by“.

Importovanie RDT prebieha podobne ako pri dokumentoch. RDT sa „vkladajú“ do príslušného variantu. Ak variant existuje, dopĺňajú sa RDT doňho. Vo variante nesmú byť rovnaké RDT. Pri importovaní sa najprv overuje, či RDT neexistuje pomocou metódy „find\_by“.

Z archívu sa importujú len tie vzťahy, ktoré sa v systéme ešte nenachádzajú. Metódou „find\_by“ sa overujú všetky atribúty vzťahu. V rôznych variantoch sa totiž môžu nachádzať rôzne RDT.

Priloženie obrázkov k dokumentu vykonáva gem „PaperClip“. Ukladá ich v tvare: identifikačnéčíslo\_photo.formát (napr. 14\_photo.jpg). Názvy obrázkov v archíve nemajú takýto tvar. Pred priložením k dokumentu sa ich názvy pretransformujú do požadovaného tvaru. Následne sú skopírované do adresára pre priložené súbory.

### 5.4.4 Testovanie

Keďže proces importovania si vyžaduje, aby neporušil integritu dátového modelu, vytváranie hodnôt v databáze musí podliehať podrobnej kontrole. Ide napríklad o sledovanie priradenia správnych cudzích kľúčov a ostatných atribútov. K otestovaniu počas vývoja bola preto použitá metóda bielej skrinky.

Testy prebiehali na množine údajov poskytnutých z portálu ALEF. K dispozícii je archív s výukovým kurzom „Lisp“.

<b>Názov testu</b>	Import	<b>ID</b>	5-4-1
<b>Rozhranie</b>	Obrazovka repozitárov	<b>autor</b>	Martin Habdák
<b>Účel</b>	Importovanie obsahu z archívu		
<b>Vstupné podmienky</b>	Používateľ je prihlásený		
<b>Výstupné podmienky</b>	Obsah archívu vložený do systému		
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>
1	Klikneme na odkaz „Add“.	Presmerovanie na obrazovku nového repozitára.	Presmerované na obrazovku nového repozitára.
2	Klikneme na výber archívu.	Zobrazenie okna pre výber archívu.	Zobrazené okno pre výber archívu.
3	Vyberieme archív a potvrdíme tlačidlom.	Zatvorenie okna výberu archívu a vloženie cesty súborového systému k archívu.	Zatvorené okno výberu archívu a vložená cesta súborového systému k archívu.
4	Klikneme na tlačidlo „Import“.	Zobrazenie indikátora importovania, presmerovanie na obrazovku repozitárov.	Zobrazený indikátor importovania, presmerované na obrazovku repozitárov.



## 5.5 Export repozitára

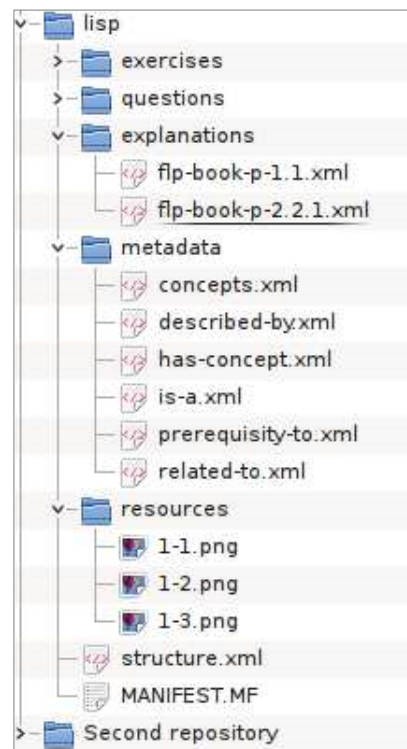
Používateľ chce exportovať repozitár do *ALEF-DocBook* formátu.

### 5.5.1 Analýza

Export repozitára je jeho prevedenie na štruktúrovaný ZIP archív. Obsah dokumentov je uložený v samostatných XML súboroch, rozdelených v adresároch podľa typu dokumentu. Štruktúra dokumentov je tiež zapísaná v súbore *structure.xml*. Metadáta majú vytvorené súbory v adresári metadata. Zoznam RDT je uložený v súbore *concepts.xml*. Vzťahy sú podľa typov rozdelené v samostatných súboroch. K dokumentom môžu byť priradené obrázky, ktoré sú v archíve uložené v adresári resources.

Obrázok znázorňuje štruktúru súborov v exportovanom archíve. Požiadavkou zákazníka je, aby mohol exportovať viac repozitárov naraz. Repozitáre potom na obrázku predstavujú adresáre „lisp“ a „Second repository“.

Bude potrebné vytvoriť opísanú štruktúru adresárov a XML súborov a vytvoriť ZIP archív. Ten bude používateľovi poskytnutý na stiahnutie.



Obr. 9. Štruktúra súborov.

### 5.5.2 Návrh

Používateľ si bude môcť vybrať jeden alebo viac repozitárov, ktoré bude exportovať. V systéme je vytvorený vzťah medzi repozitárom a variantom metadát, na základe ktorého môžu byť repozitáru priradené metadáta a vzťahy.

Používateľ bude pred získaním výsledného súboru informovaný o problémových dokumentoch: nevalidné, s konfliktami v anotáciách a bez priradeného typu.

### 5.5.3 Implementácia

Knižnice na prácu s XML súbormi sú už v projekte používané, využil som ich preto aj na vytváranie potrebných súborov pre export.

Pri exporte sa prechádzajú existujúce typy dokumentov a získavajú sa dokumenty daného typu a repozitára, ktoré sú validné a neboli označené ako vymazané. Obsah týchto súborov je zapisovaný do samostatných XML súborov pomenovaných podľa ich ALEF ID atribútu.

RDT a vzťahy medzi nimi som vyberal podľa asociácie variantov s repozitármi. Vzťahy medzi RDT a dokumentmi nie sú do variantov zahrnuté, preto sa získavajú, tie ktoré prislúchajú k dokumentom v danom repozitári.

Proces vytvorenia archívu bolo možné previesť priamo v pamäti, alebo vytvorením dočasnej štruktúry v súborovom systéme servera. Nakoľko niektoré repozitáre môžu byť obsiahle, vytváranie v pamäti by nemuselo byť efektívne. Vytvorenie súborov má nevýhody v tom, že pri chybe, ktorá proces exportu preruší, zostanú súbory na serveri. Tento problém som riešil označovaním dočasných adresárov podľa

používateľa, ktorý export vykonáva a adresáre môžu byť neskôr vymazané bez rizika, že sa odstráni dočasná štruktúra iného, práve vytváraného archívu.

#### 5.5.4 Testovanie

Jednotkovými testami boli overované čiastkové úlohy, ako získanie správnych dát pre export a ich zápis do XML súborov.

<b>Názov testu</b>		Export repozitárov	<b>ID</b>	5-5-1
<b>Rozhranie</b>	Hlavná obrazovka		<b>autor</b>	Martin Franta
<b>Účel</b>	Vytvorenie archívu s exportovanými repozitármi			
<b>Vstupné podmienky</b>	Používateľ je prihlásený, existujú minimálne 2 repozitáre			
<b>Výstupné podmienky</b>	Archív obsahuje všetky súbory podľa špecifikácie			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Kliknutím vyberieme repozitáre.	Riadky tabuľky sa zvýrazia, tlačidlo exportu bude aktívne.	Sú zvýraznené riadky tabuľky repozitárov, tlačidlo exportu je aktívne.	
2	Zvolíme „Export selected“.	Presmerovanie na obrazovku exportu, podľa obsahu repozitára budú zobrazené problematické dokumenty alebo informácia „All documents will be exported“.	Presmerované na obrazovku exportu, informácia „All documents will be exported“.	
3	Zvolíme odkaz Export.	Bude ponúknutý zip archív na stiahnutie.	Bol ponúknutý zip archív na stiahnutie.	
4	Otvoríme archív.	Archív bude obsahovať dva adresáre s obsahom daným špecifikáciou.	Archív obsahuje dva adresáre s názvami repozitárov, každý z nich obsahuje structure.xml, MANIFEST.MF, adresáre metadata, resources a adresáre typov dokumentov.	

## 5.6 Prepojenie tabuľky a grafu

Používateľ chce, aby sa po zobrazení grafu zobrazila aj tabuľka, ktorá obsahuje zoznam RDT zobrazovaných v grafe. Po kliknutí na RDT z tabuľky sa má v grafe zvýrazniť zodpovedajúci uzol.

### 5.6.1 Analýza

Pri zobrazení grafu bude potrebné vytvoriť novú tabuľku s RDT odlišnú od tabuľky na obrazovke V1. Po kliknutí na RDT v tejto tabuľke sa v grafe zobrazí v strede a zvýrazní zodpovedajúci uzol.

### 5.6.2 Návrh

Tabuľka s RDT pri grafe bude riešená ako nezoradený HTML zoznam, ktorý bude možné abecedne zoradiť pomocou JavaScriptu. Rozloženie prvkov bude podľa návrhu obrazovky VG1. Navyše pribudne panel v pravej časti obrazovky, v ktorom bude zobrazený označený uzol a jeho prepojenia (vzťahy) s inými uzlami.

### 5.6.3 Implementácia

Bola implementovaná funkcia grafu, ktorá animuje centrovanie označeného uzla v grafe. Bol implementovaný HTML zoraditeľný zoznam s RDT a pravý panel grafu, kde sa zobrazuje aktuálny uzol.

### 5.6.4 Testovanie

Akceptačné testy:

<b>Názov testu</b>		Zobrazenie RDT v grafe	<b>ID</b>	5-6-1
<b>Rozhranie</b>	Stránka s grafom variantu (obrazovka VG1).		<b>autor</b>	Petra Vrablecová
<b>Účel</b>	Overenie funkcionality pre zvýrazňovanie uzla v grafe.			
<b>Vstupné podmienky</b>	Používateľ chce zobrazíť RDT zo zoznamu v grafe.			
<b>Výstupné podmienky</b>	Uzol je zvýraznený a zobrazený v strede grafu.			
<b>Krok</b>	<b>Akcia</b>	<b>Očakávaná akcia</b>	<b>Skutočná akcia</b>	
1	Používateľ zvolí RDT zo zoznamu RDT pri grafe.	Zodpovedajúci uzol v grafe sa animuje na stred a zvýrazní sa. V pravom paneli sa zobrazí názov uzla (RDT) a jeho spojenia (vzťahy) k iným uzlom (RDT).	Zodpovedný uzol sa vycentroval a zvýraznil. V pravom paneli boli informácie o uzle (názov, vzťahy k ostatným uzlom).	



## 6 Opis prototypu

---

### 6.1 Architektúra systému

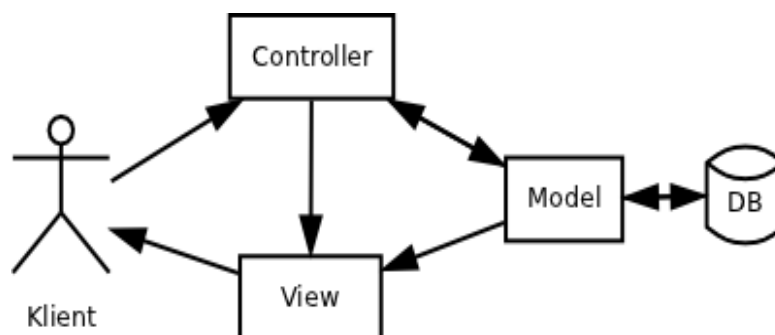
Aplikácia pracuje na princípe klient – server. Na strane servera komunikuje s databázovou vrstvou, vykonáva sa biznis logika a vytvára sa prezentačná vrstva. Na strane klienta poskytuje prezentačná vrstva prostredníctvom klientskych skriptov doplnujúce funkcie, napríklad pre zjednodušenie práce so zobrazenými dátami.

Architektúra aplikácie na serveri je daná použitím aplikačného rámca Ruby on Rails, ktorý využíva vzor *Model – View – Controller*.

Časť Model je rozhraním medzi databázou a samotnou aplikáciou, získava dáta z databázy, prípadne ich aj ďalej spracováva. Triedy modelu viac-menej zodpovedajú entitám z doménového modelu.

Controller je časť aplikácie, ktorá spracováva požiadavky klienta. Metódy sú namapované na konkrétne URL, čím je dané, aká akcia sa vykonáva pri danej požiadavke. Úlohou controllerov v aplikácii je získanie dát na základe parametrov požiadavky a výber šablóny, v ktorej budú tieto dáta zobrazené. Controllery sú v aplikácii odľahčené o biznis logiku, ktorú prevažne vykonáva model.

View je prezentačná vrstva, tvoria ju šablóny, väčšinou v jazyku HTML, do ktorých sú vložené premenné s dátami z controllera.



Obr. 10. MVC v Ruby on Rails.

## 6.2 Dátový model systému

Dátový model sa nachádza na obr. 11. Zobrazuje entity existujúce v systéme a ich vzájomné vzťahy. Popis modelu je rozdelený na štyri hlavné časti.

### 6.2.1 Document

Learning Object je v systéme chápaný ako dokument. Reprezentuje ho entita *Document*. Dokument je vždy len jedného typu. Patrí do jedného adresára. K dokumentu sa priradujú viaceré súbory, ktoré sú implementované pomocou entity *Asset*. Dokument môže mať priradených viacero súborov.

Medzi dokumentmi môže existovať vzťah. Definuje ho entita *DocumentRelationship*. Vzťah má jeden typ a patrí do jedného variantu. Na rozdiel od ostatných vzťahov nemá žiadnu váhu.

K dokumentu je možné priradovať anotácie. Reprezentované sú pomocou entity *Annotation*. Anotácia má jeden typ. Rozsah textu, ktorý anotácia popisuje sa určuje atribútmi *from* a *to*. Dokument môže mať viac anotácií.

### 6.2.2 RDT

Entita *RDT* predstavuje *Related Domain Term*. Je definovaná názvom a variantom, do ktorého patrí. Entita *Variant* je definovaná názvom. Medzi jednotlivými RDT existujú vzťahy, ktoré spolu s RDT slúžia na reprezentovanie ľahkej sémantiky. Rozdeľujeme ich na:

- *RDT-RDT* - vzťah medzi dvomi RDT. Je reprezentovaný entitou *RdtsRelationship*. Každý vzťah má určitú váhu. Jej hodnota patrí do intervalu  $<0,1>$ . Typ vzťahu je určuje entita *RdtsRelationshipType*.
- *RDT-Document* - vzťah medzi RDT a dokumentom. Reprezentovaný entitou *RdtDocumentRelationship*. Váha vzťahu je číslo z intervalu  $<0,1>$ . Typ vzťahu určuje entita *RdtDocumentRelationshipType*. Vzťahy medzi RDT a dokumentom nepatria do variantu.

Vzťah definuje vždy prepojenie len medzi dvoma RDT alebo medzi jedným RDT a jedným dokumentom. Vzťah má vždy len jeden typ, pričom viacero z nich môže byť toho istého typu.

### 6.2.3 Repository

Entita *Repository* predstavuje repozitár obsahujúci dokumenty. Do repozitára patrí viacero dokumentov. Neobsahuje dva rovnaké dokumenty. K repozitáru môže byť pridelených viacero variantov.

### 6.2.4 User

Pre identifikáciu používateľov sa používa entita *User*. Používateľ je identifikovaný pomocou atribútu *email*. Žiadny dvaja používatelia nemajú tento atribút rovnaký. Pre prihlasovanie k účtu slúži heslo, ktorí je v systéme zašifrované. K používateľovi sú priradené viaceré dokumenty, dokument však patrí len k jednému používateľovi.







## **Prílohy**



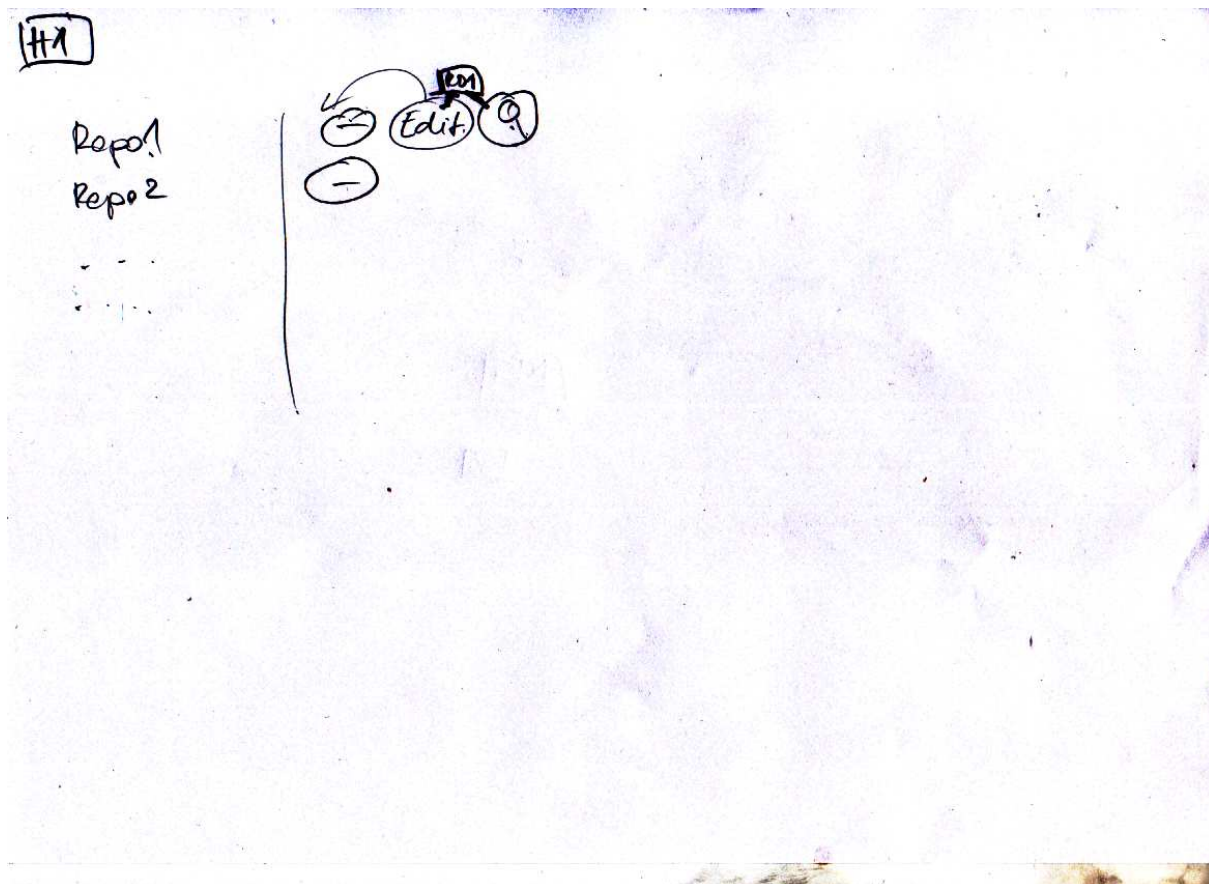
## Príloha A: Obrazovky

---

V tejto prílohe sú opísané návrhy obrazoviek systému zo stretnutia, ktoré sa konalo 8.11.2011.

### A.1 Obrazovka H1

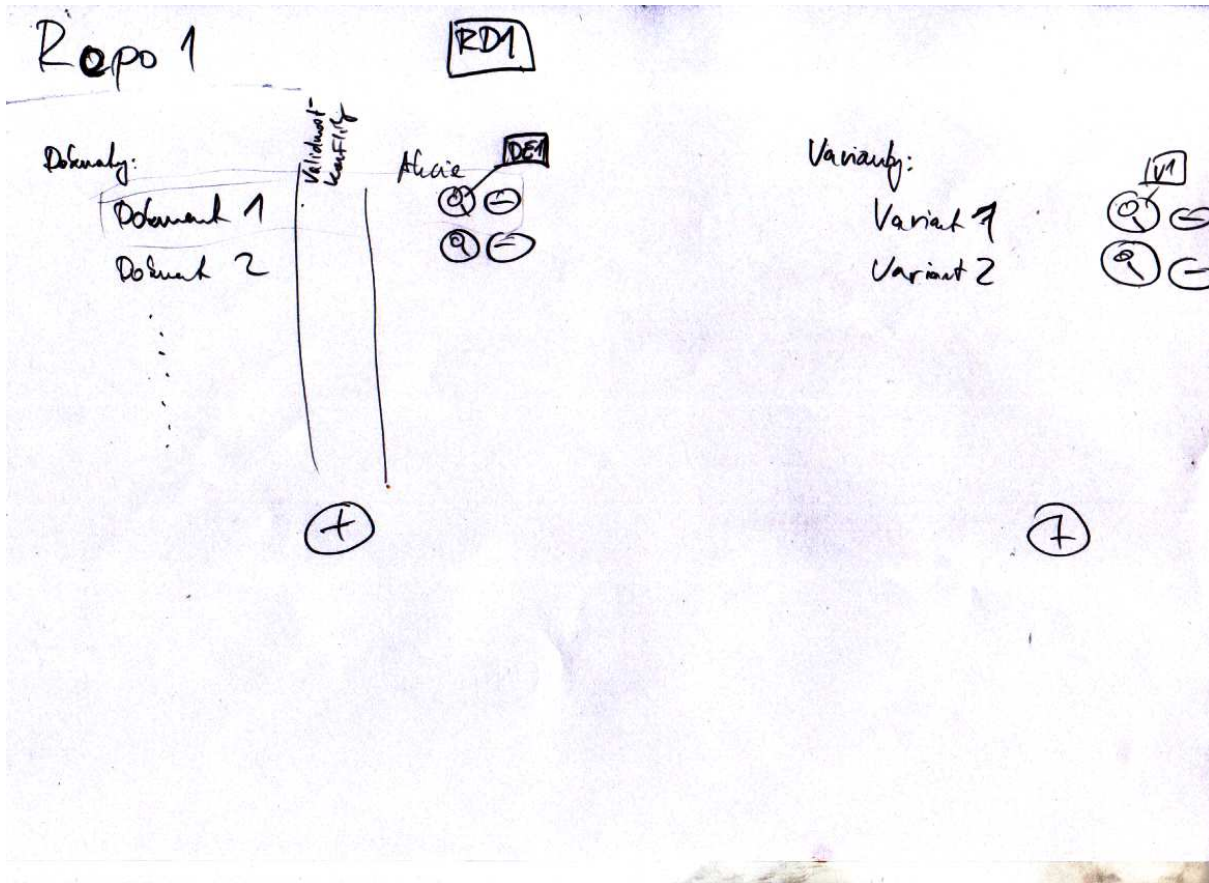
Táto obrazovka sa zobrazí po prihlásení do systému. Obsahuje zoznam všetkých repozitárov a k nim prislúchajúce akcie. Neskôr bola táto obrazovka doplnená o zoznam všetkých variantov metadát.



Obr. 12. Obrazovka H1.

## A.2 Obrazovka RD1

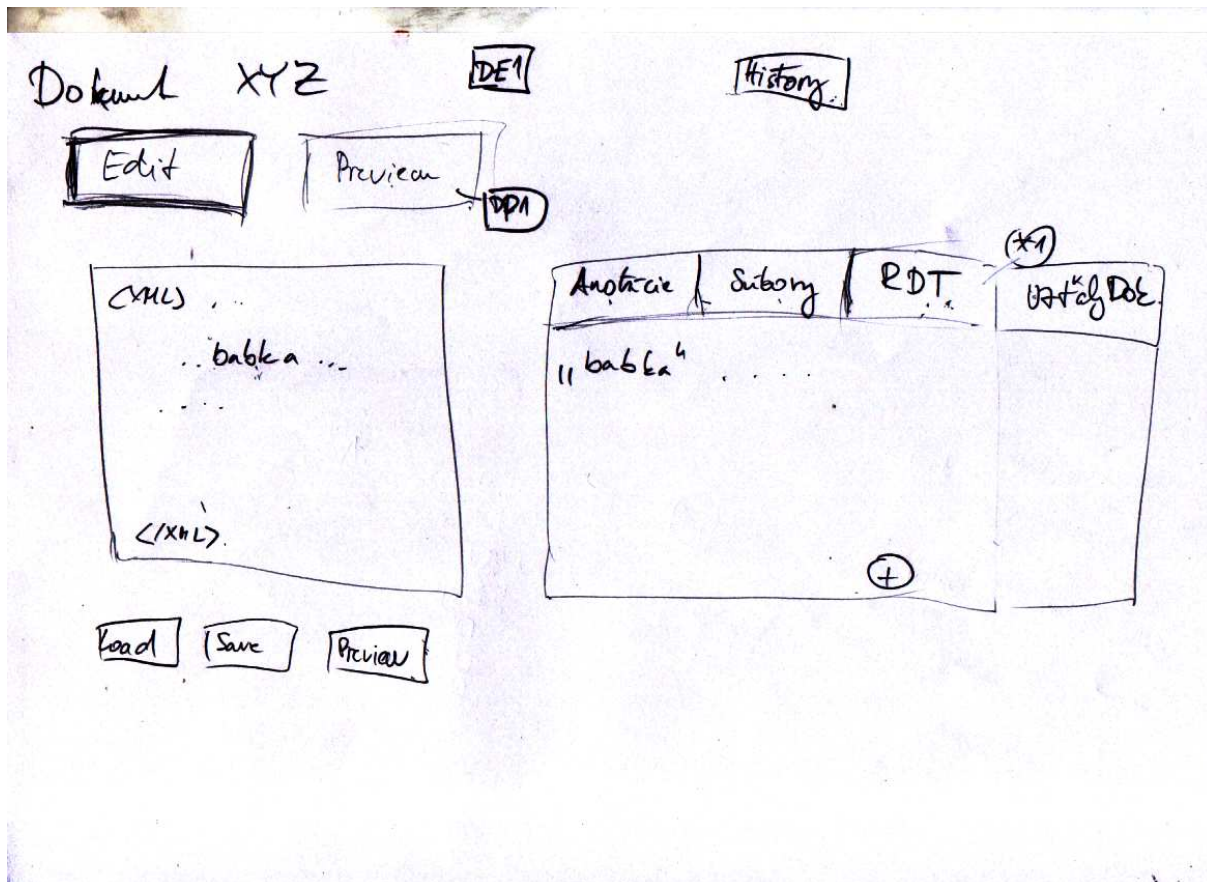
Na tejto obrazovke je znázornený detail repozitára. Obsahuje zoznam dokumentov nachádzajúcich sa v repozitári a k nim prislúchajúce akcie. Obsahuje aj zoznam variantov priradených k dokumentom tohto repozitára.



Obr. 13. Obrazovka RD1

### A.3 Obrazovka DE1

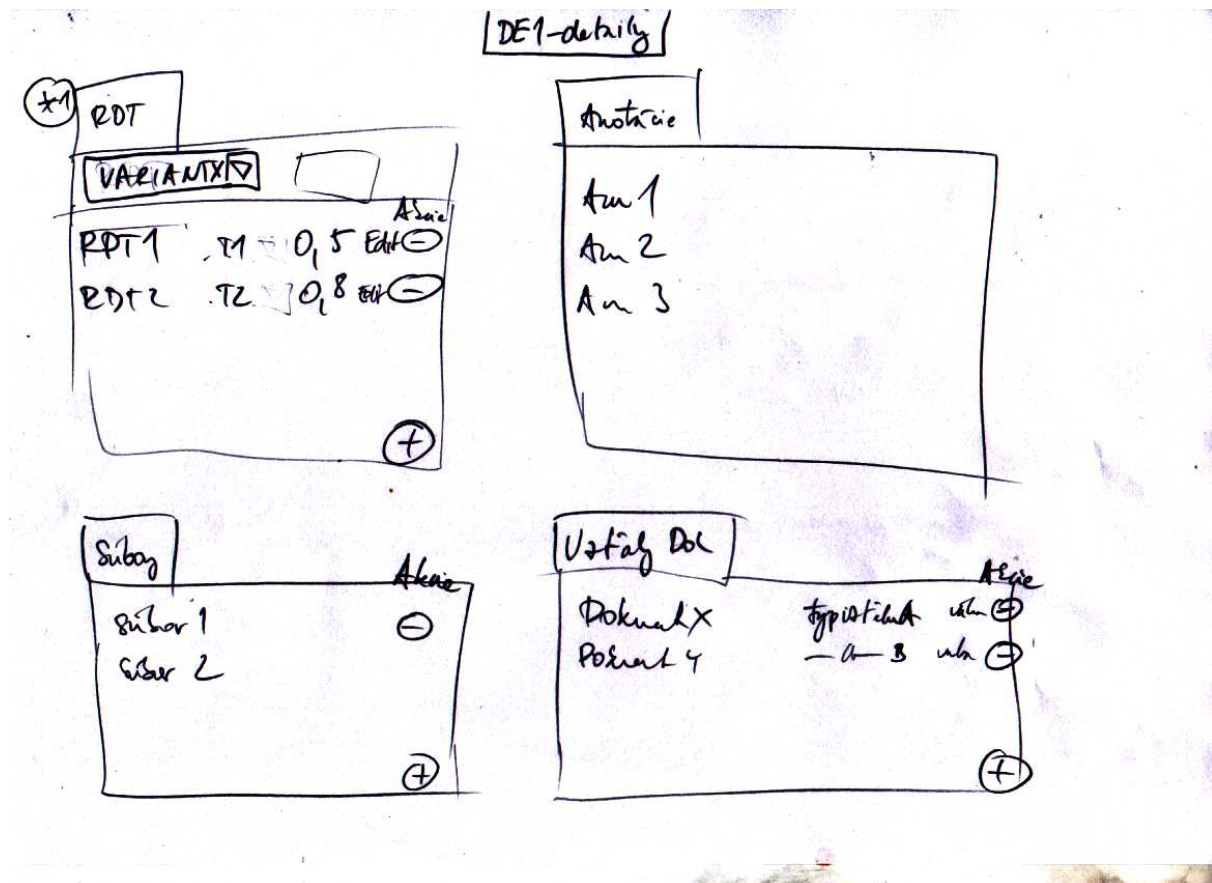
Obrazovka DE1 zobrazuje detail dokumentu v režime editovania. Je na nej znázornené editovacie okno pre úpravu obsahu dokumentu. Pod oknom sa nachádzajú akcie pre načítanie obsahu zo súboru, uloženie dokumentu a náhľad dokumentu. Na pravej strane sa nachádzajú taby s informáciami, ktoré sa týkajú dokumentu. Obsah tabov je vysvetlený na obrazovke DE1-detaily. Nad editovacím oknom sa nachádza možnosť prepnutia do režimu prezerania dokumentu a možnosť vidieť históriu dokumentu (predchádzajúce verzie).



Obr. 14. Obrazovka DE1.

### A.3.1 DE1-detaily

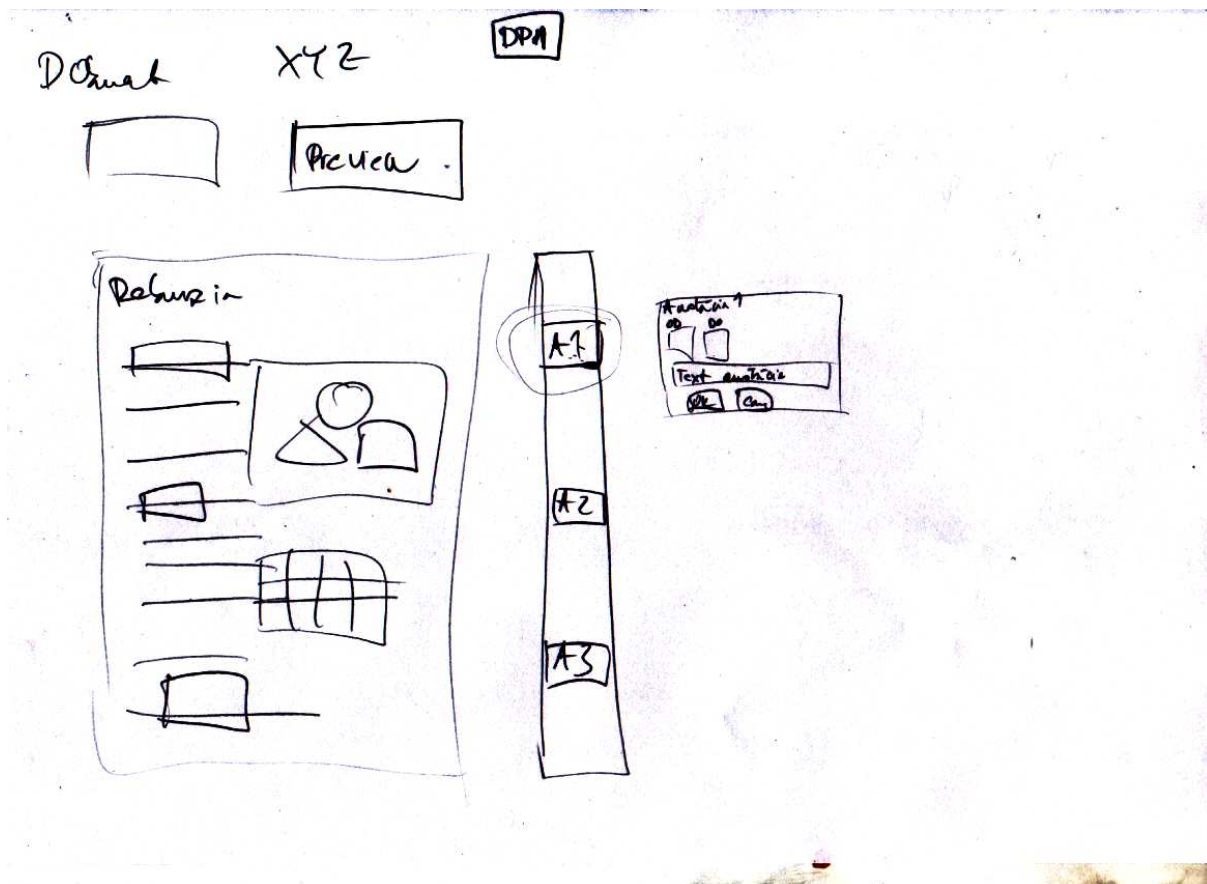
Pri každom dokumente sú zobrazené štyri taby zobrazené na tejto obrazovke. Tab RDT zobrazuje RDT priradené tomuto dokumentu. V tomto tabe sa dajú zobrazovať RDT z rôznych variantov. Tab anotácie zobrazuje anotácie označené v dokumente. Tab súbory obsahuje zoznam súborov (napr. obrázky), ktoré dokument obsahuje. Tab vzťahy s dokumentami zobrazuje aké vzťahy má tento dokument s ostatnými dokumentmi v repozitári.



Obr. 15. Obrazovka DE1-detaily.

## A.4 Obrazovka DP1

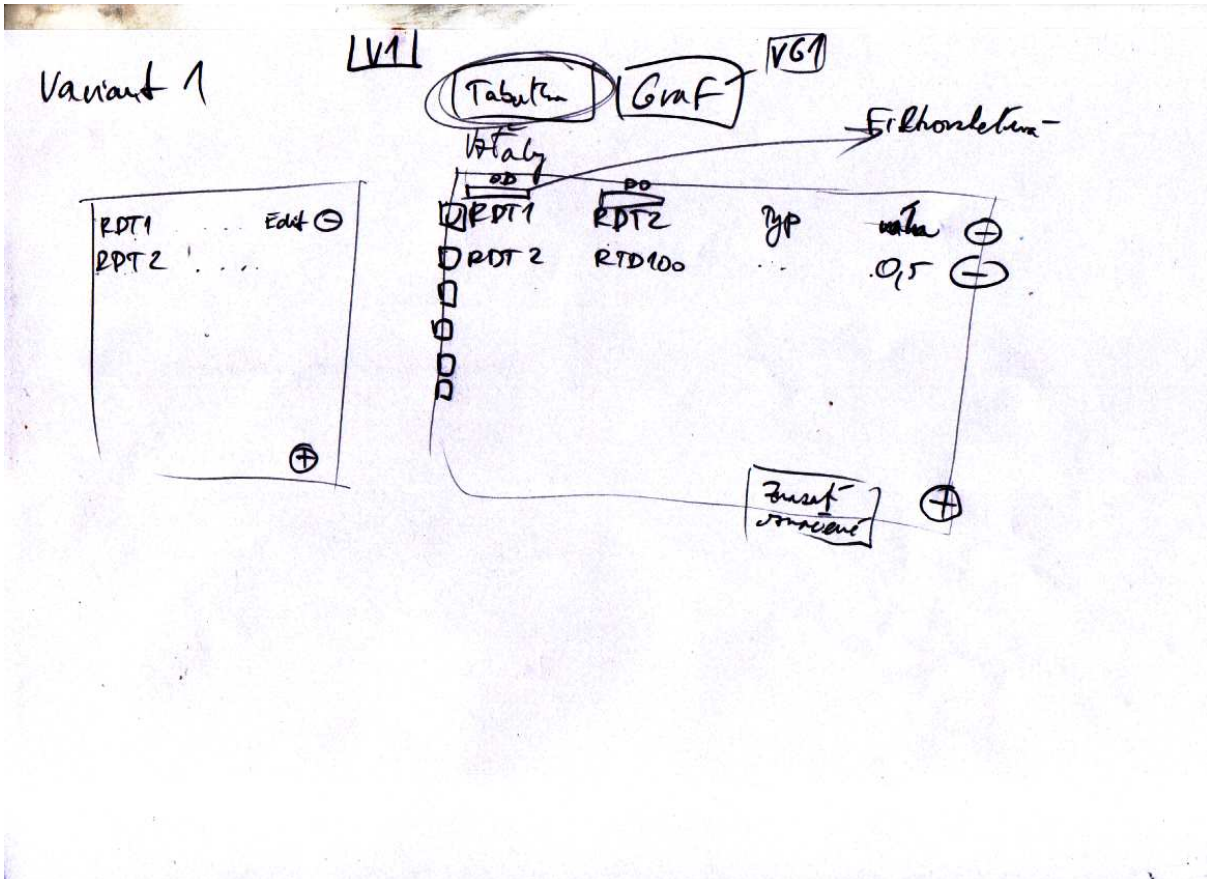
Obrazovka DP1 zobrazuje dokument v režime prezerania. V tomto režime sa vedľa dokumentu nachádza tzv. anotačný pásik, na ktorom sú zobrazené označené anotácie v dokumente. Po kliknutí na anotáciu v pásiku je možné anotáciu editovať.



Obr. 16. Obrazovka DP1.

## A.5 Obrazovka V1

Táto obrazovka obsahuje detail variantu metadát. Nachádzajú sa tu dve tabuľky. Prvá obsahuje zoznam RDT nachádzajúcich sa v tomto variante, druhá obsahuje vzťahy medzi týmito RDT. Každá tabuľka obsahuje príslušné akcie. Obrazovka obsahuje aj tlačidlo na zobrazenie variantu vo forme grafu (obrazovka VG1).

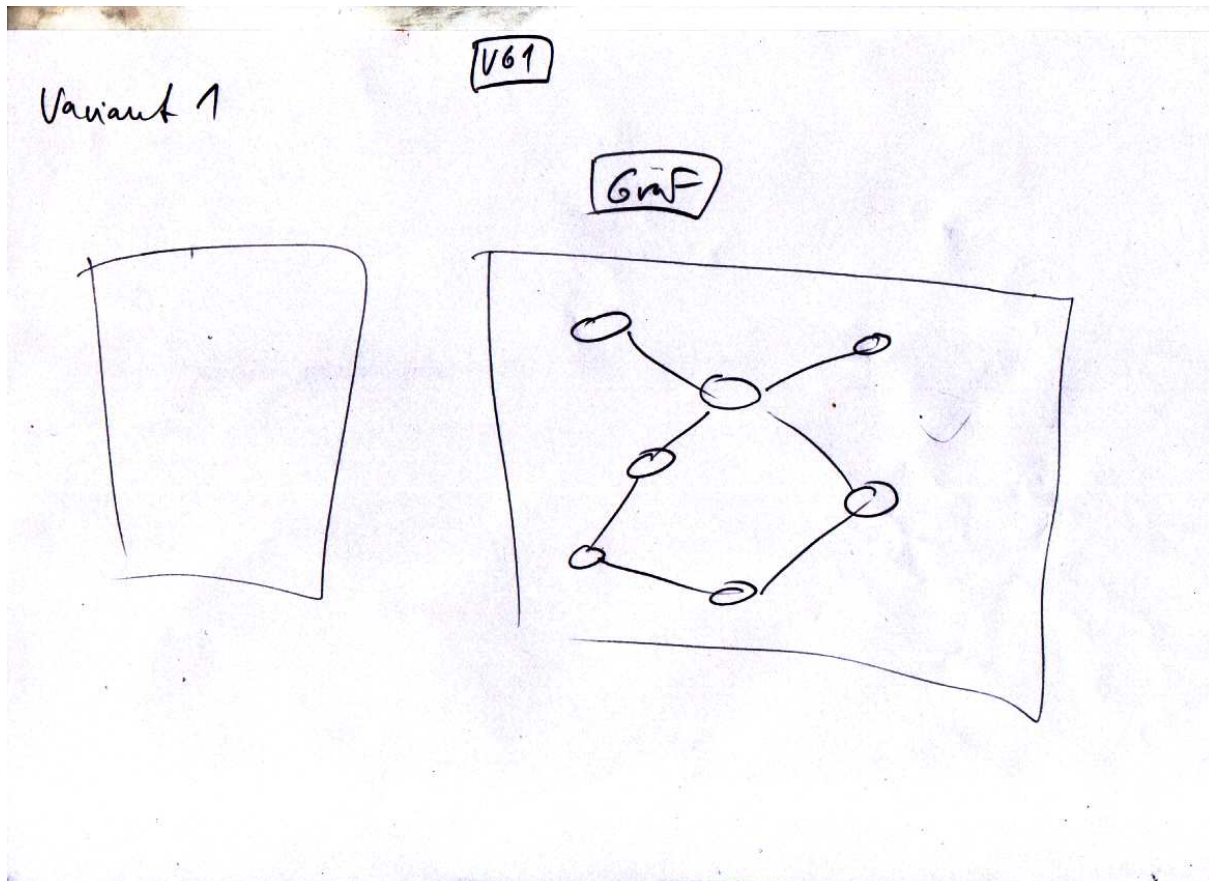


Obr. 17. Obrazovka V1.



## A.6 Obrazovka VG1

Na tejto obrazovke sa nachádza variant zobrazený vo forme grafu. Vľavo je zoznam RDT, ktoré patria do variantu, vpravo je graf, ktorý zobrazuje vzťahy medzi RDT.



Obr. 18. Obrazovka VG1.

## A.7 Tabuľky v systéme

Systém obsahuje niekoľko zobrazení vo forme tabuliek. Tu je uvedený ich zoznam s požadovanými funkcionalitami a obsahom.

Obrázovka	Tabuľka-Názov	Sĺpce	Akcie-Riadok	Akcie-Tabuľka	Typ
H1	Repositories	Name, #Documents, Actions	Show (RD2), Edit Properties, Delete	Add, Import, Export	Sortable, Scrollable
H1	Metadata	Name, #RDTs, #RDT Relationships, #Repositories, Actions	Show (V1), Edit Properties, Delete	Add, Import, Export	Sortable, Scrollable
RD2	Documents	Name, Valid, #RDTs, #Annotations, #Conflicts, Actions	Show (DP1), Show-Edit (DE1), Edit Properties, Move, Delete	Move, Delete	Sortable, Selectable, Filterable, Scrollable
RD2	Assigned Metadata	Name, #Document-RDT Relationships, Actions	Show, Edit Relationships, Remove	Add	-
DE1-detaily	Document RDTs	Name, Type, Weight, Actions	Edit, Remove	Add	Pivot – Metadata, Sortable, Selectable, Filterable, Scrollable
DE1-detaily	Document Annotations	Text + Selection	-	-	Sortable, Filterable
DE1-detaily	Document Files	Filename, Size, Actions	Remove	Choose File, Add	-
DE1-detaily	Document Relationships	Name, Type, Weight, Actions	Remove	Add	Sortable, Filterable, Scrollable
V1	RDTs	Name, Actions	Edit, Delete	Add, Delete	Sortable, Selectable, Filterable, Scrollable
V1	Relationships	From, To, Type, Weight, Actions	Edit, Delete	Add, Delete	Sortable, Selectable, Filterable, Pageable

Pozn.: V zátvorke je odkaz na obrázovku, kam sa dostať. Nie je to súčasť názvu akcie.