Tallinna Tehnikaülikool
Infotehnoloogia teaduskond
Arvutiteaduse instituut
Võrgutarkvara õppetool

Erinevatest algallikatest Eesti sündmuste automaatse kaevandamise ja ühtsele andmekoosseisule viimise laiendatav süsteem

Bakalaureusetöö

Üliõpilane: Oliver Kell
Üliõpilaskood: 112148
Juhendaja: Ago Luberg

Tallinn
2014

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

---------- ----------

(*kuupäev*) (*allkiri*)

Annotatsioon

Loodud töö on projektitöö, mille käigus luuakse süsteem, mis koondab Eestis toimuvad sündmused erinevatest veebiportaalidest ja andmeallikatest. Esmalt tõmmatakse andmed alla ning seejärel normaliseeritakse ühtsele struktuurile. Lõpptulemuseks on terviklik Eesti sündmuste andmebaas. Süsteem on laiendatav uute allikate lisamiseks. Projekti kasutatakse firma Postium positsioneerimissüsteemi projektis. Saadud andmebaasi kasutatakse ühes teises projektis, et ühendada positsioneerimisandmed toimuvate üritustega.

This work is made as a project. A system is created which collocates events in Estonia from different websites and data sources. First data is downloaded and then normalized to a unified structure. As a result, a united database of Estonian events is created. The system is expandable for adding new data sources. The database of gathered Estonian events will be used in another project to match positioning data with real events.

Sisukord

Lühendite ja mõistete selgitus

- **Andmeallikas/allikas**

Veebileht või –teenus milles on infot toimuvate sündmuste kohta

- **XML (*Extensible Markup Language*)**

Suvaliste andmete struktureerimiseks mõeldud märgistuskeel.

- **XSD (*XML Schema Definition*)**

XML dokumendi sisu definitsioon.

- **HTML (*HyperText Markup Languag*e)**

Enimlevinud veebilehtede märgendamise keel.
Töös kasutatud ka kui veebilehe lähtekoodi mõistena.

- **UML (*Unified Modeling Language*)**

Üldotstarbeline noteeringu keel tarkvara visualiseerimiseks.

- **MVC (*Model-View-Controller*)**

Tarkvara arhitektuurimuster veebirakenduste tegemiseks.

- **ORM (*Object Relational Mapping*)**

Objekt-relatsiooniline kaardistamine. Andmete viimine ühelt struktuurilt teisele objektorienteeritud programmeerimiskeeltes(siin töös andmebaasist Java objektideks ja vastupidi). Loob virtuaalse andmebaasi, mida saab kasutada rakenduse sees.

- **Scraper**

Programm, mis kaevab andmeid lehtedelt.

- **CSS (*Cascading Style Sheets*)**

Keel kirjeldamaks, kuidas HTML dokumente esitada.

1. Sissejuhatus

1.1 Töö tüüp ja taust

Loodud töö koondab Eestis toimuvad sündmused erinevatest veebiportaalidest ja andmeallikatest. Saadud andmed normaliseeritakse ühtsele kujule – tekib Eesti sündmuste andmebaas. Loodud süsteem läheb kasutusse kahes projektis.

Firma Positium tegeleb anonüümse positsioneerimise andmete kogumisega ja nende baasil erinevate teenuste pakkumisega. Neil on teada anonüümne mobiiliseadmete positsiooni informatsioon: koordinaadid ja ajatempel. Positsioneerimine sellisel kujul ei ole väga täpne, tegelik positsioon võib varieeruda 200 meetri jagu. Neil on soov tuvastada, miks mingil hetkel on mingil alal ebatavaliselt palju inimesi. Näiteks huvitab neid, kas tegemist võib olla ummiku või mõne sündmusega. Kui tegemist on sündmusega, tahaksid nad täpsemalt informatsiooni sündmuse kohta (mis sündmus, kus toimus jne). Selleks, et anda täpsem ülevaade, kus ja miks inimesed on käinud, kasutavad nad täiendavate andmeallikatena sotsiaalmeediat ja veebis olevat informatsiooni. Üks selline sisend antud projektile on kirjeldatud ja realiseeritud selles töös. Kui positsioneerimise andmetest on näha erandlik käitumine (mõnes kohas on võrreldes tavalisega rohkem inimesi), saab antud piirkonna kohta uurida täpsemat infot sündmuste andmebaasidest. Kui antud alale jääb vaid üks sündmus, võib järeldada, et see sündmus on inimeste arvu suurendanud. Sellisel puhul kahandatakse 200-meetrine ala konkreetse sündmuspaiga peale. Muidugi võib juhtuda, et samasse alasse jääb sama perioodi jaoks mitu erinevat sündmust. Selliselt juhul kasutatakse täiendavaid sisendandmeid muudest andmeallikatest (näiteks sotsiaalmeedia).

Sündmuste andmeid kogutakse ka projekti Sightsmap.com jaoks, mis koondab informatsiooni turismiobjektida kohta üle kogu maailma. Esialgu sisaldab Sightsmap.com andmebaas informatsiooni vaid staatiliste objektide (kirikud, väljakud, monumendid jne) kohta. Kuna aga antud rakendus võimaldab koostada reisiplaane, on projekti jaoks vaja sündmuste andmebaasi. Selliste kohalike veebilehtede süsteemne tõmbamine on väga veebilehe-, asukoha- ja keelepõhine. Süsteemi luues on arvestatud sellega, et sama süsteemiga oleks võimalik andmeid tõmmata ka muu riigi ja muu keele jaoks.

1.2 Töö eesmärgid

Järgnevalt on loetletud tööle püstitatud üldised eesmärgid:

1. Allikatest kõikide andmete allatõmbamine andmebaasi

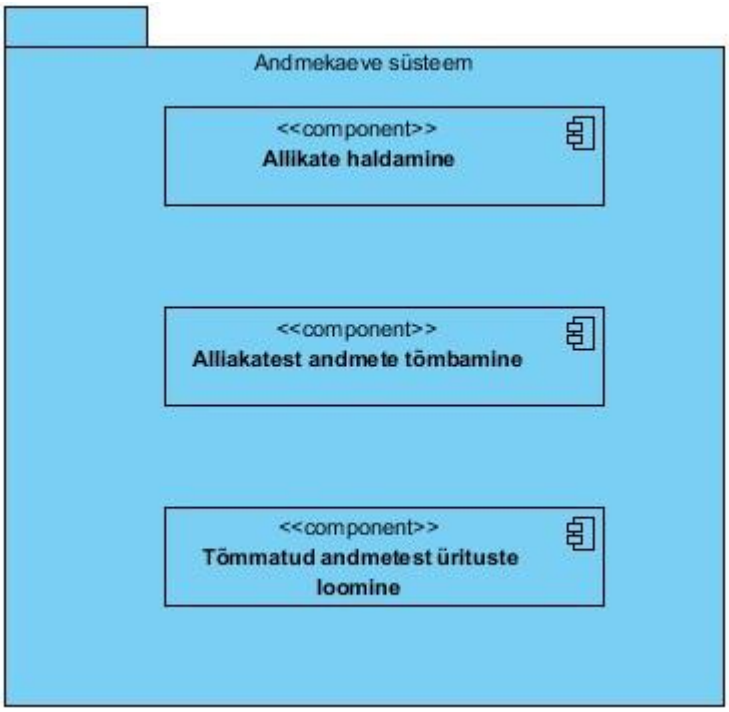
2. Tõmmatud andmetest vajaliku info välja liigendamine ja töötlemine meie andmebaasile sobivale struktuurile
3. Haldamisliides allikatele
4. Laiendatav süsteem uute allikate lisamiseks mõistliku vaevaga

1.3 Töö metoodika

Töö on kirjutatud eesti keeles, kuid tööl põhinev rakenduse kasutajaliides on loodud inglise keeles. Seetõttu ka teatud joonistel ja näidetel on kasutuskeeleks inglise keel.
Töös kasutatakse diagrammide tegemiseks UML modelleerimisformaati, diagrammid on realiseeritud kasutades programmi *Visual Paradigm for UML*.

2. Süsteemi disain

2.1 Süsteemi alamsüsteemid



Joonis 1 – süsteemi komponentdiagramm

Loodava andmekaeve süsteemi võib jagada kolmeks alamsüsteemiks, mida on kujutatud joonisel 1.

2.1.1 Alamsüsteemide kirjeldused

Allikate haldamine

Alamsüsteem tegeleb andmeallikate haldamisega ja samuti järgmise kahe alamsüsteemi protsesside käimapanemisega. Sellel alamsüsteemil on ainukesena kasutajaliides ja süsteemi haldajal on läbi selle võimalus juhtida kogu süsteemi protsesse.

Allikatest andmete tõmbamine

Alamsüsteem tegeleb veebilehtedest andmete tõmbamisega andmebaasi. Tõmmatakse infot ürituste kohta – terviklik detailvaate (leht ühe kindla ürituse kohta) lähtekood ürituse infoga ja muu info, mis saadakse selle ürituse kohta väljast poolt ürituse detailvaadet.

Tõmmatud andmetest ürituste loomine

Alamsüsteem tegeleb eelmises alamsüsteemis saadud andmete liigendamisega. Tõmmatud HTML lehtedest leitakse ürituse kohta kõik andmed, mis viiakse siis ühtsele struktuurile ja salvestatakse meie andmebaasi üritustena.

2.2 Allikate haldamine

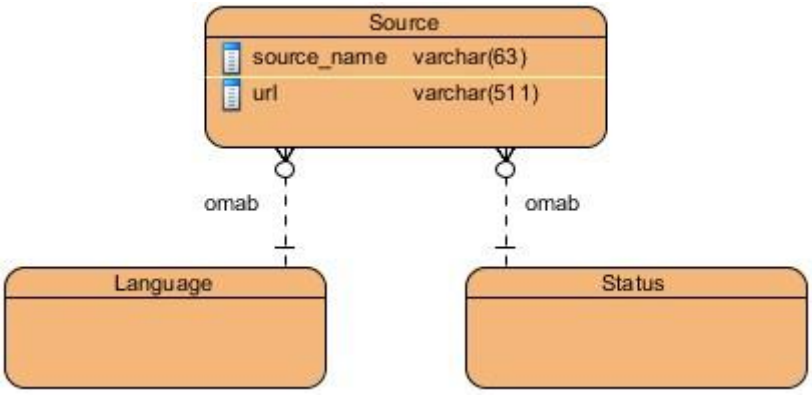
2.2.1 Alamsüsteemi eesmärgid

Allikate haldamise alamsüsteemil on järgmised eesmärgid:

- Allikaid oleks võimalik lisada, vaadata, muuta ja kustutada
- Allikatel oleks võimalik käivitada tõmbamis- ja liigendamisprotsesse
- Kasutajaliides haldamiseks oleks mugav ja lihtne

2.2.2 Allika kontsept

Allikas(andmeallikas) on andmekaeve süsteemi keskne osa. Allikas kirjeldab ühte kindlat veebilehte või –teenust, kust andmeid kaevandatakse.

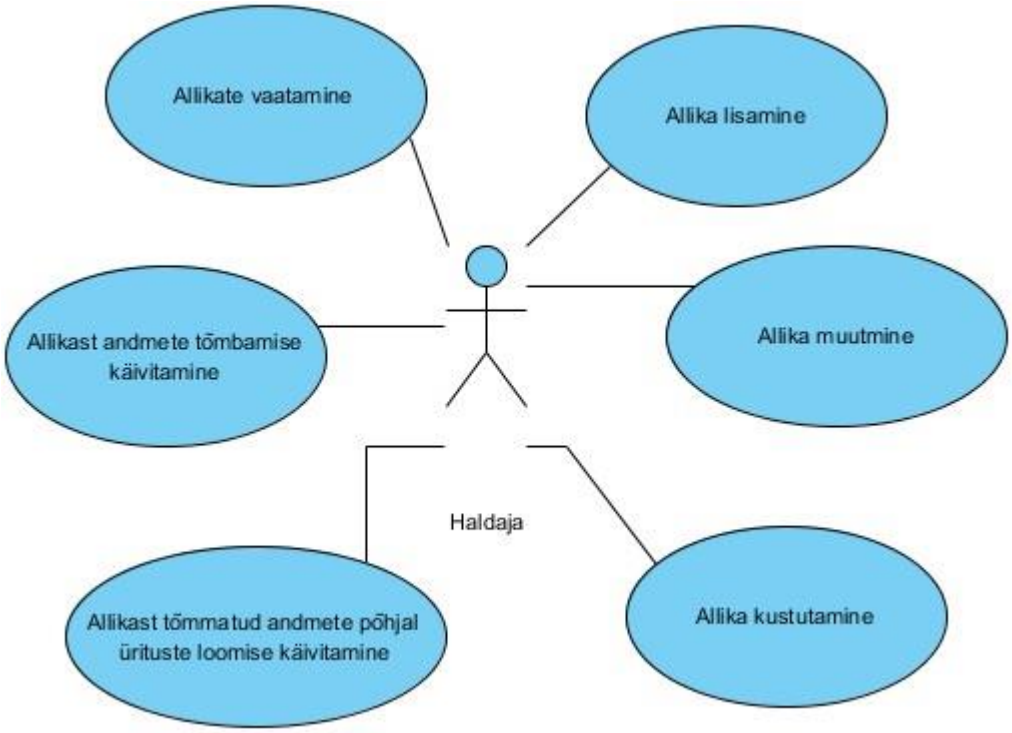


Joonis 2 – allika olemi-suhte diagramm

Joonis 2 kirjeldab allikat, nagu jooniselt näha on igal allikal olemas keel ja staatus.
Keel näitab, mis keelse veebilehega on tegemist ja mis keelset informatsiooni saadakse lehelt. Staatus näitab allika hetkelist seisundit. Kui allikaga käivitatakse mingi protsess, siis see ka muutub.

Täpsemat tabelite struktuuri saab vaadata andmebaasidiagrammilt peatükis 2.5.

2.2.3 Allikate haldamise kasutusjuhud



Joonis 3 – allikate haldamise kasutusjuhtude diagramm

Kasutusjuht 1: Allikate vaatamine
Kirjeldus: süsteemi haldajal on võimalus vaadata kõiki allikaid listina. Listis on kirjeldatud allika nimi, lehe aadress, keel, staatus ja viimase tõmbamise ning ürituste loomise aeg. Samuti saab käivitada erinevaid protsesse allikatega(vt kasutusjuht 5 ja 6).

Kasutusjuht 2: Allika lisamine
Kirjeldus: süsteemi haldajal on võimalus lisada uut allikat. Uue allika lisamisel tuleb märkida allika nimi, lehekülje aadress ja keel. Allika lisamisel määratakse talle automaatselt tegevuseta staatus.

Kasutusjuht 3: Allika muutmine
Kirjeldus: süsteemi haldajal on võimalus olemasoleva allika andmeid muuta. Allika muutmiseks on põhjust, kui uueneb allika koduleht või nimi.

Kasutusjuht 4: Allikate kustutamine
Kirjeldus: süsteemi haldajal on võimalus kustutada olemasolevat allikat. Kustutamisel kustutatakse allikas jäädavalt. Kustutamine on kinnitamisega – enne kustutamist küsitakse teistkordset nõusolekut. Kasutusjuhuks on vajadus, kui allikas lakkab eksisteerimast. Allikast tõmmatud andmed jäävad kustutamisel alles.

Kasutusjuht 5: Allikast andmete tõmbamise käivitamine
Kirjeldus: süsteemi haldajal on võimalus käivitada andmete tõmbamist allikast. Võimalus selleks on allikate listivaates. Tõmbamise käivitamisel muudetakse allika staatust ja see kajastub ka allikate listis. Andmete tõmbamise protsessi on võimalik alustada ainult siis kui allikas on tegevuseta olekus. Andmete tõmbamise protsess on asünkroonne – samaaegselt võib käivitada mitu tõmbamist.

Kasutusjuht 6: Allikast tõmmatud andmete põhjal ürituste loomise käivitamine
Kirjeldus: süsteemi haldajal on võimalus käivitada ürituste loomist. Ürituste loomise eelduseks on allikast tõmmatud andmed. Ürituste loomise käivitamisel muudetakse allika staatust ja see kajastub ka allikate listis. Protsessi on võimalik käivitada ainult siis, kui allikas on tegevuseta olekus. Protsess on asünkroonne – võib samaaegselt käivitada mitu ürituste loomist.

2.3 Allikatest andmete tõmbamine

2.3.1 Alamsüsteemi eesmärgid

Allikatest andmete tõmbamise alamsüsteemil on järgmised eesmärgid:

- Ürituste detailvaadete terviklike lehtede alla tõmbamine
- Ürituste kohta asuva informatsiooni, mis asub väljaspool detaillehti, talletamine
- Ühtse struktuuri loomine, mis teeks uutest allikatest tõmbamise mugavaks

2.3.2 Allika tüüpstruktuur

Allikatest tõmbamise süsteemi ülesehituse aluseks on leitud tüüpiline allika struktuur. See tüüpstruktuur leiti võrreldes visuaalselt erinevaid allikate lehti.
Tüüpiline struktuur lehel on nimekiri üritustest(tavaliselt nimekiri üle mitme lehe) ja iga nimekirja element omab infot ürituse kohta, alati on olemas link detaillehele. Mõnikord pole võimalik detailvaate lehelt leida kogu infot, mis on olemas ürituste nimekirjas, sellel põhjusel tuleb ka sealt info salvestada.
Juhul, kui üritused on üle mitme lehekülje, siis on lehel ka link, kuidas minna järgmisele lehele(viimasel lehel puudub).

Tüüpilise struktuuri näide:

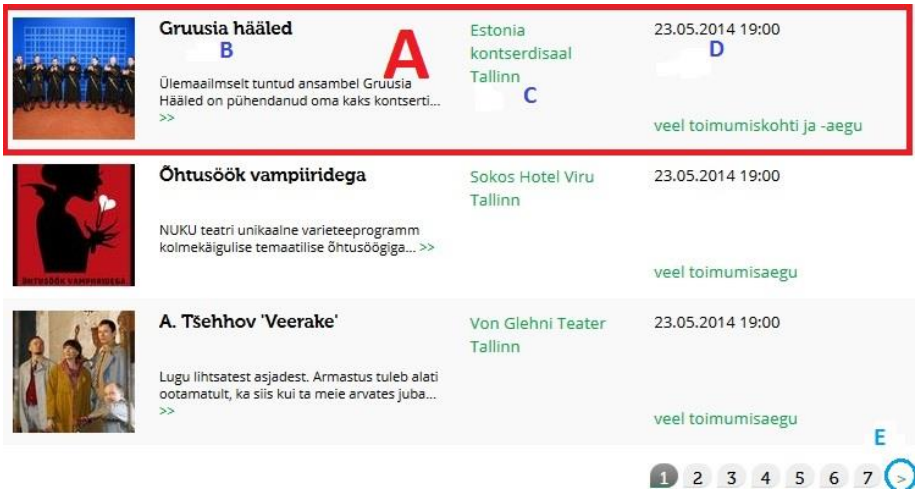
A – Ürituse blokk

B – Link järgmisele lehele
Ürituse pealkiri

C – Ürituse toimumiskoht

D – Ürituse toimumisaeg

E – Link järgmisele lehele



Joonis 4 – tüüpilise allika struktuur

Osad A, B ja E olid olemas igas vaadatud allikas. Nimekirjas olev info ürituse kohta võib varieeruda sõltuvalt allikast. Joonisel 4 on nimekirjas toimumiskoht(C) ja toimumisaeg(D).

2.3.3 Allikast tõmbamise konfiguratsioonifail

Vastavalt eelmises peatükis kirjeldatud struktuurile kirjutatakse iga allika kohta konfiguratsioonifail, mis kirjeldab elementide asukohti lehel.

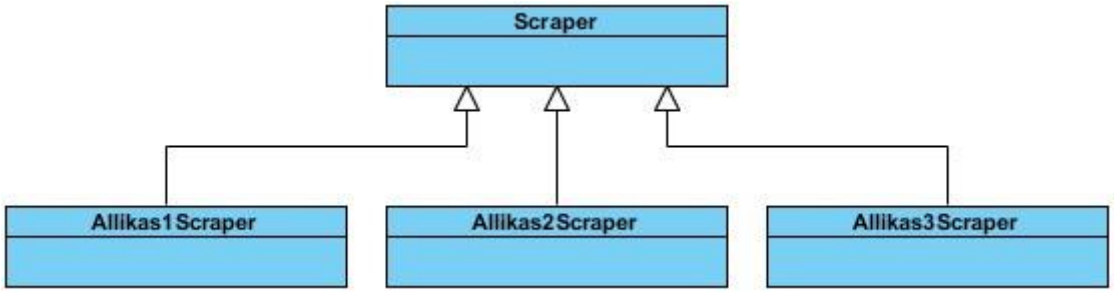
Konfiguratsioonifaili hoiab järgmisi elemente:

- URL kust alustatakse tõmbamist – see peaks olema leht, kus asub ürituste list
- Üritusblokkide asukoht
- Üritusblokki sees olevate elementide asukohad(asukoht bloki suhtes)
    - Link detailvaate lehele
    - Ürituse nimi
    - Ürituse toimumiskoht
    - Ürituse toimumisaeg
- Järgmisele lehele minev link

Kui allikal puuduvad teatud eelpool mainitud osad, siis võib jätta vastava elemendi tühjaks. Kuna kogu ülejäänud loogika tõmbamisprotsessi juures on samasugune, siis paljudel juhtudel piisabki uue allika lisamisel ainult konfiguratsioonifaili kirjutamisest. Selline struktuur muudab süsteemi kiireks ja mugavaks.

2.3.4 Tüüpstruktuurist erinevad lehed

Sarnasustest vaatamata on teatud allikaid, mille kõik osad ei vasta tüüpstruktuurile. Taolised olukorrad vajavad individuaalset lähenemist allikale.
On võimalus luua igale allikale oma *scraper*, mis laiendavad tüüpklassi *Scraper*. Nendes on võimalus üle kirjutada soovitud funktsionaalsus.



Joonis 5 - laiendatava *scraperi* klassidiagramm

Kogu loodud süsteemi funktsionaalsust on võimalik laiendada ja üle kirjutada. Selline lähenemine tagab selle, et igat andmeallikat on võimalik realiseerida.

2.3.5 Tõmmatud andmete salvestamine

Kõik tõmmatud andmed, kaasa arvatud terviklik HTML leht, salvestatakse andmebaasi, tabelisse *downloaded_event*. Täpsemat tabeli struktuuri saab vaadata andmebaasidiagrammilt peatükis 2.5.

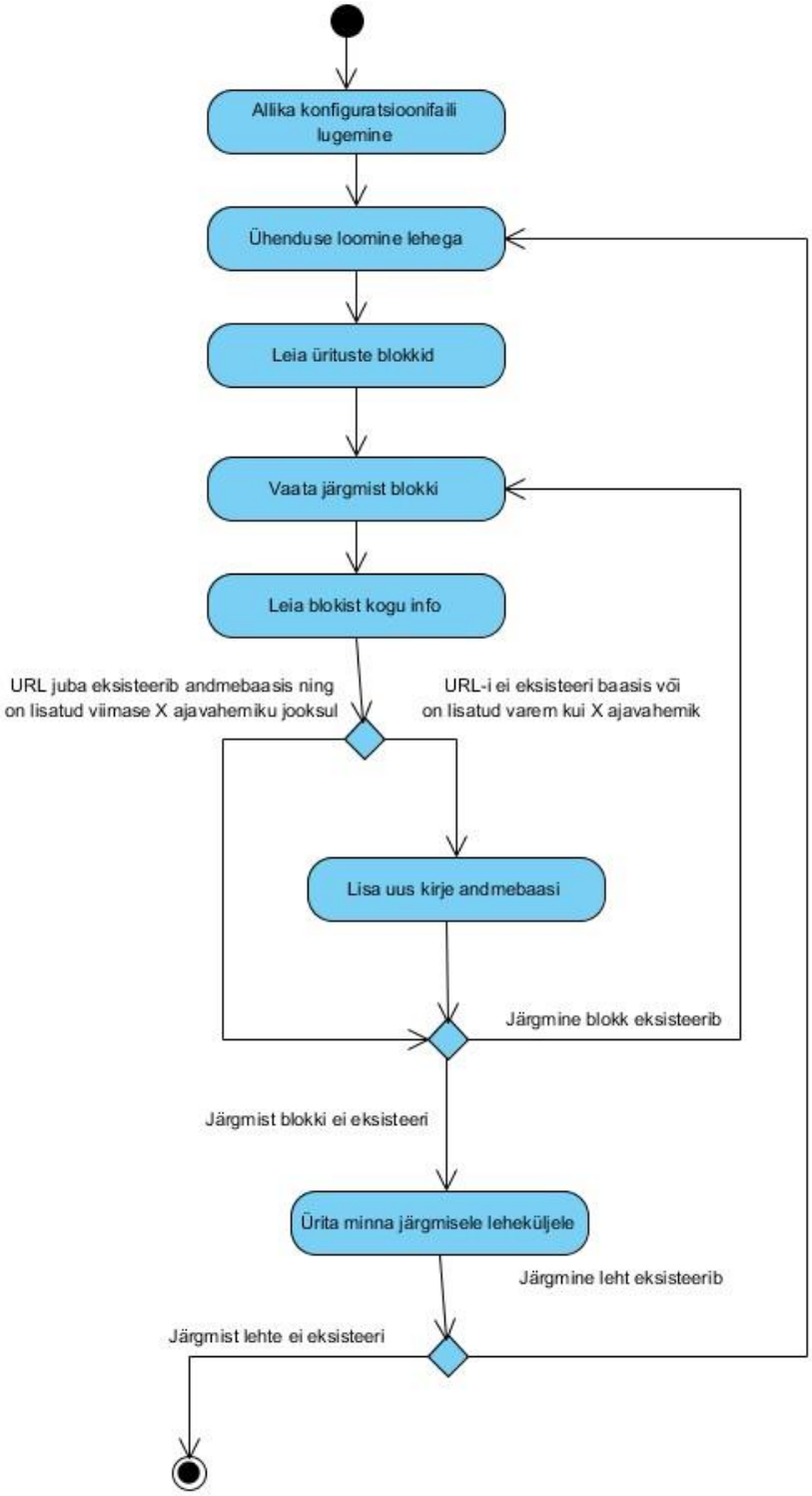
2.3.6 Korduvate andmete käsitlemine

Kui tõmmata ühest allikast mitu korda andmeid hakkavad andmed korduma. Samuti võib ühes allikas olla mitu üritust, mis viitavad samale lehele.
Et takistada suures koguses duplikaatürituste teket andmebaasi, siis andmete salvestamise loogika on järgmine:

- Kui leitakse üritus, mille linki pole andmebaasis, siis üritus salvestatakse
- Kui leitakse üritus, mille link on juba andmebaasis olemas, siis käitutakse järgmiselt:
    - Juhul kui sündmus on viimase X ajavahemiku jooksul loodud, siis jäetakse vahele
    - Juhul kui sündmus on hiljem loodud, kui X ajavahemik, siis lisatakse baasi täiesti uus kirje

2.3.7 Allikatest tõmbamise tööprotsess

Eelmiste alampeatükkide sisu arvestades saab koostada üldise tööprotsessi kirjelduse, mis on esitatud joonisel 6.



Joonis 6 – allikatest tõmbamise tegevusdiagramm arvestamata veaolukordade teket

Joonisel 6 pole arvestatud erinevate veaolukordade teket. Reaalselt võib igas faasis tekkida veaolukordi, mis tuleb logida. Veaolukordadeks programmis võivad olla näiteks:

- Leht on maas ja ühendust ei saa luua
- Lehe struktuur on muutunud ja elemente ei suudeta leida
- Ei suudeta luua ühendust andmebaasi
- Konfiguratsioonifaili pole kirjutatud allikale

2.4 Tõmmatud andmetest ürituste loomine

2.4.1 Alamsüsteemi eesmärgid

Alamsüsteemil on järgnevad eesmärgid:

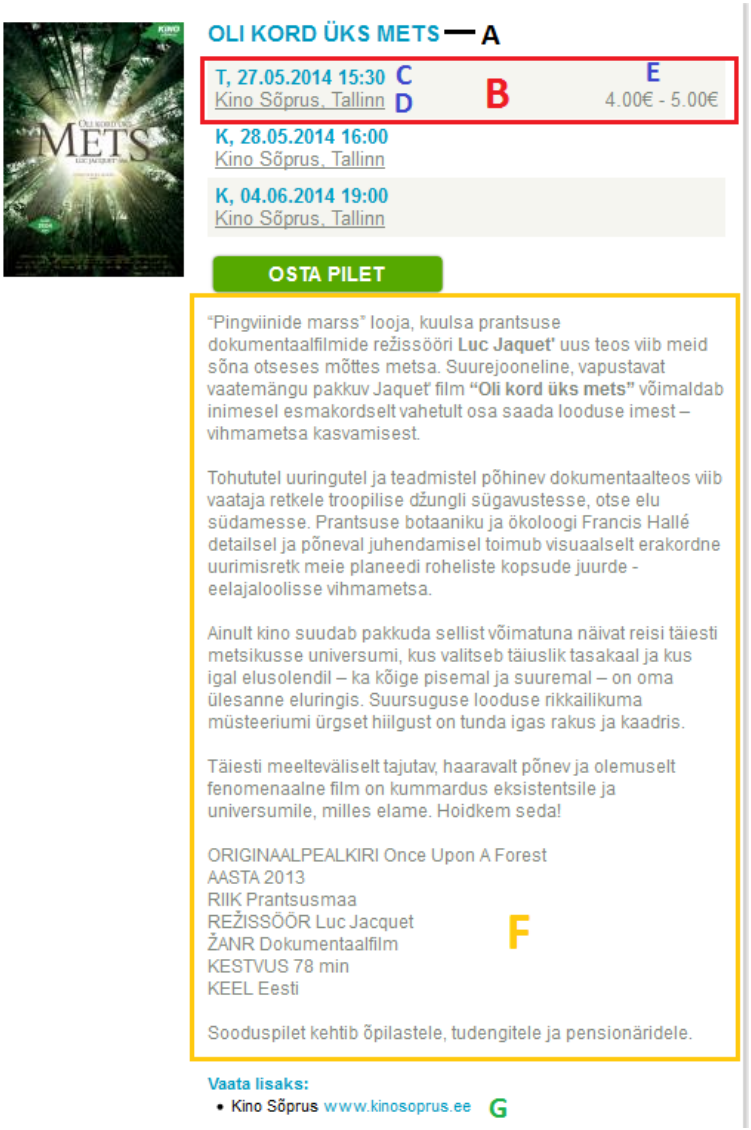
- Erinevatest allikatest tõmmatud ürituste viimine ühtsele struktuurile ja salvestamine
- Laiendatav süsteem uute allikate lisamiseks

2.4.2 Ürituse detaillehe tüüpstruktuur

Sarnaselt andmete tõmbamise loogikale on ka siin disaini aluseks lehe tüüpstruktuur.
Detailleht koosneb tüüpiliselt üldisest informatsioonist lehe ürituse kohta nagu nimi ja kirjeldus, mille juures on nimekiri ürituse toimumistest. Toimumised omavad infot koha, kellaaja ja hinna kohta. Sõltuvalt üritusest ja toimumisest võib informatsiooni olla vähem või rohkem.

Tüüpilise allika struktuuri näide on esitatud joonisel 7:

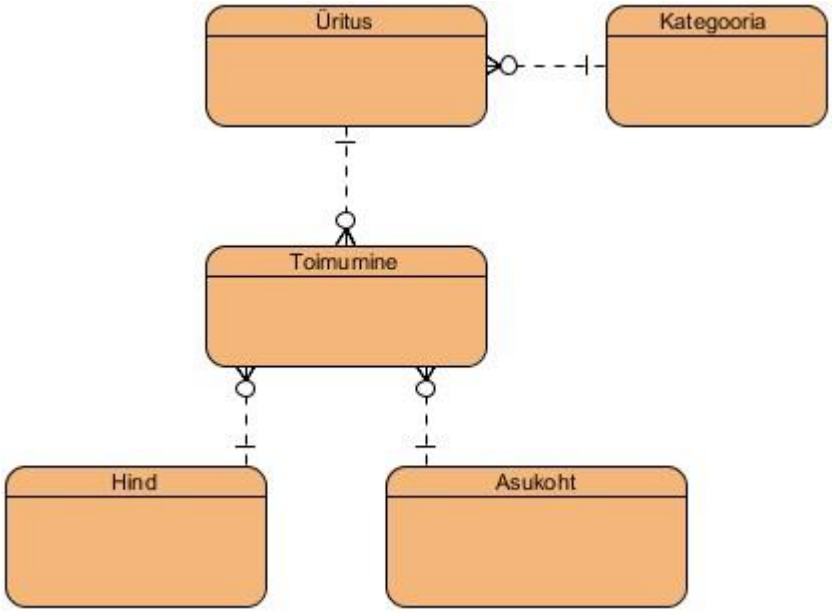
A – Ürituse pealkiri
B – Ürituse toimumise blokk
C – Toimumise aeg
D – Toimumise koht
E – Toimumise hind
F – Ürituse informatsioon
G – Link lisainfoks



Joonis 7 – allika detaillehe tüüpstruktuur

2.4.3 Ürituse struktuur

Võttes arvesse eelmises peatükis kujutatud tüüpilist ürituste struktuuri allikate lehel loome struktuuri ürituste talletamiseks andmebaasi.



Joonis 8 – ürituse kontseptuaalne mudel

Struktuuri loomise juures on arvesse võetud veel potentsiaalseid täiendusi tulevikus, näiteks eraldi andmete kaevandamine lehtedest, mis hoiavad informatsiooni asukoha kohta. Antud süsteemi realiseerimiseks võiks ka toimumise, asukoha ja hinna tabelid liita, kuid need on loodud eraldi just potentsiaalseteks laiendusteks.

Kontseptuaalse mudeli alusel luuakse ka andmebaasitabelid, mida on võimalik vaadata peatükis 2.5.

2.4.4 Konfiguratsioonifail

Konfiguratsioonifail hoiab ürituse detaillehel olevate elementide asukohti.
Konfiguratsioonifailis peaksid olema kajastatud järgmiste elementide asukohad:

- Ürituse nimi
- Ürituse kategooria
- Ürituse kirjeldus
- Link lisainformatsiooni jaoks
- Toimumisblokk
    - Toimumise hind

    - Toimumise asukoht
    - Toimumise kuupäev

Lisaks eelnenud tüüpinformatsioonile võib teatud allikatel leidud ka järgnevat infot:

- Toimumiskoha aadress
- Toimumiskoha koordinaadid
- Ürituse alguskellaaeg
- Ürituse lõppemiskellaaeg

2.4.5 Tüüpstruktuurist erinevad detaillehed

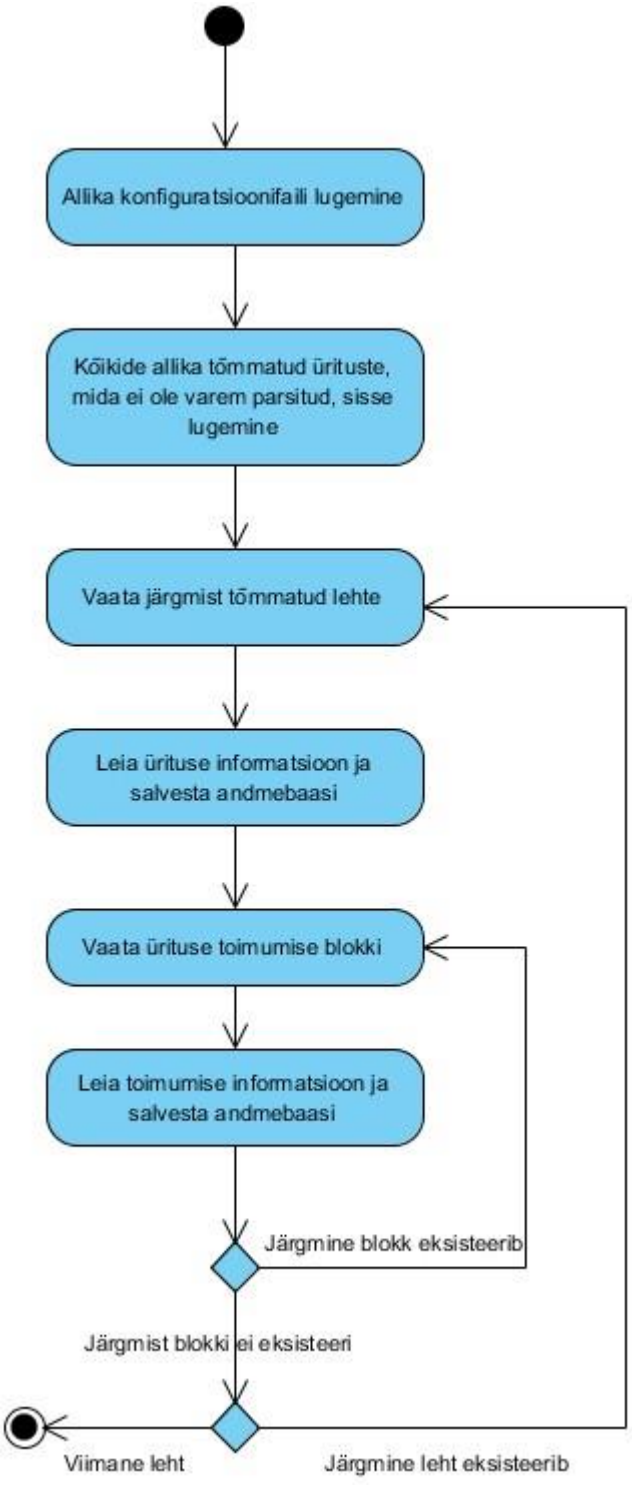
Sarnaselt peatükis 2.3.4 käsitletule võib ka detaillehtede liigendamisel tekkida olukordi, mis nõuavad individuaalset lähenemist. Siin olukorras kehtib identne loogika – on võimalik teatud funktsionaalsust üle kirjutada.

2.4.6 Andmete salvestamine

Kõik leitud informatsioon salvestatakse andmebaasi. Kõik väljad mille kohta informatsiooni ei leita jäetakse tühjaks. Täpset andmetabelite struktuuri on võimalik vaadata peatükis 2.5.
Käesolev töö ei tegele ürituste uuendamisega. Samuti ei tegele töö erinevatest allikatest leitud ürituste ühendamise funktsionaalsuse loomisega. Andmebaasi võib sattuda duplikaate ühest ja samast üritusest.

2.4.7 Ürituste loomise tegevusdiagramm

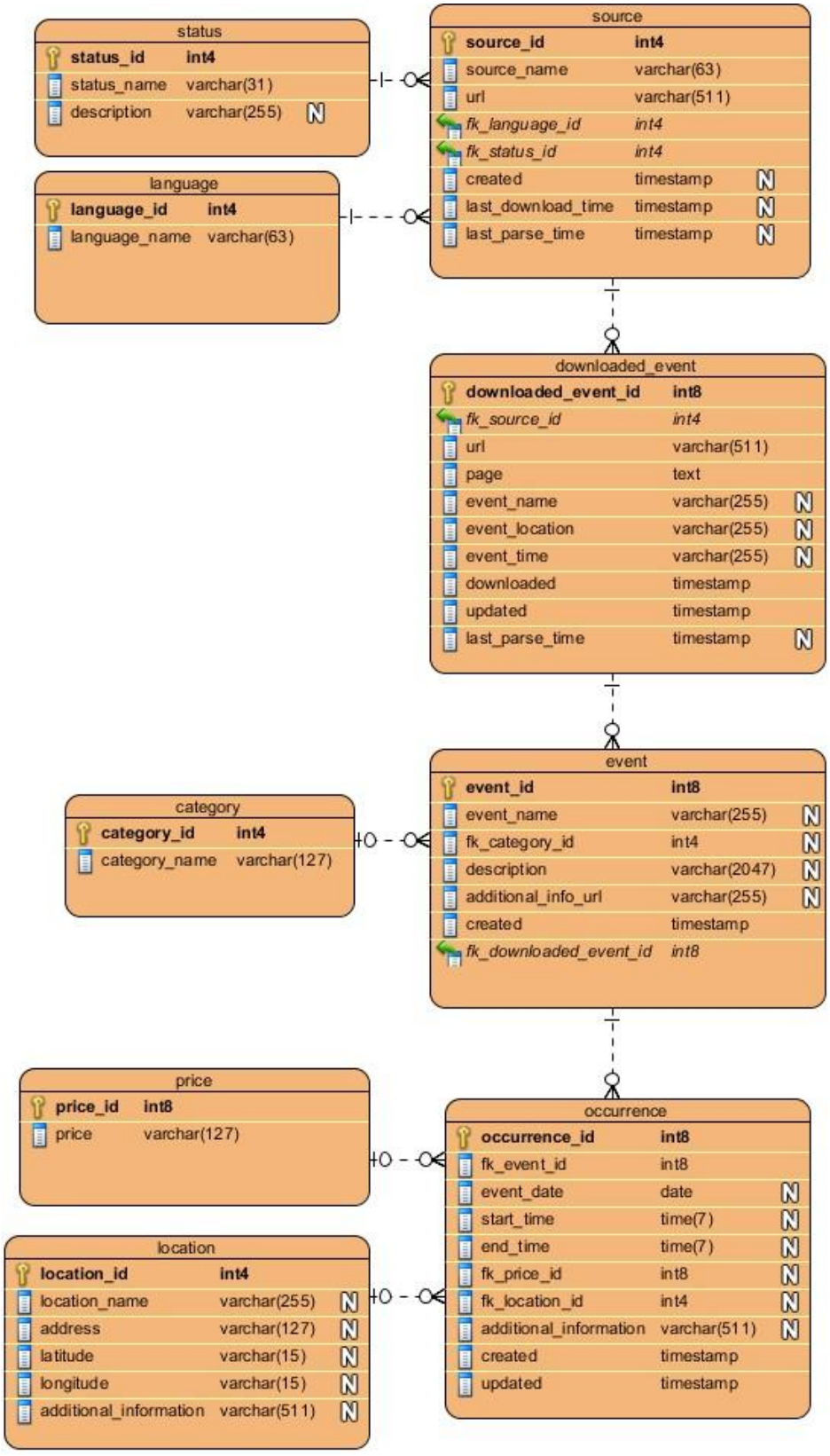
Eelmistes sektsioonides kirjeldatud alamsüsteem on kujutatud joonisel 9.



Joonis 9 – ürituste loomise tegevusdiagramm

2.5 Andmebaas

2.5.1 Andmebaasi diagramm



Joonis 10 – andmebaasidiagramm

2.5.2 Tabelite kirjeldused

SEQ – lühend tähistamaks unikaalseid automaatselt genereeritud täisarve sammuga 1.

status
Tabel, mis kirjeldab staatust.

Veeru nimi	Veeru tüüp	Vaikeväärtus	Kohustuslik	Veeru kirjeldus
status_id	täisarv	SEQ	Jah	Unikaalne identifikaator
status_name	string		Jah	Staatuse nimi, see on nähtav ka kasutajaliideses.
description	string			Juhul kui tekib vajadus staatust täpsemalt kirjeldada.

Primaarvõti – *status_id*
Unikaalsuskitsendused

- *status_name*

language
Tabel, mis kirjeldab keelt.

Veeru nimi	Veeru tüüp	Vaikeväärtus	Kohustuslik	Veeru kirjeldus
language_id	täisarv	SEQ	Jah	Unikaalne identifikaator
language_name	string		Jah	Keele nimi

Primaarvõti – *language_id*
Unikaalsuskitsendused

- *language_name*

source
Tabel, mis kirjeldab allikat.

Veeru nimi	Veeru tüüp	Vaikeväärtus	Kohustuslik	Veeru kirjeldus
source_id	täisarv	SEQ	Jah	Unikaalne identifikaator
source_name	string		Jah	Allika nimi
url	string		Jah	Allika kodulehe veebiaadress

fk_language_id	täisarv	1	Jah	Viide allika keelele
fk_status_id	täisarv	1	Jah	Viide allika staatusele
created	ajatempel	Praegune kellaaeg	Jah	Kellaaeg, millal allikas loodi
last_download_time	ajatempel			Kellaaeg, millal viimati lõpetati andmete tõmbamine
last_parse_time	ajatempel			Kellaaeg, millal viimati lõpetati ürituste loomine

Primaarvõti – *source_id*
Välisvõtmed

- *fk_language_id* viitab veerule *language*.*language_id*
- *fk_status_id* viitab veerule *status*.*status_id*

Unikaalsuskitsendused

- *source_name, url, fk_language_id*

downloaded_event

Tabel, mis hoiab allikast tõmmatud andmeid.

Veeru nimi	Veeru tüüp	Vaikeväärtus	Kohustuslik	Veeru kirjeldus
downloaded_event_id	suur täisarv	SEQ	Jah	Unikaalne identifikaator
fk_source_id	täisarv		Jah	Viide allikale, mille kohta need andmed käivad
url	string		Jah	Lehe aadress, kust tõmmati detailleht
page	tekst		Jah	Terviklik ürituse kohta käiva detaillehe HTML
event_name	string			Ürituse nimi, mis leiti allika listivaatest
event_location	string			Ürituse toimumiskoht, mis leiti allika listivaatest
event_time	string			Ürituse toimumisaeg, mis leiti allika listivaatest
downloaded	ajatempel	Praegune kellaaeg	Jah	Kellaaeg, millal kirje loodi
updated	ajatempel	Praegune kellaaeg	Jah	Kellaaeg, millal kirje loodi või seda muudeti

last_parse_time	ajatempel			Kellaaeg, millal viimati sellest kirjest üritus loodi

Primaarvõti – *downloaded_event_id*
Välisvõtmed

- *fk_source_id* viitab veerule *source.source_id*

category
Tabel, mis kirjeldab ürituse kategooriat.

Veeru nimi	Veeru tüüp	Vaikeväärtus	Kohustuslik	Veeru kirjeldus
category_id	täisarv	SEQ	Jah	Unikaalne identifikaator
category_name	string (127)		Jah	Kategooria nimi

Primaarvõti – *category_id*

event
Tabel, mis kirjeldab üritust.

Veeru nimi	Veeru tüüp	Vaikeväärtus	Kohustuslik	Veeru kirjeldus
event_id	täisarv	SEQ	Jah	Unikaalne identifikaator
event_name	string (255)			Ürituse nimi
fk_category_id	täisarv			Viide ürituse kategooriale
description	string (10000)			Kirjeldus üritusest
additional_info_url				Lehekülg lisainformatsiooniks ürituse kohta
created	ajatempel	Praegune kellaaeg	Jah	Kellaaeg, millal üritus loodi
fk_downloaded_event_id	suur täisarv			Viide tõmmatud lehele, mille alusel üritus loodi

Primaarvõti – *event_id*
Välisvõtmed

- *fk_category_id* viitab veerule *category.category_id*
- *fk_downloaded_event_id* viitab veerule *downloaded_event.downloaded_event_id*

price
Tabel, mis kirjeldab ürituse hinda.

Veeru nimi	Veeru tüüp	Vaikeväärtus	Kohustuslik	Veeru kirjeldus
price_id	suur täisarv	SEQ	Jah	Unikaalne identifikaator
price	string (127)		Jah	Hind

Primaarvõti – *price_id*

location
Tabel, mis kirjeldab ürituse asukohta.

Veeru nimi	Veeru tüüp	Vaikeväärtus	Kohustuslik	Veeru kirjeldus
location_id	täisarv	SEQ	Jah	Unikaalne identifikaator
location_name	string (255)			Kohanimi
address	string (127)			Koha aadress
latitude	string (15)			Koha laiuskraad
longitude	string (15)			Koha pikkuskraad
additional_information	string (511)			Koha kohta käiv lisainformatsioon

Primaarvõti – *location_id*

occurrence
Tabel, mis kirjeldab ürituse toimumist.

Veeru nimi	Veeru tüüp	Vaikeväärtus	Kohustuslik	Veeru kirjeldus
occurrence_id	suur täisarv	SEQ	Jah	Unikaalne identifikaator
fk_event_id	suur täisarv		Jah	Viide üritusele, mille toimumist kirje kirjeldab
event_date	string (255)			Ürituse toimumise kuupäev(võib sisaldada ka kellaaega)
start_time	string (63)			Ürituse algusaeg
end_time	string (63)			Ürituse lõppaeg
fk_price_id	suur täisarv			Viide hinnale
fk_location_id	täisarv			Viide toimumiskohale

additional_information	string (511)			Lehekülg lisainformatsiooniks ürituse kohta
created	ajatempel	Praegune kellaaeg	Jah	Kellaaeg, millal kirje loodi
updated	ajatempel	Praegune kellaaeg	Jah	Kellaaeg, millal kirjet uuendati

Primaarvõti – *occurrence_id*
Välisvõtmed

- *fk_price_id* viitab veerule *price.price_id*
- *fk_location_id* viitab veerule *location.location_id*

3. Süsteemi realisatsioon

3.1 Kasutatud tehnoloogiad

3.1.1 Andmebaas

PostgreSQL 9.3
Valitud sai kuna süsteemi teised osad on loodud sama andmebaasisüsteemiga.

3.1.2 Programmeerimiskeel

Java 8
Isiklik kogemus oli suurim selle programmeerimiskeelega. Samuti tundsin erinevaid teeke, mis teeksid arendamise mugavamaks.

3.1.3 Server

Tomcat 8
Populaarseim server Java veebirakenduste jooksutamiseks.

3.1.4 Muu

Tehnoloogia:	Apache Maven 3.1.1
Koduleheküg:	http://maven.apache.org/
Lühiülevaade:	Tarkvara Java projektide automaatehitamiseks
Valiku põhjus:	Mugav projektis vajaminevate teekide haldamine Automaatne üles seadmine serverisse

Tehnoloogia:	Spring MVC 4.0.3
Koduleheküg:	http://spring.io/
Lühiülevaade:	Java raamistik veebirakenduste loomiseks
Valiku põhjus:	Varasem kogemus Sisse ehitatud MVC arhitektuuri printsiip, mis aitab rakenduse eri osasid lahus hoida

Rakenduse komponentide sidumine(„*autowiring*“)

Tehnoloogia:	Hibernate 4.3.5
Koduleheküg:	http://hibernate.org/orm/
Lühiülevaade:	Teek objekt-relatsioonilise kaardistamise jaoks
Valiku põhjus:	Varasem kogemus Andmebaasiloogika lahushoid äriloogikast

Tehnoloogia:	JSoup 1.7.3
Koduleheküg:	http://jsoup.org/
Lühiülevaade:	Vabavaraline Java teek veebilehtede tõmbamiseks ja liigendamiseks
Valiku põhjus:	Veebilehtede tõmbamise ja HTML-i liigendamiseks valmis loodud funktsionaalsus

Tehnoloogia:	log4j 1.2
Koduleheküg:	http://logging.apache.org/log4j/1.2/
Lühiülevaade:	Vabavaraline Java teek logimiseks
Valiku põhjus:	Populaarseim teek logimiseks Java jaoks

3.2 Andmevahetus andmebaasiga

Ühendus andmebaasi on loodud Springi ja Hibernate’i kasutades. Esmalt on vaja konfiguratsioonifaili kirjutada andmebaasi ühendumiseks vajalikud parameetrid – aadress, kasutajanimi, parool. Rakenduse käivitumisel loeb *Spring* selle automaatselt sisse ja loob ühenduse kasutades Hibernate’i.

Andmebaasiga andmevahetusel on kasutatud Hibernate raamistiku objekt-relatsiooniliseks kaardistamiseks. Lisaks on Hibernate’il kasutada omaenda päringukeel HQL (*Hibernate Query Language*), mida on kasutatud päringute tegemiseks.
Igale andmebaasitabelile loodi vastavad Java klassid, kus veerud on klassimuutujateks. Java mudelites kasutati JPA (*Java Persistence API*) annotatsioone, millega konfigureeriti mudel andmetabelile vastavaks.

```
@Entity
@Table( name="source" )
public class Source {

    private Language language;
```

```
    @ManyToOne
    @JoinColumn(name = "fk_language_id", referencedColumnName = "language_id",
nullable = false)
    public Language getLanguage() {
        return language;
    }

}
```

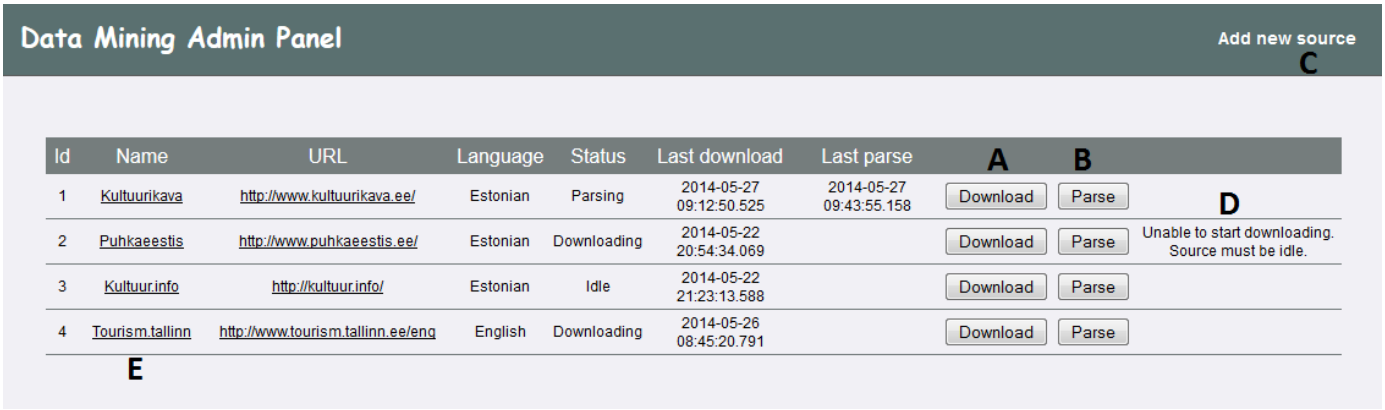
Koodinäide 1 – JPA annotatsioonide kasutamine välisvõtme realiseerimsieks

Koodinäide 1 iseloomustab Hibernate'i pakutavat ORM funktsionaalsust. Annotatsioon *@JoinColumn* viib vastavusse veeru *fk_language_id* tabelis *source* veeruga *language_id* tabelis *language*.

3.3 Allikate haldamine

Allikate haldamisliides on realiseeritud kasutades teeki Spring MVC.

3.3.1 Haldamisliidese peavaade



Joonis 11 – haldamisliidese peavaade

Joonisel 11 on kujutatud loodud haldamisliidese peavaadet, kus on informatsioon kõikide olemasolevate andmeallikate kohta. Tähtedega on märgitud haldajale huvi pakkuvad funktsionaalsused:

A – nupp, mis alustab andmete tõmbamist allikast
B – nupp, mis alustab ürituste loomist tõmmatud andmetest
C – link, mis viib uue allika lisamise lehele
D – vastus süsteemilt, kui üritatakse alustada protsessi punktide A või B all
E – link allika detailvaatesse, kust saab seda muuta ja kustutada

3.4 Allikatest tõmbamine

3.4.1 Konfiguratsioonifail

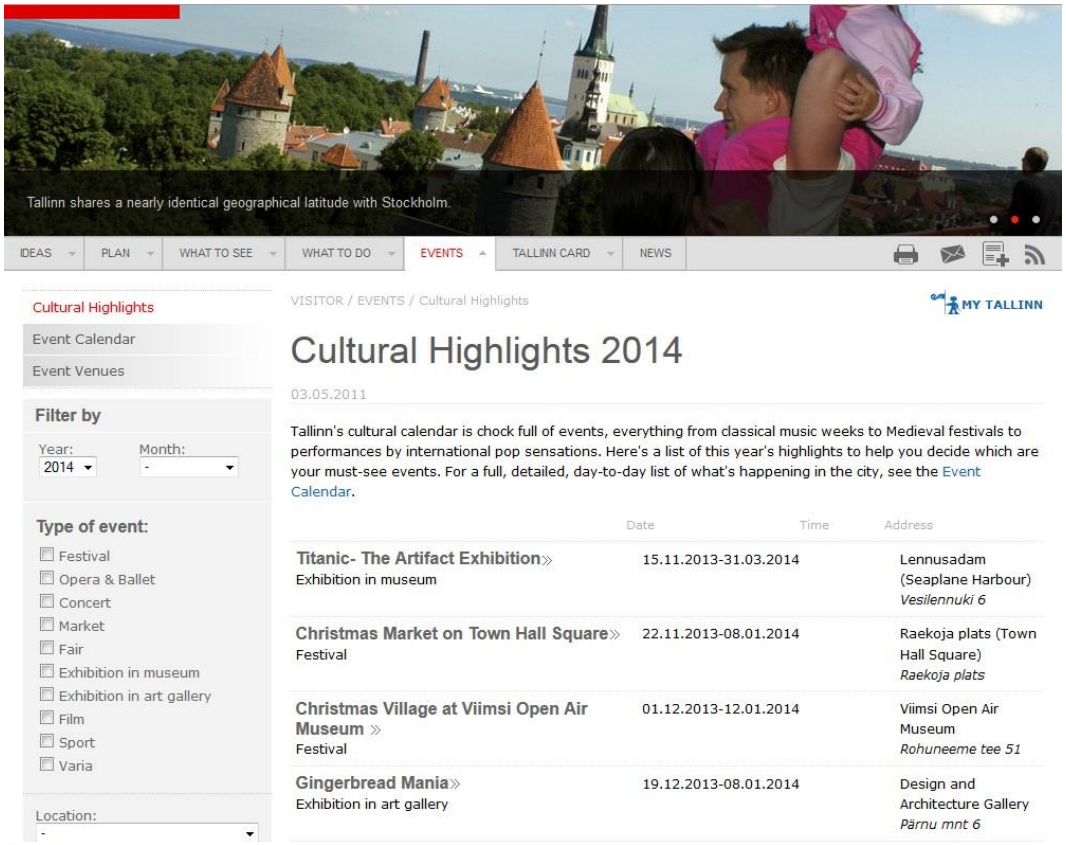
Konfiguratsioonifailide loodud struktuuri kirjeldab järgnev XSD fail:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="page">
        <xs:complexType>
            <xs:sequence>
                <xs:element type="xs:anyURI" name="startUrl"/>
                <xs:element type="xs:string" name="eventBlock"/>
                <xs:element type="xs:string" name="downloadLink"/>
                <xs:element type="xs:string" name="name"/>
                <xs:element type="xs:string" name="location"/>
                <xs:element type="xs:string" name="time"/>
                <xs:element type="xs:string" name="nextPage"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Koodinäide 2 – allikatest andmete tõmbamise konfiguratsioonifaili struktuur XSD failis

Kõik string tüüpi XML elemendid, ehk need, mis hoiavad endas teatud HTML elemendi asukohta tuleb kirjutada kasutatud teegi JSoup vastavas süntaksis. Süntaks on CSS süntaks, millele on lisatud teatud lisavõimalused. Minu loodud failides pole neid lisavõimalusi kordagi läinud vaja.
XML elemendid, mis viitavad linkidele(*downloadLink*, *nextPage*) on viited HTML elementidele, mis hoiavad endas *href* atribuuti lehe aadressiga.

Näide tourism.tallinn allikale loodud konfiguratsioonifailist lehe põhjal:




```
<table id="convention" class="sorting_table" cellspacing="0">
  <thead>
  <tbody>
    <tr class="linker" place="175252" event_type="24" href="#!4864_10298_24">
      <td class="first">
        <h3>
          <a href="#!4864_10298_24">
            Titanic- The Artifact Exhibition
            <img src="/static/imgs/sorting_table_arrows.png">
          </a>
        </h3>
        <span class="category">Exhibition in museum</span>
      </td>
      <td>15.11.2013-31.03.2014 </td>
      <td class="can_hide"></td>
      <td>
        Lennusadam (Seaplane Harbour)
        <br>
        <i>Vesilennuki 6</i>
      </td>
    </tr>
    <tr class="linker" place="26907" event_type="1" href="#!3096_10318_1">
    <tr class="linker" place="177833" event_type="1" href="#!3859_10323_1">
    <tr class="linker" place="175372" event_type="12" href="#!3269_10324_12">
```

Joonis 12 – tourism.tallinn allika leht ja osa lähtekoodist

Nagu jooniselt 12 näha siis on lehe struktuuriks tabel, mille keha iga rida kirjeldab ühte üritust.
Kuna rea veergudel pole klassi nimesid (välja arvatud esimesel veerul), siis tuleb kasutada n-nda järglase süntaksit. Kui lehe lähtekoodi täpselt järgida, siis saab loodud järgmine konfiguratsioonifail:

```
<?xml version="1.0" encoding="UTF-8"?>
<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:noNamespaceSchemaLocation="scraper.xsd">
    <startUrl>http://www.tourism.tallinn.ee/fpage/experience/cultural_highlights#
    </startUrl>
    <eventBlock>#convention tbody tr</eventBlock>
    <downloadLink>.first a</downloadLink>
    <name>.first a</name>
    <location>td:nth-child(4)</location>
    <time>td:nth-child(2)</time>
    <nextPage></nextPage>
</page>
```

Koodinäide 3 – tourism.tallinn allikast andmete tõmbamise konfiguratsioonifail

3.4.2 Üldine *scraper*

Põhiline andmete tõmbamise loogika on kõik loodud ühte klassi. Süsteemi tehnilisel realiseerimisel on kasutatud vabavaralist JSoup teeki. Seda on kasutatud nii veebilehtede allatõmbamisel kui ka HTML-i liigendamisel.
Andmete tõmbamisel on oluline mitte üle koormata allika serverit, sellepärast tehakse iga päringu vahel mõne sekundiline paus.

3.4.3 Alam-*scraper*, funktsionaalsuse ülekirjutamine

Scraperite laiendatavus on lahendatud Java pärimise ja polümorfismiga. Alam-*scraperi* funktsionaalsus päritakse tüüp-*scraperist*. Alamklassis saab muuta tüüpklassist pärinevat funktsionaalsust kirjutades üle vajalikud osad.

Vajadus tüüpklassi laiendamiseks tekib kultuurikava allikas. Kultuurikava on allikas, kus üritusi tuleb otsida kuupäevade järgi: tuleb kirjutada alguskuupäev ja lõppkuupäev. Sellist lehe aadressi pole võimalik staatiliselt konfiguratsioonifaili kirjutada.

```
public class Scraper{

    protected String getStartUrl(){
        return scraperPage.getStartUrl();
```

```
    }

}

public class KultuurikavaScraper extends Scraper{

    private static final String DATE_FORMAT_FOR_URL = "yyyyMMdd";

    @Override
    protected String getStartUrl(){
        String url = source.getUrl();
        url += "events/startdate=";
        url += DateAndTimeUtility.getCurrentDateAsString(DATE_FORMAT_FOR_URL);
        url += "/enddate=";
        url += DateAndTimeUtility.getFutureDateAsString(DATE_FORMAT_FOR_URL, 30);

        return url;
    }
}
```

Koodinäide 4 – alglehe aadressi tavaloogika ülekirjutamine

Koodinäites 4 on kirjutatud üle alglehe aadressi leidmise loogika. Tüüpjuhul võetakse see konfiguratsioonifailist. Näites aga luuakse dünaamiliselt aadress, kus on sees praegune kuupäev ja kuupäev 30 päeva pärast.

3.5 Lehtedest andmete liigendamine

3.5.1 Konfiguratsioonifail

Konfiguratsioonifailile loodud struktuuri kirjeldab järgnev XSD fail:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
           xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="event">
        <xs:complexType>
            <xs:sequence>
                <xs:element type="xs:string" name="name"/>
                <xs:element type="xs:string" name="description"/>
                <xs:element type="xs:string" name="additional"/>
                <xs:element type="xs:string" name="occurrence_block"/>
                <xs:element type="xs:string" name="occurrence_date"/>
                <xs:element type="xs:string" name="occurrence_start"/>
                <xs:element type="xs:string" name="occurrence_end"/>
                <xs:element type="xs:string" name="occurrence_additional"/>
                <xs:element type="xs:string" name="location_name"/>
                <xs:element type="xs:string" name="location_address"/>
                <xs:element type="xs:string" name="location_latitude"/>
```

```
                <xs:element type="xs:string" name="location_longitude"/>
                <xs:element type="xs:string" name="location_additional"/>
                <xs:element type="xs:string" name="price"/>
                <xs:element type="xs:string" name="category"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Koodinäide 5 – ürituse detaillehest liigendamise konfiguratsioonifaili struktuur XSD failina

Koodinäites 5 on tõmmatud detaillehtede liigendamiseks loodava konfiguratsioonifaili struktuur.
Sarnaselt allikatest andmete tõmbamisele tuleb konfiguratsioonifailis määrata veebilehe elementide asukohad, kasutades JSoup süntaksit (vt kirjeldust peatükis 3.3.1).
Kõik asukohad, mis asuvad toimumise blokki (vt 2.3.2) sees tuleb kirjutada bloki suhtes. Tegevusbloki sees asuvad väljad *occurrence_date* elemendist kuni *location_additional* elemendini.

Ürituse nime, toimumisaega ja/või -kohta võib olla soovi võtta tõmmatud andmete tabelist. Selline olukord on vajalik, kui detaillehelt ei saa kätte informatsiooni, mida saab kätte veebilehe ürituste listivaatest.
Taoliseks olukorraks on loodud kokkuleppeline väärtus, mille sisestamisel seda ka tehakse. Kirjutades XML elementi teksti DB, ignoreeritakse detaillehte selle elemendi suhtes ja väärtus võetakse tabelist *downloaded_event*.

Näide kultuurikava allikale loodud XML konfiguratsioonifailist:

```
<?xml version="1.0" encoding="UTF-8"?>
<event xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="parser.xsd">
    <name>DB</name>
    <description>.evright .description</description>
    <additional>.evright .links a</additional>
    <occurrence_block>.evright .times .time</occurrence_block>
    <occurrence_date>.row:nth-child(1) .kn_timestamp</occurrence_date>
    <occurrence_start></occurrence_start>
    <occurrence_end></occurrence_end>
    <occurrence_additional></occurrence_additional>
    <location_name>.row:nth-child(2) a</location_name>
    <location_address></location_address>
    <location_latitude></location_latitude>
    <location_longitude></location_longitude>
    <location_additional></location_additional>
    <price>.row:nth-child(2) .price</price>
    <category></category>
</event>
```

Koodinäide 6 – kultuurikava allikast ürituste liigendamise konfiguratsioonifail

Nagu koodinäitest 6 on näha siis ürituse nimi (element *name*) ei võeta mitte detaillehelt vaid andmebaasist, tabelist *downloaded_event*.

3.5.2 Liigendamine

Informatsiooni leidmise eest tõmmatud HTML lehest ning ürituseks loomise eest on vastutav loodud liigendaja (klass *Parser*).
Põhiline funktsionaalsus liigendaja jaoks, elementide leidmine, on tehtud kasutades JSoup teeki.

3.6 Realiseeritud andmeallikad

Realiseerisin andmete kaevamise järgmistest allikatest:

- **Kultuurikava** (http://www.kultuurikava.ee/)
  Allikas omab suurel hulgal üritusi erinevatest valdkondadest. Allikast saab kätte ürituse nime, toimumisaja, toimumiskoha nime, ürituse kirjelduse, lisainformatsiooni lingi ning mõnikord ka hinna.

- **Kultuur.info** (http://kultuur.info/)
  Allikas omab kultuurivaldkonna üritusi. Vähem rõhku on laiemale publikule mõeldud meelelahutusüritustel ja rohkem üritustel nagu väärtfilmid, kirjandusüritused jms. Allikast saab kätte ürituse nime, toimumisaja, toimumiskoha nime, aadressi ja ürituse kirjelduse.

- **Puhkaeestis** (http://www.puhkaeestis.ee/)
  Allikas omab üritusi, mis on suunitletud turistidele. Puuduvad paljud igapäevaüritused, aga on olemas palju pikemaajalisi üritusi nagu festivalid, näitused jne. Allikast saab kätte ürituse nime, toimumisaja, toimumiskoha nime, koordinaadid, hinna, ürituse kirjelduse ja lisainformatsiooni lingi.

- **Tourism.tallinn** (http://www.tourism.tallinn.ee)
  Allikas omab väheses koguses üritusi. Allikas on suunitletud välisturistidele. On välja toodud vaid aasta tähtsamad sündmused Tallinnas. Allikast saab kätte ürituse nime, toimumisaja, toimumiskoha nime, ürituse kirjeldus ja mõnikord ka toimumiskoha aadressi ja koordinaadid.

4. Kogutud andmed

Kogutud andmete laiema analüüsiga see töö ei tegele. Järgnevalt on välja toodud mõned erinevad faktid leitud andmete kohta, et eeskätt näidata, et töö eesmärk täideti edukalt.

Loodud rakendus suutis leida üle tuhande ürituse toimumise lähitulevikus, järgnevalt on välja toodud valik neist.

	Ürituse nimi character varying(255)	Toimumisaeg character varying(2	Toimumiskoht character varying(255)	Hind character varying(127)
1	Hobustest ja inimestest	Jun 3, 2014	Kino Sõprus, Tallinn	4.00€ - 5.00€
2	Katusekino 2014 - Kurb Jasmine / Blue Jasmine	Jun 18, 2014	Viru Keskuse katuseterrass, Tallinn	6.00€ - 8.00€
3	Ninasarvik Otto (Otto er et Næsehorn)	Jun 4, 2014	Coca-Cola Plaza, Tallinn	
4	Koit Toome - Võitleja.	Jun 1, 2014	Mooste mõis, Mooste vald	
5	Rio 2 EST 2D	Jun 2, 2014	Solaris kino, Tallinn	
6	Godzilla (Godzilla)	Jun 2, 2014	Kinokeskus Cinamon.., Tartu	
7	Mina olin veel väikene	Jun 2, 2014	Palmse mõis, Vihula vald	10.50€ - 12.00€
8	Gigolo maski all	May 29, 2014	Solaris kino, Tallinn	
9	Paras pähkel (The Nut Job)	Jun 1, 2014	Forum Cinemas Astri, Narva	
10	Mina olin veel väikene	Jun 28, 2014	Palmse mõis, Vihula vald	10.50€ - 12.00€
11	Pahatar (Maleficent)	May 31, 2014	Kinokeskus Cinamon.., Tartu	
12	Katusekino 2014 - Kurb Jasmine / Blue Jasmine	Jun 27, 2014	Viru Keskuse katuseterrass, Tallinn	6.00€ - 8.00€
13	Pahatar 3D	May 30, 2014	Solaris kino, Tallinn	
14	VANAD JA NOORED / Ugala teater	Jun 27, 2014	A. H. Tammsaare muuseum Vargamäel, Albu vald	
15	Pikk tee alla (A Long Way Down)	Jun 4, 2014	Forum Cinemas Astri, Narva	
16	Neetud naabrid (Neighbors)	Jun 4, 2014	Kinokeskus Cinamon.., Tartu	
17	Godzilla 3D	May 29, 2014	Solaris kino, Tallinn	
18	Красавица и чудовище RU	Jun 2, 2014	Solaris kino, Tallinn	
19	Pahatar (Maleficent)	Jun 4, 2014	Kinokeskus Cinamon.., Tartu	
20	KADUMISE JÄRJEKORRAS	Jun 1, 2014	Männimäe külalistemaja, Viljandi	

Joonis 13 – hulk suvaliselt valitud kaevatud üritusi

	Toimumiskoht character varying(255)	Ürituste toimumiste arv bigint
1	Solaris kino, Tallinn	161
2	Kinokeskus Cinamon.., Tartu	138
3	Forum Cinemas Astri, Narva	62
4	Kino Artis, Tallinn	58
5	Coca-Cola Plaza, Tallinn	45
6	Viru Keskuse katuseterrass, Tallinn	28
7	VEF, Ūnijas iela 8, K 7, Riia	17
8	Palmse mõis, Vihula vald	14
9	Rahvusooper Estonia, Tallinn	13
10	Hüüru mõis, Saue vald	10

Joonis 14 – 10 populaarseimat ürituste toimumiskohta juunis 2014

	Üritus character varying(255)	Ürituse toimumise arv bigint
1	Pikk tee alla (A Long Way Down)	35
2	Pahatar (Maleficent)	29
3	Katusekino 2014 - Kurb Jasmine / Blue Jasmine	28
4	X-Mehed: Tulevase möödaniku päevad (X-Men: Days of Future Past)	24
5	Pahatar 3D	20
6	Pikk tee alla	20
7	Miljon moodust Läänes kärvata	20
8	Ninasarvik Otto (Otto er et Næsehorn)	19
9	PSAIHO.LV - Horribly-positive emotion, theatrical-game with participation and fearing of it.	17
10	Paras pähkel (The Nut Job)	16

Joonis 15 – 10 enimtoimuvat üritust juunis 2014

Nagu näha on populaarseimateks ürituspaikadeks erinevad Eesti kinod. Populaarseimateks üritusteks on nendes toimuvad filmid.

5. Kokkuvõte

Lõputöös disainiti ja realiseeriti rakendus erinevatest Eesti veebilehtedest ürituste kaevandamiseks. Realiseeriti järgmistelt veebilehtedelt andmete tõmbamine ja töötlemine: kultuurikava.ee, puhkaeestis.ee, kultuur.info, tourism.tallinn.ee. Loodud süsteem on laiendatav uute veebilehtede lisamiseks.
Töö käigus leiti tuhandeid lähitulevikus toimuvaid üritusi Eestis.

Edasised arendused:

- Ürituste ühendamine
  Erinevad allikad annavad samu üritusi. Esiteks aitaks see duplikaatüritustest vabaneda, aga samuti aitaks see luua terviklikumat ülevaadet üritusest, sest üks allikas võib anda infot, mida teine ei anna.
- Liides tõmmatud andmete ja loodud ürituste vaatamiseks
  Lisada saab ka näiteks kaardirakenduse, kus kuvatakse üritusi kaardil.
- Andmete tõmbamise täiendused
  Kõiki andmeid(nt toimumiskoha koordinaate) ei saa kätte ürituse detaillehelt, selleks peab minema sügavamale lehele sisse.

6. Viited

1. JSoup. [www] http://jsoup.org/ (29.05.2014)
2. Apache Maven [www] http://maven.apache.org/ (05.06.2014)
3. Spring [www] http://spring.io/ (05.06.2014)
4. Hibernate [www] http://hibernate.org/orm/ (05.06.2014)
5. log4j [www] http://logging.apache.org/log4j/1.2/ (05.06.2014)
6. PostgreSQL [www] http://www.postgresql.org/ (05.06.2014)
7. Java Development Kit [www] http://www.oracle.com/technetwork/java/javase/overview/index.html (05.06.2014)
8. Apache Tomcat [www] http://tomcat.apache.org/ (05.06.2014)
9. Kultuurikava. [www] http://www.kultuurikava.ee/ (29.05.2014)
10. Tourism.tallinn. 2014 aasta ürituste nimekiri. [www] http://www.tourism.tallinn.ee/fpage/experience/cultural_highlights# (29.05.2014)
11. Kultuur.info. [www] http://kultuur.info/ (29.05.2014)
12. Puhkaeestis. [www] http://www.puhkaeestis.ee/et/ (29.05.2014)
13. Visual Paradigm. Andmete modelleerimine. [www] http://www.visual-paradigm.com/features/data-modeling/ (29.05.2014)

7. Lisad

7.1 Rakenduse üles seadmine

7.1.1 Vajalik tarkvara

Rakenduse jooksutamiseks peavad olema arvutis järgmised tarkvarad:

1. Java Development Kit 8
2. Apache Tomcat 8
3. Apache Maven 3.1.1
4. PostgreSQL 9.3

Versiooninumbrid on need, mida rakenduse tegemisel kasutati. Vanemate versioonide kasutamisel pole ootuspärane käitumine tagatud.

7.1.2 Tomcati ülesseadmine

Tomcatis peab olema kasutaja manager-script privileegidega.

Windowsi keskkonnas tuleb selleks *[TOMCAT_HOME]/conf/tomcat-users.xml* faili lisada järgmine rida(kasutajanimi ja parool muuta sobivaks):

```
<user username="manager" password="manager" roles="manager-script"/>
```

7.1.3 Maveni ülesseadmine

Windows keskkonnas tuleb luua keskkonnamuutuja ja *path variable*, et käsurealt oleks võimalik käivitada mavenit.

Lähtekoodi juurkaustas tuleb muuta pom.xml faili. Sinna tuleb sisestada oma tomcati konfiguratsioon. Punasega on märgitud muuta tulevad andmed. Kasutajanimi ja parool on eelpool mainitud manager-script õigustega kasutaja.

```
<plugin>
	<groupId>org.codehaus.mojo</groupId>
	<artifactId>tomcat-maven-plugin</artifactId>
	<configuration>
		<username>manager</username>
		<password>manager</password>
```

```
            <server>TomcatServer</server>
            <url>http://localhost:8888/manager/text</url>
        </configuration>
</plugin>
```

7.1.4 Andmebaasi üles seadmine

1. Tuleb luua PostgreSQL andmebaasi uus andmebaas, nime võib ise valida.
2. Lähtekoodi */src/main/resources/sql/* kaustast tuleb käivitada mõlemad SQL failid andmebaasis. Esimesena *tables.sql* ja teisena *insert.sql*
3. Edasi tuleb seadistada rakenduse andmebaasiühendus:

   Konfiguratsioonifail asub kaustas */src/main/config/hibernate.properties*

   ```
   jdbc.driver_class = org.postgresql.Driver
   jdbc.url = jdbc:postgresql://localhost:5432/mining
   jdbc.username = postgres
   jdbc.password = postgres
   hibernate.dialect = org.hibernate.dialect.PostgreSQL9Dialect
   ```

   Punasega märgitud tuleb panna enda vastavalt enda konfiguratsioonile. Mining on näites andmebaasi nimi. Kasutajanimi ja parool PostgreSQL kasutaja omad.

7.1.5 Logimise üles seadmine

Konfiguratsioonifail asub kaustas */src/main/config/log4j.properties*

Muuta tuleb järgmiste ridade sisu:

```
log4j.appender.A1.File=J:/Log/Mining/scraper.log
log4j.appender.A2.File=J:/Log/Mining/error.log
```

Sisestada tuleb endale sobivad logifailide asukohad.

7.1.6 Tööle panemine

Käivitamiseks tuleb käsurealt minna lähtekoodi juurkataloogi(seal kus asub pom.xml) ja käivitada käsklus:

mvn tomcat:deploy	esimesel korral
mvn tomcat:redeploy	edaspidistel kordadel

Lehel http://localhost:PORT/mining/ peaks nüüd olema töötav rakendus(tomcati port tuleb sisestada url-i).