

Универзитет у Београду
Математички факултет

Дизајн и прототип регистра дигитализоване културне баштине

Мастер рад

Сања Животић

Београд, 30.09.2010.

Универзитет у Београду - Математички факултет
Мастер рад

Аутор: Сања Животић
Наслов: Дизајн и прототип регистра дигитализоване културне баштине
Ментор: др Саша Малков
Чланови
комисије: др Ненад Митић
др Жарко Мијајловић
Датум: 30.09.2010.

Садржај

1 Увод.....	3
2 Методологија.....	5
2.1 Случајеви коришћења.....	6
2.2 Дијаграми класа.....	7
2.3 Дијаграми ентитета и односа.....	9
3 Анализа и дефинисање захтева.....	11
3.1 Опис проблема и дефинисање захтева.....	11
3.1.1 Садржај регистра.....	11
3.1.2 Одржавање регистра.....	13
3.1.3 Претраживање регистра.....	13
3.2 Случајеви коришћења.....	14
3.2.1 Пријављивање / одјављивање корисника.....	14
3.2.2 Управљање корисницима.....	16
3.2.3 Управљање изворима ресурса.....	26
3.2.4 Управљање садржајима.....	30
3.2.5 Управљање дигиталним ресурсима.....	37
3.2.6 Управљање збиркама ресурса.....	43
3.2.7 Увоз података.....	49
3.2.8 Претраживање.....	51
3.3 Улоге.....	53
4 Пројектовање.....	55
4.1 Класе података.....	55
4.2 Моделовање класа података дијаграмима класа.....	62
4.3 База података.....	64
5 Имплементација.....	66
5.1 Програмски језик.....	66
5.2 Веб компоненте.....	67
5.3 База података.....	67
5.4 Објектно-релационо пресликавање.....	67
5.5 XML схема.....	68
5.5.1 Елементи.....	69
5.5.2 Атрибути.....	76
5.5.3 Типови.....	77
5.6 Претраживање.....	77
5.6.1 Текстуално претраживање.....	77
5.6.2 Претраживање по атрибутима.....	83
5.7 Развојно окружење.....	85

6	Закључак.....	87
7	Додаци.....	89
7.1	Додатак А – Структура базе података.....	89
7.2	Додатак Б - XML схема регистра.....	96
7.3	Додатак В - Пример XML документа.....	103
8	Референце.....	106

1 Увод

Националну културну баштину чине духовна и материјална добра народа, наслеђена од претходних генерација. Она се махом чувају у библиотекама, музејима, галеријама, архивима и сличним установама широм земље. Већина поменутих добара има велики национални значај, па би њихово оштећење или чак уништење могло бити ненадокнадиви губитак за један народ. Зато се, ради једноставније и безбедније употребе, веће јавне доступности и лакшег претраживања садржаја, јавила потреба за дигитализацијом културних добара.

Ради очувања културног идентитета и подизања свести о значају културног наслеђа за друштво и будуће генерације, Министарство Републике Србије подстиче употребу информационих технологија и дигитализацију националних културних добара. Све више установа, као што су нпр. Народна библиотека Србије, Републички завод за заштиту споменика културе, Библиотека Матице српске, Архив Србије, Народни Музеј у Београду, Етнографски музеј, Југословенска кинотека и многе друге, презентују своје садржаје у дигитализованом облику.

Како дигитализација узима маха и повећава се број и разноликост типова дигитализованих културних добара, тако расте и проблем проналажења истих. Централни регистар би омогућио претраживање и проналажење свих дигиталних ресурса на једном месту. Централизација података би олакшала приступ, презентацију и употребу националних културних добара. Координација рада установа и појединаца који дигитализују културне садржаје, спречила би дигитализацију истих садржаја више пута, чиме би процес дигитализације националне баштине био убрзан.

Централни регистар не би садржао саме дигиталне ресурсе, већ податке који их на неки начин описују и лоцирају. Ти подаци могу бити специфични за различите формате дигиталних ресурса, као и за типове културних добара на које се односе. Обезбеђивали би их извори ресурса, односно институције и појединци који располажу дигиталним ресурсима. Регистар би се попуњавао подацима ручно или аутоматским увозом датотека у неком преносивом формату. Тај формат би требало да буде такав да се лако може прилагођавати различитим изворима ресурса. Дигитални ресурси би се у регистру једноставно проналазили на основу задатих критеријума или помоћу речи и фраза које их описују.

У овом раду је разматрано шта би све требало да омогући регистар дигитализоване културне баштине и на које проблеме је могуће наићи при његовој имплементацији. Резултат разматрања је израда прототипа централног регистра.

2 Методологија

При развоју прототипа регистра коришћене су одабране технике различитих методологија, од којих већина потиче из објектно оријентисаних методологија [OOD].

Једна од специфичности регистра су концептуално једноставни процеси. Зато, није било потребе за применом техника које се односе на моделирање процеса, као што су, на пример, дијаграми активности и стања из објектних, или дијаграми тока података из структурних методологија. На основу постављених захтева и дефинисаних случајева коришћења је било могуће издвојити класе података. Модел класа података је додатно преведен у модел ентитета и односа, што није својство објектно оријентисаних методологија, али олакшава дефинисање схеме базе података.

Пројекат је развијан кроз следеће фазе:

- **Дефинисање захтева** (дефинисање граница система) је прва фаза развоја пројекта у којој се описују потенцијални проблеми у изради регистра и постављају захтеви које пројекат треба да испуни. Врши се упознавање са могућностима које треба да пружи централни регистар дигитализоване културне баштине.
- **Анализа** обухвата моделовање корисничких захтева дефинисаних у првој фази, израдом дијаграма случајева коришћења. Издавају се могући учесници, њихови међусобни односи и основна улога у систему.
- **Пројектовање** је фаза развоја пројекта у којој се, анализом случајева коришћења, издавају класе, односно скупови објеката са истим својствима и понашањем које постоје у систему. Уочене класе података и односи међу њима се моделују дијаграмима класа. Дефинише се структура релационе базе података и представља дијаграмом ентитета и односа. За израду *UML* (енгл. *Unified Modeling Language*) дијаграма, дијаграма случајева коришћења и дијаграма класа, као и за моделирање структуре базе података дијаграмом ентитета и односа, коришћен је *Visual Paradigm For UML 7.2 Community Edition*[UML, VP].
- **Имплементација** је последња фаза развоја у којој се врши израда веб апликације, односно софтверског решења за разматрани проблем. Алати коришћени у имплементацији прототипа регистра су детаљније описани у Поглављу 5. Примаран вид попуњавања регистра је увоз

веће количине података коришћењем *XML* докумената, па је у истом поглављу дат и опис елемената, атрибута и типова, као и графичка презентација одговарајуће *XML* схеме. Описано је и на који начин се дигитални ресурси претражују и проналазе у централном регистру.

2.1 Случајеви коришћења

Прва фаза у изради пројекта је упознавање са проблемом и дефинисање тога шта корисник очекује од система и које захтеве је потребно остварити током развоја пројекта. Различита коришћења система се моделују у фази анализе пројекта, израдом дијаграма случајева коришћења.

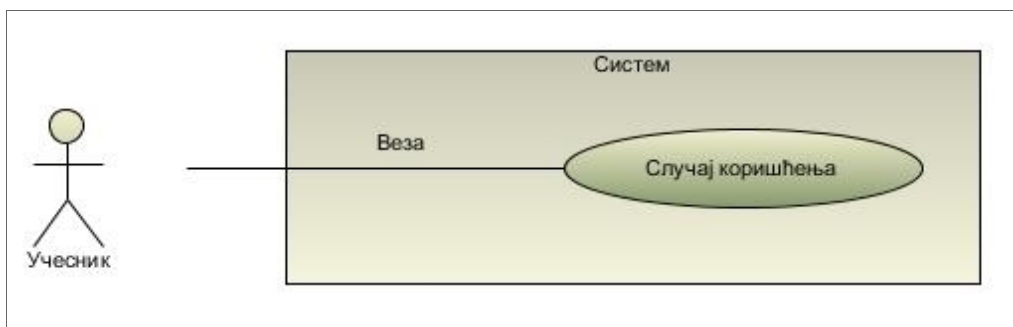
Случајеви коришћења (енгл. *use cases*) представљају технику моделирања понашања система, из аспекта интеракције између корисника и система.

Дијаграм случајева коришћења (енгл. *use case diagram*) приказује односе између учесника (енгл. *Actor*), као и између случајева коришћења. Описује шта систем треба да ради из угла корисника. Једноставан је за разумевање чак и без формалног познавања нотације, што има велики значај у комуникацији између развојног тима и корисника, као и у дискусији са клијентима.

При дефинисању случајева коришћења, потребно је уочити:

- границе система
- учеснике који су у интеракцији са системом
- односе између учесника и система

Основу случаја коришћења представља опис једне или више секвенци корака који воде једном заједничком циљу. Тај низ корака се назива „основни сценарио“. Могу постојати и додатни алтернативни сценарији, који описују низове корака који се обављају у неким посебним случајевима.



Слика 1: Дијаграм случајева коришћења

Опис једног случаја коришћења обухвата:

- назив
- кратак опис
- учеснике
- предуслове који морају бити задовољени да би случај коришћења могао да почне да се извршава
- постуслове који морају бити задовољени да би случај коришћења био успешно извршен
- основни сценарио одвијања случаја коришћења
- алтернативне сценарије одвијања случаја коришћења

2.2 Дијаграми класа

Једна од основних активности у фази пројектовања је препознавање класа, односно скупова објеката са истим својствима (атрибутима) и понашањем (операцијама). Класе се препознају на основу резултата анализе, а при изради овог пројекта, првенствено на основу случајева коришћења. Уочене класе и односи међу њима се моделују коришћењем дијаграма класа (енгл. *class diagram*). Дијаграми класа приказују статичку структуру система. На њима је представљена структура класа и различите врсте односа међу њима.

Име класе
-атрибут : тип податка
+операција (листа_аргумената) : тип резултата

Слика 2: Нотација класе

Дијаграм класа представља један од најчешће употребљаваних дијаграма UML-а. Класа података се приказује правоугаоником, подељеним хоризонтално на три дела (слика 2). У део на врху се уписује име класе, у део у средини атрибути класе, а у доњи део операције класе. Ознака испред назива атрибута или операције говори о њиховој доступности (приватни -, јавни +, заштићени #).

На дијаграму се представљају различите врсте односа међу класама:

Асоцијација (енгл. *association*) је најопштији однос међу класама, који означава да је за пуну функционалност објекта једне од класа неопходан један или више објеката друге класе.



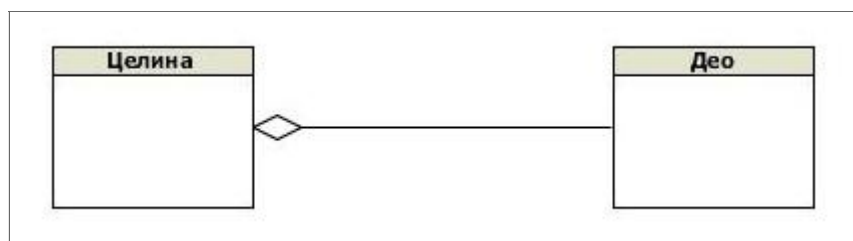
Слика 3: Асоцијација

Асоцијација може бити именована. На претходној слици, улоге на крајевима асоцијације представљају начин на које класе виде једна другу. Бројност представља број објеката једне класе које могу бити у односу са једним објектом друге класе.

Бројност	Значење
0..1	0 или 1 објекат n..m значи између n и m објеката
0..*	без ограничења броја објеката
1	тачно један објекат
1..*	бар један објекат

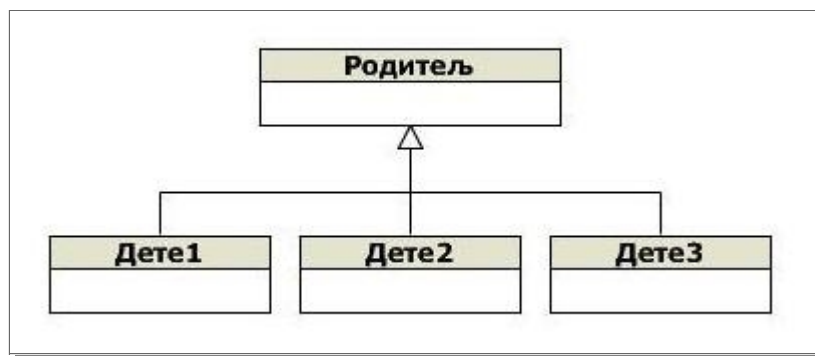
Слика 4: Бројност

Агрегација (енгл. *agregation*) је врста асоцијације у којој се објекат једне класе (целина) састоји од објеката друге класе (део). Другачије се назива однос целина/део.



Слика 5: Агрегација

Генерализација (енгл. *generalization*) се користи за приказивање односа општи случај/посебан случај. Уобичајено је да се назива и "наслеђивање", у ком случају се општи случај назива "родитељ", а посебан случај "дете". У контексту класа, класа родитељ се назива "наткласа", а класа дете "поткласа". Поткласа садржи ("наслеђује") све атрибуте, операције и односе који постоје у наткласи.

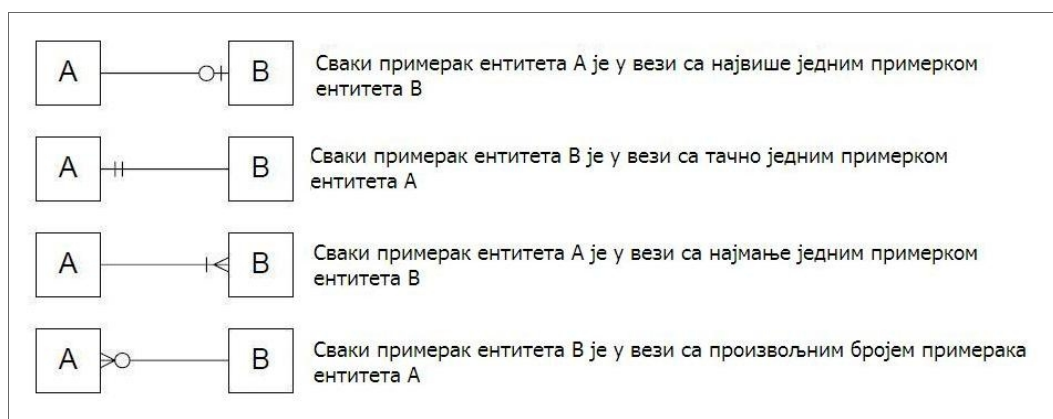


Слика 6: Генерализација

2.3 Дијаграми ентитета и односа

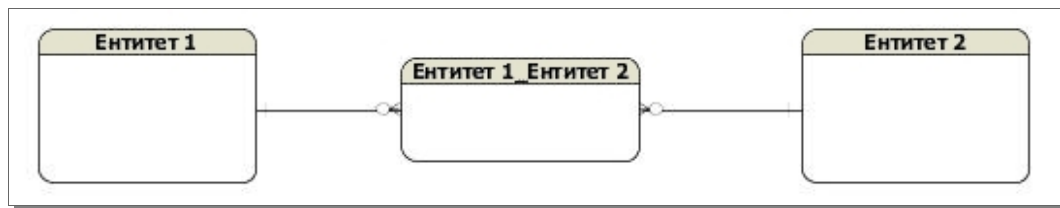
Дијаграм ентитета и односа (енгл. *Entity Relationship Diagram, ERD*) је врста дијаграма која се користи за моделирање релационе базе података [ERD]. Сваки ентитет на дијаграму представља табелу базе података, а сваки примерак ентитета је ред у одговарајућој табели. Односи између ентитета представљају стране кључеве.

Ентитети се на дијаграму представљају правоугаоником, у чијем се горњем делу налази назив ентитета, а у доњем, листа атрибута тог ентитета. Односи између ентитета се представљају испрекиданим линијама на чијем крају се налазе симболи који означавају обавезност и бројност везе. Круг (O) указује на то да је веза опциона, тј. да је најмањи број примерака једног ентитета који је у вези са примерком другог ентитета нула. Црта (|) означава да је веза између ентитета обавезна. Други симбол на линији означава бројност, где црта (|) значи да је највише један примерак првог ентитета у вези са примерком другог ентитета, док симбол (≤) указује да је највећи број примерака једног ентитета, везаних за примерак другог ентитета, произвољан.



Слика 7: Обавезност и бројност односа ентитета

У случају да примерак једног ентитета може бити у вези са произвољним бројем примерака другог ентитета, а важи и обрнуто, примерак другог ентитета може бити везан за произвољан број примерака првог (енгл. *many-to-many relationship*), тада се однос између ентитета представља новим ентитетом, а везе између њих пуном линијом.



Слика 8: Дијаграм ентитета и односа

3 Анализа и дефинисање захтева

3.1 Опис проблема и дефинисање захтева

С обзиром на експерименталну природу пројекта, прикупљање захтева је обављено углавном као самостално истраживање у одговарајућој области, уз неформалне разговоре са различитим субјектима који су активни у области дигитализације културне баштине.

При изградњи централног регистра дигиталних ресурса, јавља се више различитих проблема и питања. Технички проблеми се састоје од прављења схеме метаподатака, дизајна базе података и корисничког интерфејса, имплементације и сличног. Међутим, незаобилазна су и питања социјалног карактера која се односе на дизајнирање система који је једноставан за коришћење и заиста доприноси очувању и презентацији националних културних добара.

3.1.1 Садржај регистра

Централни регистар би требало да садржи податке који потичу од више различитих извора ресурса. То би заправо требало да буду метаподаци, односно подаци о постојећим дигиталним ресурсима који објашњавају, лоцирају или на неки други начин омогућавају лакшу употребу ресурса. Потребно је успоставити сарадњу са што већим бројем извора ресурса, како би регистар располагао већим бројем разноврсних садржаја.

Сваки извор ресурса, био то музеј, библиотека, галерија или нека друга установа, располаже одређеним бројем садржаја. Временом, све више тих садржаја се преводи у дигитални облик. Један садржај може бити представљен у више различитих дигиталних формата. На пример, књига „Благо цара Радована“, аутора Јована Дучића, може бити доступна у текстуалном формату *pdf* и *doc*, док је слику „Поље среће“, аутора Драгана Малешевића Тапија, могуће представити у више формата, као што су на пример *jpg* и *png*.

Постоји велики број различитих типова садржаја, као што су књига, часопис, слика, фотографија, видео, аудио записи итд. Без обзира на тип, сваки садржај мора имати назив и, по могућству, кратак опис. Сваки тип садржаја описује се својим атрибутима. На пример, када се говори о некој књизи, најчешће је потребна и информација о њеном аутору, издавачу или њен

*ISBN**. Битно је и на ком језику је књига написана и у ком периоду је издата. За сваки тип садржаја, постоји прегршт сличних одлика. С обзиром на њихову бројност и разноликост, прототип централног регистра подржава само основне типове и њихове најзначајније особине. У случају реалне примене регистра, број типова би био већи, па је могућност надоградње система још један од проблема који се поставља при изради овог пројекта.

Дигитални облик неког садржаја називамо и дигиталном репрезентацијом тог садржаја или дигиталним ресурсом. Дигитални ресурси такође морају имати назив и краћи опис, ако је познат. За разлику од садржаја, они се описују и *MIME* типом*, величином и локацијом на којој се заправо налазе. Локација представља јединствену и недвосмислену референцу на дигитални ресурс, путем које корисник може приступити садржају ресурса. За одржавање локација на којима се налазе дигитални ресурси и поступака за њихово преузимање, одговорни су извори ресурса. Најпожељнији општи облик реферисања јесте путем *URI-ја***.

Један дигитални ресурс може презентовати више од једног садржаја. На пример, можемо имати једну слику формата *jpg* на којој се налазе два различита акварела или један *pdf* документ који садржи једну књигу и неколико фотографија.

Дигиталне ресурсе који потичу било од истог, било од више различитих извора, могуће је груписати по збиркама. Осим што свака збирка може имати већи број ресурса, важи и обрнуто. Један дигитални ресурс може се наћи у више различитих збирки. На пример, дигитализована књига „Благо цара Радована“ може да припада збирци „Изабрана дела Јована Дучића“, као и збирци „Најчитаније књиге у Србији“.

Дигиталне репрезентације одређених садржаја не морају бити јавне. Извори ресурса могу имати различиту политику објављивања и услова коришћења

* **ISBN** (енгл. *International Standard Book Number*, међународни стандардни број књиге) је јединствени идентификатор књиге. Најпре се користио као *SBN* (енгл. *Standard Book Number*, стандардни број књиге) у Великој Британији. Седамдесетих година прошлог века постаје опште прихваћен и преименован у *ISBN*.

* **MIME** (енгл. *Multipurpose Internet Mail Extensions*, вишенаменска проширења за пошту) је интернет стандард који је изворно коришћен за проширење могућности електронске поште на слање: текста који не мора бити *ASCII*, нетекстуалних података закачених за поруку и порука чија тела чини више делова и информација у заглављу поруке које нису *ASCII*. Данас је свеприсутан у размени датотека на Интернету.

** **URI** (енгл. *Uniform Resource Identifier*, јединствени идентификатор ресурса) је низ карактера који се користи за лоцирање ресурса на Интернету. Веб адреса се често наводи као синоним.

тј. лиценцирања ресурса. С обзиром да се у регистру налазе само подаци који описују и лоцирају ресурсе, нема потребе да се старање о правима приступа остварује на страни регистра. Питањима овлашћења се бави сам извор ресурса приликом приступа регистрованим локацијима.

3.1.2 Одржавање регистра

Администратори система и представници извора ресурса могу на два начина извршити унос и ажурирање података. Први и основни би био увозом датотеке у неком преносивом формату. Систем треба да омогући проверу исправности, парсирање и увоз, односно издвајање података из датотеке и смештање у базу података. Потребно је обезбедити довољно преносив формат који трпи даља допуњавања, као и различитости специфичне за одређене изворе ресурса.

Други начин управљања подацима је „ручно“, односно путем интерфејса који омогућује да се додају, мењају и бришу појединачни подаци.

Постоје одређене разлике у могућностима приступа одређеним подацима од стране различитих корисника. Администратори система уз одговарајућа овлашћења, могу имати право увоза, као и манипулисања подацима свих извора ресурса. Представници извора могу то чинити само за своје ресурсе, док су подаци других извора за њих доступни само за читање и претраживање. Обични корисници немају права додавања, ажурирања и брисања садржаја. Они могу само да читају, прегледају и претражују све јавне податке.

За додавање других корисника, било администратора, представника извора ресурса или обичних корисника, неопходно је имати дозволу за управљање корисницима.

Због постојања више различитих корисничких улога са различитим овлашћењима за управљање системом и подацима, неопходно је омогућити и аутентификацију корисника. На основу корисничког имена и лозинке, систем зна улогу пријављеног корисника и омогућује или забрањује одговарајуће операције.

3.1.3 Претраживање регистра

Не треба заборавити једну од најзначајнијих ставки при изради централног регистра, а то је претраживање садржаја регистра. Треба обезбедити претрагу

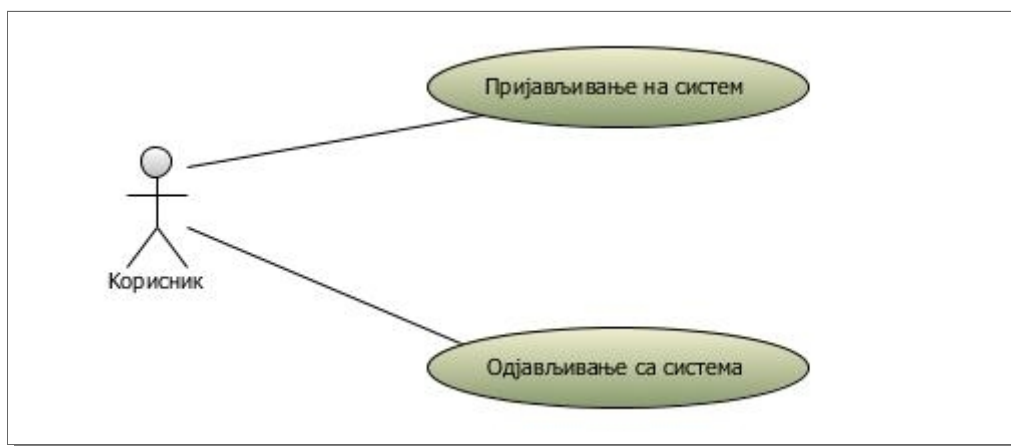
ресурса по задатим критеријумима. Пожељно је да се за критеријуме претраге могу задати бар основна својства, као што су назив, опис, аутори и тип садржаја, а могло би да се омогући и задавање детаљнијих података у зависности од типа садржаја. Могућност једноставне, брзе и квалитетне претраге је један од најважнијих проблема у изради овог пројекта. Из угла обичног корисника регистра, претраживање је уједно и основна функција регистра.

3.2 Случајеви коришћења

Међу захтевима које систем треба да испуни, могу се уочити следеће групе послова:

- пријављивање/одјављивање корисника
- управљање корисницима
- управљање изворима ресурса
- управљање садржајима
- управљање дигиталним ресурсима
- управљање збиркама ресурса
- увоз података
- претраживање

3.2.1 Пријављивање / одјављивање корисника



Слика 9: Дијаграм случајева коришћења – Пријављивање / одјављивање корисника

3.2.1.1 Пријављивање на систем

Опис

Обичан корисник не мора бити пријављен на систем, али да би могао да користи посебне привилегије, он мора бити идентификован.

Учесници

- Корисник са посебним привилегијама

Предуслови

Потребно је да корисник има отворен налог (видети случај коришћења 3.2.2.8. Додавање појединачног корисника).

Постуслови

Корисник је пријављен на систем и може радити све послове за које је овлашћен. По завршетку употребе система, потребно је да се корисник одјави (видети случај коришћења 3.2.1.2. Одјављивање са система).

Основни сценарио

1. Корисник уноси своје корисничко име и лозинку.
2. Систем потврђује исправност унетог корисничког имена и лозинке.
3. Систем евидентира податке о пријављеном кориснику како би могао да управља ауторизацијама и врши редирекцију на почетну страну.

Алтернативни сценарио

- Корисник је унео неисправно корисничко име и лозинку. Систем исписује одговарајућу поруку о грешци и поново приказује формулар за пријављивање.

3.2.1.2 Одјављивање са система

Опис

Корисник се одјављује са система.

Учесници

- Корисник са посебним привилегијама

Предуслови

Корисник је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Корисник је одјављен са система. Од овог тренутка може користити регистар само као обичан неауторизован корисник.

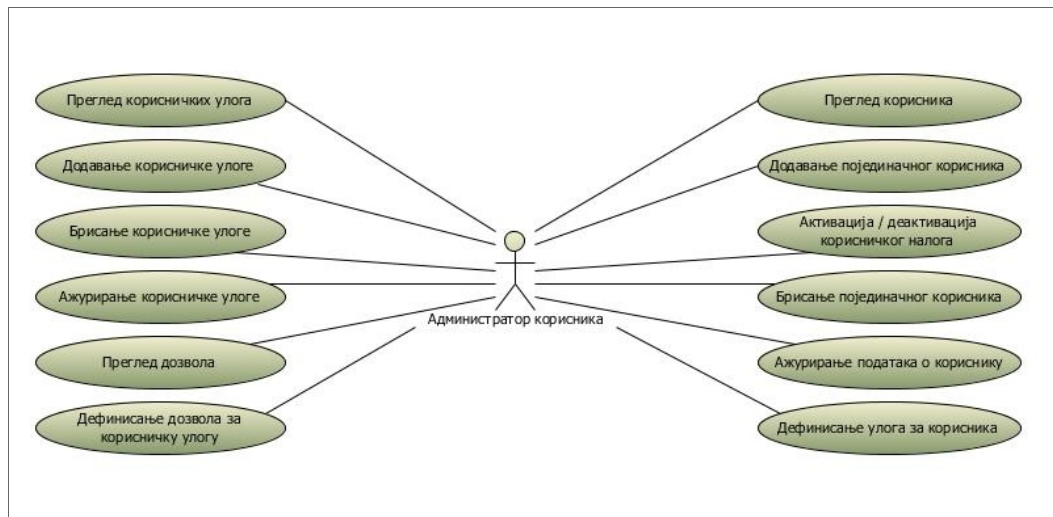
Основни сценарио

1. Корисник бира операцију одјављивања са система.
2. Систем евидентира да корисник више није пријављен на систем како би могао да управља ауторизацијама и врши редирекцију на почетну страну.

Алтернативни сценарио

- Не постоји

3.2.2 Управљање корисницима



Слика 10: Дијаграм случајева коришћења: Управљање корисницима

3.2.2.1 Преглед корисничких улога

Опис

Корисник прегледа податке о корисничким улогама.

Учесници

- Администратор корисника

Предуслови

Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Приказани су подаци о корисничким улогама.

Основни сценарио

1. Систем приказује листу свих расположивих корисничких улога.
2. Корисник бира детаљнији преглед података за жељену корисничку улогу.
3. Систем приказује назив и опис корисничке улоге, све њене дозволе и кориснике који је имају. За сваког корисника, приказано је његово име, презиме и корисничко име, а за сваку дозволу, њен назив и опис.

Алтернативни сценарио

Не постоји.

3.2.2.2 Додавање корисничке улоге

Опис

Додавање нове корисничке улоге.

Учесници

- Администратор корисника

Предуслови

Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о новој корисничкој улози су евидентирани.

Основни сценарио

1. Систем приказује листу претходно дефинисаних корисничких улога.
2. Корисник бира опцију додавања нове корисничке улоге.
3. Отвара се формулар и корисник уноси податке о новој корисничкој улози:
 - назив
 - опис
4. Систем генерише интерни идентификатор улоге, додељује јој га и евидентира податке о новој корисничкој улози.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од додавања нове корисничке улоге у току уноса података. Систем поново покреће основни сценарио.
- Недостатак података: Нису унете све неопходне информације о корисничкој улози. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података: Већ постоји корисничка улога са унетим називом. Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.2.3 Брисање корисничке улоге

Опис

Брисање изабране корисничке улоге.

Учесници

- Администратор корисника

Предуслови

- Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).
- Не постоји ниједан корисник са улогом која се брише.

Постуслови

Корисничка улога је обрисана.

Основни сценарио

1. Систем приказује листу претходно дефинисаних корисничких улога. Опција брисања постоји само за улоге без корисника.
2. Корисник бира операцију брисања жељене корисничке улоге.
3. Систем захтева потврђивање брисања изабране корисничке улоге.
4. Корисник потврђује брисање изабране корисничке улоге.
5. Систем брише изабрану корисничку улогу.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од брисања корисничке улоге, у кораку потврђивања (3). Систем поново покреће основни сценарио.

3.2.2.4 Ажурирање корисничке улоге

Опис

Ажурирање података о изабраној корисничкој улози.

Учесници

- Администратор корисника

Предуслови

Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о изабраној корисничкој улози су промењени.

Основни сценарио

1. Систем приказује листу претходно дефинисаних корисничких улога.
2. Корисник бира операцију промене жељене корисничке улоге.
3. Отвара се формулар са подацима о изабраној корисничкој улози и корисник врши жељене измене.
4. Систем ажурира информације о изабраној корисничкој улози.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од промене корисничке улоге у току измене података. Систем поново покреће основни сценарио.
- Недостатак података: Нису унете све неопходне информације о корисничкој улози. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података: Већ постоји нека друга корисничка улога са унетим називом. Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.2.5 Преглед дозвола

Опис

Корисник прегледа податке о дозволама.

Учесници

- Администратор корисника

Предуслови

Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Приказани су подаци о дозволама.

Основни сценарио

1. Систем приказује листу свих расположивих дозвола.
2. Корисник бира детаљан преглед података за жељену дозволу.
3. Систем приказује назив и опис дозволе и све кориснике и корисничке улоге који је имају. За сваког корисника, приказано је његово име, презиме и корисничко име, а за сваку корисничку улогу, њен назив и опис.

Алтернативни сценарио

Не постоји.

3.2.2.6 Дефинисање дозвола за корисничку улогу

Опис

Дефинисање дозвола за изабрану корисничку улогу.

Учесници

- Администратор корисника

Предуслови

Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Дефинисане су дозволе које важе за изабрану корисничку улогу.

Основни сценарио

1. Систем приказује листу свих корисничких улога.
2. Корисник бира преглед дозвола за жељену корисничку улогу.
3. Систем приказује листу свих расположивих дозвола. Дозволе које важе за изабрану корисничку улогу су штиклиране.
4. На приказаној листи, корисник штиклира/одштиклира дозволе које ће/неће важити за корисничку улогу и потврђује измену.
5. Систем евидентира које дозволе важе за изабрану корисничку улогу.

Алтернативни сценарио

- Одустајање од операције: Током корака 4, корисник одлучује да одустане од дефинисања дозвола за корисничку улогу. Систем поново покреће основни сценарио.

3.2.2.7 Преглед корисника

Опис

Корисник прегледа податке о корисницима.

Учесници

- Администратор корисника

Предуслови

Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Приказани су подаци о корисницима.

Основни сценарио

1. Систем приказује листу активних и листу неактивних корисника. За сваког корисника, приказано је његово име, презиме и корисничко име, као и опције детаљнијег прегледа личних података и корисничких улога.

Алтернативни сценарио

Не постоји.

3.2.2.8 Додавање појединачног корисника

Опис

Додавање новог корисника.

Учесници

- Администратор корисника

Предуслови

Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о новом кориснику су евидентирани.

Основни сценарио

1. Систем приказује листу активних и листу неактивних корисника.
2. Корисник бира опцију додавања новог корисника.
3. Отвара се формулар и корисник уноси податке о новом кориснику:
 - корисничко име
 - лозинка (и потврда лозинке)
 - име
 - презиме
 - е-пошта
 - датум рођења
 - пол
 - коментар

Подразумева се да је сваки нови корисник активан (погледати случај коришћења 3.2.2.9. Активација / деактивација корисничког налога).

4. Систем евидентира податке о новом кориснику.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од додавања новог корисника у току уноса података. Систем поново покреће основни сценарио.
- Недостатак података: Нису унете све неопходне информације о новом кориснику. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података:
 - Потврда лозинке се не поклапа са самом лозинком. Систем приказује поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.
 - Већ постоји корисник са унетим корисничким именом. Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.
 - Лозинка је превише једноставна (недовољан број карактера, без цифара...). Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.2.9 Активација / деактивација корисничког налога

Опис

Активација / деактивација налога за изабраног корисника.

Учесници

- Администратор корисника

Предуслови

Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Налог за изабраног корисника је активиран / деактивиран.

Основни сценарио

1. Систем приказује листу активних и листу неактивних корисника. За сваког корисника, приказано је његово име, презиме и корисничко име. За активне кориснике постоји и опција деактивације њиховог налога. Такође, могуће је активирати налог било ког неактивног корисника.
2. Корисник активира / деактивира налог за изабраног корисника.
3. Систем евидентира да је изабрани корисник активан / неактиван.

Алтернативни сценарио

Не постоји

3.2.2.10 Брисање појединачног корисника

Опис

Брисање изабраног корисника.

Учесници

- Администратор корисника

Предуслови

Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Корисник је обрисан.

Основни сценарио

1. Систем приказује листу активних и листу неактивних корисника.
2. Корисник бира операцију брисања жељеног корисника (опција брисања не постоји за тренутног корисника).
3. Систем захтева потврђивање брисања изабраног корисника.
4. Корисник потврђује брисање изабраног корисника.
5. Систем брише изабраног корисника.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од брисања корисника, у кораку потврђивања (3). Систем поново покреће основни сценарио.

3.2.2.11 Ажурирање података о кориснику

Опис

Ажурирање података о изабраном кориснику.

Учесници

- Администратор корисника

Предуслови

Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о изабраном кориснику су промењени.

Основни сценарио

1. Систем приказује листу активних и листу неактивних корисника.
2. Корисник бира операцију промене података о жељеном кориснику.
3. Отвара се формулар са подацима о изабраном кориснику. Корисник врши жељене измене података. У случају да корисник не промени лозинку, она остаје иста као пре измене података.
4. Систем ажурира информације о изабраном кориснику.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од операције током измене података о кориснику. Систем поново покреће основни сценарио.
- Недостатак података: Нису унети сви неопходни подаци о кориснику. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података:
 - Корисник је променио лозинку и унео потврду лозинке која се не поклапа са њом. Систем приказује поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.
 - Већ постоји неки други корисник са унетим корисничким именом. Систем приказује поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.2.12 Дефинисање улога за корисника

Опис

Дефинисање улога за изабраног корисника.

Учесници

- Администратор корисника

Предуслови

- Администратор корисника је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).
- У систему постоје евидентиране корисничке улоге.

Постуслови

Дефинисане су улоге корисника.

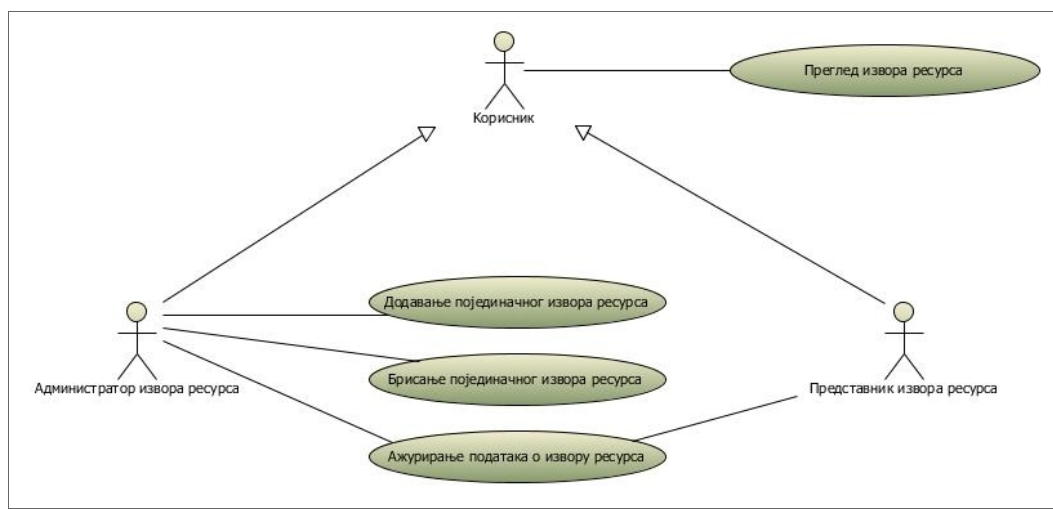
Основни сценарио

1. Систем приказује листу активних и листу неактивних корисника.
2. Корисник бира преглед улога изабраног корисника.
3. Систем приказује листу свих улога корисника, опцију за брисање сваке од њих, као и листу корисничких улога које корисник не поседује (измена улога тренутног корисника није могућа).
 - а) Из приказане листе улога које корисник нема, администратор корисника бира ону коју ће доделити кориснику. Поједине улоге корисника се везују за извор ресурса тј. представљају послове који се односе само на одређени извор. На пример, ако је у питању улога представника извора ресурса, систем омогућује избор извора за који ће корисник бити представник. У том случају, администратор корисника бира извор ресурса и потврђује измене. Систем евидентира нову улогу корисника.
Напомена: Један корисник може бити представник не само једног већ и више извора ресурса.
 - б) Из приказане листе улога корисника, администратор корисника бира улогу коју ће одузети кориснику. Систем брише улогу корисника.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од дефинисања улога за изабраног корисника, током корака 3. Систем поново покреће основни сценарио.

3.2.3 Управљање изворима ресурса



Слика 11: Дијаграм случајева коришћења – Управљање изворима ресурса

3.2.3.1 Преглед извора ресурса

Опис

Корисник прегледа податке о постојећим изворима ресурса.

Учесници

- Било који корисник

Предуслови

Не постоје.

Постуслови

Приказани су основни подаци о свим изворима ресурса.

Основни сценарио

1. Систем приказује листу постојећих извора ресурса. За сваки извор ресурса, приказују се његов назив, број дигиталних ресурса, опис и веб адреса. У зависности од права корисника пријављеног на систем, постоје опције за додавање, измену и брисање извора ресурса.

Алтернативни сценарио

Не постоји.

3.2.3.2 Додавање појединачног извора ресурса

Опис

Додавање новог извора ресурса.

Учесници

- Администратор извора ресурса

Предуслови

Администратор извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о новом извору ресурса су евидентирани.

Основни сценарио

1. Систем приказује листу постојећих извора ресурса.
2. Корисник бира опцију додавања новог извора.
3. Отвара се формулар и корисник уноси податке о новом извору ресурса:
 - назив
 - опис
 - веб адреса извора
4. Систем генерише интерни идентификатор извора, додељује му га и евидентира нови извор ресурса.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од додавања новог извора ресурса у току уноса података. Систем поново покреће основни сценарио.
- Недостатак података: Нису унети сви неопходни подаци о извору ресурса. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података: Већ постоји неки други извор ресурса са унетим називом. Систем приказује поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.3.3 Брисање појединачног извора ресурса

Опис

Брисање изабраног извора ресурса.

Учесници

- Администратор извора ресурса

Предуслови

Администратор извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Извор ресурса, све његове збирке, садржаји, дигитални ресурси и аутори су обрисани.

Основни сценарио

1. Систем приказује листу постојећих извора ресурса.
2. Корисник бира операцију брисања жељеног извора.
3. Систем захтева потврђивање брисања изабраног извора ресурса.
4. Корисник потврђује брисање изабраног извора.
5. Систем брише изабрани извор ресурса, као и све његове збирке, садржаје, дигиталне ресурсе и ауторе .

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од брисања извора ресурса, у кораку потврђивања (3). Систем поново покреће основни сценарио.

3.2.3.4 Ажурирање података о извору ресурса

Опис

Ажурирање података о изабраном извору ресурса.

Учесници

- Администратор извора ресурса
- Представник извора ресурса

Предуслови

Администратор извора ресурса или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о изабраном извору ресурса су промењени.

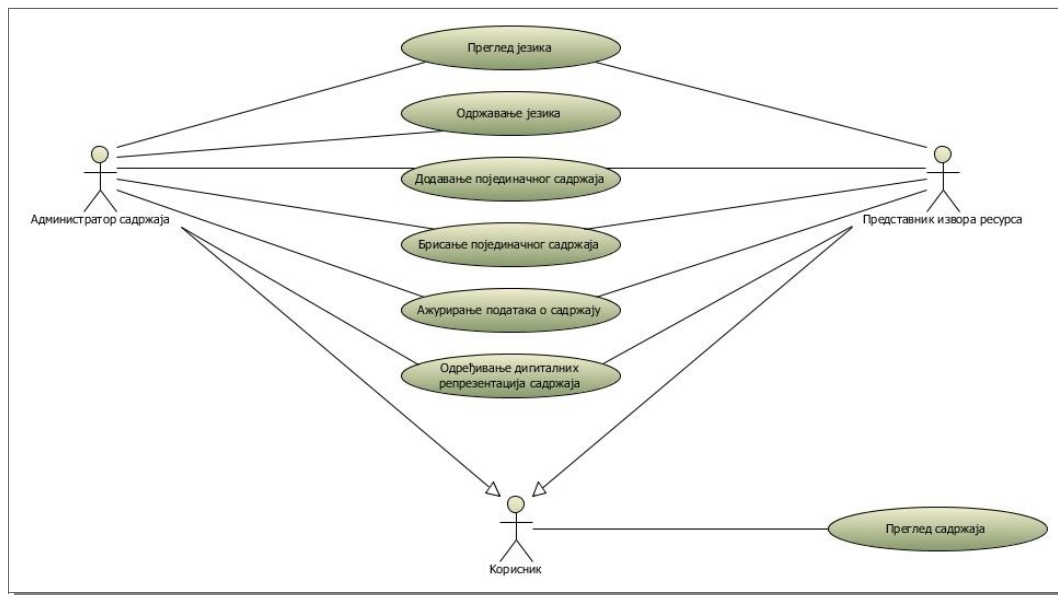
Основни сценарио

1. Систем приказује листу постојећих извора ресурса. Администратори извора ресурса могу да мењају податке свих извора, док представници извора ресурса могу да мењају само изворе које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира операцију промене жељеног извора ресурса.
3. Отвара се формулар са подацима о изабраном извору ресурса и корисник врши жељене измене.
4. Систем ажурира информације о изабраном извору ресурса.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од промене извора ресурса у току измене података. Систем поново покреће основни сценарио.
- Недостатак података: Корисник није унео све неопходне информације о извору ресурса. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података: Већ постоји неки други извор ресурса са унетим називом. Систем приказује поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.4 Управљање садржајима



Слика 12: Дијаграм случајева коришћења – Управљање садржајима

3.2.4.1 Одржавање језика

Опис

Одржавање допуштених језика садржаја.

Учесници

- Администратор садржаја

Предуслови

Администратор садржаја је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Евидентирани су подаци о језицима који могу бити заступљени у неком садржају.

Основни сценарио

- Систем приказује листу свих језика и њихових ознака по стандарду ISO 639-1. За језике који се не налазе у систему, постоји опција за њихово додавање у систем. Постојећи језик могуће је обрисати из система ако не постоји ниједан садржај на њему.
- Додавање и брисање језика:
 - Корисник бира опцију додавања жељеног језика. Систем генерише

интерни идентификатор, додељује га језику и евидентира његове податке.

- б) Корисник бира опцију брисања жељеног језика. Систем брише изабрани језик.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од одређивања допуштених језика садржаја.

3.2.4.2 Преглед језика

Опис

Преглед података о допуштеним језицима садржаја.

Учесници

- Администратор садржаја
- Представник извора ресурса

Предуслови

Администратор садржаја или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Приказани су основни подаци о допуштеним језицима садржаја.

Основни сценарио

1. Систем приказује листу свих допуштених језика садржаја. За сваки језик, приказују се његова ознака и назив. Систем нуди избор уређивања листе по називу и по ознаци језика. Језици су подразумевано уређени по називу, а у случају да корисник промени начин уређивања, систем приказује листу уређену по изабраном критеријуму.

Алтернативни сценарио

Не постоји.

3.2.4.3 Преглед садржаја

Опис

Корисник прегледа податке о садржајима.

Учесници

- Било који корисник

Предуслови

Не постоје.

Постуслови

Приказани су подаци о садржајима.

Основни сценарио

1. Систем омогућује избор извора ресурса.
2. Корисник бира жељени извор ресурса из листе понуђених.
3. Систем приказује листу садржаја изабраног извора ресурса, као и њихов укупан број. За сваки садржај, приказан је његов назив, врста и основни подаци у зависности од врсте садржаја. У зависности од права корисника пријављеног на систем, постоје опције за додавање, измену и брисање садржаја.
4. Корисник бира садржај за који су му потребне детаљније информације.
5. Систем приказује детаљније податке о садржају, као и листу дигиталних репрезентација истог. За сваку дигиталну репрезентацију садржаја, наведена је локација на којој се она заправо налази (веб адреса), тип фајла и величина.

Алтернативни сценарио

Не постоји.

3.2.4.4 Додавање појединачног садржаја

Опис

Додавање новог садржаја.

Учесници

- Администратор садржаја
- Представник извора ресурса

Предуслови

Администратор садржаја или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о новом садржају су евидентирани.

Основни сценарио

1. Систем омогућује избор извора ресурса. Администратори извора ресурса могу да додају садржаје свих извора, док представници извора

ресурса то могу чинити само за изворе које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).

2. Корисник бира жељени извор из листе понуђених.
3. Систем приказује листу садржаја изабраног извора ресурса.
4. Корисник бира операцију додавања новог садржаја.
5. Отвара се формулар и корисник уноси податке о новом садржају:
 - извор ресурса (иницијално је изабран извор ресурса који је корисник претходно одабрао, али могуће је променити извор на неки други из листе извора за који корисник има одговарајуће привилегије)
 - локални идентификатор (идентификатор садржаја на страни извора ресурса)
 - назив
 - опис
 - врста (књига, часопис, звук, видео, слика, фотографија....)
 - подаци о ауторима (локални идентификатор, име, презиме, надимак, редни број и улога сваког аутора)
 - додатни подаци у зависности од врсте садржаја
6. Систем генерише интерни идентификатор садржаја, додељује му га и евидентира податке о новом садржају.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од додавања новог садржаја у току уноса података. Систем поново покреће основни сценарио.
- Недостатак података: Корисник није унео све неопходне информације о садржају. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података: Већ постоји садржај изабраног извора ресурса са наведеним локалним идентификатором. Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.4.5 Брисање појединачног садржаја

Опис

Брисање изабраног садржаја.

Учесници

- Администратор садржаја
- Представник извора ресурса

Предуслови

Администратор садржаја или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Садржај је обрисан.

Основни сценарио

1. Систем омогућује избор извора ресурса. Администратори извора ресурса могу да бришу садржаје свих извора, док представници извора ресурса могу да бришу само садржаје извора које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира жељени извор ресурса из листе понуђених.
3. Систем приказује листу садржаја изабраног извора.
4. Корисник бира операцију брисања жељеног садржаја.
5. Систем захтева потврђивање брисања изабраног садржаја.
6. Корисник потврђује брисање изабраног садржаја.
7. Систем брише изабрани садржај.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од брисања садржаја, у кораку потврђивања (5). Систем поново покреће основни сценарио.

3.2.4.6 Ажурирање података о садржају

Опис

Ажурирање података о изабраном садржају.

Учесници

- Администратор садржаја
- Представник извора ресурса

Предуслови

Администратор садржаја или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о изабраном садржају су промењени.

Основни сценарио

1. Систем омогућује избор извора ресурса. Администратори извора ресурса могу да мењају податке о садржајима свих извора, док представници извора ресурса то могу чинити само за садржаје извора које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира жељени извор ресурса из листе понуђених.
3. Систем приказује листу садржаја изабраног извора.
4. Корисник бира операцију промене жељеног садржаја.
5. Отвара се формулар са подацима о изабраном садржају и корисник врши жељене измене.
6. Систем ажурира информације о изабраном садржају.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од промене садржаја у току измене података. Систем поново покреће основни сценарио.
- Недостатак података: Корисник није унео све неопходне информације о садржају. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података: Већ постоји неки други садржај изабраног извора са наведеним локалним идентификатором. Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.4.7 Одређивање дигиталних репрезентација садржаја

Опис

Одређивање дигиталних ресурса који презентују изабрани садржај.

Учесници

- Администратор садржаја
- Представник извора ресурса

Предуслови

Администратор садржаја или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Одређено је који дигитални ресурси представљају изабрани садржај.

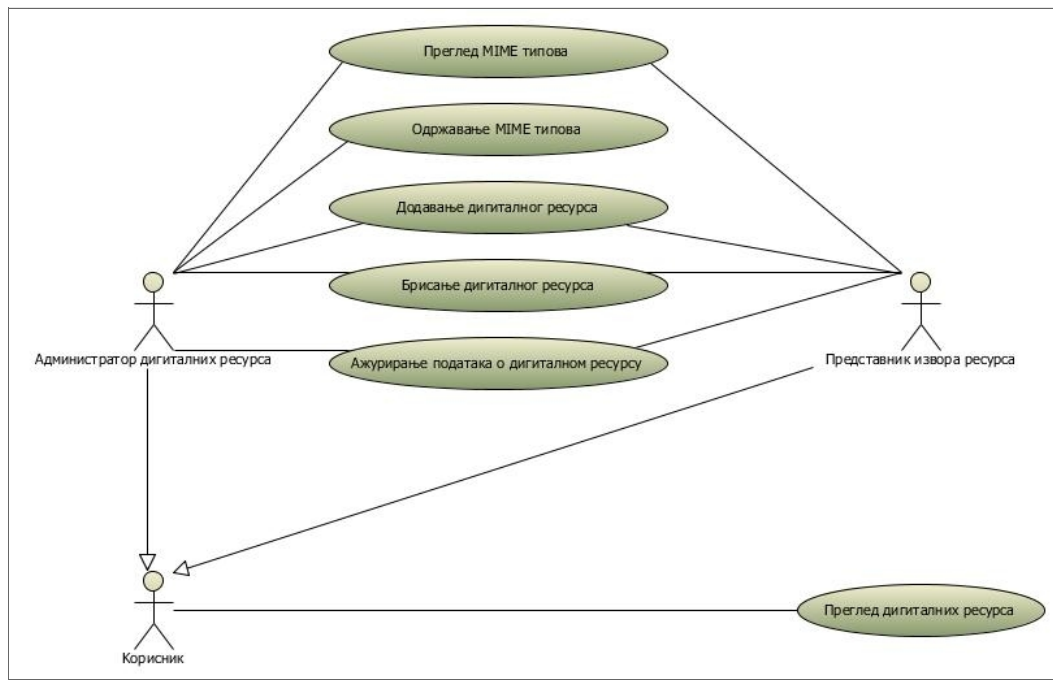
Основни сценарио

1. Систем омогућује избор извора ресурса.
2. Корисник бира жељени извор ресурса из листе понуђених.
3. Систем приказује листу садржаја изабраног извора ресурса, као и њихов укупан број. За сваки садржај, приказан је његов назив, врста и основни подаци у зависности од врсте садржаја.
4. Корисник бира садржај за који жели да одреди дигиталне репрезентације.
5. Систем приказује детаљније податке о садржају, као и листу дигиталних репрезентација истог. За сваку дигиталну репрезентацију садржаја, наведена је локација на којој се она заправо налази (веб адреса), тип фајла и величина.
6. Додавање и брисање дигиталних репрезентација садржаја:
 - а) Корисник бира операцију додавања новог дигиталног ресурса који ће презентовати изабрани садржај. Систем омогућује унос података о новом дигиталном ресурсу (видети случај коришћења 3.2.5.4. Додавање дигиталног ресурса). Корисник уноси податке о новом дигиталном ресурсу. Систем генерише интерни идентификатор дигиталног ресурса, додељује му га и евидентира податке о новом дигиталном ресурсу. Придружује нови дигитални ресурс изабраном садржају.
 - а) Корисник бира операцију додавања већ евидентираног дигиталног ресурса као презентације изабраног садржаја. Ради проналажења жељеног дигиталног ресурса, систем омогућује избор извора ресурса, а затим збирке и садржаја тог извора. На основу изабраног извора, збирке и садржаја, систем приказује листу одговарајућих ресурса. Није могуће изабрати дигитални ресурс који већ презентује изабрани садржај. Корисник бира жељени дигитални ресурс из листе понуђених. Систем евидентира да изабрани дигитални ресурс презентује изабрани садржај.
 - а) Корисник бира операцију брисања дигиталне репрезентације садржаја. Систем одваја изабрану дигиталну репрезентацију од садржаја.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од дефинисања дигиталних репрезентација садржаја, током корака 6. Систем поново покреће основни сценарио.

3.2.5 Управљање дигиталним ресурсима



Слика 13: Дијаграм случајева коришћења – Управљање дигиталним ресурсима

3.2.5.1 Одржавање *MIME* типова

Опис

Одржавање допуштених *MIME* типова дигиталних ресурса.

Учесници

- Администратор дигиталних ресурса

Предуслови

Администратор дигиталних ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Евидентирани су подаци о допуштеним *MIME* типовима.

Основни сценарио

- Систем приказује листу регистрованих *MIME* типова^{*}. За сваки *MIME* тип приказани су *MIME* тип садржаја, тип фајла и опис. За оне који не постоје у систему, постоји опција њиховог додавања, док је постојећи

^{*} За регистрацију *MIME* типова задужена је организација IANA (енгл. *Internet Assigned Numbers Authority*) [IANA]

MIME тип могуће обрисати из система само ако не постоји ниједан дигитални ресурс тог типа.

2. Додавање и брисање *MIME* типова:
 - а) Корисник бира опцију додавања жељеног *MIME* типа. Систем генерише интерни идентификатор, додељује га *MIME* типу и евидентира податке о њему.
 - б) Корисник бира опцију брисања жељеног *MIME* типа. Систем брише изабрани *MIME* тип.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од одређивања допуштених *MIME* типова.

3.2.5.2 Преглед *MIME* типова

Опис

Преглед података о допуштеним *MIME* типовима.

Учесници

- Администратор дигиталних ресурса
- Представник извора ресурса

Предуслови

Администратор дигиталних ресурса или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Приказани су основни подаци о допуштеним *MIME* типовима.

Основни сценарио

1. Систем приказује листу свих допуштених *MIME* типова. За сваки *MIME* тип, приказују се *MIME* тип садржаја, тип фајла и опис. Систем нуди избор уређивања листе по типу садржаја и по типу фајла. *MIME* типови су подразумевано уређени по типу садржаја, а они са истим типом садржаја, по типу фајла. Ако корисник промени начин уређивања, систем приказује листу уређену по изабраном критеријуму.

Алтернативни сценарио

Не постоји.

3.2.5.3 Преглед дигиталних ресурса

Опис

Преглед података о дигиталним ресурсима.

Учесници

- Било који корисник

Предуслови

Не постоје.

Постуслови

Приказани су основни подаци о дигиталним ресурсима.

Основни сценарио

1. Систем омогућује избор извора ресурса, збирке и садржаја.
2. Корисник бира жељени извор ресурса, збирку и садржај из листе понуђених.
3. Систем приказује листу дигиталних ресурса који задовољавају изабране критеријуме. За сваки дигитални ресурс, приказана је веб адреса односно локација на којој се он налази, тип фајла и величина. У зависности од права корисника пријављеног на систем, постоје опције за додавање, измену и брисање ресурса.
4. Корисник бира детаљнији преглед информација за жељени дигитални ресурс.
5. Систем приказује све информације о изабраном дигиталном ресурсу и о садржајима на које се односи.

Алтернативни сценарио

Не постоји.

3.2.5.4 Додавање дигиталног ресурса

Опис

Додавање новог дигиталног ресурса.

Учесници

- Администратор дигиталних ресурса
- Представник извора ресурса

Предуслови

Администратор дигиталних ресурса или представник извора ресурса је

пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о новом дигиталном ресурсу су евидентирани.

Основни сценарио

1. Систем омогућује избор извора ресурса, збирке и садржаја. Администратори дигиталних ресурса могу да додају нове дигиталне ресурсе свих извора, док представници извора ресурса то могу чинити само за изворе које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира жељени извор ресурса, збирку и садржај из листе понуђених.
3. Систем приказује листу дигиталних ресурса који задовољавају изабране критеријуме.
4. Корисник бира операцију додавања новог дигиталног ресурса
5. Отвара се формулар и корисник уноси податке о новом дигиталном ресурсу:
 - извор ресурса (претходно одабран, па се само приказује)
 - локални идентификатор (идентификатор дигиталног ресурса на страни извора ресурса)
 - збирка дигиталних ресурса којој припада (уколико је претходно одабрана, само се приказује, у супротном дигитални ресурс не припада ниједној збирци ресурса)
 - садржај који презентује (претходно одабран, па се само приказује)
 - *MIME* тип
 - величина
 - веб адреса односно локација на којој се заправо налази
 - назив
 - опис
 - информација о могућностима употребе (јаван или не)
 - кључне речи

Напомена: Један дигитални ресурс може да презентује не само један већ и више садржаја (видети случај коришћења 3.2.4.7. Одређивање дигиталних репрезентација садржаја).
6. Систем генерише интерни идентификатор дигиталног ресурса, додељује му га и евидентира податке о новом дигиталном ресурсу.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од додавања новог дигиталног ресурса у току уноса података. Систем поново покреће основни сценарио.
- Недостатак података: Корисник није унео све неопходне информације о дигиталном ресурсу. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података: Већ постоји дигитални ресурс изабраног извора са наведеним локалним идентификатором. Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи податке унете у формулар.

3.2.5.5 Брисање дигиталног ресурса

Опис

Брисање изабраног дигиталног ресурса.

Учесници

- Администратор дигиталних ресурса
- Представник извора ресурса

Предуслови

Администратор дигиталних ресурса или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Дигитални ресурс је обрисан.

Основни сценарио

1. Систем омогућује избор извора ресурса, збирке и садржаја. Администратори дигиталних ресурса могу да бришу дигиталне ресурсе свих извора, док представници извора ресурса могу да бришу само дигиталне ресурсе извора које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира жељени извор ресурса, збирку и садржај из листе понуђених.
3. Систем приказује листу дигиталних ресурса који задовољавају изабране критеријуме.
4. Корисник бира операцију брисања жељеног дигиталног ресурса.

5. Систем захтева потврђивање брисања изабраног дигиталног ресурса. Уколико дигитални ресурс презентује више од једног садржаја, систем приказује упозорење да ће брисањем изабраног дигиталног ресурса, он бити одвојен од свих садржаја на које се односи.
6. Корисник потврђује брисање изабраног дигиталног ресурса.
7. Систем брише изабрани дигитални ресурс и одваја га од свих садржаја са којима је повезан.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од брисања дигиталног ресурса, у кораку потврђивања (5). Систем поново покреће основни сценарио.

3.2.5.6 Ажурирање података о дигиталном ресурсу

Опис

Ажурирање података о изабраном дигиталном ресурсу.

Учесници

- Администратор дигиталних ресурса
- Представник извора ресурса

Предуслови

Администратор дигиталних ресурса или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о изабраном дигиталном ресурсу су промењени.

Основни сценарио

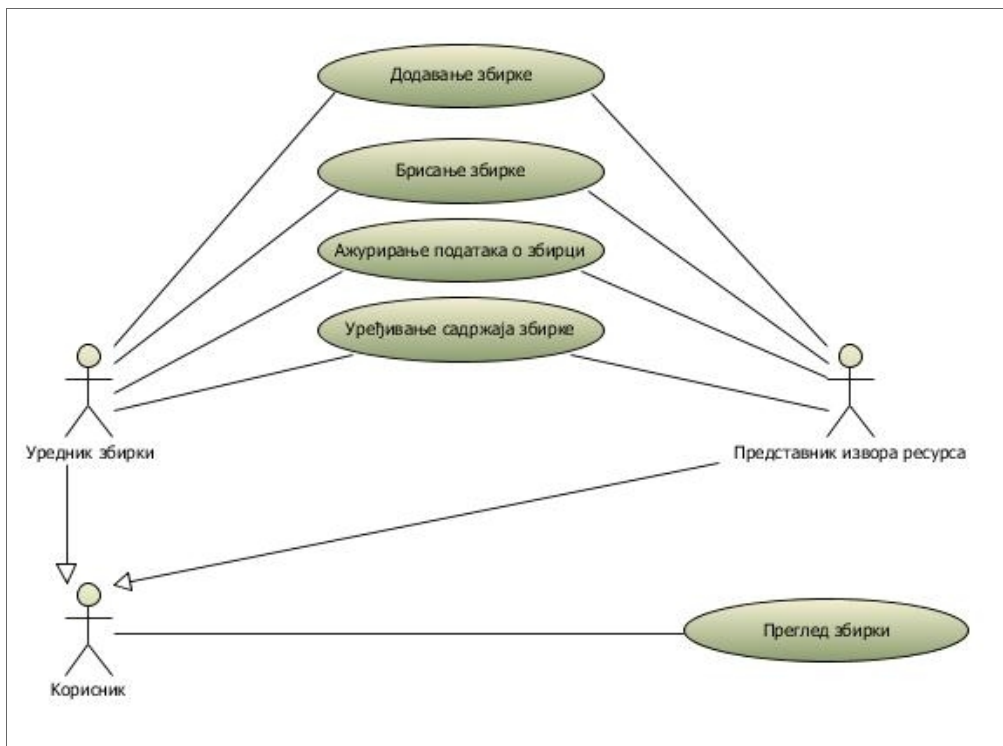
1. Систем омогућује избор извора ресурса, збирке и садржаја. Администратори дигиталних ресурса могу да мењају податке о дигиталним ресурсима свих извора, док представници извора ресурса могу да мењају само дигиталне ресурсе извора које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира жељени извор ресурса, збирку и садржај из листе понуђених.
3. Систем приказује листу постојећих дигиталних ресурса који задовољавају изабране критеријуме.
4. Корисник бира операцију промене жељеног дигиталног ресурса.

5. Отвара се формулар са подацима о изабраном дигиталном ресурсу и корисник врши жељене измене.
6. Систем ажурира информације о изабраном дигиталном ресурсу.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од промене дигиталног ресурса у току измене података. Систем поново покреће основни сценарио.
- Недостатак података: Корисник није унео све неопходне информације о дигиталном ресурсу. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података: Већ постоји неки други дигитални ресурс изабраног извора са наведеним локалним идентификатором. Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.6 Управљање збиркама ресурса



Слика 14: Дијаграм случајева коришћења – Управљање збиркама диг. ресурса

3.2.6.1 Преглед збирки

Опис

Корисник прегледа податке о збиркама дигиталних ресурса.

Учесници

Било који корисник

Предуслови

Не постоје.

Постуслови

Приказани су подаци о збиркама дигиталних ресурса.

Основни сценарио

1. Систем омогућује избор извора ресурса.
2. Корисник бира жељени извор ресурса из листе понуђених.
3. Систем приказује листу већ дефинисаних збирки изабраног извора, укључујући и збирке дигиталних ресурса из више извора, где је један од њих и изабрани извор ресурса. За сваку збирку се приказују њен назив, опис и број дигиталних ресурса које садржи. У случају да збирка садржи и дигиталне ресурсе других извора, то је назначено. У зависности од права корисника пријављеног на систем, постоје опције за додавање, измену и брисање збирки.
4. Корисник бира збирку за коју жели да добије детаљније информације.
5. За изабрану збирку, систем приказује њен назив, опис, назив извора ресурса коме припада и листу дигиталних ресурса које садржи. У случају да збирка садржи дигиталне ресурсе из више различитих извора, то је назначено. За сваки дигитални ресурс, наведена је локација на којој се он заправо налази (веб адреса), тип фајла и величина.

Алтернативни сценарио

Не постоји.

3.2.6.2 Додавање збирке

Опис

Додавање нове збирке дигиталних ресурса.

Учесници

- Уредник збирки
- Представник извора ресурса

Предуслови

Уредник збирки или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о новој збирци дигиталних ресурса су евидентирани.

Основни сценарио

1. Систем омогућује избор извора ресурса. Уредници збирки могу да додају нове збирке свих извора ресурса, док представници извора ресурса то могу чинити само за изворе које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира жељени извор ресурса из листе понуђених.
3. Систем приказује листу већ дефинисаних збирки изабраног извора, укључујући и збирке дигиталних ресурса из више извора, где је један од њих и изабрани извор ресурса.
4. Корисник бира опцију додавања нове збирке дигиталних ресурса.
5. Отвара се формулар и корисник уноси податке о новој зирци дигиталних ресурса:
 - извор ресурса (иницијално је изабран извор ресурса који је корисник претходно одабрао, али могуће је променити извор на неки други из листе извора за које корисник има одговарајуће привилегије)
 - локални идентификатор (идентификатор збирке дигиталних ресурса на страни извора ресурса)
 - назив
 - описУ случају да нису изабрани извор ресурса и локални идентификатор збирке, подразумева се да збирка може садржати дигиталне ресурсе из више различитих извора.
6. Систем генерише интерни идентификатор збирке, додељује јој га и евидентира податке о новој збирци дигиталних ресурса.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од додавања нове збирке у току уноса података. Систем поново покреће основни сценарио.
- Недостатак података: Корисник није унео све неопходне информације о зирци дигиталних ресурса. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да

настави са попуњавањем формулара са подацима.

- Неисправност унетих података: Већ постоји збирка изабраног извора ресурса са наведеним локалним идентификатором. Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.6.3 Брисање збирке

Опис

Брисање изабране збирке дигиталних ресурса.

Учесници

- Уредник збирки
- Представник извора ресурса

Предуслови

Уредник збирки или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Збирка је обрисана.

Основни сценарио

1. Систем омогућује избор извора ресурса. Уредници збирки могу да бришу збирке свих извора ресурса, док представници извора ресурса могу да бришу само збирке дигиталних ресурса из извора које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира жељени извор ресурса из листе понуђених.
3. Систем приказује листу свих збирки изабраног извора ресурса, укључујући и збирке дигиталних ресурса из више извора, где је један од њих и изабрани извор ресурса.
4. Корисник бира операцију брисања жељене збирке дигиталних ресурса.
5. Систем захтева потврђивање брисања изабране збирке.
6. Корисник потврђује брисање изабране збирке.
7. Систем брише изабрану збирку дигиталних ресурса.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од брисања збирке дигиталних ресурса, у кораку потврђивања (5). Систем поново покреће основни сценарио.

3.2.6.4 Ажурирање података о збирци

Опис

Ажурирање података о изабраној збирци дигиталних ресурса.

Учесници

- Уредник збирки
- Представник извора ресурса

Предуслови

Уредник збирки или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци о изабраној збирци ресурса су промењени.

Основни сценарио

1. Систем омогућује избор извора ресурса. Уредници збирки могу да мењају податке о збиркама свих извора ресурса, док представници извора ресурса могу да мењају само збирке дигиталних ресурса из извора које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира жељени извор ресурса из листе понуђених.
3. Систем приказује листу већ дефинисаних збирки изабраног извора, укључујући и збирке дигиталних ресурса из више извора, где је један од њих и изабрани извор ресурса.
4. Корисник бира операцију промене жељене збирке дигиталних ресурса.
5. Отвара се формулар са подацима о изабраној збирци и корисник врши жељене измене.
6. Систем ажурира информације о изабраној збирци.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од промене збирке дигиталних ресурса у току измене података. Систем поново покреће основни сценарио.
- Недостатак података: Корисник није унео све неопходне информације о збирци дигиталних ресурса. Систем приказује одговарајућу поруку за сваки обавезан податак који недостаје и омогућује кориснику да настави са попуњавањем формулара са подацима.
- Неисправност унетих података: Већ постоји нека друга збирка

изабраног извора са наведеним локалним идентификатором. Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи податке које је унео у формулар.

3.2.6.5 Уређивање садржаја збирке

Опис

Уређивање садржаја збирке дигиталних ресурса.

Учесници

- Уредник збирки
- Представник извора ресурса

Предуслови

Уредник збирки или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Одређен је садржај збирке дигиталних ресурса.

Основни сценарио

1. Систем омогућује избор извора ресурса. Уредници збирки могу да мењају садржај збирки свих извора ресурса, док представници извора ресурса могу да уређују само збирке дигиталних ресурса из извора које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира жељени извор ресурса из листе понуђених.
3. Систем приказује листу свих збирки изабраног извора ресурса, укључујући и збирке дигиталних ресурса из више извора, где је један од њих и изабрани извор ресурса.
4. Корисник бира детаљнији преглед података о жељеној збирци дигиталних ресурса.
5. Систем приказује назив и опис збирке, назив извора ресурса коме она припада и листу дигиталних ресурса које садржи. У случају да збирка садржи дигиталне ресурсе више различитих извора, то је назначено.
6. Додавање и брисање дигиталног ресурса из збирке:
 - а) Корисник бира операцију додавања неког дигиталног ресурса у збирку. Систем омогућује избор извора ресурса, збирке и садржаја на који се дигитални ресурс односи, а затим и избор дигиталног ресурса који ће бити додат у збирку. Корисник бира дигитални ресурс, уписује његов редни број у збирци и коментар по потреби.

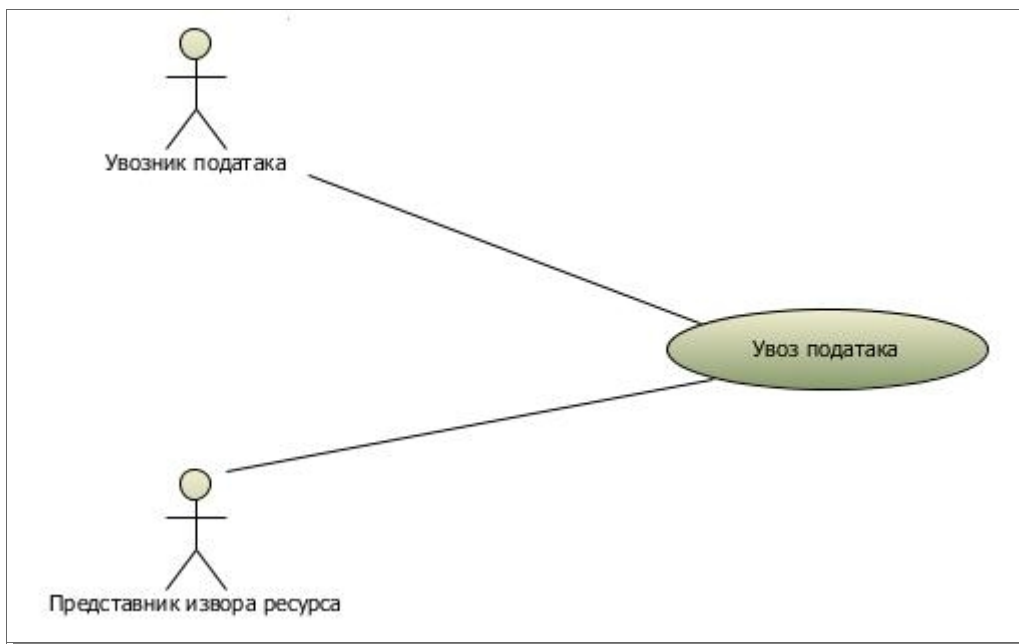
Систем додаје избрани дигитални ресурс у збирку.

- б) Корисник бира операцију брисања жељеног дигиталног ресурса из збирке. Систем уклања дигитални ресурс из збирке.

Алтернативни сценарио

- Одустајање операције: Корисник одлучује да одустане од уређивања садржаја збирке дигиталних ресурса, током корака (6). Систем поново покреће основни сценарио.

3.2.7 Увоз података



Слика 15: Дијаграм случајева коришћења – Увоз података

Опис

Унос и измена већег броја података неког извора из дате *XML* датотеке.

Учесници

- Увозник података
- Представник извора ресурса

Предуслови

Увозник података или представник извора ресурса је пријављен на систем (видети случај коришћења 3.2.1.1. Пријављивање на систем).

Постуслови

Подаци су увезени.

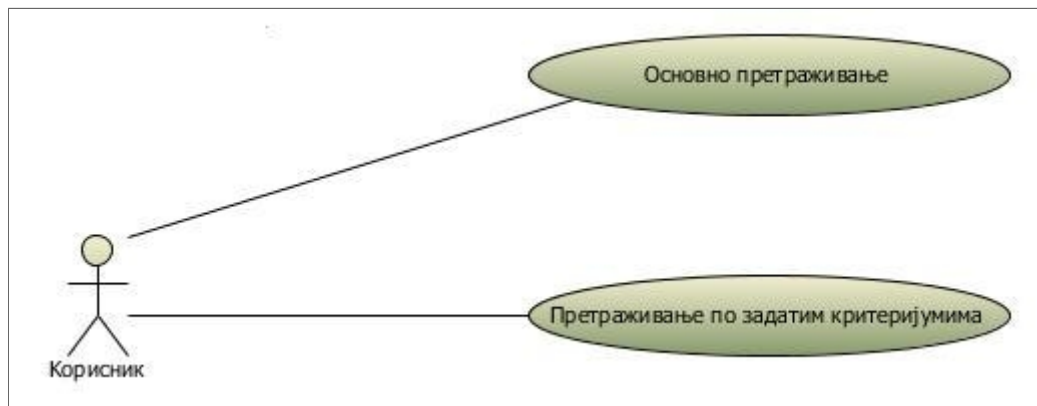
Основни сценарио

1. Систем омогућује избор извора ресурса, датотеке са подацима и опције брисања података који нису наведени у датотеци. На основу локалних идентификатора збирки, садржаја и дигиталних ресурса (идентификатора од стране извора), систем може да препозна да ли су у питању нови или већ евидентирани подаци и у складу са тиме, изврши операцију додавања или ажурирања. Уколико се изабере опција брисања, систем ће из регистра обрисати све збирке, садржаје и дигиталне ресурсе извора који нису наведени у датотеци. Увозници података могу да увозе податке за све изворе ресурса, било појединачно за сваки извор или за све изворе одједном (при избору извора ресурса, постоји и опција „Сви“ за увоз података свих извора одједном). Представници извора ресурса могу увозити податке само за изворе које представљају, појединачно за сваки извор или истовремено за све изворе које представљају (видети случај коришћења 3.2.2.12. Дефинисање улога за корисника).
2. Корисник бира извор ресурса из листе понуђених, затим датотеку у формату *XML* са одговарајућим подацима и уколико је потребно, опцију брисања података који нису наведени у датотеци.
3. Систем проверава исправност *XML* документа на основу дефинисане *XML* схеме, парсира садржај документа и додаје, брише или ажурира податке у зависности од тога да ли је изабрана опција брисања и да ли подаци већ постоје у систему или не. Након евидентирања података, систем приказује поруку да је увоз успешно извршен и исписује број нових, број измењених, број обрисаних и укупан број увезених података.

Алтернативни сценарио

- Одустајање од операције: Корисник одлучује да одустане од увоза података. Систем поново покреће основни сценарио.
- Недостатак података: Корисник није унео све неопходне податке. Систем приказује одговарајућу поруку за податак који недостаје и омогућује кориснику да настави са уносом података.
- Неисправност документа: Корисник није изабрао документ у формату *XML* или јесте, али он не задовољава правила дефинисана *XML* схемом. Систем приказује одговарајућу поруку о грешци и омогућује кориснику да исправи унете податке.

3.2.8 Претраживање



Слика 16: Дијаграм случајева коришћења – Претраживање

3.2.8.1 Основно претраживање

Опис

Корисник претражује дигиталне ресурсе.

Учесници

- Било који корисник

Предуслови

Не постоје.

Постуслови

Приказани су резултати претраге.

Основни сценарио

1. Систем омогућује унос услова тражења.
2. Корисник уноси услов тражења.
3. Систем проналази и приказује листу свих дигиталних ресурса који задовољавају услов тражења. Приказује се и укупан број резултата претраге, а за сваки дигитални ресурс, локација на којој се он заправо налази (веб адреса), назив и тип садржаја који презентује, аутори и детаљније информације у зависности од врсте садржаја.

Алтернативни сценарио

Не постоји.

3.2.8.2 Претраживање по задатим критеријумима

Опис

Претраживање дигиталних ресурса по задатим критеријумима.

Учесници

- Било који корисник

Предуслови

Не постоје.

Постуслови

Приказује се листа дигиталних ресурса који задовољавају задате критеријуме претраге.

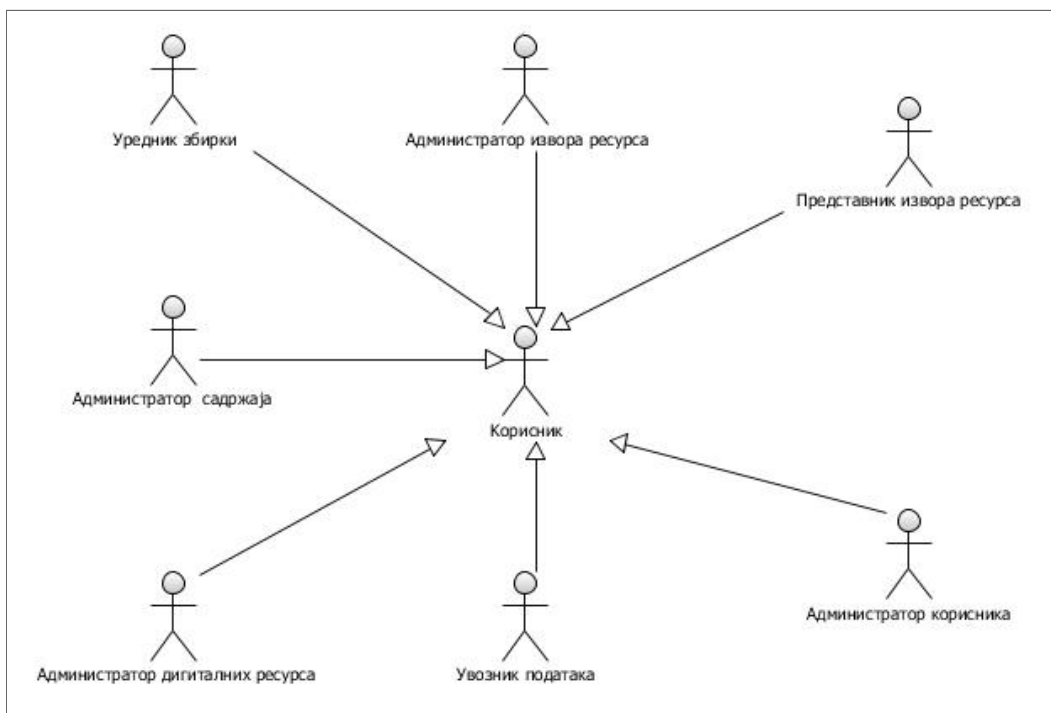
Основни сценарио

1. Систем омогућује задавање критеријума по којима се врши претрага:
 - извор ресурса
 - збирка
 - *MIME* тип ресурса
 - врста садржаја који ресурс презентује
 - назив
 - опис
 - аутор
 - детаљније информације у зависности од врсте садржаја
2. Корисник уноси жељене критеријуме.
3. Систем проналази и приказује листу свих дигиталних ресурса који задовољавају унете критеријуме, као и укупан број резултата претраге. За сваки дигитални ресурс, приказује се локација на којој се он заправо налази (веб адреса), назив и тип садржаја који презентује, аутори и детаљније информације у зависности од врсте садржаја.

Алтернативни сценарио

Не постоји.

3.3 Улоге



Слика 17: Улоге

У описаним случајевима коришћења појављује се неколико врста актера:

- **Корисник**
Прегледа и претражује податке о дигиталним ресурсима, садржајима, збиркама и изворима ресурса.
- **Представник извора ресурса**
Ажурира и увози податке о дигиталним ресурсима, садржајима и збиркама извора који представља.
- **Администратор извора ресурса**
Управља подацима о изворима ресурса.
- **Администратор садржаја**
Управља подацима о садржајима.
- **Администратор дигиталних ресурса**
Управља подацима о дигиталним ресурсима.
- **Уредник збирки**
Управља подацима о збиркама дигиталних ресурса. Групише дигиталне ресурсе по збиркама.

- **Увозник податка**

Увози податке о дигиталним ресурсима, садржајима, збиркама и изворима ресурса путем датотеке у формату *XML*.

- **Администратор корисника**

Управља подацима о корисницима и корисничким улогама.

Један корисник може имати права да обавља више различитих послова, тј. да учествује у раду система као више актера, па у подацима о корисницима постоји апстракција "улоге". Једна улога одговара једној врсти актера у описаним случајевима коришћења. Један ауторизовани корисник може имати једну или више улога. Неауторизован корисник има само улогу Корисник.

4 Пројектовање

4.1 Класе података

Анализом случајева коришћења, могу се уочити следеће класе односно скупови објеката са истим својствима (атрибутима) и понашањем (операцијама):

- **Дозвола** (класа *Permission*) представља овлашћење које се може доделити кориснику за обављање појединачне операције у систему. Описује се подацима:
 - идентификатор (атрибут *id*)
 - назив (атрибут *name*)
 - опис (атрибут *description*)
 - време првог уноса (атрибут *date_of_insert*)
 - време последње измене (атрибут *date_of_update*)

Свака дозвола је јединствено одређена својим идентификатором и од других дозвола се разликује својим називом. Опис садржи детаљније информације о дозволи. Време првог уноса и последње измене представљају записе тренутка прављења дозволе и тренутка њене последње измене.

- **Корисничка улога** (класа *Role*) је модел улоге која се може доделити корисницима система. Атрибути ове класе су:
 - идентификатор (атрибут *id*)
 - назив (атрибут *name*)
 - опис (атрибут *description*)
 - време првог уноса (атрибут *date_of_insert*)
 - време последње измене (атрибут *date_of_update*)

Улога представља функцију корисника у систему, односно описује врсту посла који он обавља. Свака корисничка улога је јединствено одређена својим идентификатором. Оно што је разликује од других, јесте и њен назив, док се детаљније информације о њој могу наћи у опису. Време првог уноса представља запис тренутка када је корисничка улога направљена, а време последње измене, тренутак када су последњи пут мењани њени подаци.

Једној корисничкој улози одговара скуп дозвола за појединачне

операције. Једну дозволу могу имати различите улоге. Да би се представио однос између корисничке улоге и дозволе, додаје се следећа класа:

- **Дозвола за улогу** (класа ***RolePermission***) са атрибутима:
 - идентификатор улоге (атрибут *role_id*)
 - идентификатор дозволе (атрибут *permission_id*)
 - време уноса (атрибут *date_of_insert*)
- **Корисник** (класа ***User***) представља модел корисника система. Атрибути ове класе су:
 - идентификатор (атрибут *id*)
 - корисничко име (атрибут *username*)
 - лозинка (атрибут *password*)
 - име (атрибут *first_name*)
 - презиме (атрибут *last_name*)
 - е-пошта (атрибут *email*)
 - датум рођења (атрибут *date_of_birth*)
 - пол (атрибут *gender*)
 - податак да ли је корисник активан или не (атрибут *active*)
 - коментар (атрибут *comment*)
 - време првог уноса (атрибут *date_of_insert*)
 - време последње измене (атрибут *date_of_update*)

Сваки корисник је јединствено одређен својим идентификатором. Описују га име, презиме, е-пошта, датум рођења и пол, а корисничко име и лозинка служе за његово представљање систему и ауторизацију. Корисничко име је јединствено за сваког корисника. Време првог уноса је време отварања корисничког налога за корисника, а време последње измене, тренутак последње измене његових података. Налог корисника је могуће деактивирати и поново активирати. Корисник се сматра активним ако је његов налог активиран. Деактивацијом налога, корисник постаје неактиван.

У случају потребе да корисник обавља више различитих послова, може му бити додељено више корисничких улога. Исту улогу може имати више корисника.

Поједине улоге корисника се везују за неки извор ресурса тј. омогућавају обављање послова који се односе само на одређени извор. На пример, ако корисник има улогу представника неког извора

ресурса, тада је потребно да постоји податак о томе о ком извору је реч. У случају да извор ресурса није наведен, подразумева се да корисник може обављати одређену врсту посла за све изворе. За описивање односа између корисника и корисничке улоге, додаје се класа:

- **Улога корисника** (класа *UserRole*) са атрибутима:

- идентификатор (атрибут *id*)
- идентификатор корисника (атрибут *user_id*)
- идентификатор улоге (атрибут *role_id*)
- време уноса (атрибут *date_of_insert*)

За описивање улога корисника које се односе само на одређени извор ресурса, додаје се и поткласа:

Улога корисника за извор ресурса (класа *ResourceSourceUserRole*) са додатним атрибутом:

- идентификатор извора ресурса (атрибут *resource_source_id*)

- **Језик** (класа *Language*) представља опис језика на којем може бити неки садржај. Својства језика су:

- идентификатор (атрибут *id*)
- ознака (атрибут *code*)
- назив (атрибут *name*)
- време првог уноса (атрибут *date_of_insert*)
- време последње измене (атрибут *date_of_update*)

Сваки језик је јединствено одређен својим идентификатором. Има своју ознаку по стандарду ISO 639-1 и назив. Време првог уноса представља запис тренутка када је језик додат у систем, а време последње измене, запис тренутка када су последњи пут мењани његови подаци.

- **MIME тип** (класа *MimeFormat*) је модел *MIME* типа дигиталних ресурса и представља се подацима:

- идентификатор (атрибут *id*)
- тип садржаја (атрибут *content_type*)
- ознака типа фајла (атрибут *file_extension*)
- опис (атрибут *description*)
- време првог уноса (атрибут *date_of_insert*)
- време последње измене (атрибут *date_of_update*)

Сваки *MIME* тип је јединствено одређен својим идентификатором. Описују га тип садржаја, ознака типа фајла и опис са детаљнијим објашњењем о типу. Постоје *MIME* типови са истим типом садржаја, али различитим типом фајла. Време првог уноса представља запис тренутка када је *MIME* тип додат у систем, а време последње измене, запис тренутка када су последњи пут мењани подаци који га описују.

- **Извор ресурса** (класа *ResourceSource*) представља извор информација, тј. података о дигиталним ресурсима. То је најчешће нека установа културе или образовна установа, али може бити и појединац. Описује се подацима:
 - идентификатор (атрибут *id*)
 - назив (атрибут *name*)
 - опис (атрибут *description*)
 - веб адреса (атрибут *uri*)
 - време првог уноса (атрибут *date_of_insert*)
 - време последње измене (атрибут *date_of_update*)

Извори ресурса су јединствено одређени својим идентификатором. Имају свој назив, детаљнији опис и адресу на вебу. Време првог уноса извора је време његовог додавања у систем, а време последње измене, тренутак последње измене неког од података који га описују.

Да би корисник управљао подацима неког извора, треба му доделити корисничку улогу представника тог извора ресурса.

Једној установи може одговарати један или више извора ресурса, у зависности од организације рада и потребних ауторизација.

- **Садржај** (класа *Content*) представља модел културног добра са подацима:
 - идентификатор (атрибут *id*)
 - идентификатор извора ресурса (атрибут *resource_source_id*)
 - локални идентификатор (атрибут *local_id*)
 - назив (атрибут *name*)
 - опис (атрибут *description*)
 - идентификатор језика (атрибут *language_id*)
 - време првог уноса (атрибут *date_of_insert*)
 - време последње измене (атрибут *date_of_update*)

Сваки садржај је јединствено одређен својим идентификатором. Идентификатор извора ресурса је идентификатор извора од којег

садржај потиче, а локални идентификатор је онај који тај садржај одређује међу локалним подацима тог извора. Садржај има свој назив, опис, време првог уноса у систем и време када су његови подаци последњи пут промењени. Сваки садржај је одређеног типа. У зависности од типа садржаја могу се издвојити поткласе:

Књига (класа *Book*) са додатним атрибутима:

- издавач (атрибут *publisher*)
- датум издавања (атрибут *pub_date*)
- издање (атрибут *edition*)
- ISBN (атрибут *isbn*)

Часопис (класа *Magazine*) са додатним атрибутима:

- издавач (атрибут *publisher*)
- датум издавања (атрибут *pub_date*)
- ISSN (атрибут *issn*)

Звук (класа *Audio*) са додатним атрибутима:

- дужина трајања (атрибут *duration*) у секундама

Видео (класа *Video*) са додатним атрибутима:

- дужина трајања (атрибут *duration*) у секундама

Слика (класа *Painting*) са додатним атрибутима:

- ширина (атрибут *width*) у милиметрима
- висина (атрибут *height*) у милиметрима
- техника (атрибут *technique*)

Фотографија (класа *Photo*) са додатним атрибутима:

- ширина (атрибут *width*) у милиметрима
- висина (атрибут *height*) у милиметрима

- **Аутор** (класа *Author*) представља модел аутора културних добара са атрибутима:

- идентификатор (атрибут *id*)
- идентификатор извора ресурса (атрибут *resource_source_id*)
- локални идентификатор (атрибут *local_id*)
- име (атрибут *first_name*)
- презиме (атрибут *last_name*)
- надимак (атрибут *nickname*)
- време првог уноса (атрибут *date_of_insert*)
- време последње измене (атрибут *date_of_update*)

Аутори су једнозначно одређени својим идентификатором. Имају име, презиме, надимак, податак о времену када су први пут унети у систем и време њихове последње измене. Описује их и идентификатор извора ресурса од којег потичу, као и локални идентификатор, тј. онај који их једнозначно одређује на страни извора.

Аутори могу бити ствараоци мноштва различитих садржаја. Један садржај може бити дело више аутора. За представљање односа између садржаја и аутора, додаје се класа:

- **Аутор садржаја** (класа *AuthorContent*) са атрибутима:
 - идентификатор аутора (атрибут *author_id*)
 - идентификатор садржаја (атрибут *content_id*)
 - редни број (атрибут *rank*)
 - улога (атрибут *role*)
 - време уноса (атрибут *date_of_insert*)

Однос између аутора и садржаја детаљније описују редни број и улога аутора при стварању садржаја.

- **Дигитални ресурс** (класа *DigitalResource*) је модел дигиталне репрезентације културног добра. Његова својства су:
 - идентификатор (атрибут *id*)
 - идентификатор извора ресурса (атрибут *resource_source_id*)
 - локални идентификатор (атрибут *local_id*)
 - адреса (атрибут *uri*)
 - идентификатор *MIME* типа (атрибут *mime_format_id*)
 - величина (атрибут *size*) у бајтовима
 - назив (атрибут *name*)
 - опис (атрибут *description*)
 - јавност (атрибут *public*)
 - кључне речи (атрибут *keywords*)
 - време првог уноса (атрибут *date_of_insert*)
 - време последње измене (атрибут *date_of_update*)

Дигитални ресурс је јединствено одређен својим идентификатором. Он припада једном извору међу чијим ресурсима га једнозначно одређује локални идентификатор. Сваки дигитални ресурс се налази на некој адреси, одређеног је *MIME* типа, а његова величина је пожељан, али не и обавезан податак. Има свој назив и по могућству

краћи опис. У зависности од политике извора, дигитални ресурс може, али и не мора бити јаван. Време првог уноса представља време када су подаци о дигиталном ресурсу први пут унети у регистар, а време последње измене, тренутак када су они последњи пут мењани.

- **Однос садржаја и његове дигиталне репрезентације** (класа *ContentRepresentation*)
 - идентификатор садржаја (атрибут *content_id*)
 - идентификатор дигиталног ресурса (атрибут *digital_resource_id*)
 - време уноса (атрибут *date_of_insert*)

Садржаји могу имати више различитих дигиталних репрезентација. Један дигитални ресурс може представљати више различитих садржаја.

- **Збирка** (класа *Collection*) представља модел збирке дигиталних ресурса са подацима:
 - идентификатор (атрибут *id*)
 - идентификатор извора ресурса (атрибут *resource_source_id*)
 - локални идентификатор (атрибут *local_id*)
 - назив (атрибут *name*)
 - опис (атрибут *description*)
 - време првог уноса (атрибут *date_of_insert*)
 - време последње измене (атрибут *date_of_update*)

Ради једноставнијег претраживања и употребе, дигиталне ресурсе је могуће груписати по збиркама. Свака збирка је једнозначно одређена својим идентификатором. Идентификатор извора ресурса и локални идентификатор су опциони атрибути. У случају да збирка садржи дигиталне ресурсе само једног извора, онда постоје подаци о идентификатору тог извора, као и идентификатор који одређује ту збирку међу локалним подацима извора, тј. локални идентификатор. У супротном, уколико збирка садржи ресурсе више различитих извора, ти подаци се изостављају. Збирка се одликује и својим називом, описом, временом када је направљена и временом последње измене њених података.

Збирка може да садржи више различитих дигиталних ресурса. Један дигитални ресурс може да припада различитим збиркама. Да би се представио однос између збирке и ресурса, додаје се следећа класа:

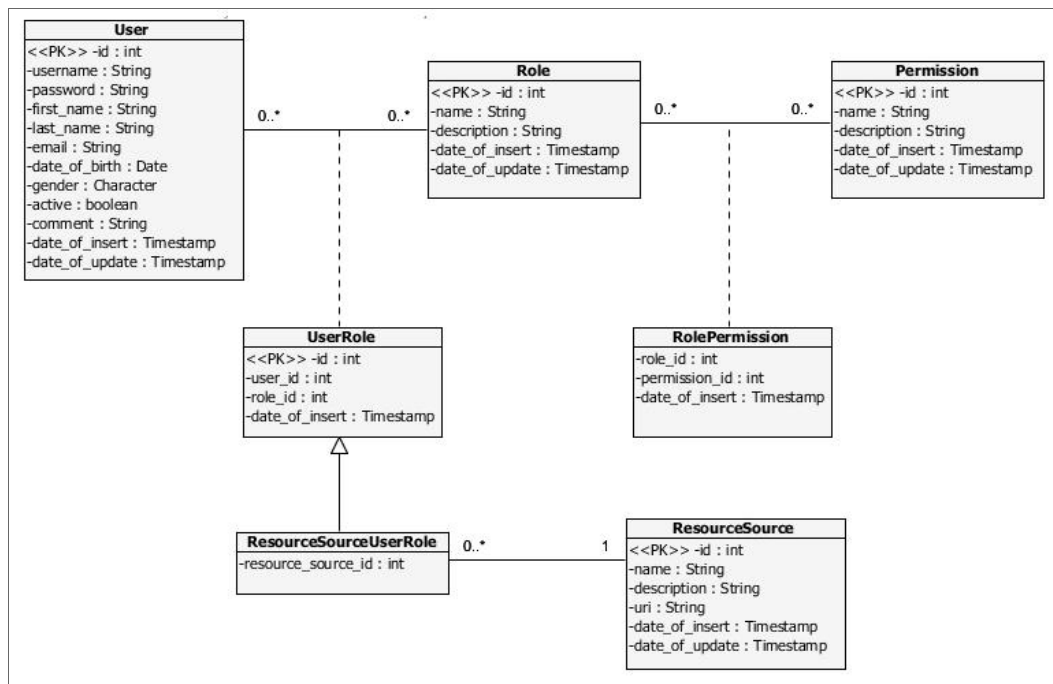
- **Члан збирке дигиталних ресурса** (класа *CollectionMember*) са атрибутима:
 - идентификатор збирке (атрибут *collection_id*)
 - идентификатор дигиталног ресурса (атрибут *digital_resource_id*)
 - редни број (атрибут *sequence_number*)
 - коментар (атрибут *comment*)
 - време уноса (атрибут *date_of_insert*)

Сваки дигитални ресурс има свој редни број у збирци, а специфичности у односу збирке и ресурса се уписују у коментар.

4.2 Моделовање класа података дијаграмима класа

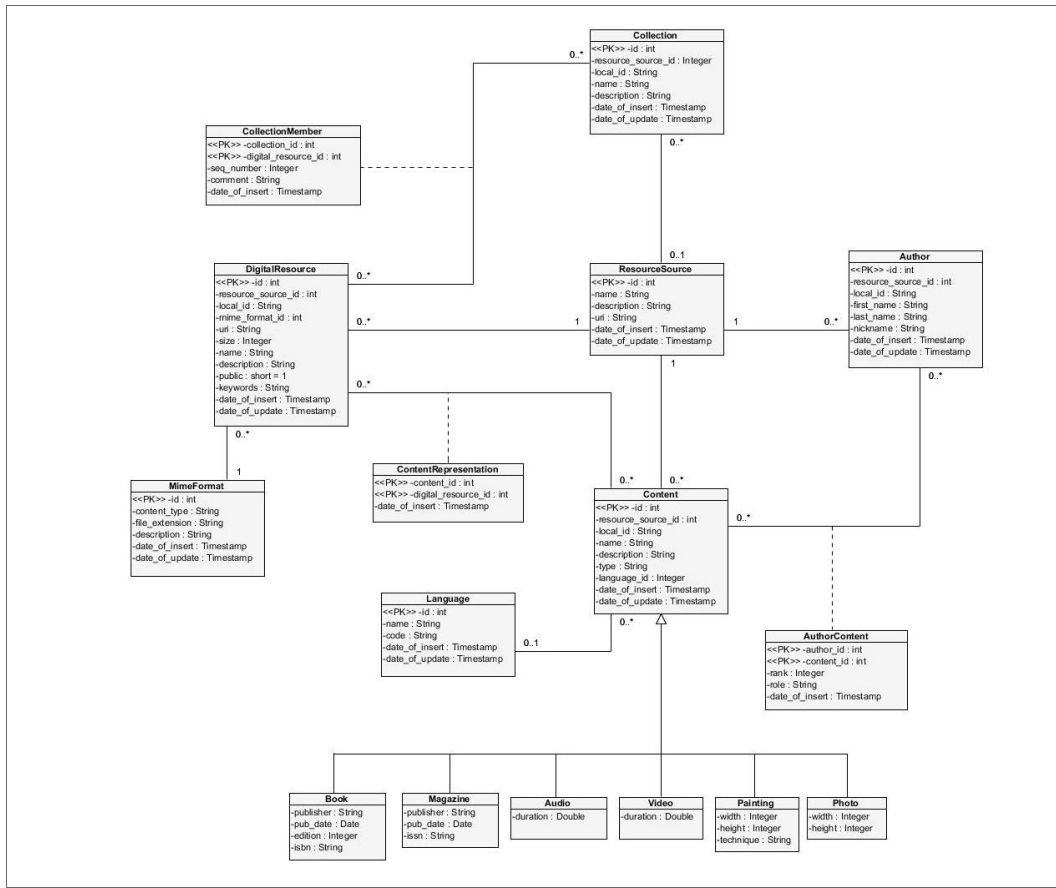
Класе података које су уочене анализом случајева коришћења се моделују и представљају дијаграмима класа.

На слици 18 су представљене класе које описују изворе ресурса, кориснике, корисничке улоге, дозволе и основне односе између њих. Класа *UserRole* описује улогу корисника. Њена поткласа *ResourceSourceUserRole* описује улогу корисника која важи само за одређени извор ресурса. Поред атрибута, метода и односа наслеђених од родитељске класе, класа *ResourceSourceUserRole* има и додатни атрибут, идентификатор извора ресурса (*resource_source_id*) који везује улогу корисника за одређени извор ресурса.



Слика 18: Дијаграм класа – Управљање корисницима

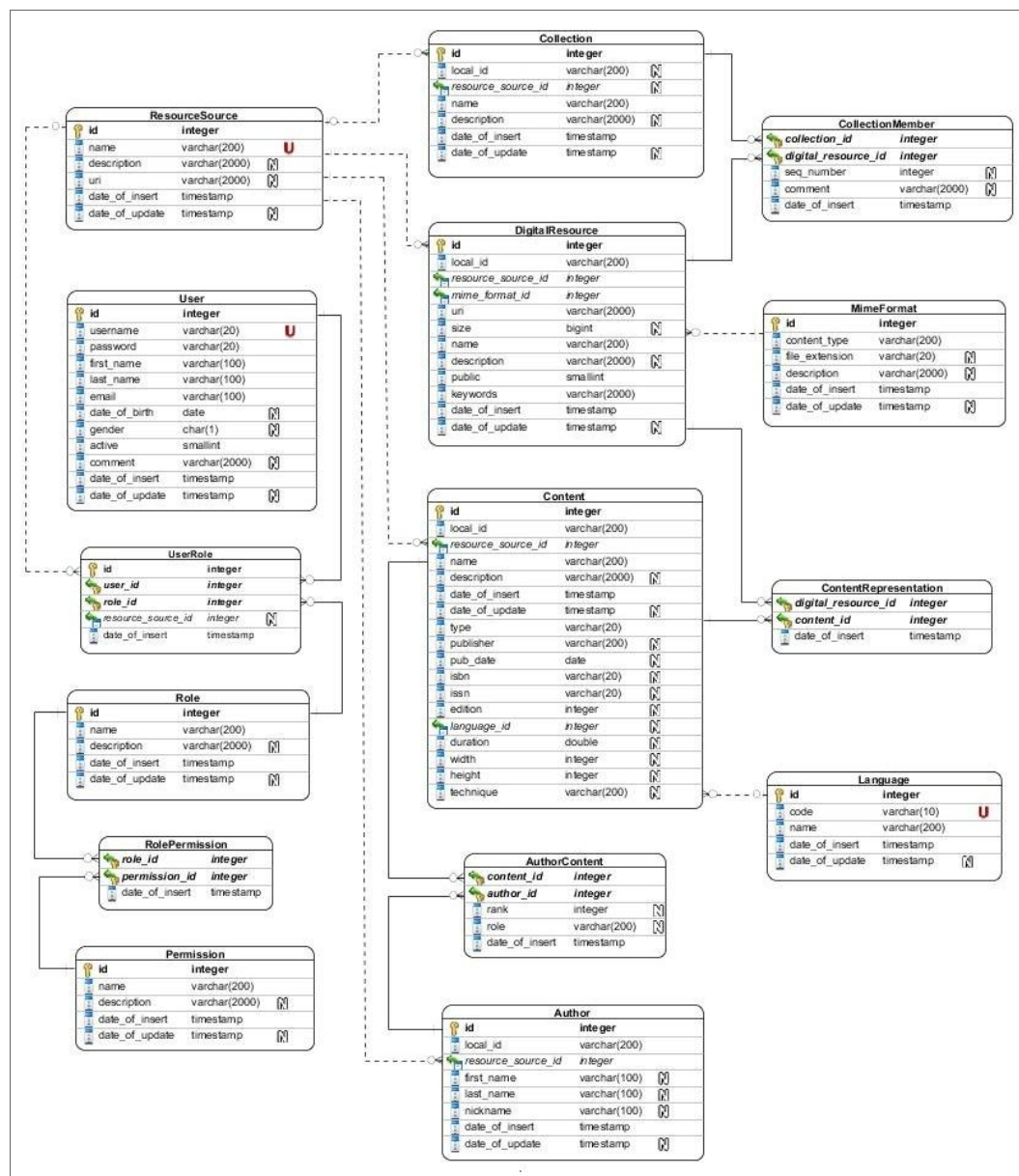
На слици 19 приказане су класе неопходне за дефинисање и управљање изворима ресурса, збиркама, садржајима, дигиталним ресурсима, ауторима и свим основним односима између њих. Класа *Content* је наткласа, а класе *Book*, *Magazine*, *Audio*, *Video*, *Painting* и *Photo* су њене поткласе. Оне од ње наслеђују атрибуте, методе и односе. На пример, без обзира на тип, сваки садржај има свој назив и опис, али само књига може имати *ISBN*, па класа *Book* има одговарајући додатни атрибут.



Слика 19: Дијаграм класа: Управљање ресурсима

4.3 База података

На основу издвојених ентитета, њихових атрибута и односа, одређена је структура релационе базе података. Структура базе података је представљена на слици 20.



Слика 20: Дијаграм ентитета и односа

При имплементацији, сваком ентитету приказаном на дијаграму, одговара по једна табела релационе базе података, а сваком примерку ентитета, ред у табели која је еквивалент том ентитету. Линије које повезују ентитете

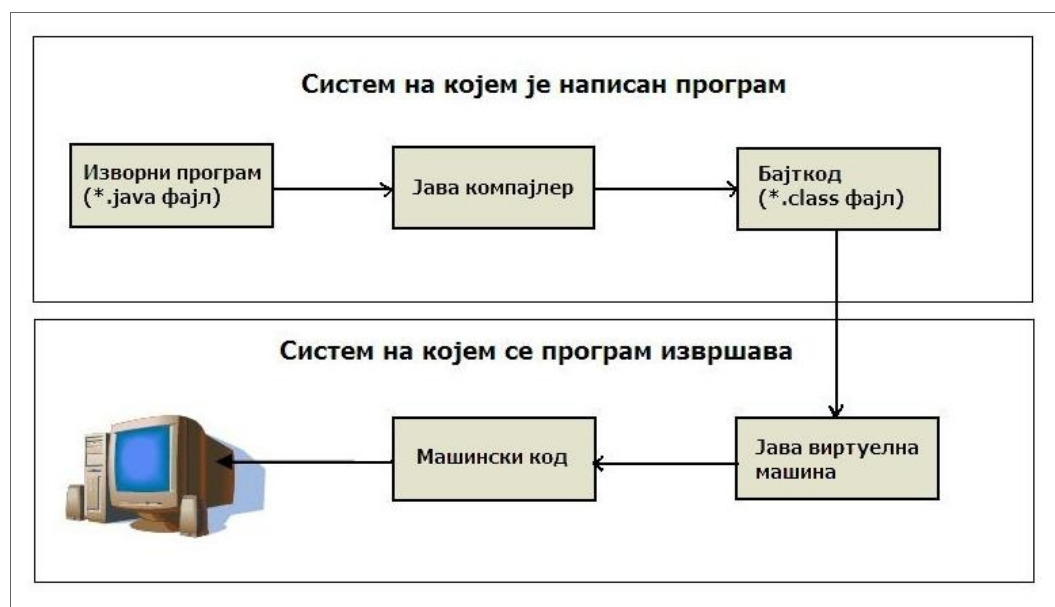
представљају односе између њих, односно стране кључеве.

У бази података су дефинисани и окидачи (енгл. *triggers*). При промени примерка сваког основног ентитета, активира се окидач који бележи време када је та измена извршена. Резултат тога је да за сваки примерак ентитета постоји податак о томе када је он последњи пут мењан (атрибут *date_of_update*). Такође, сваки примерак има и додатни атрибут који представља време када је он направљен (атрибут *date_of_insert*). Наредбе за прављење табела базе података и окидача су наведене у Додатку А.

5 Имплементација

5.1 Програмски језик

Прототип регистра је имплементиран на програмском језику Јава (енгл. *Java*) [*JAVA*]. Јава је објектно оријентисани програмски језик, који је развила фирма *Sun Microsystems* (данас у власништву фирме *Oracle*). Једна од битнијих одлика Јаве је преносивост, што омогућује потпуну независност од платформе на којој се извршава. Јава компајлер преводи јава изворни код у тзв. бајткод (енгл. *bytecode*) који даље интерпретира Јава виртуелна машина (енгл. *Java Virtual Machine, JVM*), пресликавајући Јава наредбе у наредбе конкретног рачунарског система.



Слика 21: Процес превођења програма

Поред Јава виртуелне машине, друга компонента Јава платформе је *Java API* (*Java Application Programming Interface*). *Java API* је стандардна библиотека програмског језика Јава. Чини је велики број класа груписаних по пакетима (енгл. *packages*).

Јава аутоматски ослобађа заузету меморију помоћу скупљача отпадака (енгл. *garbage collector*), поуздана је и подржава истовремено извршавање више нити (енгл. *threads*) програма.

5.2 Веб компоненте

Динамички аспекти веба су имплементирани применом Јава технологије *Java Server Pages (JSP)* [JSP]. *JSP* стране су текстуални документи који се састоје од статичких података у неком текстуалном формату (*HTML*, *XML*...) и од *JSP* елемената којима се представља динамички садржај. Делови Јава кода у оквиру *JSP* страна називају се скриптлети (енгл. *scriptlets*). Након измена, *JSP* стране се аутоматски компајлирају и поново учитавају на веб сервер. Да би се *JSP* странице покретале, потребно је извршавати их на веб серверу који подржава Јаву и *JSP*. При имплементацији прототипа централног регистра је коришћен *Apache Tomcat 6.0 Servlet/JSP container*, Јава *HTTP* веб сервер који развија организација *Apache Software Foundation*[TOM].

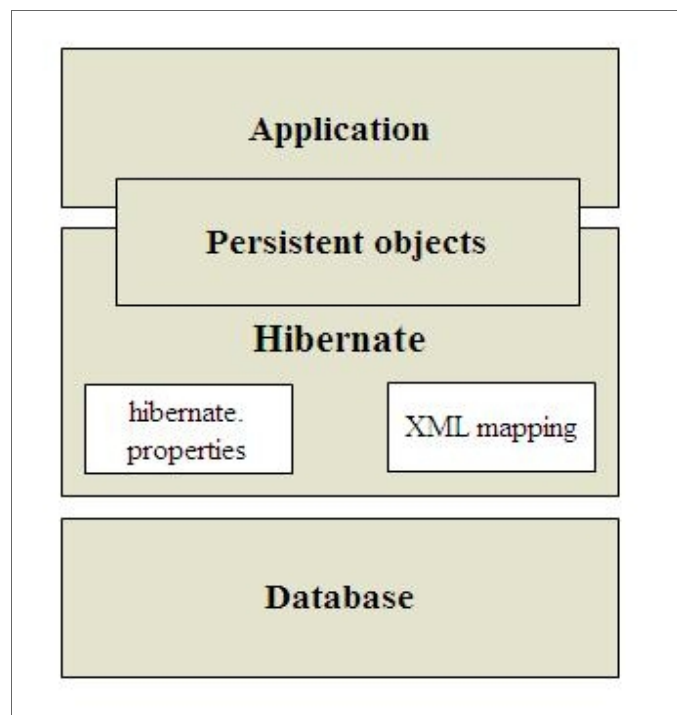
5.3 База података

База података је имплементирана на релационом систему за управљање базама података *IBM DB2* [DB2]. Релациони систем за управљање базама података (енгл. *relational database management system*, *RDBMS*) је скуп програма који омогућава поуздано и ефикасно прављење, измену и коришћење база података.

За омогућавање претраге текстуалних поља у бази података, на нивоу речи (енгл. *full text search*), коришћен је додатни алат *DB2 Net Search Extender* [NSE] (Детаљније у Поглављу 5.6.1).

5.4 Објектно-релационо пресликавање

Објектно-релационо пресликавање (енгл. *Object relational mapping*, *ORM*) је изведено применом технологије *Hibernate* [HIB]. Објектно-релационо пресликавање је техника пресликавања класа апликације у табеле релационе базе података. *Hibernate* је Јава алат отвореног кода (енгл. *open source*) који пресликава класе и типове података у релациону базу података на основу *XML* конфигурационих докумената или коришћењем специјалних ознака (енгл. *annotations*). Алат пружа могућност рада на нивоу Јава објеката, а подржава и класични приступ у којем се, помоћу упита на језику *SQL*, директно приступа објектима у релационој бази података. На слици 22 приказана је вишеслојна архитектура *Hibernate*-а.



Слика 22: Вишеслојна архитектура *Hibernate*-а

5.5 XML схема

XML је стандардизован језик који је једноставан за читање и човеку и рачунару [*XML*]. Дефинише општу синтаксу за означавање података помоћу одговарајућих етикета (енгл. *tags*) које имају разумљиво значење и описују садржај који се налази унутар њих. *XML* датотеке су текстуалне и независне од конкретног рачунарског или оперативног система. Једна од најважнијих особина *XML*-а је то што он представља метајезик, па нема фиксан скуп ознака и елемената већ се може прилагођавати и проширивати како би задовољио различите потребе корисника.

За проверу исправности *XML* докумената, користе се *XML* схеме. *XML* схема (или граматика) је опис, односно скуп правила структурирања *XML* докумената. Њеним коришћењем могуће је врло детаљно одредити опис садржаја (кардиналност, тип...). У случају да *XML* документ задовољава сва ограничења задата схемом, проглашава се исправним у односу на ту схему. Неки од језика за дефинисање схеме су *DTD* (*Document Type Definition*), *XML Schema* (*WXS* или *XSD*), *RELAX NG* (*Regular Language description for XML Next Generation*)[*DTD*, *XSD*, *RLX*].

Да би се подаци размењивали посредством *XML* докумената, пожељно је дефинисати *XML* схему. На тај начин се омогућава провера исправности

докумената пре њиховог парсирања и употребе. При увозу треба издвојене податке сместити у базу података. На основу локалних идентификатора ресурса датог извора, систем треба да препозна да ли је у питању нови или већ постојећи податак и у складу са тим изврши операцију додавања или ажурирања тог податка. За израду схеме коришћен је језик *XML Schema* познат и као *XSD (XML Schema Definition)*, препоручен од *W3C (енгл. World Wide Web Consortium)* [W3C].

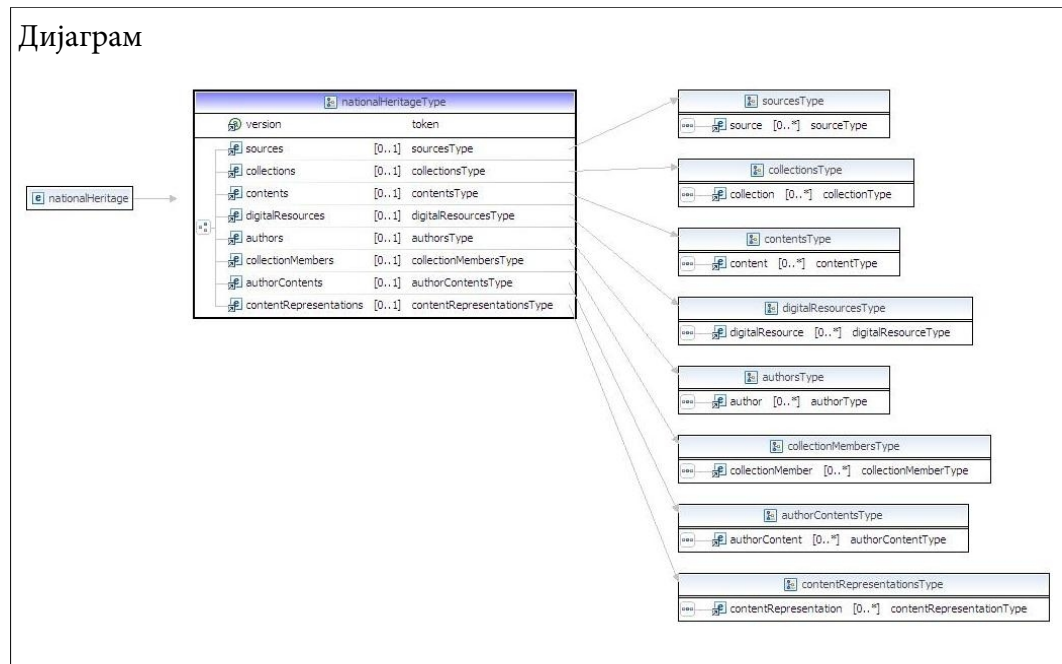
Увоз података путем *XML* докумената представља примаран вид попуњавања регистра, па су у даљем тексту детаљније описани елементи, атрибути и типови у *XML* схеми. За графичко представљање елемената *XML* схеме, коришћен је додатак за развојно окружење *Eclipse, XML Editors and tools 3.1.1* [ECL]. Целокупна *XML* схема се може наћи у Додатку Б, а пример *XML* документа који задовољава структуру схеме у Додатку В.

5.5.1 Елементи

Свакој од издвојених класа података одговара по један елемент или колекција елемената у *XML* схеми.

5.5.1.1 *nationalHeritage*

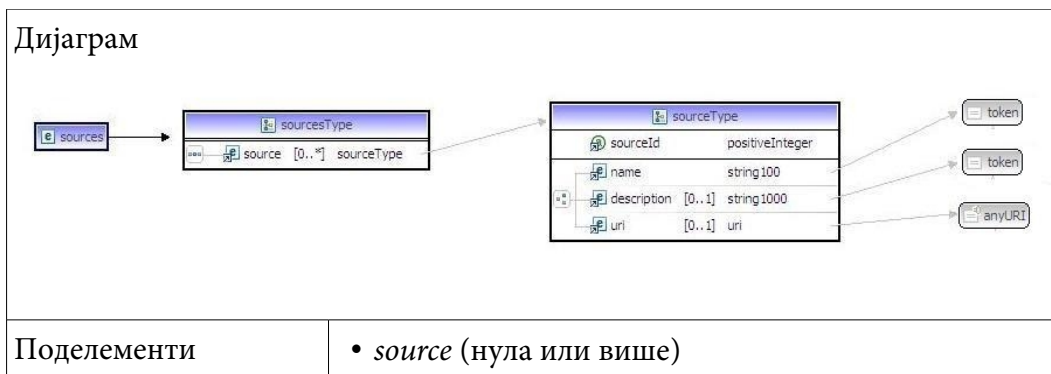
Корени елемент *nationalHeritage* је основни елемент у којем су угњеждени сви остали елементи и њихов садржај. Има један обавезан атрибут (*version*) чија вредност представља ознаку верзије *XML* схеме.



Поделементи (редослед није битан)	<ul style="list-style-type: none"> • <i>sources</i> (нула или један) • <i>collections</i> (нула или један) • <i>contents</i> (нула или један) • <i>authors</i> (нула или један) • <i>digitalResources</i> (нула или један) • <i>authorContents</i> (нула или један) • <i>contentRepresentations</i> (нула или један) • <i>collectionMembers</i> (нула или један) 	
Атрибути	<ul style="list-style-type: none"> • <i>version</i> (тачно један) 	<ul style="list-style-type: none"> • тип: <i>token</i>

5.5.1.2 *sources*

Елемент *sources* представља колекцију елемената *source*, који садрже податке о изворима ресурса.



5.5.1.3 *source*

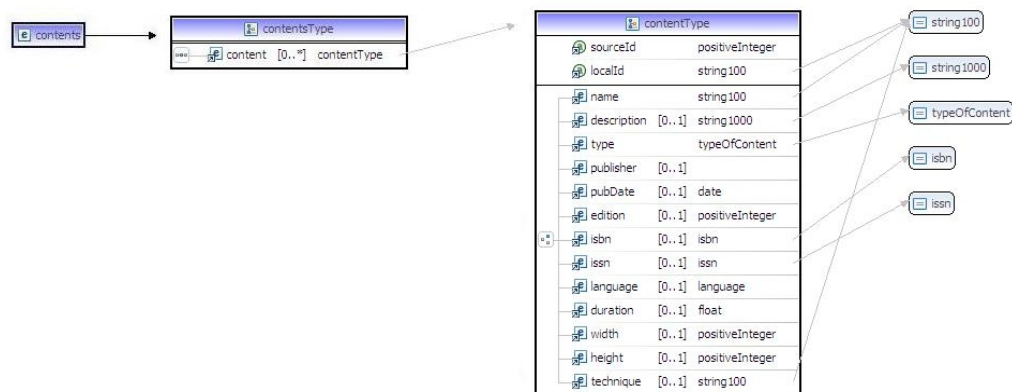
Елемент *source* садржи податке о неком извору ресурса. Редослед навођења података о извору није битан.

Поделементи (редослед није битан)	<ul style="list-style-type: none"> • <i>name</i> (тачно један) • <i>description</i> (нула или један) • <i>uri</i> (нула или један) 	<ul style="list-style-type: none"> • тип: <i>string100</i> • тип: <i>string1000</i> • тип: <i>uri</i>
Атрибути	<ul style="list-style-type: none"> • <i>sourceId</i> (тачно један) 	<ul style="list-style-type: none"> • тип: <i>positiveInteger</i>

5.5.1.4 *contents*

Елемент *contents* представља колекцију елемената *content*, који садрже податке о културним добрима.

Дијаграм



Поделементи

- *content* (нула или више)

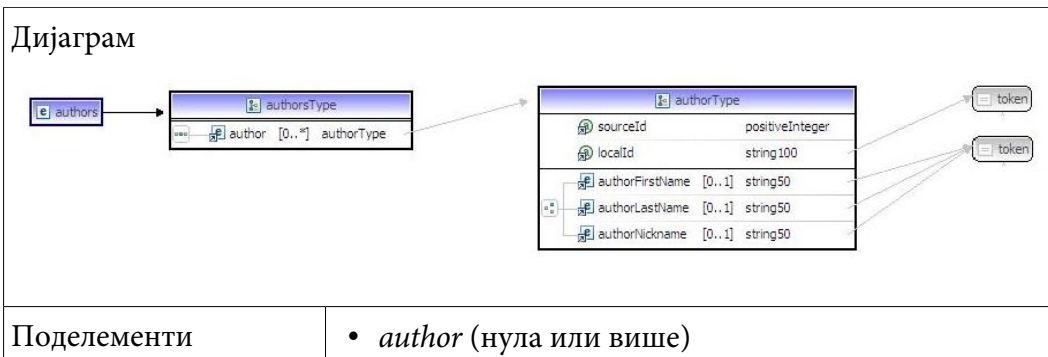
5.5.1.5 *content*

Елемент *content* садржи податке о културном добру неког извора ресурса. Редослед навођења података о културном добру није битан.

Поделементи (редослед није битан)	<ul style="list-style-type: none"> • <i>name</i> (тачно један) • <i>description</i> (нула или један) • <i>type</i> (тачно један) • <i>language</i> (нула или један) • <i>publisher</i> (нула или један) • <i>pubDate</i> (нула или један) • <i>edition</i> (нула или један) • <i>isbn</i> (нула или један) • <i>issn</i> (нула или један) • <i>duration</i> (нула или један) • <i>width</i> (нула или један) • <i>height</i> (нула или један) • <i>technique</i> (нула или један) 	<ul style="list-style-type: none"> • тип: <i>string100</i> • тип: <i>string1000</i> • тип: <i>typeOfContent</i> • тип: <i>language</i> • тип: <i>string100</i> • тип: <i>date</i> • тип: <i>positiveInteger</i> • тип: <i>isbn</i> • тип: <i>issn</i> • тип: <i>float</i> • тип: <i>positiveInteger</i> • тип: <i>positiveInteger</i> • тип: <i>string100</i>
Атрибути	<ul style="list-style-type: none"> • <i>sourceId</i> (тачно један) • <i>localId</i> (тачно један) 	<ul style="list-style-type: none"> • тип: <i>positiveInteger</i> • тип: <i>string100</i>

5.5.1.6 *authors*

Елемент *authors* представља колекцију елемената *author*, који садрже податке о ауторима односно ствараоцима културних добара.



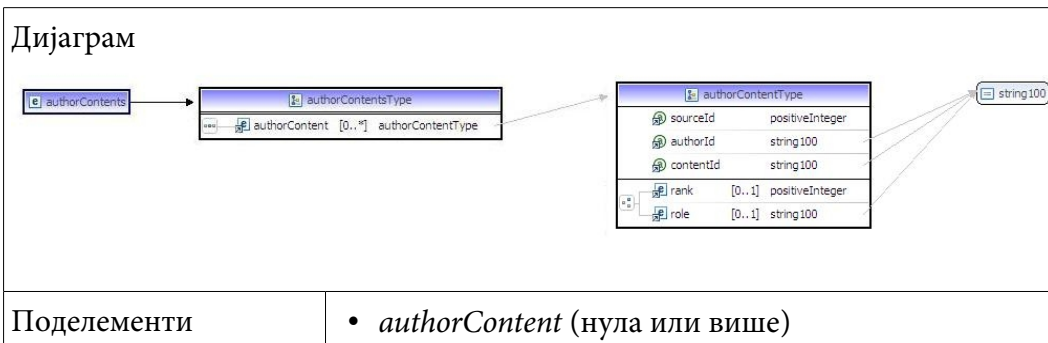
5.5.1.7 *author*

Елемент *author* садржи податке о неком аутору. Редослед навођења података о аутору није битан.

Подеlementи (редослед није битан)	<ul style="list-style-type: none"> <i>authorFirstName</i> (нула или један) <i>authorLastName</i> (нула или један) <i>authorNickname</i> (нула или један) 	<ul style="list-style-type: none"> тип: <i>string50</i> тип: <i>string50</i> тип: <i>string50</i>
Атрибути	<ul style="list-style-type: none"> <i>sourceId</i> (тачно један) <i>localId</i> (тачно један) 	<ul style="list-style-type: none"> тип: <i>positiveInteger</i> тип: <i>string100</i>

5.5.1.8 *authorContents*

Елемент *authorContents* представља колекцију елемената *authorContent*, који садрже податке о односу аутора и културних добара.



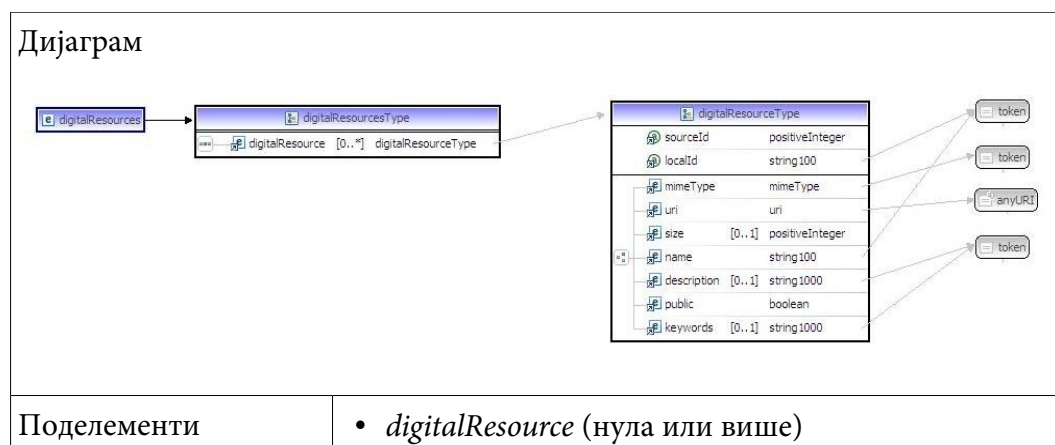
5.5.1.9 *authorContent*

Елемент *authorContent* садржи податке који везују аутора за одређено културно добро. Подаци говоре о томе који аутор је стваралац ког културног добра, са којом улогом и којим степеном учешћа. Редослед навођења података није битан.

Поделементи (редослед није битан)	<ul style="list-style-type: none"> • <i>seqNumber</i> (нула или један) • <i>role</i> (нула или један) 	<ul style="list-style-type: none"> • тип: <i>positiveInteger</i> • тип: <i>string100</i>
Атрибути	<ul style="list-style-type: none"> • <i>sourceId</i> (тачно један) • <i>authorId</i> (тачно један) • <i>contentId</i> (тачно један) 	<ul style="list-style-type: none"> • тип: <i>positiveInteger</i> • тип: <i>string100</i> • тип: <i>string100</i>

5.5.1.10 *digitalResources*

Елемент *digitalResources* представља колекцију елемената *digitalResource*, који садрже податке о дигиталним ресурсима.



5.5.1.11 *digitalResource*

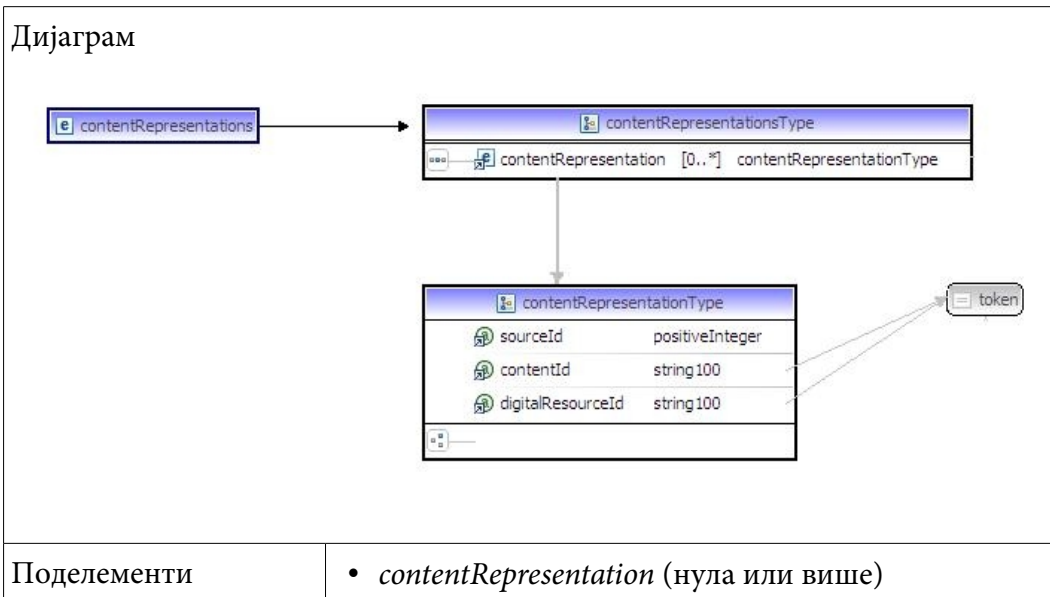
Елемент *digitalResource* садржи податке о дигиталном ресурсу неког извора. Редослед навођења података о дигиталном ресурсу није битан.

Поделементи (редослед није битан)	<ul style="list-style-type: none"> • <i>mimeType</i> (тачно један) • <i>uri</i> (тачно један) • <i>size</i> (нула или један) • <i>name</i> (тачно један) • <i>description</i> (нула или један) • <i>public</i> (тачно један) • <i>keywords</i> (нула или један) 	<ul style="list-style-type: none"> • тип: <i>mimeType</i> • тип: <i>uri</i> • тип: <i>positiveInteger</i> • тип: <i>string100</i> • тип: <i>string1000</i> • тип: <i>boolean</i> • тип: <i>string1000</i>
--------------------------------------	--	--

Атрибути	<ul style="list-style-type: none"> • <i>sourceId</i> (тачно један) • <i>contentId</i> (тачно један) • <i>digitalResourceId</i> (тачно један) 	<ul style="list-style-type: none"> • тип: <i>positiveInteger</i> • тип: <i>string100</i> • тип: <i>string100</i>
----------	---	---

5.5.1.12 *contentRepresentations*

Елемент *contentRepresentations* представља колекцију елемената *contentRepresentation*, који садрже податке о односу културних добара и њихових дигиталних репрезентација.



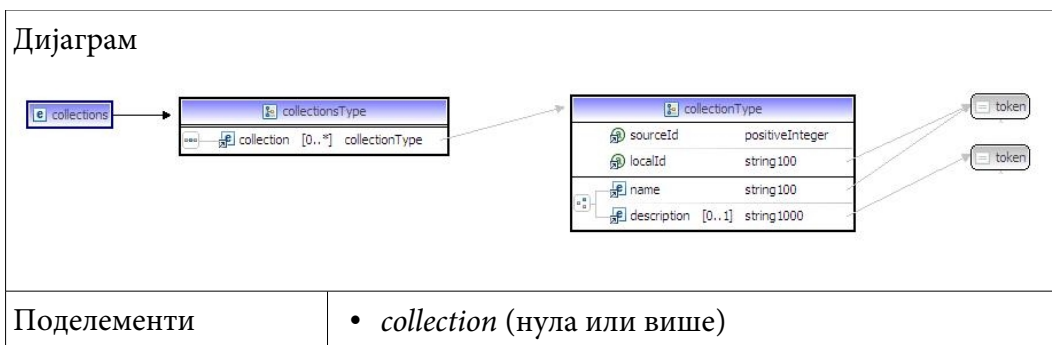
5.5.1.13 *contentRepresentation*

Елемент *contentRepresentation* садржи податке о односу културног добра и његове дигиталне репрезентације. Редослед навођења података није битан.

Атрибути	<ul style="list-style-type: none"> • <i>sourceId</i> (тачно један) • <i>contentId</i> (тачно један) • <i>digitalResourceId</i> (тачно један) 	<ul style="list-style-type: none"> • тип: <i>positiveInteger</i> • тип: <i>string100</i> • тип: <i>string100</i>
----------	---	---

5.5.1.14 *collections*

Елемент *collections* представља колекцију елемената *collection*, који садрже податке о збиркама дигиталних ресурса.



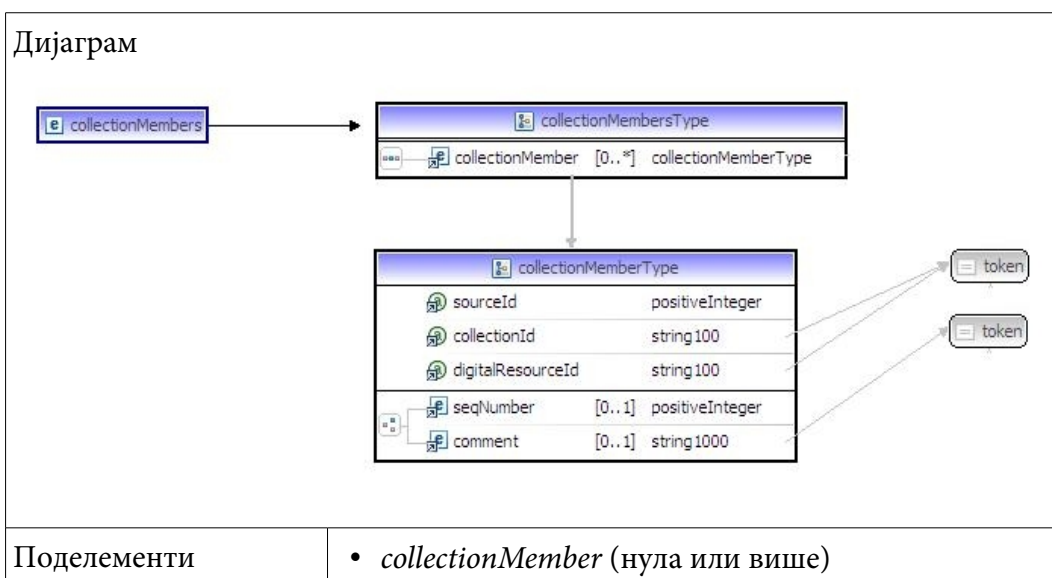
5.5.1.15 *collection*

Елемент *collection* садржи податке о збирци дигиталних ресурса. Редослед навођења података о збирци није битан.

Поделементи (редослед није битан)	<ul style="list-style-type: none"> <i>name</i> (тачно један) <i>description</i> (нула или један) 	<ul style="list-style-type: none"> тип: <i>string100</i> тип: <i>string1000</i>
Атрибути	<ul style="list-style-type: none"> <i>sourceId</i> (тачно један) <i>localId</i> (тачно један) 	<ul style="list-style-type: none"> тип: <i>positiveInteger</i> тип: <i>string100</i>

5.5.1.16 *collectionMembers*

Елемент *collectionMembers* представља колекцију елемената *collectionMember*, који садрже податке о односу збирки и дигиталних ресурса.



5.5.1.17 *collectionMember*

Елемент *collectionMember* садржи податке који везују дигитални ресурс за одређену збирку. Редослед навођења података није битан.

Поделементи (редослед није битан)	<ul style="list-style-type: none"> • <i>seqNumber</i> (нула или један) • <i>comment</i> (нула или један) 	<ul style="list-style-type: none"> • тип: <i>positiveInteger</i> • тип: <i>string1000</i>
Атрибути	<ul style="list-style-type: none"> • <i>sourceId</i> (тачно један) • <i>collectionId</i> (тачно један) • <i>digitalResourceId</i> (тачно један) 	<ul style="list-style-type: none"> • тип: <i>positiveInteger</i> • тип: <i>string100</i> • тип: <i>string100</i>

5.5.2 Атрибути

<i>version</i>	<i>xs:token</i>
<i>localId</i>	<i>string100</i>
<i>sourceId</i>	<i>xs:positiveInteger</i>
<i>contentId</i>	<i>string100</i>
<i>authorId</i>	<i>string100</i>
<i>digitalResourceId</i>	<i>string100</i>
<i>collectionId</i>	<i>string100</i>

5.5.3 Типови

<i>string1000</i>	<i>xs:token, maxlength=1000</i>
<i>string100</i>	<i>xs:token, maxlength=100</i>
<i>string50</i>	<i>xs:token, maxlength=50</i>
<i>uri</i>	<i>xs:anyUri, maxlength=1000</i>
<i>typeOfContent</i>	<i>xs:token, enumeration{book, magazine, audio, video, painting, photo}</i>
<i>isbn</i>	<i>xs:NMTOKEN, maxlength=20</i> <i>pattern:[0-9xX-]+</i>
<i>issn</i>	<i>xs:NMTOKEN, maxlength=20</i> <i>pattern:[0-9xX-]+</i>
<i>contentType</i>	<i>xs:token, enumeration{application/pdf...}</i>

5.6 Претраживање

Једна од најважнијих софтверских компоненти централног регистра је претраживање садржаја регистра. Прототип регистра обезбеђује два различита начина претраживања дигиталних ресурса:

- Текстуално претраживање на нивоу речи (енгл. *full text search*) које омогућује проналажење дигиталних ресурса на основу задатих кључних речи или фразе.
- Претраживање по атрибутима односно проналажење дигиталних ресурса који задовољавају задате критеријуме.

5.6.1 Текстуално претраживање

Уобичајено решење за претрагу неког текстуалног поља је коришћење оператора *LIKE* у упитима на *SQL*-у. На пример, када је потребно наћи све дигиталне ресурсе у чијем називу се налази текст „uneti_tekst“, може да се користи следећи упит:

```
select * from digital_resource
where name like '%uneti_tekst%'
```

Претрага по више задатих критеријума, може се решити на сличан начин, уз примену одговарајућих логичких операција. Проблем је у томе што коришћење тзв. џокера (у претходном случају %), онемогућује коришћење индекса, што има за последицу спорије извршавање упита. Код мањих система то може бити занемарљиво. С обзиром на то да ће централни регистар дигиталних ресурса располагати огромном количином података, оваква претрага би била видно спорија. За додавање могућности текстуалне претраге односно технике претраживања на нивоу речи, може се користити *DB2 Net Search Extender* [NSE] . Још једна од предности овог начина претраживања јесте могућност рангирања резултата, а за одређене језике, *DB2 Net Search Extender* подржава и препознавање корена речи и њених различитих облика. У даљем тексту је описано како се реализују поменуте могућности.

5.6.1.1 *DB2 Net Search Extender*

Покретање сервиса

Да би користили текстуалну претрагу, потребно је покренути *NSE* сервис наредбом:

```
db2text START
```

Сервис се може зауставити наредбом:

```
db2text STOP
```

Омогућавање текстуалне претраге на бази података

Да би база података могла да користи *DB2 Net Search Extender*, потребно је то експлицитно омогућити:

```
db2text ENABLE DATABASE FOR TEXT CONNECT TO ime_base
```

За једну базу података, довољно је извршити ову наредбу само једанпут.

Прављење текстуалних индекса

За сваку колону по којој се претражује, потребно је направити по текстуални индекс:

```
db2text CREATE INDEX ime_sheme.ime_indeksa FOR TEXT ON  
ime_sheme.ime_tabele(ime_kolone)
```

За попуњавање направљеног индекса, користи се наредба за синхронизацију индекса и садржаја текстуалне колоне на којој је он направљен:

```
db2text UPDATE INDEX ime_sheme.ime_indeksa FOR TEXT
```

Одржавање текстуалних индекса

Након прављења и првог попуњавања текстуалног индекса подацима, индекс је потребно и даље одржавати. Након операција додавања, измене или брисања над одређеном табелом, потребно је ажурирати и садржај индекса тако да одговара новим подацима. При прављењу индекса, могуће је навести опцију `RECREATE INDEX ON UPDATE` којом се при сваком ажурирању обнавља садржај индекса. У случају већих табела, ова операција може бити прилично скупа, па је тада боље освежавати индекс периодично. За периодично ажурирање индекса, потребно је навести учесталост и минимални број измена да би се ажурирање индекса извршило. На пример, ако је потребно да се већ направљени индекс промени тако да се освежава сваког радног дана у 12:00 и то само ако је измењено бар 100 вредности у одговарајућој колони, то се може учинити наредбом:

```
db2text ALTER INDEX ime_sheme.ime_indeksa FOR TEXT  
UPDATE FREQUENCY d(1,2,3,4,5) h(12) m(00)  
UPDATE MINIMUM 100
```


Методe претраживања

DB2 Net Search Extender допуњава SQL додатним скаларним функцијама за текстуално претраживање:

- **CONTAINS** – претражује колону на којој је изграђен текстуални индекс и за сваки ред, у случају да поље колоне садржи задати услов претраге, враћа вредност 1, а у супротном враћа 0.

```
SELECT *  
FROM ime_sheme.ime_tabele  
WHERE CONTAINS(ime_kolone, '"uslov_trazenja"') = 1
```

- **NUMBEROFMATCHES** – на основу задатог услова, претражује колону на којој је изграђен текстуални индекс и за сваки ред израчунава број појављивања услова претраге.

```
SELECT t.*, NUMBEROFMATCHES(ime_kolone, '"uslov_trazenja"')  
FROM ime_sheme.ime_tabele AS t  
WHERE NUMBEROFMATCHES(ime_kolone, '"uslov_trazenja"') > 0
```

- **SCORE** – на основу задатог услова, претражује колону на којој је изграђен текстуални индекс и за сваки ред израчунава колико добро је поље колоне описано условом претраге у односу на друга поља у индексу. Квалитет је одређен бројем погодака у односу на величину поља.

```
WITH temp(id, score) AS (  
    SELECT id, SCORE(ime_kolone, '"uslov_trazenja"')  
    FROM ime_sheme.ime_tabele  
)  
  
SELECT *  
FROM temp  
WHERE score > 0  
ORDER BY score ASC
```

DB2 Net Search Extender пружа и могућност тзв. „непрецизног“ (енгл. *fuzzy*) претраживања које проналази и речи које су у наведеној мери сличне услову претраге.

```
SELECT *  
FROM ime_sheme.ime_tabele  
WHERE CONTAINS(ime_kolone,  
    'FUZZY FORM OF 80 "uslov_trazenja"') = 1
```

Непрецизна претрага се користи за препознавање погрешно унетог текста, где се степен подударања представља вредношћу од 1 до 100. Ако је степен подударања 100, онда се препознају само речи у тексту које се у потпуности подударају са условом претраге.

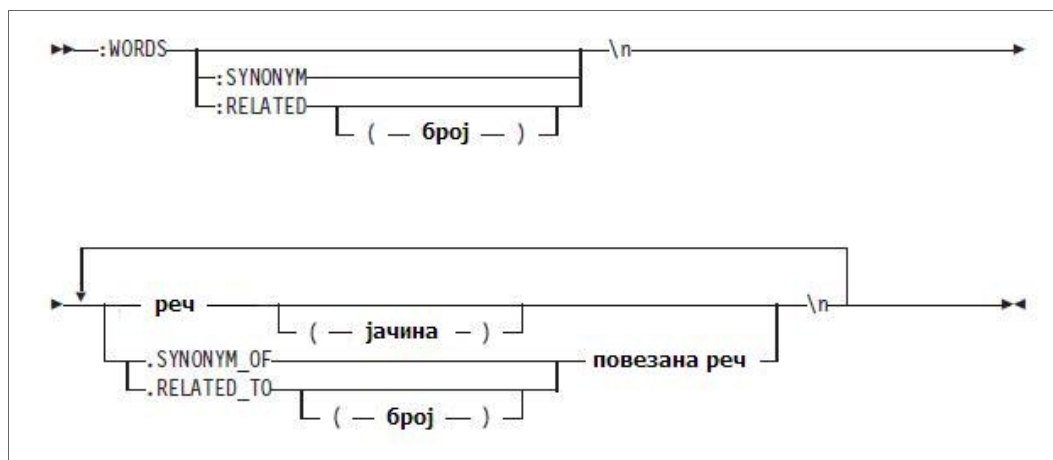
Речник синонима

Веома корисна метода претраживања јесте на основу речника синонима. Могуће је дефинисати синониме, а сем њих и речи које су међусобно повезане тј. асоцирају једна на другу. Синоними су речи које су идентичне или врло сличне по значењу, али се разликују по свом облику, као нпр. ученик – ђак или кућа – дом, а пример речи које асоцирају једна на другу би био рекет – тенис или ученик - школа.

```
SELECT *
FROM ime_sheme.ime_tabele
WHERE CONTAINS(ime_kolone,
    'THESAURUS "ime_fajla_sa_recnikom_sinonima"
    EXPAND SYNONYM TERM OF "uslov_trazenja"' ) = 1

SELECT *
FROM ime_sheme.ime_tabele
WHERE CONTAINS(ime_kolone,
    'THESAURUS "ime_fajla_sa_recnikom_sinonima"
    EXPAND RELATED TERM OF "uslov_trazenja"' ) = 1
```

За дефинисање листе синонима и речи које асоцирају једна на другу, неопходно је направити фајл у којем ће се налазити та листа у прописаном формату:



Слика 23: Формат речника синонима

Након прављења фајла, потребно је копирати га на одређену локацију (<sqllib>\db2ext\thes на Windows системима, <instance_owner_home>/sqllib/db2ext/thes на Unix системима) и компајлирати га:

```
db2extth -f recnik.def -ccsid 1208
```

5.6.1.2 Имплементација текстуалног претраживања

Да би се обезбедила што боља претрага дигиталних ресурса, бази података регистра је додата могућност текстуалног претраживања. Направљен је текстуални индекс на колони *keywords* табеле *digital_resource* што омогућује проналажење дигиталних ресурса на основу задатих кључних речи:

```
db2text CREATE INDEX nch.index_search FOR TEXT ON
nch.digital_resource(keywords)
CCSID 1208 LANGUAGE SR_SP FORMAT TEXT
UPDATE FREQUENCY D(*) H(0) M(0)
UPDATE MINIMUM 10
```

Текстуални индекс ће се освежавати периодично, сваког дана у 00:00 и то само ако има бар 10 измена. Врши се иницијално попуњавање индекса:

```
db2text UPDATE INDEX nch.index_search FOR TEXT
```

За одређене језике, *DB2 Net Search Extender* подржава препознавање корена речи и њених различитих облика, али на жалост не и за српски језик. Када би постојала подршка за српски језик, очекивано би било да, ако је реч по којој се претражују дигитални ресурси на пример „цар“, резултати претраге буду и „Благо цара Радована“ и „Цареви и краљеви“. С обзиром да нема ове могућности, проблему се може приступити и на други начин. Дефинише се листа синонима, где ће синоними неке речи бити и сви њени облици и различита писма (ћирилица и латиница).

...

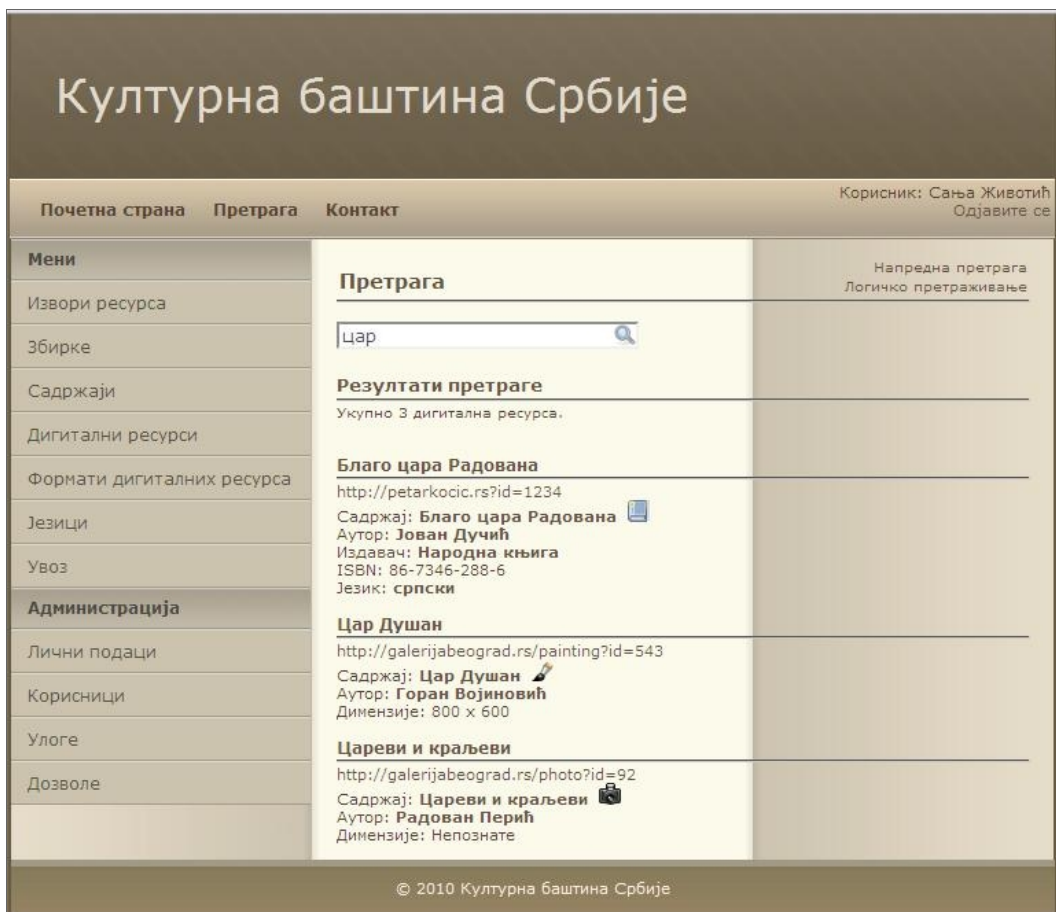
:WORDS:SYNONIM

```
цар
car
цара
cara
цару
caru
царе
care
царем
carem
цареви
carevi
царева
careva
царевима
carevima
цареве
careve
...
```

Првобитна листа речи преузета је из пакета *MySpell* [MS]. То су заправо две

листе речи, прва са 343242 речи на ћирилици и друга са истим бројем речи на латиници. Затим су ове две листе спојене тако да исте речи у различитом писму представљају синониме. Имајући у виду да је развијан прототип регистра, овде је примењен релативно једноставан алгоритам, који не даје увек тачне резултате, али је проценат успешности задовољавајући (штавише, изнад прага који би се очекивао од прототипа). На пример, једно од правила груписања синонима је да су речи које се разликују у свом последњем слову синоними. За групу речи цар – цара – цару – царе је то у реду, али резултат није тачан за на пример речи аван – аванс, које имају потпуно различито значење. Са друге стране ово аутоматски представља и један вид „непрецизне“ претраге тако да не представља неминовно слабост у приступу.

Листа синонима нема ни приближно све речи српског језика и све њихове облике, али се може постепено проширивати, у случају реалне примене регистра. Ради демонстрације начина рада текстуалне претраге, на слици 24 приказана је страна за претраживање дигиталних ресурса.



Слика 24: Текстуална претрага

5.6.2 Претраживање по атрибутима

Претраживање по атрибутима је заправо претрага дигиталних ресурса по задатим критеријумима. Критеријуми претраге могу бити основна својства дигиталних ресурса и садржаја који они презентују, као што су извор ресурса, назив, опис, величина и формат ресурса, аутор, тип садржаја и др. У зависности од изабраног типа садржаја, омогућено је и задавање додатних података. На слици 25 је дат пример напредног претраживања, са могућношћу уноса додатних критеријума, специфичних за изабрани тип садржаја.

Културна баштина Србије

Почетна страна Претрага Контакт

Корисник: Сања Животић
Одјавите се

Мени

- Извори ресурса
- Збирке
- Садржаји
- Дигитални ресурси
- Формати дигиталних ресурса
- Језици
- Увоз

Администрација

- Лични подаци
- Корисници
- Улоге
- Дозволе

Напредна претрага

Основна претрага
Логичко претраживање

Извор ресурса

Збирка

Врста садржаја

МIME тип

Назив

Аутор

Издавач

Издата од до

ISBN

Језик

Резултати претраге

Укупно 2 дигитална ресурса.

Благо цара Радована

<http://petarkocic.rs?id=1234>

Садржај: **Благо цара Радована**

Аутор: **Јован Дучић**

Издавач: **Народна књига**

ISBN: 86-7346-288-6

Језик: **српски**

Јутра са Леутара

<http://petarkocic.rs?id=43>

Садржај: **Јутра са Леутара**

Аутор: **Јован Дучић**

Издавач: **Народна књига, 01.08.2006**

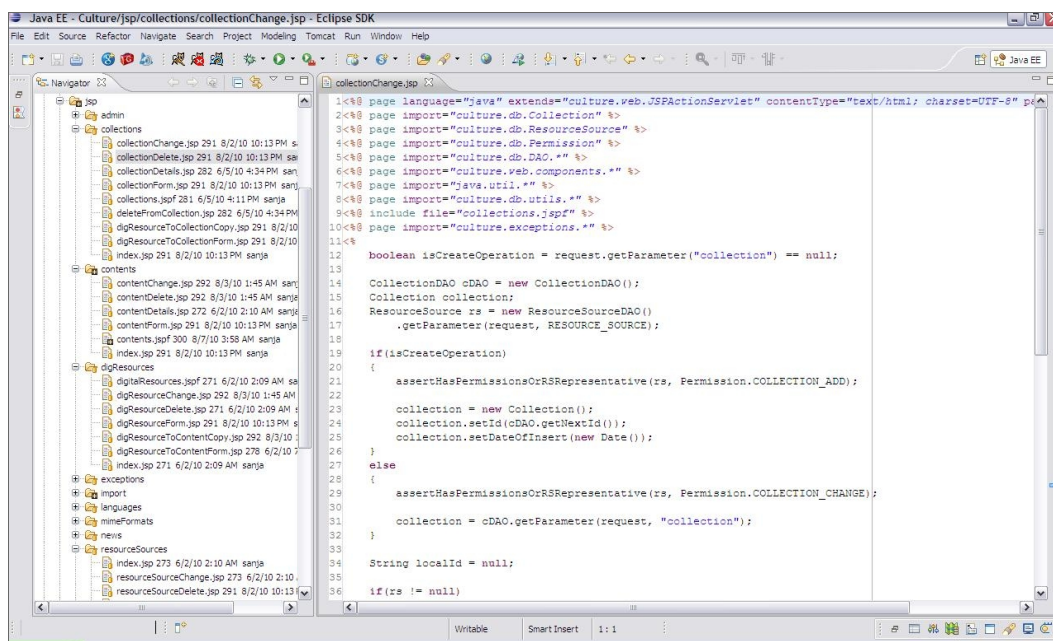
Језик: **српски**

© 2010 Културна баштина Србије

Слика 25: Напредна претрага

5.7 Развојно окружење

При имплементацији прототипа регистра, коришћено је развојно окружење *Eclipse 3.5.1*. [ECL]. Јава програми се могу компајлирати и покренути из командне линије, али и помоћу Јава развојних алата, као што су *Borland Jbuilder*, *Net Beans*, *Eclipse* и други. Јава развојни алати подржавају интегрисано развојно окружење (енгл. *Integrated Development Environment, IDE*) за ефикасан развој Јава програма. Омогућавају измену изворног кода, компајлирање, отклањање грешака и друго, што у великој мери повећава продуктивност рада.



Слика 26: Eclipse

Eclipse је развојно окружење отвореног кода које развија заједница *The Eclipse Foundation*. Посебно је прилагођено развоју апликација на програмском језику Јава, али није ограничено само на један програмски језик. Функционалност развојног окружења се може проширити инсталирањем различитих додатака (енгл. *plugin*), међу којима су и развојни алати за друге програмске језике. Веома користан додатак је *Web Tools Platform (WTP)* који обезбеђује већину алата потребних за развој Јава веб апликација. *Eclipse WTP* поједностављује руковање *XML* датотекама, *HTML* и *JSP* страницама и др. Постоји и додатни алат за управљање *Tomcat* сервером унутар самог развојног окружења *Eclipse*, као и подршка алату за објектно-релационо пресликавање, *Hibernate*.

6 Закључак

Прототип регистра дигитализоване културне баштине је пројектован и имплементиран применом савремених метода за пројектовање информационих система и развој апликација. Могућности система су недвосмислено представљене случајевима коришћења, на основу којих су у фази пројектовања једноставно издвојене класе података, као и односи између њих. Уочене класе су представљене дијаграмима класа података и директно су пресликане у класе објектно оријентисаног језика. На основу издвојених ентитета, њихових атрибута и односа, одређена је и структура релационе базе података која је представљена дијаграмом ентитета и односа. На основу дијаграма ентитета и односа, имплементирана је база података прототипа централног регистра дигиталних ресурса.

Прототип регистра је имплементиран применом Јава технологија које су обезбедиле преносивост, флексибилност, ефикасност и безбедност апликације. Релациони систем за управљање базама података *IBM DB2* се показао као добар избор за имплементацију регистра, а додатни алат *IBM DB2 Net Search Extender* омогућио је ефикасније проналажење жељених дигиталних ресурса у регистру. За аутоматизацију процеса пресликавања објеката из Јава апликације у релациону базу података (објектно релационо пресликавање) је коришћен алат *Hibernate*. Употреба *Hibernate*-а је умерено смањила перформансе софтвера, али је са друге стране знатно смањила време потребно за имплементацију перзистентног слоја и у великој мери олакшала одржавање постојећег кода. Продуктивност рада је значајно повећана употребом развојног окружења *Eclipse*.

Битан корак у имплементацији прототипа регистра је било дефинисање *XML* схеме за проверу исправности *XML* докумената са метаподацима. Тиме је омогућен увоз велике количине података посредством *XML* докумената што представља примаран начин попуњавања регистра подацима.

Прототип централног регистра садржи специфичности само најчешћих типова садржаја. Без значајних промена у структури, база података се може проширити тако да подржава нове типове садржаја, употребом *Entity-Attribute-Value (EAV)* модела, познатог и као *Object-Attribute-Value* модел и *Open Schema [EAV]*. *EAV* је модел података који се често користи када је број различитих атрибута ентитета неограничен, али се само мали број атрибута заправо користи у пракси. Бази података регистра би била додата табела са могућим атрибутима, односно особинама садржаја, као и још једна табела

која чува податке о идентификатору садржаја, идентификатору атрибута и вредности атрибута.

Примена централног регистра у пракси би омогућила преглед и претраживање великог броја дигитализованих културних добара на једном месту. Централизација метаподатака би објединила институције и појединце који дигитализују културна добра и тиме спречила дигитализацију истих добара више пута. Једноставнија и безбеднија употреба и претраживање, као и повећање јавне доступности добара, дало би велики подстрек процесу дигитализације широм земље, а самим тим и очувању националне културне баштине.

Централни регистар дигиталних ресурса омогућује проналажење жељених ресурса на основу информација које дају извори. Да би се регистар попунио подацима, потребно је успоставити сарадњу са изворима – установама културе, образовним установама и појединцима. Систем је дизајниран тако да трпи даље измене и допуњавања, па би у договору са институцијама и појединцима, било могуће додати нове типове садржаја, својства дигиталних ресурса и разне специфичности сваког извора.

Објектно оријентисани приступ у развоју пројекта је омогућио јасно издвајање захтева и њихову трансформацију у софтверски систем. Извршена је декомпозиција проблема што је у великој мери повећало квалитет развоја. Изабране технологије и алати примењени у имплементацији, показали су се као добар избор у изради прототипа регистра. Наредни изазов је примена централног регистра дигитализоване културне баштине у пракси.

7 Додаци

7.1 Додатак А – Структура базе података

```
create table nch.language (
    id                integer not null,
    code              varchar(10) not null,
    name              varchar(200) not null,
    date_of_insert    timestamp not null with default
                      current timestamp,
    date_of_update    timestamp,

    constraint pk_language primary key (id),

    constraint uk_code unique (code)
);

create trigger nch.tg_language_u
no cascade before update on nch.language
referencing new as new
for each row mode db2sql
set new.date_of_update = current timestamp
;

create table nch.resource_source (
    id                integer not null,
    name              varchar(200) not null,
    description        varchar(2000),
    uri               varchar(2000),
    date_of_insert    timestamp not null with default
                      current timestamp,
    date_of_update    timestamp,

    constraint pk_resource_source primary key (id)
);

create trigger nch.tg_resource_source_u
no cascade before update on nch.resource_source
referencing new as new
for each row mode db2sql
set new.date_of_update = current timestamp
;

create table nch.collection (
    id                integer not null,
    local_id          varchar(200),
    resource_source_id integer,
    name              varchar(200) not null,
    description        varchar(2000),
    date_of_insert    timestamp not null with default
                      current timestamp,
    date_of_update    timestamp,

    constraint pk_collection primary key (id),

    foreign key fk_collection_rs (resource_source_id)
        references nch.resource_source
);
```

```

create trigger nch.tg_collection_u
no cascade before update on nch.collection
referencing new as new
for each row mode db2sql
set new.date_of_update = current timestamp
;

create table nch.author (
    id                integer not null,
    local_id          varchar(200) not null,
    resource_source_id integer not null,
    first_name        varchar(100),
    last_name         varchar(100),
    nickname          varchar(100),
    date_of_insert    timestamp not null with default
                        current timestamp,
    date_of_update    timestamp,

    constraint pk_author primary key (id),

    constraint uk_local_author unique (
        local_id,
        resource_source_id
    ),

    foreign key fk_author_rs (resource_source_id)
        references nch.resource_source
);

create trigger nch.tg_author_u
no cascade before update on nch.author
referencing new as new
for each row mode db2sql
set new.date_of_update = current timestamp
;

create table nch.content (
    id                integer not null,
    local_id          varchar(200) not null,
    resource_source_id integer not null,
    name              varchar(200) not null,
    description        varchar(2000),
    language_id       integer,

    type              varchar(20) not null,

    publisher          varchar(200),
    pub_date           date,
    edition            integer,
    isbn               varchar(20),
    issn               varchar(20),
    duration           float,
    width              integer,
    height             integer,
    technique          varchar(200),
    date_of_insert    timestamp not null with default
                        current timestamp,
    date_of_update    timestamp,

    constraint pk_content primary key (id),

```

```

constraint uk_local_content unique (
    local_id,
    resource_source_id
),

constraint chk_type check (
    type in (
        'book',
        'magazine',
        'audio',
        'video',
        'painting',
        'photo'
    )
),

foreign key fk_content_rs (resource_source_id)
references nch.resource_source,

foreign key fk_content_language (language_id)
references nch.language
);

create trigger nch.tg_content_u
no cascade before update on nch.content
referencing new as new
for each row mode db2sql
set new.date_of_update = current timestamp
;

create table nch.author_content (
    author_id            integer not null,
    content_id           integer not null,
    rank                integer,
    role                varchar(200),

    date_of_insert       timestamp not null with default
                        current timestamp,

    constraint pk_author_content primary key (
        author_id,
        content_id
    ),

    foreign key fk_ac_author (author_id)
    references nch.author (id)
    on delete cascade,

    foreign key fk_ac_content (content_id)
    references nch.content (id)
    on delete cascade
);

create table nch.mime_format (
    id                    integer not null,
    content_type          varchar(200) not null,
    file_extension        varchar(20),
    description           varchar(2000),
    date_of_insert       timestamp not null with default
                        current timestamp,
    date_of_update        timestamp,

    constraint pk_mime_format primary key (id)
);

```

```

create trigger nch.tg_mime_format_u
no cascade before update on nch.mime_format
referencing new as new
for each row mode db2sql
set new.date_of_update = current timestamp
;

create table nch.digital_resource (
    id                integer not null,
    local_id          varchar(200) not null,
    resource_source_id integer not null,
    mime_format_id    integer not null,
    size              bigint, -- bytes
    uri               varchar(2000) not null,
    name              varchar(200) not null,
    description        varchar(2000),
    public            smallint not null with default 1,
    keywords           varchar(2000) not null with default '',
    date_of_insert     timestamp not null with default
                        current timestamp,
    date_of_update     timestamp,

    constraint pk_digital_resource primary key (id),

    constraint uk_local_dr unique (
        local_id,
        resource_source_id
    ),

    foreign key fk_dr_rs (resource_source_id)
        references nch.resource_source,

    foreign key fk_dr_mf (mime_format_id)
        references nch.mime_format (id)
);

create trigger nch.tg_digital_resource_u
no cascade before update on nch.digital_resource
referencing new as new
for each row mode db2sql
set new.date_of_update = current timestamp
;

create table nch.content_representation (
    content_id          integer not null,
    digital_resource_id integer not null,

    date_of_insert      timestamp not null with default
                        current timestamp,

    constraint pk_content_representation primary key (
        content_id,
        digital_resource_id
    ),

    foreign key fk_cr_content (content_id)
        references nch.content (id)
        on delete cascade,

    foreign key fk_cr_dr (digital_resource_id)
        references nch.digital_resource (id)
        on delete cascade
);

```

```

create table nch.collection_member (
    collection_id        integer not null,
    digital_resource_id   integer not null,
    seq_number            integer,
    comment               varchar(2000),

    date_of_insert        timestamp not null with default
                        current timestamp,

    constraint pk_collection_member primary key (
        collection_id,
        digital_resource_id
    ),

    foreign key fk_cm_collection (collection_id)
        references nch.collection (id)
        on delete cascade,

    foreign key fk_cm_dig_resource (digital_resource_id)
        references nch.digital_resource (id)
        on delete cascade
);

create table nch.permission (
    id                    integer not null,
    name                  varchar(200) not null,
    description            varchar(2000),

    date_of_insert        timestamp not null with default
                        current timestamp,
    date_of_update        timestamp,

    constraint pk_permission primary key (id),

    constraint uk_perm_name unique (name)
);

create trigger nch.tg_permission_u
no cascade before update on nch.permission
referencing new as new
for each row mode db2sql
set new.date_of_update = current timestamp
;

create table nch.role (
    id                    integer not null,
    name                  varchar(200) not null,
    description            varchar(2000),
    date_of_insert        timestamp not null with default
                        current timestamp,
    date_of_update        timestamp,

    constraint pk_role primary key (id),

    constraint uk_role_name unique (name)
);

create trigger nch.tg_role_u
no cascade before update on nch.role
referencing new as new
for each row mode db2sql
set new.date_of_update = current timestamp
;

```

```

create table nch.role_permission (
    role_id            integer not null,
    permission_id      integer not null,

    date_of_insert     timestamp not null with default
                        current timestamp,

    constraint pk_role_permission primary key (
        role_id,
        permission_id
    ),

    foreign key fk_rp_role (role_id)
        references nch.role
        on delete cascade,

    foreign key fk_rp_permission (permission_id)
        references nch.permission
        on delete cascade
);

create table nch.user (
    id                 integer not null,
    username           varchar(20) not null,
    password           varchar(20) not null,
    first_name         varchar(100) not null,
    last_name          varchar(100) not null,
    email              varchar(100) not null,
    date_of_birth       date,
    gender             char,
    active             smallint not null with default 1,
    comment            varchar(2000),
    date_of_insert     timestamp not null with default
                        current timestamp,
    date_of_update     timestamp,

    constraint pk_user primary key (id),

    constraint uk_username unique (username)
);

create trigger nch.tg_user_u
no cascade before update on nch.user
referencing new as new
for each row mode db2sql
set new.date_of_update = current timestamp
;

create table nch.user_role (
    id                 integer not null,
    user_id            integer not null,
    role_id            integer not null,
    resource_source_id integer,
    date_of_insert     timestamp not null with default
                        current timestamp,

    constraint pk_user_role primary key (id),

    foreign key fk_ur_user (user_id)
        references nch.user
        on delete cascade,

```

```
foreign key fk_ur_role (role_id)
  references nch.role
  on delete cascade,

foreign key fk_ur_rs (resource_source_id)
  references nch.resource_source
);
```

7.2 Додатак Б - XML схема регистра

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- Definition of types -->
  <xs:simpleType name="string1000">
    <xs:restriction base="xs:token">
      <xs:maxLength value="1000"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="string100">
    <xs:restriction base="xs:token">
      <xs:maxLength value="100"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="string50">
    <xs:restriction base="xs:token">
      <xs:maxLength value="50"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="uri">
    <xs:restriction base="xs:anyURI">
      <xs:maxLength value="1000"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="typeOfContent">
    <xs:restriction base="xs:token">
      <xs:maxLength value="20"/>
      <xs:enumeration value="book"/>
      <xs:enumeration value="magazine"/>
      <xs:enumeration value="audio"/>
      <xs:enumeration value="video"/>
      <xs:enumeration value="painting"/>
      <xs:enumeration value="photo"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="isbn">
    <xs:restriction base="xs:NMTOKEN">
      <xs:maxLength value="20"/>
      <xs:pattern value="[0-9xX-]+"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="issn">
    <xs:restriction base="xs:NMTOKEN">
      <xs:maxLength value="20"/>
      <xs:pattern value="[0-9xX-]+"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="mimeType">
    <xs:restriction base="xs:token">
      <xs:enumeration value="application/andrew-inset"/>
      ...
    </xs:restriction>
  </xs:simpleType>
```



```

        <xs:enumeration value="x-epoc/x-sisx-app"/>
    </xs:restriction>
</xs:simpleType>

<!-- Definition of attributes -->
<xs:attribute name="version" type="xs:token"/>
<xs:attribute name="localId" type="string100"/>
<xs:attribute name="sourceId" type="xs:positiveInteger"/>
<xs:attribute name="collectionId" type="string100"/>
<xs:attribute name="contentId" type="string100"/>
<xs:attribute name="digitalResourceId" type="string100"/>
<xs:attribute name="authorId" type="string100"/>

<!-- Definition of simple elements -->
<xs:element name="name" type="string100"/>
<xs:element name="description" type="string1000"/>
<xs:element name="uri" type="uri"/>
<xs:element name="type" type="typeOfContent"/>
<xs:element name="authorFirstName" type="string50"/>
<xs:element name="authorLastName" type="string50"/>
<xs:element name="authorNickname" type="string50"/>
<xs:element name="publisher" type="string100"/>
<xs:element name="pubDate" type="xs:date"/>
<xs:element name="edition" type="xs:positiveInteger"/>
<xs:element name="language" type="xs:language"/>
<xs:element name="width" type="xs:positiveInteger"/>
<xs:element name="height" type="xs:positiveInteger"/>
<xs:element name="isbn" type="isbn"/>
<xs:element name="issn" type="issn"/>
<xs:element name="duration" type="xs:float"/>
<xs:element name="technique" type="string100"/>
<xs:element name="mimeType" type="mimeType"/>
<xs:element name="size" type="xs:positiveInteger"/>
<xs:element name="public" type="xs:boolean"/>
<xs:element name="keywords" type="string1000"/>
<xs:element name="seqNumber" type="xs:positiveInteger"/>
<xs:element name="role" type="string100"/>
<xs:element name="comment" type="string1000"/>

<!-- Definition of complex elements -->
<xs:element name="sources" type="sourcesType"/>
<xs:element name="source" type="sourceType"/>
<xs:element name="collections" type="collectionsType"/>
<xs:element name="collection" type="collectionType"/>
<xs:element name="contents" type="contentsType"/>
<xs:element name="content" type="contentType"/>
<xs:element name="digitalResource" type="digitalResourcesType"/>
<xs:element name="digitalResource" type="digitalResourceType"/>
<xs:element name="authors" type="authorsType"/>
<xs:element name="author" type="authorType"/>
<xs:element name="collectionMembers" type="collectionMembersType"/>
<xs:element name="collectionMember" type="collectionMemberType"/>
<xs:element name="authorContents" type="authorContentsType"/>
<xs:element name="authorContent" type="authorContentType"/>
<xs:element name="contentRepresentations"
    type="contentRepresentationsType"/>
<xs:element name="contentRepresentation"
    type="contentRepresentationType"/>

<!-- Definition of complex Types -->
<xs:complexType name="nationalHeritageType">
    <xs:all>

```

```

        <xs:element ref="sources" minOccurs="0"/>
        <xs:element ref="contents" minOccurs="0"/>
        <xs:element ref="authors" minOccurs="0"/>
        <xs:element ref="authorContents" minOccurs="0"/>
        <xs:element ref="digitalResources" minOccurs="0"/>
        <xs:element ref="contentRepresentations" minOccurs="0"/>
        <xs:element ref="collections" minOccurs="0"/>
        <xs:element ref="collectionMembers" minOccurs="0"/>
    </xs:all>
    <xs:attribute ref="version" use="required"/>
</xs:complexType>

<xs:complexType name="sourceType">
    <xs:all>
        <xs:element ref="name"/>
        <xs:element ref="description" minOccurs="0"/>
        <xs:element ref="uri" minOccurs="0"/>
    </xs:all>
    <xs:attribute ref="sourceId" use="required"/>
</xs:complexType>

<xs:complexType name="collectionType">
    <xs:all>
        <xs:element ref="name"/>
        <xs:element ref="description" minOccurs="0"/>
    </xs:all>
    <xs:attribute ref="sourceId"/>
    <xs:attribute ref="localId"/>
</xs:complexType>

<xs:complexType name="contentType">
    <xs:all>
        <xs:element ref="name"/>
        <xs:element ref="description" minOccurs="0"/>
        <xs:element ref="type"/>
        <xs:element ref="publisher" minOccurs="0"/>
        <xs:element ref="pubDate" minOccurs="0"/>
        <xs:element ref="edition" minOccurs="0"/>
        <xs:element ref="isbn" minOccurs="0"/>
        <xs:element ref="issn" minOccurs="0"/>
        <xs:element ref="language" minOccurs="0"/>
        <xs:element ref="duration" minOccurs="0"/>
        <xs:element ref="width" minOccurs="0"/>
        <xs:element ref="height" minOccurs="0"/>
        <xs:element ref="technique" minOccurs="0"/>
    </xs:all>
    <xs:attribute ref="sourceId" use="required"/>
    <xs:attribute ref="localId" use="required"/>
</xs:complexType>

<xs:complexType name="digitalResourceType">
    <xs:all>
        <xs:element ref="mimeType"/>
        <xs:element ref="uri"/>
        <xs:element ref="size" minOccurs="0"/>
        <xs:element ref="name"/>
        <xs:element ref="description" minOccurs="0"/>
        <xs:element ref="public"/>
        <xs:element ref="keywords" minOccurs="0"/>
    </xs:all>
    <xs:attribute ref="sourceId" use="required"/>
    <xs:attribute ref="localId" use="required"/>
</xs:complexType>

```

```

<xs:complexType name="authorType">
  <xs:all>
    <xs:element ref="authorFirstName" minOccurs="0"/>
    <xs:element ref="authorLastName" minOccurs="0"/>
    <xs:element ref="authorNickname" minOccurs="0"/>
  </xs:all>
  <xs:attribute ref="sourceId" use="required"/>
  <xs:attribute ref="localId" use="required"/>
</xs:complexType>

<xs:complexType name="collectionMemberType">
  <xs:all>
    <xs:element ref="seqNumber" minOccurs="0"/>
    <xs:element ref="comment" minOccurs="0"/>
  </xs:all>
  <xs:attribute ref="sourceId" use="required"/>
  <xs:attribute ref="collectionId" use="required"/>
  <xs:attribute ref="digitalResourceId" use="required"/>
</xs:complexType>

<xs:complexType name="authorContentType">
  <xs:all>
    <xs:element ref="seqNumber" minOccurs="0"/>
    <xs:element ref="role" minOccurs="0"/>
  </xs:all>
  <xs:attribute ref="sourceId" use="required"/>
  <xs:attribute ref="authorId" use="required"/>
  <xs:attribute ref="contentId" use="required"/>
</xs:complexType>

<xs:complexType name="contentRepresentationType">
  <xs:all></xs:all>
  <xs:attribute ref="sourceId" use="required"/>
  <xs:attribute ref="contentId" use="required"/>
  <xs:attribute ref="digitalResourceId" use="required"/>
</xs:complexType>

<xs:complexType name="sourcesType">
  <xs:sequence>
    <xs:element ref="source" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="collectionsType">
  <xs:sequence>
    <xs:element ref="collection" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="contentsType">
  <xs:sequence>
    <xs:element ref="content" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="digitalResourcesType">
  <xs:sequence>
    <xs:element ref="digitalResource" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="authorsType">
  <xs:sequence>
    <xs:element ref="author" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="collectionMembersType">
  <xs:sequence>
    <xs:element ref="collectionMember" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="authorContentsType">
  <xs:sequence>
    <xs:element ref="authorContent" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="contentRepresentationsType">
  <xs:sequence>
    <xs:element ref="contentRepresentation"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="nationalHeritage" type="nationalHeritageType">

  <!-- Make sure that each source has a unique id -->
  <xs:key name="keySource">
    <xs:selector xpath="sources/source"/>
    <xs:field xpath="@sourceId"/>
  </xs:key>

  <!-- Make sure that each source has a unique name -->
  <xs:unique name="nameSource">
    <xs:selector xpath="sources/source"/>
    <xs:field xpath="name"/>
  </xs:unique>

  <!-- Make sure that each collection has a unique
  (sourceId, localId) -->
  <xs:unique name="keyCollection">
    <xs:selector xpath="collections/collection"/>
    <xs:field xpath="@sourceId"/>
    <xs:field xpath="@localId"/>
  </xs:unique>

  <!-- Make sure that the sourceId of each collection element is
  identifier for existing source -->
  <xs:keyref name="refCollectionSource" refer="keySource">
    <xs:selector xpath="collections/collection"/>
    <xs:field xpath="@sourceId"/>
  </xs:keyref>

  <!-- Make sure that each content has a unique
  (sourceId, localId) -->
  <xs:key name="keyContent">
    <xs:selector xpath="contents/content"/>
    <xs:field xpath="@sourceId"/>
    <xs:field xpath="@localId"/>
  </xs:key>

```

```

<!-- Make sure that the sourceId of each content element is
identifier for existing source -->
<xs:keyref name="refContentSource" refer="keySource">
  <xs:selector xpath="contents/content"/>
  <xs:field xpath="@sourceId"/>
</xs:keyref>

<!-- Make sure that each digitalResource has a unique
(sourceId, localId) -->
<xs:key name="keyDigitalResource">
  <xs:selector xpath="digitalResources/digitalResource"/>
  <xs:field xpath="@sourceId"/>
  <xs:field xpath="@localId"/>
</xs:key>

<!-- Make sure that the sourceId of each digitalResource element
is identifier for existing source -->
<xs:keyref name="refDigitalResourceSource" refer="keySource">
  <xs:selector xpath="digitalResources/digitalResource"/>
  <xs:field xpath="@sourceId"/>
</xs:keyref>

<!-- Make sure that each author has a unique
(sourceId, localId) -->
<xs:key name="keyAuthor">
  <xs:selector xpath="authors/author"/>
  <xs:field xpath="@sourceId"/>
  <xs:field xpath="@localId"/>
</xs:key>

<!-- Make sure that the sourceId of each author element is
identifier for existing source -->
<xs:keyref name="refAuthorSource" refer="keySource">
  <xs:selector xpath="authors/author"/>
  <xs:field xpath="@sourceId"/>
</xs:keyref>

<!-- Make sure that each collectionMember has a unique
(sourceId, collectionId, digitalResourceId) -->
<xs:unique name="keyCollectionMember">
  <xs:selector xpath="collectionMembers/collectionMember"/>
  <xs:field xpath="@sourceId"/>
  <xs:field xpath="@collectionId"/>
  <xs:field xpath="@digitalResourceId"/>
</xs:unique>

<!-- Make sure that each authorContent has a unique
(sourceId, contentId, authorId) -->
<xs:key name="keyAuthorContent">
  <xs:selector xpath="authorContents/authorContent"/>
  <xs:field xpath="@sourceId"/>
  <xs:field xpath="@contentId"/>
  <xs:field xpath="@authorId"/>
</xs:key>

<!-- Make sure that each contentRepresentation has a unique
(sourceId, contentId, digitalResourceId) -->
<xs:key name="keyContentRepresentation">
  <xs:selector
    xpath="contentRepresentations/contentRepresentation"/>
  <xs:field xpath="@sourceId"/>
  <xs:field xpath="@contentId"/>
  <xs:field xpath="@digitalResourceId"/>
</xs:key>

```

```

<!-- Make sure that the (sourceId, collectionId) of each
collectionMember element is identifier for existing
collection -->
<xs:keyref name="refCollectionInCollectionMember"
    refer="keyCollection">
    <xs:selector xpath="collectionMembers/collectionMember"/>
    <xs:field xpath="@sourceId"/>
    <xs:field xpath="@collectionId"/>
</xs:keyref>

<!-- Make sure that the (sourceId, digitalResourceId) of each
collectionMember element is identifier for existing
digitalResource -->
<xs:keyref name="refDigitalResourceInCollectionMember"
    refer="keyDigitalResource">
    <xs:selector xpath="collectionMembers/collectionMember"/>
    <xs:field xpath="@sourceId"/>
    <xs:field xpath="@digitalResourceId"/>
</xs:keyref>

<!-- Make sure that the (sourceId, contentId) of each
authorContent element is identifier for existing content -->
<xs:keyref name="refContentInAuthorContent" refer="keyContent">
    <xs:selector xpath="authorContents/authorContent"/>
    <xs:field xpath="@sourceId"/>
    <xs:field xpath="@contentId"/>
</xs:keyref>

<!-- Make sure that the (sourceId, authorId) of each
authorContent element is identifier for existing author -->
<xs:keyref name="refAuthorInAuthorContent" refer="keyAuthor">
    <xs:selector xpath="authorContents/authorContent"/>
    <xs:field xpath="@sourceId"/>
    <xs:field xpath="@authorId"/>
</xs:keyref>

<!-- Make sure that the (sourceId, contentId) of each
contentRepresentation element is identifier for existing
content -->
<xs:keyref name="refContentInContentRepresentation"
    refer="keyContent">
    <xs:selector
        xpath="contentRepresentations/contentRepresentation"/>
    <xs:field xpath="@sourceId"/>
    <xs:field xpath="@contentId"/>
</xs:keyref>

<!-- Make sure that the (sourceId, digitalResourceId) of each
contentRepresentation element is identifier for existing
digitalResource -->
<xs:keyref name="refDigitalResourceInContentRepresentation"
    refer="keyDigitalResource">
    <xs:selector
        xpath="contentRepresentations/contentRepresentation"/>
    <xs:field xpath="@sourceId"/>
    <xs:field xpath="@digitalResourceId"/>
</xs:keyref>
</xs:element>
</xs:schema>

```

7.3 Додатак В - Пример XML документа

Ради лакшег разумевања XML схеме прототипа регистра, дат је пример XML документа, исправног у односу на схему.

```
<?xml version="1.0" encoding="UTF-8"?>
<nationalHeritage version="1.0">
  <sources>
    <source sourceId="1">
      <name>Библиотека Петар Кочић</name>
      <uri>http://www.petarkocic.rs</uri>
    </source>
    <source sourceId="2">
      <name>Галерија Београд</name>
      <description>
        Галерија слика из Београда.
      </description>
      <uri>http://www.galerijabeograd.rs</uri>
    </source>
  </sources>
  <contents>
    <content sourceId="1" localId="c1">
      <name>Кратка историја Београда</name>
      <type>book</type>
      <isbn>8673464668</isbn>
    </content>
    <content sourceId="1" localId="c2">
      <name>Хазарски речник</name>
      <description>...</description>
      <type>book</type>
      <publisher>Народна књига</publisher>
      <pubDate>2006-09-02</pubDate>
      <edition>2</edition>
      <isbn>978-8673464565</isbn>
      <language>SR</language>
    </content>
    <content sourceId="2" localId="c1">
      <name>Калемегдан ноћу</name>
      <type>photo</type>
    </content>
    <content sourceId="2" localId="c2">
      <name>Теразије 2010.</name>
      <technique>акварел</technique>
      <type>painting</type>
    </content>
  </contents>
  <authors>
    <author sourceId="1" localId="a1">
      <authorFirstName>Милорад</authorFirstName>
      <authorLastName>Павић</authorLastName>
    </author>
    <author sourceId="2" localId="a1">
      <authorFirstName>Милена</authorFirstName>
      <authorLastName>Илић</authorLastName>
    </author>
    <author sourceId="2" localId="a2">
      <authorFirstName>Стефан</authorFirstName>
      <authorLastName>Јосиповић</authorLastName>
    </author>
  </authors>
</nationalHeritage>
```

```

<authorContents>
  <authorContent sourceId="1" contentId="c1" authorId="a1">
    <role>писац</role>
  </authorContent>
  <authorContent sourceId="1" contentId="c2" authorId="a1">
    <role>писац</role>
  </authorContent>
  <authorContent sourceId="2" contentId="c1" authorId="a1">
    <role>фотограф</role>
  </authorContent>
  <authorContent sourceId="2" contentId="c2" authorId="a1">
    <role>сликар</role>
    <seqNumber>2</seqNumber>
  </authorContent>
  <authorContent sourceId="2" contentId="c2" authorId="a2">
    <role>сликар</role>
    <seqNumber>1</seqNumber>
  </authorContent>
</authorContents>

<digitalResources>
  <digitalResource sourceId="1" localId="dr1">
    <mimeType>application/pdf</mimeType>
    <uri>http://www.hazrecnik1.rs</uri>
    <name>Хазарски речник</name>
    <public>0</public>
  </digitalResource>
  <digitalResource sourceId="1" localId="dr2">
    <mimeType>application/pdf</mimeType>
    <size>10000</size>
    <uri>http://www.hazrecnik2.rs</uri>
    <name>Хазарски речник</name>
    <description>...</description>
    <public>1</public>
  </digitalResource>
  <digitalResource sourceId="1" localId="dr3">
    <mimeType>application/pdf</mimeType>
    <uri>http://www.serbianhistory.rs</uri>
    <name>Дигитализована Кратка историја Београда</name>
    <public>1</public>
  </digitalResource>
  <digitalResource sourceId="2" localId="dr1">
    <mimeType>image/jpeg</mimeType>
    <uri>http://www.galerija.rs/kalemegdan</uri>
    <name>Калемегдан</name>
    <public>1</public>
  </digitalResource>
  <digitalResource sourceId="2" localId="dr2">
    <mimeType>image/jpeg</mimeType>
    <uri>http://www.galerija.rs/terazije</uri>
    <name>Теразије</name>
    <public>1</public>
  </digitalResource>
  <digitalResource sourceId="2" localId="dr3">
    <mimeType>application/pdf</mimeType>
    <uri>http://www.galerija.rs/beograd</uri>
    <name>Слике из Београда</name>
    <public>1</public>
  </digitalResource>
</digitalResources>

```



```

<contentRepresentations>
  <contentRepresentation sourceId="1" contentId="c2"
    digitalResourceId="dr1"/>
  <contentRepresentation sourceId="1" contentId="c2"
    digitalResourceId="dr2"/>
  <contentRepresentation sourceId="1" contentId="c1"
    digitalResourceId="dr3"/>
  <contentRepresentation sourceId="2" contentId="c1"
    digitalResourceId="dr1"/>
  <contentRepresentation sourceId="2" contentId="c2"
    digitalResourceId="dr2"/>
  <contentRepresentation sourceId="2" contentId="c1"
    digitalResourceId="dr3"/>
  <contentRepresentation sourceId="2" contentId="c2"
    digitalResourceId="dr3"/>
</contentRepresentations>

<collections>
  <collection sourceId="1" localId="coll1">
    <name>Изабрана дела Милорада Павића</name>
    <description>
      Збирка садржи изабрана дела Милорада Павића.
    </description>
  </collection>
  <collection sourceId="1" localId="coll3">
    <name>Најчитаније књиге</name>
    <description>
      Збирка најчитанијих књига.
    </description>
  </collection>
  <collection sourceId="2" localId="coll1">
    <name>Збирка слика из 2010. године</name>
  </collection>
</collections>

<collectionMembers>
  <collectionMember sourceId="1" collectionId="coll1"
    digitalResourceId="dr1"/>
  <collectionMember sourceId="1" collectionId="coll1"
    digitalResourceId="dr2"/>
  <collectionMember sourceId="1" collectionId="coll1"
    digitalResourceId="dr3"/>
  <collectionMember sourceId="1" collectionId="coll3"
    digitalResourceId="dr1"/>
  <collectionMember sourceId="1" collectionId="coll3"
    digitalResourceId="dr2"/>
</collectionMembers>
</nationalHeritage>

```

8 Референце

[OOD] David Avison, Guy Fitzgerald, Information Systems Development, McGraw Hill Higher Education, 2003. ISBN 978-0077096267

[UML] Martin Fowler, Kendall Scott, UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley, 1999. ISBN 978-0201657838

[VP] Visual Paradigm International Ltd., Visual Paradigm for UML User's Guide <http://www.visual-paradigm.com>

[ERD] Sighe Bagui, Richard Earp, Database Design Using Entity-Relationship Diagrams, Auerbach Publications, 2003. ISBN 978-0849315480

[IANA] Internet assigned numbers authority, MIME Media Types <http://www.iana.org/assignments/media-types/>

[JAVA] Ken Arnold, James Gosling, The Java Programming Language, Java Series, Sun Microsystems, 1996. ISBN 0-201-63455-4

[JSP] Hans Bergsten, JavaServer pages, O'Reilly Media, 2003. ISBN 978-0596005634

[TOM] Jason Brittain, Ian F. Darwin, Tomcat: The Definitive Guide, O'Reilly Media, 2007. ISBN 978-0596101060

[DB2] IBM DB2 Version 9.7: SQL Reference, IBM Corporation, 2009.

[NSE] IBM DB2 Version 9.7: DB2 Net Search Extender documentation, IBM Corporation, 2009.

[HIB] Christian Bauer, Gavin King, Java Persistence with Hibernate, Manning Publications, 2006, ISBN 978-1932394887

[XML] Eric T. Ray, Learning XML, O'Reilly Media, 2001. ISBN 978-0596000462

[DTD] Elliotte Rusty Harold, W. Scott Means, XML in a Nutshell, O'Reilly Media, 2004. ISBN 978-0596007645

[XSD] Eric van der Vlist, XML Schema: The W3C's Object-Oriented Descriptions for XML, O'Reilly Media, 2002. ISBN 978-0596002527

[RLX] Eric van der Vlist, Relax Ng, O'Reilly Media, 2003. ISBN 978-0596004217

[W3C] World Wide Web Consortium <http://www.w3schools.com/>

[MS] Горан Ракић, MySpell пакет за српски језик у програму GNU Aspell
<http://srpski.org/aspell/>

[ECL] Eclipse documentation, The Eclipse Foundation, 2010.
<http://www.eclipse.org/documentation>

[EAV] John Sinard, Practical Pathology Informatics: Demystifying informatics for the practicing anatomic pathologist, Springer, 2005. ISBN 978-0387280578