

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Arvutiteaduse Instituut

Võrgutarkvara õppetool

**Eesti sündmuste andmete kogumise ja  
väljastamise teenus ning seda kasutav  
Androidi rakendus**

Bakalaureusetöö

Üliõpilane: Liis Harjo  
Üliõpilaskood: 123628IAPB  
Juhendaja: Ago Luberg

Tallinn  
2015

## **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**

Antud lõputöö eesmärk on teostada sündmuste soovitamise teenus, mis võimaldab kuvada kasutajale tema läheduses toimuvaid sündmusi ja otsida sündmusi muul ajal ja asukohas. Lõputöö käigus teostatakse serveri poole peal Eesti sündmuste andmete kogumine. Selleks kasutatakse meetodit, mille puhul laaditakse alla veebilehekülje lähtekood ja eraldatakse sellest vajalikud andmed. Serveri poole peal teostatakse ka sündmuste andmete väljastamine. Androidi rakendus kuvab kasutajale tema läheduses lähiajal toimuvaid sündmusi. Kasutajal on võimalik määrata, millised sündmuste kategooriad teda huvitavad ning talle kuvatakse ainult vastavaid sündmusi. Kasutajal on võimalik otsida sündmusi muul ajal ja asukohas. Androidi rakenduses toimub ka sündmuste andmete puhverdamine.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 32 leheküljel, 5 peatükki, 10 joonist, 4 tabelit.

## **Abstract**

The objective of this thesis is to implement an events recommendation service that enables to display to the user events taking place nearby and search for events at other times and locations. Part of this thesis is the gathering of the events data on a server. This is implemented using a method that involves downloading the source of a web page and extracting the necessary data. The delivering of the events data on a server is also implemented. The Android application displays to the user events taking place nearby. It is possible for the user to indicate what events categories he is interested in and he is shown only corresponding events. The user can search for events at other times and locations. Caching of the events data is also implemented in the Android application.

The thesis is in Estonian and contains 32 pages of text, 5 chapters, 10 figures, 4 tables.

## Lühendite ja mõistete sõnastik

<b>CGI</b>	<i>Common Gateway Interface</i> Veebiserveri ja CGI programmi vaheliides, võimaldab CGI programmidel vastata veebiserverisse tulevatele päringutele.
<b>HTTP</b>	<i>HyperText Transfer Protocol</i> Hüpertexti edastusprotokoll, protokoll teabe edastamiseks arvutivõrkudes.
<b>JSON</b>	<i>JavaScript Object Notation</i> Kergekaaluline andmevahetusformaad. Andmed on esitatud tekstilisel kujul nimi-väärtus paaridena või järjestatud väärtuste jadadena.
<b>web scraping</b>	Meetod andmete eraldamiseks veebilehekülgedelt.
<b>XPath</b>	<i>XML Path Language</i> Päringukeel, mis võimaldab XML dokumendist elemente välja valida.

## Jooniste nimekiri

Joonis 1 Vasakult Eventbrite, All Events in City, Nearify, Eventful .....	12
Joonis 3 Süsteemi arhitektuur.....	14
Joonis 2 Andmebaasi skeem.....	15
Joonis 4 Serveri jadadiagramm.....	19
Joonis 5 Sündmuste vaatamise tegevusdiagramm.....	21
Joonis 6 Profiili vaade .....	22
Joonis 7 Vasakult avakuva, detailvaade, kaardi vaade .....	23
Joonis 8 Vasakult otsingu vaade, otsingu eelistuste muutmise vaade.....	23
Joonis 10 Rakenduse klassidiagramm .....	26

## **Tabelite nimekiri**

Tabel 1 Rakenduste võrdlus .....	11
Tabel 2 Andmebaasitabel event.....	15
Tabel 3 Andmebaasitabel event_venue .....	15
Tabel 4 Andmebaasitabel event_session .....	16

# Sisukord

1. Sissejuhatus .....	9
1.1 Taust ja probleem .....	9
1.2 Ülesande püstitus .....	9
1.3 Ülevaade tööst .....	10
2. Ülevaade olemasolevatest rakendustest.....	11
2.1 Eventbrite.....	12
2.2 Nearify .....	12
2.3 All Events in City .....	13
2.4 Eventful .....	13
2.5 Analüüs .....	13
3. Server.....	14
3.1 Süsteemi arhitektuur .....	14
3.2 Andmebaas .....	15
3.3 Andmete kogumine.....	16
3.4 Andmete väljastamine .....	18
3.5 Edasiarendused .....	20
4. Androidi rakendus .....	21
4.1 Androidi rakenduse tegevusdiagramm .....	21
4.2 Androidi rakenduse ekraanivaated .....	22
4.3 Androidi rakenduse teostus .....	24
4.4 Androidi rakenduse klassidiagramm .....	26
4.5 Edasiarendused .....	27
5. Kokkuvõte .....	28
Summary.....	29
Kasutatud kirjandus .....	30
Lisa 1 .....	32



# 1. Sissejuhatus

Antud lõputöö eesmärgiks on luua teenus, mis võimaldab Androidi rakendusega soovitada kasutajale tema läheduses toimuvaid sündmusi. Lõputöö raames teostatakse nii Androidi rakendus kui ka serveri poole peal sündmuste andmete kogumine ja nende väljastamine. Teenus teostatakse selliselt, et seda on võimalik edaspidi täiendada.

Sündmuste andmete kogumine toimub nii, et laetakse alla vajaliku veebilehekülje lähtekood, filtreeritakse sellest välja sündmuste andmed, ning salvestatakse andmebaasi. Veebiserver võtab vastavalt päringu parameetritele andmebaasist andmed ja tagastab informatsiooni sündmuste kohta.

Androidi rakendus kuvab kasutajale tema läheduses toimuvaid sündmusi. Sündmusi kuvatakse vastavalt kasutaja asukohale, mille teeb kindlaks rakendus, ning hetke kuupäevale ja kellaajale. Kasutaja saab määrata, millised teemad teda huvitavad, ja talle kuvatakse ainult vastavaid sündmusi. Samuti on kasutajal võimalik otsida sündmusi mõnel muul ajal ja asukohas.

## 1.1 Taust ja probleem

Asukohapõhised teenused on tänapäeval populaarsed [13]. Antud lõputöö eesmärgiks on uurida, kuidas ühte sellist teenust praktiliselt teostada.

Lõputöö eesmärk on teostada asukohapõhine teenus, mis soovib kasutajale tema läheduses toimuvaid sündmusi, et tuua välja võimalikud kitsaskohad ja laienemisvõimalused. Midagi reaalselt teostades tuleb ilmsiks see, mis on võimalik, mida tuleks teha teisiti ja kuidas edasi liikuda. Esialgset versiooni on võimalik edaspidi täiendada.

## 1.2 Ülesande püstitus

Lõputöö peamine eesmärk on teostada teenus, mis võimaldab soovitada kasutajale tema läheduses toimuvaid sündmusi. Selle saavutamiseks teostatakse järgmised alameesmärgid:

- Serveri poole peal andmete kogumine.

- Serveri poole peal andmete väljastamine. Server peab vastavalt päringu parameetritele tagastama sündmuste andmeid.
- Androidi rakendus. Rakendus peab tegema päringu serverile ja kuvama kasutajale vastuseks saadud sündmuste andmeid.

### **1.3 Ülevaade tööst**

Teises peatükis annan ülevaate mõnedest olemasolevatest rakendustest, mis kasutajale erinevate kriteeriumite põhjal sündmusi soovivad. Kolmandas peatükis kirjeldan seda, kuidas serveri poole peal andmeid kogutakse ja väljastatakse. Neljas peatükk käsitleb Androidi rakenduse teostust.

## 2. Ülevaade olemasolevatest rakendustest

Järgnevalt annan ülevaate mõnedest olemasolevatest rakendustest, mis on oma funktsionaalsuse poolest sarnased antud lõputöö raames teostatava rakendusega.

Selliseid rakendusi on raske täiesti objektiivselt omavahel võrrelda, sest nad pakuvad erinevaid võimalusi, ja näiteks see, kui palju sündmusi nad soovitada oskavad, oleneb sündmuste kategooriast, asukohast ja ajast. Antud rakendusi katsetasin reedel 8. mail ja märkisin kõigis asukohaks Tallinna.

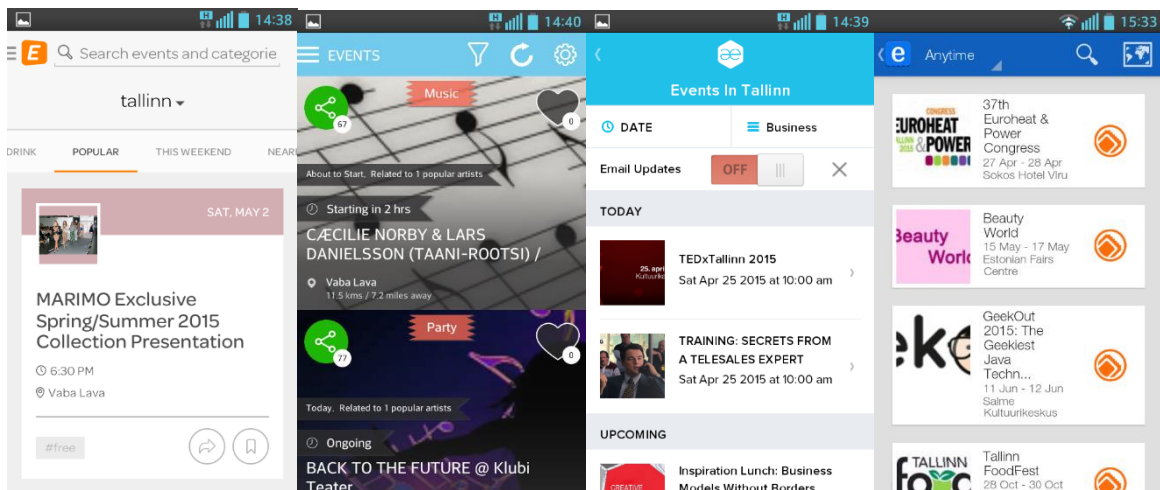
	Eventbrite [3]	Nearify [15]	All Events in City [1]	Eventful [4]
Leitud sündmusti	0	34	35	3
Sündmuse põhjalik kirjeldus	Jah	Jah	Jah	Jah
Sündmuse asukoht kaardil	Jah	Ei	Jah	Jah
Erinevate kategooriate arv	20	18	9	29
Kasutaja eelistused	Jah	Jah	Ei	Ei

**Tabel 1 Rakenduste võrdlus**

Tabeliridade tähendused:

- Leitud sündmusi – kui mitu sündmust rakendus leidis. Lugesin kokku sündmused kõigist kategooriatest, mis toimusid katse tegemisega samal päeval.
- Sündmuse põhjalik kirjeldus – kas iga sündmuse kohta on olemas kirjeldus.
- Sündmuse asukoht kaardil – kas sündmuse toimumise asukohta on võimalik vaadata kaardilt.

- Erinevate kategooriate arv – kui mitut erinevat sündmuste kategooriat rakendus pakub. Kategooriad on näiteks muusika, kino ja pidu.
- Kasutaja eelistused – kas kasutajal on võimalik oma eelistusi püsivalt kindlaks määrata. Selle all mõtlen ma seda, et kas kasutajal on võimalik kindlaks määrata, millised sündmuste kategooriad teda huvitavad ning rakendus peab kasutaja eelistusi meeles.



**Joonis 1 Vasakult Eventbrite, All Events in City, Nearify, Eventful**

## 2.1 Eventbrite

Rakenduse Eventbrite (joonis 1 vasakult esimene) esmakordsel käivitamisel saab kasutaja valida kolm teda huvitavat kategooriat. See on ebaotstarbekas, kuna kasutajat võib huvitada rohkem kui kolm erinevat teemat. Edaspidi kuvatakse sündmuse nende kategooriate hulgast ja lisaks neile populaarseid sündmuse kõigist kategooriatest. Erinevate kategooriate sündmuse kuvatakse erinevates vaadetes. Maailmas on see rakendus väga populaarne – Google Play’st[8] on seda üle miljoni korra alla laaditud. Eestis töötab hetkel halvasti, kuna leiab suhteliselt vähe erinevaid sündmuse – katse tegemise ajal ei leidnud mitte ühtegi sündmust, mis oleks toimunud katse tegemisega samal päeval. Osades kategooriates ei leia üldse sündmuse ja teistes leiab ainult mõne päeva kohta. Kujunduse poolest on rakendus lihtne ja maitsekas.

## 2.2 Nearify

Nearify (joonis 1 vasakult teine) rakendus töötab Eestis hästi, katse tegemise ajal leidis 34 erinevat sündmust. Kasutajal on võimalik oma eelistused püsivalt paika panna, ta saab määrata, millised sündmuste kategooriad teda huvitavad ja välja valitavate kategooriate arv ei ole

piiratud. Edaspidi kuvatakse sündmusi kõigist nendest kategooriatest ühes vaates üksteise järel. Rakendus pakub mugavaid võimalusi sündmuste filtreerimiseks. Erineb teistest katsetatud rakendustest oma värviküllase kujunduse poolest.

### **2.3 All Events in City**

Ka see rakendus (joonis 1 paremalt teine) töötab Eestis hästi, katse tegemise ajal leidis 35 erinevat sündmust. Kasutajal pole võimalik oma eelistusi püsivalt kindlaks määrata. Ühes vaates kuvatakse kasutajale erinevad sündmuste kategooriad, neist mõne peal klikates kuvatakse sündmusi sellest kategooriast. Teises vaates kuvatakse järjest populaarseid sündmusi.

### **2.4 Eventful**

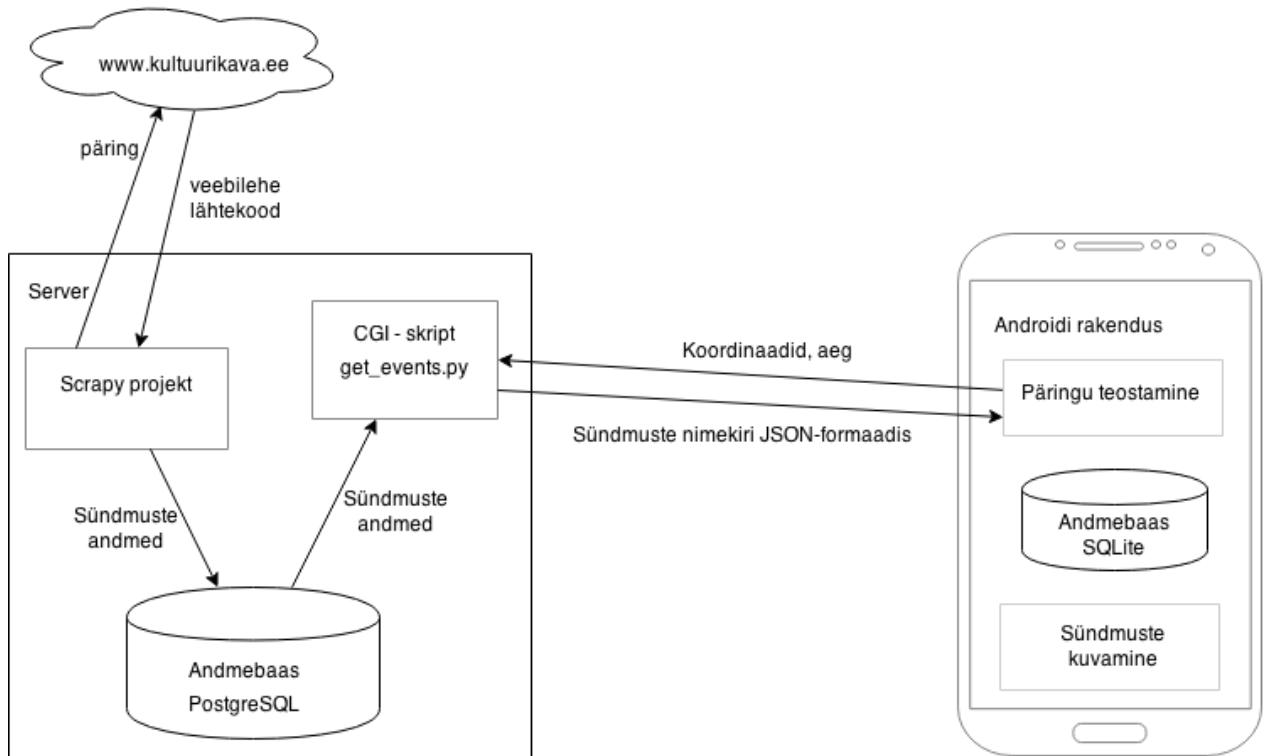
Rakendus Eventful (joonis 1 paremalt esimene) töötab Eestis halvasti, katse tegemise ajal leidis ainult kolm erinevat sündmust. Kasutaja ei saa oma eelistusi püsivalt kindlaks määrata. Samamoodi nagu rakenduses All Events in City kuvatakse ühes vaates erinevad sündmuste kategooriaid, neist mõne peal klikates kuvatakse sündmusi sellest kategooriast, teises vaates kuvatakse järjest populaarseid sündmusi.

### **2.5 Analüüs**

Katsetatud rakendused jagunevad kaheks selle poolest, kui hästi nad Eestis töötavad. Nearify ja All Events in City leidsid mõlemad üle 30 erineva sündmuse, mis toimusid katse tegemisega samal päeval. Samas Eventbrite ei leidnud mitte ühtegi sellist sündmust ning Eventful leidis neid ainult kolm. Kõigis rakendustes on sündmuse kohta olemas peamine informatsioon – toimumise kuupäev, kellaeg ja asukoht ning sündmuse kirjeldus. Kõigis rakendustes, välja arvatud Nearify, on sündmuse asukohta võimalik näha ka kaardil. Rakendused jagunevad kaheks ka selle poolest, kas nad lasevad kasutajal oma eelistused püsivalt kindlaks määrata või mitte. Rakendustes Eventbrite ja Nearify saab kasutaja määrata, millised sündmuste kategooriad teda huvitavad, ning edaspidi kuvatakse talle sündmusi nende kategooriate hulgast (kuigi Eventbrite kuvab lisaks neile ka populaarseid sündmusi kõigist kategooriatest). Rakendused All Events in City ja Eventful ei paku kasutajale võimalust oma eelistusi püsivalt kindlaks määrata.

## 3. Server

### 3.1 Süsteemi arhitektuur

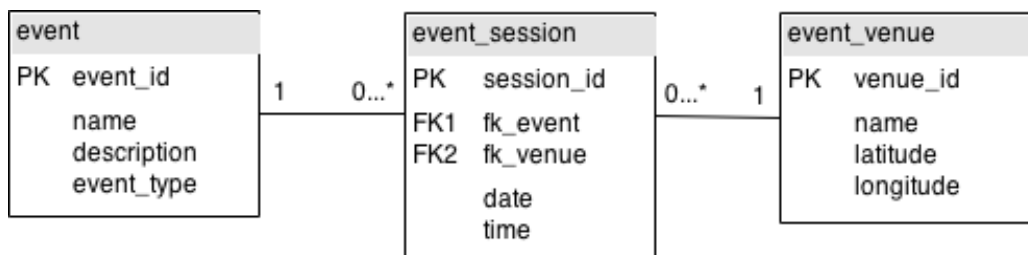


**Joonis 2 Süsteemi arhitektuur**

Sündmuste andmete kogumine toimub Scrapy[18] projektiga, mis töötab serveris eraldi Pythoni[16] virtuaalkeskonnas. Andmeid hoitakse PostgreSQL[17] andmebaasis. Andmeid väljastatakse CGI-skripti[2] abil, mis vastavalt päringu parameetritele tagastab informatsiooni sündmuste kohta. Informatsioon sündmuste kohta tagastatakse JSON-formaadis[11]. Androidi rakendus teostab päringu serverile andes parameetritena kaasa koordinaadid, kuupäeva ja kellaaja ning teisendab saadud JSON-formaadis vastuse enda sisemisteks andmetüüpideks. Androidi rakenduses salvestatakse vastuseks saadud informatsioon ka rakenduse SQLite[21] andmebaasi. Kui kasutajale on vaja kuvada sündmuste andmeid, tehakse kõigepealt päring rakenduse andmebaasi. Ka rakenduse andmebaasi tehakse päring koordinaatide, kuupäeva ja kellaaja põhjal. Kui rakenduse andmebaasist saadakse vastuseks vähemalt üks sündmus, siis ei tehta päringut serverile.

## 3.2 Andmebaas

Sündmuste andmeid hoitakse PostgreSQL andmebaasis, järgnevalt on ära toodud andmebaasi skeem (joonis 2).



**Joonis 3 Andmebaasi skeem**

Järgnevalt on ära toodud andmebaasitabelite kirjeldused:

Veeru nimi	Andmetüüp	Kirjeldus	Näitevärtus
event_id	Täisarv	Tabeli primaarvõti	1
name	Tekstistring	Sündmuse nimi	Peterburi Konservatooriumi koori kontsert
description	Tekst	Sündmuse kirjeldus	Sankt-Peterburi Konservatooriumi segakoor esitab S.Rahmaninovi 'Vesprid. Op.37'.Dirigent Anton Maksimov.
event_type	Tekstistring	Sündmuse tüüp	muusika

**Tabel 2 Andmebaasitabel event**

Unikaalsuse kitsendus hõlmab name ja event\_type veerge.

Veeru nimi	Andmetüüp	Kirjeldus	Näitevärtus
venue_id	Täisarv	Tabeli primaarvõti	1
name	Tekstistring	Toimumiskoha nimi	Jaani kirik
latitude	Komakohaga arv	Toimumiskoha laiuskraad	59.4336835
longitude	Komakohaga arv	Toimumiskoha pikkuskraad	24.7454649

**Tabel 3 Andmebaasitabel event\_venue**

Unikaalsuse kitsendus hõlmab name, latitude ja longitude veerge.

<b>Veeru nimi</b>	<b>Andmetüüp</b>	<b>Kirjeldus</b>	<b>Näiteväärtus</b>
session_id	Täisarv	Tabeli primaarvõti	1
fk_event	Täisarv	Välisvõti Event tabeli event_id veerule. Viide toimuvale sündmusele.	1
fk_venue	Täisarv	Välisvõti EventVenue tabeli venue_id veerule. Viide sündmuse toimumise asukohale.	1
date	Kuupäev	Sündmuse toimumise kuupäev	2015-05-23
time	Kellaeg	Sündmuse toimumise kellaeg	16:00:00

**Tabel 4 Andmebaasitabel event\_session**

Unikaalsuse kitsendus hõlmab kõiki veerge, välja arvatud session\_id.

### 3.3 Andmete kogumine

Andmete kogumiseks on mitmeid võimalusi. Kasutasin andmete kogumiseks meetodit, mille puhul laaditakse alla veebilehekülje lähtekood ja eraldatakse sellest vajalikud andmed (*web scraping* [22]). Reaalses rakenduses võiks sellele lisaks kasutada ka teisi variante, näiteks võib pakkuda kasutajatele võimalust lisada sündmusi. Andmete lähteallikana olen kasutanud ainult ühte veebilehekülge, reaalses rakenduses võiks kasutada mitut erinevat.

Andmete lähteallikana kasutasin Kultuurikava[12] veebilehekülge, kust on võimalik leida informatsiooni erinevat liiki sündmuste kohta. Sündmused on seal jaotatud erinevate kategooriate alla, näiteks teater, muusika ja kino. Iga sündmuse kohta on enamasti võimalik leida tema nime, toimumise kuupäeva, kellaega, asukohta ja koordinaate ning kirjeldust.

Andmete kogumiseks kasutasin raamistikku Scrapy[18]. See on vabavaraline raamistik, mis võimaldab laadida alla veebilehekülje lähtekoodi ja eraldada sellest vajaliku andmed. Scrapy projekt töötab serveris eraldi virtuaalkeskonnas ja on kirjutatud programmeerimiskeeles Python. Scrapy projektis defineerisin, millised andmed on vajalikud (laiendades klassi `Item`) ja kuidas veebilehekülje lähtekoodist vajalikud andmed üles leida (laiendades klassi `CrawlSpider`). Klassis `EventItem` defineerisin, et iga sündmuse kohta on vaja leida tema tüüp, nimi, toimumise koht, kellaeg, kuupäev ja koordinaadid ning kirjeldus. Klassis



KultuurikavaSpider defineerisin, kuidas Kultuurikava veebilehekülgi läbida ja milliste HTML-elementide sees mind huvitavad andmed asuvad.

KultuurikavaSpider klassis defineerisin veebiaadressid, millelt Kultuurikava veebilehekülgede läbimist alustada, ja reegli, mille põhjal leida veebiaadressid, kuhu edasi liikuda. Scrapy CrawlSpider käib vaikimisi läbi kõik veebiaadressid, mille kohta ta leiab lingi. Kuna minu projektis polnud vaja Kultuurikava domeeni kõiki veebilehekülgi külastada, defineerisin reegli, et tuleb külastada ainult neid veebilehekülgi, mille aadress sisaldab sõnet „start=“. Sellist reeglit saab defineerida regulaaravaldisega:

```
Rule(LinkExtractor(allow=[r'/start=']), callback='parse_list_page', follow = True)
```

Selliselt laaditakse alla veebileheküljed, mis sisaldavad nimekirjas erinevate sündmuste andmeid. Veebilehekülje lähtekoodist leitakse XPath'iga[23] HTML-elementid, mis sisaldavad sündmuste andmeid. Realiseerisin selle nii, et kõigepealt leitakse XPath'iga kõik elementid, mis sisaldavad kuupäeva andmeid. Sellistele elementidele järgnevad elementid, mis sisaldavad sündmuste andmeid:

```
selectors = response.xpath('//div[@class="dateheader rightshadowbg"]')  
  
for sel in selectors:  
  
    events = sel.xpath('./following-sibling::*')  
  
    for event in events:
```

event on muutuja, mis sisaldab sündmuse andmeid. Igast sellisest elementist eraldatakse XPath'iga sündmuse andmeväljad. Näiteks selleks, et leida sündmuse nime, kasutasin järgmist XPath'i:

```
event.xpath('./div[@class="name fixwidth"]/a/text()').extract()[0]
```

Sündmuse juures ei ole tema tüüpi määratud, selle saab teada veebilehekülje aadressist. Sündmuse kirjeldus asub eraldi veebilehel, selle leidmiseks tuleb lähtekoodist eraldada kirjelduse lehekülje aadress ja teha eraldi päring. Kui kõik sündmuse andmed on leitud, siis salvestatakse see andmebaasi.

Kogutud andmed salvestan andmebaasi kasutades teeki SQLAlchemy[20], mille abil on võimalik panna vastavusse EventItem tüüpi objektide andmeväljad ja andmebaasitabeli kirjed.

### 3.4 Andmete väljastamine

Serveripool on teostatud kasutades CGI-skripte [2]. Server võtab vastu HTTP päringu[9] ja tagastab informatsiooni sündmuste kohta JSON-formaadis[11]. Päringus peavad olema parameetritena koordinaadid, kuupäev ja kellaaeg. Tagastatakse sündmused, mis toimuvad antud kuupäeval, peale antud kellaaega, antud koordinaatide läheduses.

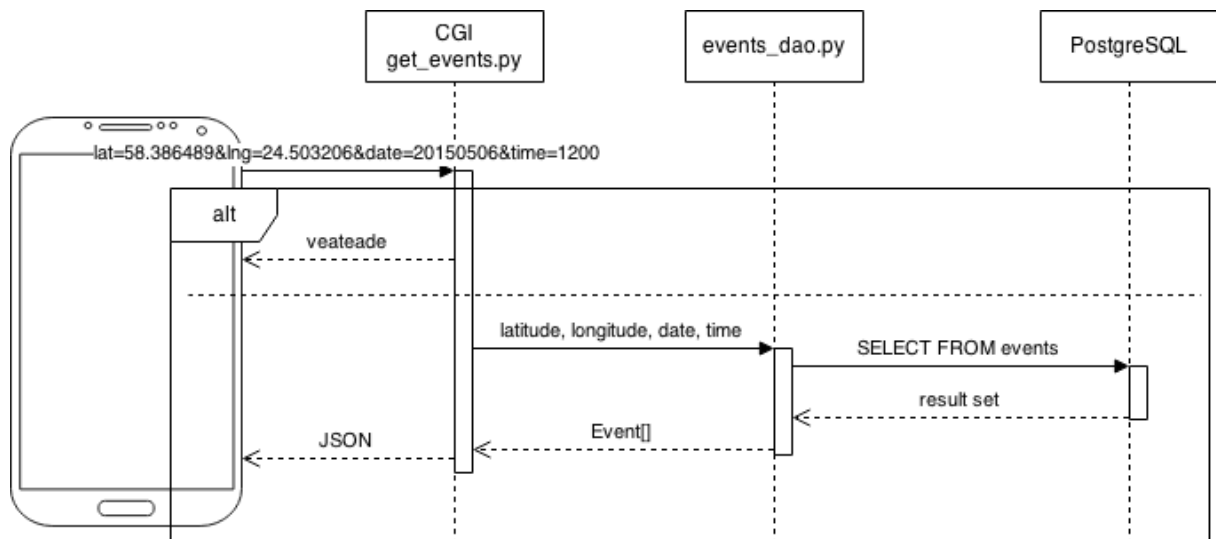
Päringu näide:

[http://dijkstra.cs.ttu.ee/~Liis.Harjo/cgi-bin/get\\_events.py?lat=59.434224&lng=24.743475&date=20150531&time=1200](http://dijkstra.cs.ttu.ee/~Liis.Harjo/cgi-bin/get_events.py?lat=59.434224&lng=24.743475&date=20150531&time=1200)

Päringu parameetrite tähendused:

- lat – laiuskraad. Tähistab selle asukoha, mille läheduses sündmusi otsitakse, laiuskraadi.
- lng – pikkuskraad. Tähistab selle asukoha, mille läheduses sündmusi otsitakse, pikkuskraadi.
- date – kuupäev. Tähistab kuupäeva, millal otsitavad sündmused toimuvad. Kuupäeva formaat koosneb kaheksast numbrist, millest neli esimest tähistavad aastat, kaks järgmist kuud ja kaks viimast päeva.
- time – kellaaeg. Tähistab kellaaega, peale mida otsitavad sündmused toimuvad. Kellaaja formaat koosneb neljast numbrist, millest kaks esimest tähistavad tundi ja kaks viimast minuteid.

Tagastatava vastuse näide on lisas 1.



#### Joonis 4 Serveri jadadiagramm

Päringu võtab vastu CGI-skript `get_events.py` (joonis 4), mis kontrollib, et oleks määratud kõik vajalikud parameetrid – kuupäev, kellaeg, laius – ja pikkuskraad. Samuti kontrollib seda, et parameetrid sisaldaksid ainult lubatud väärtusi – koosneksid ainult numbritest ja oleksid lubatud vahemikes. Kuupäeva parameeter peab sisaldama väärtust, mida oleks võimalik teisendada reaalseks kuupäevaks, kellaaja parameeter peab sisaldama väärtust, mida oleks võimalik teisendada reaalseks kellaajaks. Laius- ja pikkuskraadid peavad sisaldama koordinaate, mis asuvad enam-vähem Eesti piires.

Kui päringu parameetrid vastavad lubatud väärtustele teostab `events_dao.py` andmebaasipäringu, milles küsitakse sündmusi, mis toimuvad antud kuupäeval peale antud kellaega. Saadud vastusest filtreeritakse välja need sündmused, mis toimuvad päringu koordinaatide läheduses. Sündmustevahelise kauguse kindlaks tegemiseks kasutan teeki `GeoPy`[6]. Sündmuste tagastamine on realiseeritud nii, et kõigepealt üritatakse tagastada sündmusi, mis toimuvad päringu koordinaatidest kahe kilomeetri raadiuses. Kui selliseid sündmusi leitakse vähem kui 10, siis proovitakse järjest suuremaid raadiusi:

```

for distance in range(2000, 14000, 2000):
    events_list = get_events_within_distance(distance, latitude, longitude)
    if len(events_list) > 10:
        return events_list
events_list = get_events_within_distance(14000, latitude, longitude)
return events_list
  
```

Eesmärgiks on optimeerida tagastavate sündmuste arvu. Tagastatavate sündmuste arv on piiratud, maksimaalselt tagastatakse 200 sündmust.

Kui juhtub mõni viga, siis tagastatakse veateade. Kui sündmuseid ei leita, siis tagastatakse tühi sündmuste nimekiri.

### **3.5 Edasiarendused**

Antud lahenduses ei arvestata sellega, et sündmuste andmed võivad muutuda. Kuna kõik veebileheküljelt leitud andmed salvestatakse andmebaasi ja hiljem ei kontrollita, kas veebileheküljel on andmed muutunud, siis võib näiteks juhtuda, et mõni sündmus jääb ära, aga andmebaasi jäävad selle kohta andmed.

Andmeid võiks koguda mitmelt erinevalt veebileheküljelt, sest siis leitaks tõenäoliselt rohkem erinevaid andmeid. Sellisel juhul tuleks mõelda selle peale, kuidas erinevatest allikatest pärit andmeid omavahel ühildada, kuna erinevatel veebilehekülgedel võivad andmed sama sündmuse kohta olla esitatud erinevalt. Näiteks võib sama sündmus olla erinevatel veebilehekülgedel paigutatud erinevate kategooriate alla.

Üks sündmus võib samaaegselt sobida erinevate sündmuste kategooriate alla, näiteks lastefilm sobib nii kategooria alla „film“ kui ka „lastele“. Seetõttu tuleks mõelda selle peale, kuidas sündmus ja tema kategooriad omavahel paremini siduda. Praeguses lahenduses on siis, kui sündmus sobib mitme erineva kategooria alla, andmebaasi tabelis selle sündmuse kohta mitu rida, kus muud andmed korduvad, kuid kategooria on erinev. Parem lahendus oleks see, kui selline sündmus oleks välisvõtmete abil seotud mitme erineva kategooriaga.

Reaalses süsteemis tuleks realiseerida kasutajate autentimine, et piirata andmetele ligipääsu. Samuti võiks serverile päringut tehes anda kaasa otsitavate sündmuste tüübid ning server võiks tagastada ainult neile tüüpidele vastavaid sündmusi. See teeks süsteemi efektiivsemaks, kuna edastatakse ainult vajalikke andmeid.

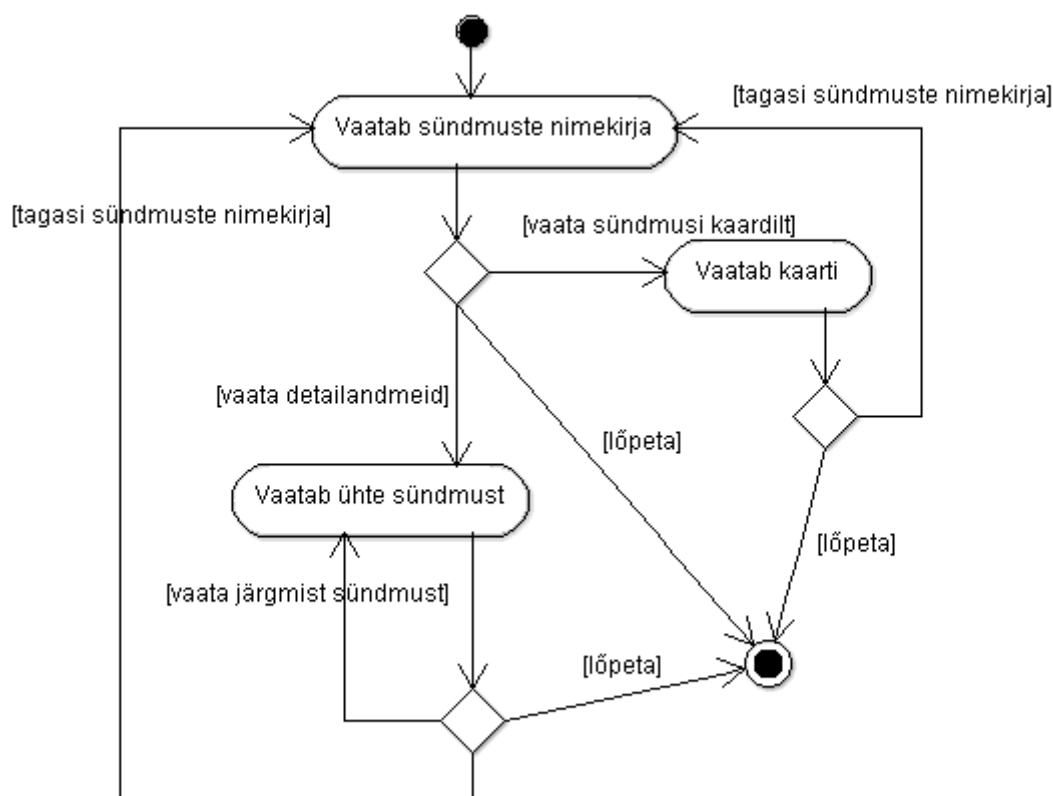
Teenuse võiks teha mitmekeelseks. Selleks tuleks Kultuurikava veebilehelt koguda andmeid erinevates keeltes ning server võiks vastavalt päringu parameetrile tagastada andmeid erinevas keeles. Ka Androidi rakendus võiks olla mitmekeelne.

## 4. Androidi rakendus

Kasutajal on võimalik oma profiilis määrata, millised sündmuste kategooriad teda huvitavad, ning talle kuvatakse ainult neile kategooriatele vastavaid sündmusi. Kui kasutaja rakenduse avab, siis kuvatakse kõigepealt tema läheduses toimuvaid sündmusi. Kasutajal on ka võimalik otsida sündmusi, mis toimuvad mõnel muul ajal ja asukohas, kui tema hetke asukoht.

Rakenduses toimub ka andmete puhverdamine, selleks salvestatakse kõik alla laetud andmed rakenduse SQLite andmebaasi.

### 4.1 Androidi rakenduse tegevusdiagramm

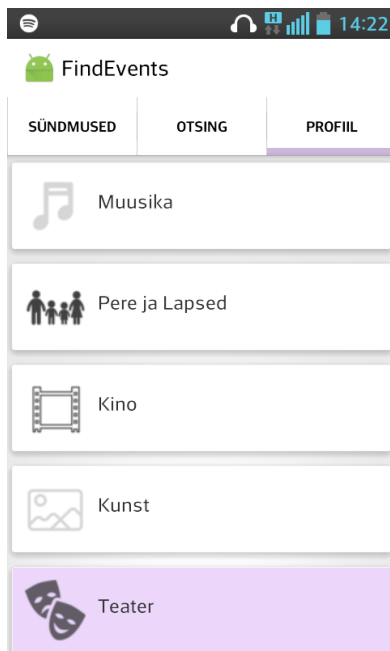


#### Joonis 5 Sündmuste vaatamise tegevusdiagramm

Kui kasutaja rakenduse avab, siis kuvatakse talle kõigepealt tema läheduses toimuvate sündmuste nimekiri (joonis 5). Kuvatakse vaid neid sündmusi, mis vastavad kasutaja poolt välja valitud kategooriatele. Rakenduse kasutamine on tehtud lihtsaks – kasutaja peab ühe korra oma profiilis määrama, millised teemad teda huvitavad, ning edaspidi näeb ta kohe peale rakenduse avamist, mis tema läheduses tema jaoks huvitavat toimub. Sündmuste nimekirjas mõne

sündmuse peal klikates avaneb selle detailvaade, kus on enamasti võimalik näha sündmuse pikemat kirjeldust. Detailvaates on *swipe*'iga võimalik liikuda järgmise või eelmise sündmuse juurde. Sündmuste toimumise asukohti on võimalik vaadata ka kaardilt.

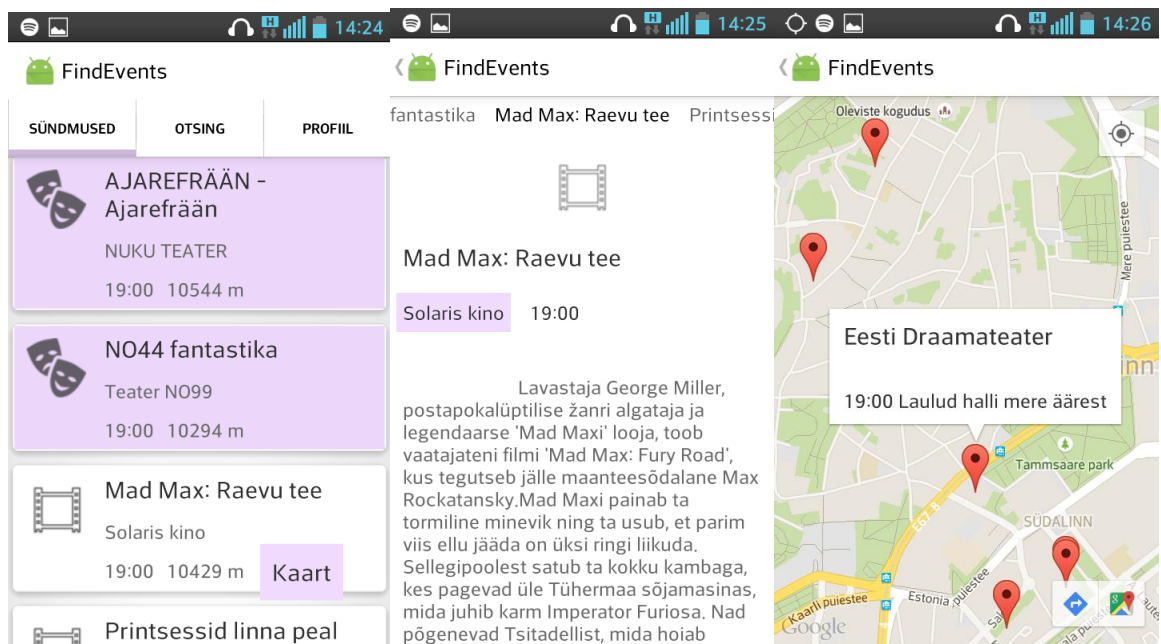
## 4.2 Androidi rakenduse ekraanivaated



### Joonis 6 Profili vaade

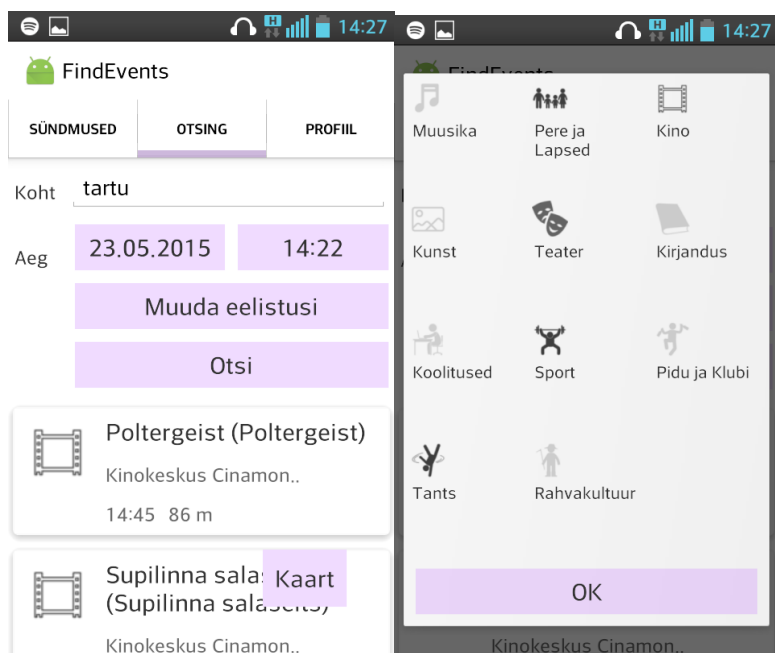
Profili vaates on kasutajal võimalik määrata, millised teemad teda huvitavad (joonis 6). Sündmuste kategooriatel on kolm olekut:

- Kategooria pilt on helehall – see kategooria kasutajat ei huvita ning sellele vastavaid sündmusi talle ei kuvata.
- Kategooria pilt on must – see kategooria huvitab kasutajat ning talle kuvatakse vastavaid sündmusi.
- Kategooria pilt on must ja kategooria on esile toodud taustavärviga – see kategooria on üks kasutaja lemmikutest, talle kuvatakse vastavaid sündmusi ning need tuuakse sündmuste nimekirjas samuti esile taustavärviga.



### Joonis 7 Vasakult avakuva, detailvaade, kaardi vaade

Joonisel 7 on näha rakenduse avakuva (vasakult esimene). Nimekirjas on iga sündmuse kohta ära toodud tema nimi, toimumiskoht ja algusaeg, samuti toimumiskoha kaugus kasutajast. Mõne sündmuse peal klikates avaneb selle detailvaade (joonis 7 vasakult teine), kus on lisaks eelpoolnimetatud informatsioonile enamasti olemas sündmuse kirjeldus. Kaardi vaates on markeritena ära toodud sündmuste toimumise asukohad (joonis 7 paremalt esimene), markeri peal klikates on võimalik näha toimumiskoha nime ja sündmuste nimekirja.



### Joonis 8 Vasakult otsingu vaade, otsingu eelistuste muutmise vaade

Joonisel 8 vasakul on näha rakenduse otsingu vaade. Kasutajal on võimalik määrata otsitavate sündmuste asukohta, kui ta jätab selle välja tühjaks teostatakse otsing tema hetke asukoha põhjal. Klikates kuupäeva või kellaaja peal on kasutajal võimalik neid parameetreid muuta, vaikumisi on välja valitud hetke kuupäev ja kellaeg. Kasutajal on võimalik määrata otsitavate sündmuste kategooriad (joonis 8 paremal). Vaikumisi on välja valitud samad kategooriad, mis kasutaja on oma profiilis välja valinud.

### 4.3 Androidi rakenduse teostus

Selleks, et sündmuste informatsiooni lihtsamalt esitada defineerisin klassi `Event`, kus hoitakse ühe sündmuse andmeid. Selleks, et teada, milliseid sündmusi kasutajale välja näidata, defineerisin klassi `EventType`. Igal `Event`-tüüpi objektile on viide ühele `EventType`-tüüpi objektile. Kui kasutaja profiili vaates mõne teda huvitava teema välja valib, siis määratakse vastav `EventType`-tüüpi objekt välja valituks. Kasutajale näidatakse välja ainult neid sündmusi, mille tüüp on määratud valituks. Kasutaja eelistusi hoitakse rakenduse erinevate käivituskordade vahel meeles kasutades `SharedPreferences` teeki [19].

Selleks, et kuvada kasutajale tema läheduses toimuvaid sündmusi, on kõigepealt vaja teada tema asukohta. Asukoha kindlaks tegemiseks on Androidis erinevaid võimalusi, näiteks võib kasutada otse GPSi või Androidi `Network Location Provider`'it. Kuid otsustasin kasutada `Google Location Services API`-t, mis on väidetavalt võimsam ja täpsem[14]. Seda kasutades tuleb implementeerida `LocationListener`, millele hakatakse edastama informatsiooni asukoha muutumise kohta. Minu realisatsioonis ootab rakendus peale tööle hakkamist kõigepealt asukoha informatsiooni. Kui see on leitud, siis lõpetatakse asukohauuenduste kuulamine ja saab kuvada sündmusi.

Alati, kui tehakse päring serverile, salvestatakse vastuseks saadud andmed ka rakenduse enda `SQLite` andmebaasi. Kasutajale sündmuste kuvamiseks tehakse kõigepealt päring rakenduse andmebaasi. Ka rakenduse andmebaasi tehakse päring kuupäeva, kellaaja ka koordinaatide põhjal. Tagastatakse sündmused, mis toimuvad antud kuupäeval, peale antud kellaega (kuni südaööni), antud koordinaatide läheduses. Koordinaatidevahelise kauguse leian kasutades teeki `Geocoder`[5]. Kui rakenduse andmebaasist leitakse vähemalt üks sündmus, siis ei tehta päringut serverile, vaid kuvatakse olemasolevaid sündmusi. See teeb rakenduse kiiremaks. Selleks, et rakenduse andmebaasi ei koguneks liiga palju andmeid, kustutakse rakenduse käivitamisel andmebaasist kõik vananenud andmed, see tähendab kõigi sündmuste andmed, mille kuupäev



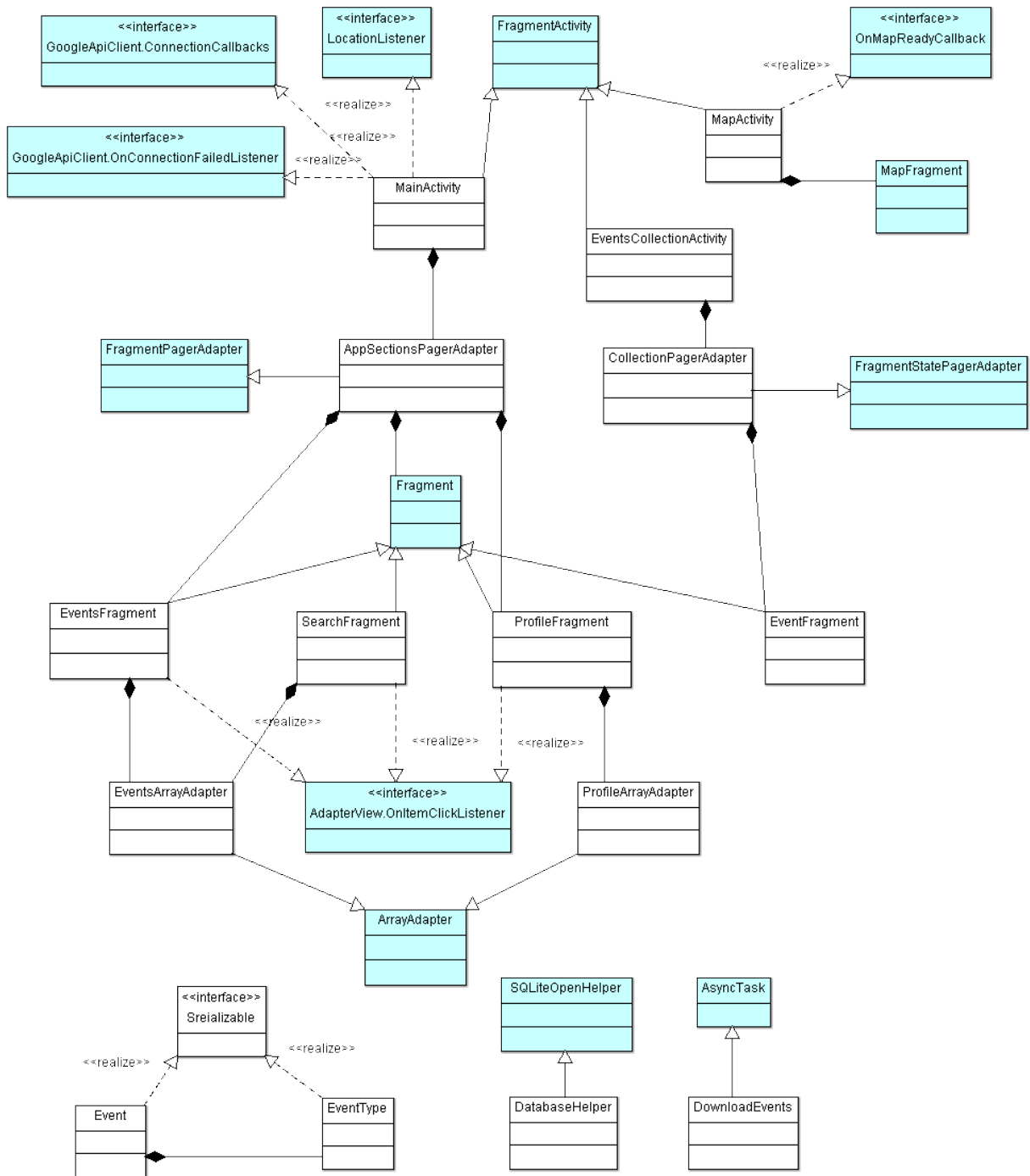
on varasem hetke kuupäevast. Kui rakenduse andmebaasist sündmusi ei leita, tuleb teha päring serverile.

Serverile tehakse HTTP-päring asukoha koordinaatide ning hetke kuupäeva ja kellaaja põhjal. Saadakse JSON-formaadis vastus sündmuste kohta, mis teisendatakse minu defineeritud `Event`-tüüpi objektideks ning salvestatakse ka rakenduse andmebaasi.

Sündmusi kuvatakse nimekirjas, mõne sündmuse peal klikates avaneb selle detailvaade. Sündmuste toimumise asukohti on võimalik vaadata ka kaardilt, selleks kasutasin API-t Google Maps Android API[7]. See võimaldab avada kaardi vaate kasutaja koordinaatide koha pealt ning lisada iga sündmuse toimumiskohta markeri.

Otsingu puhul on kasutajal võimalik sisestada otsitavate sündmuste asukoht, kellaeg ja kuupäev ning otsitavate sündmuste tüübid. Kui kasutaja jätab mõne parameetri määramata, siis kasutatakse vaikimisi väärtusi – hetke asukoht, kuupäev ja kellaeg. Asukoha põhjal leitakse koordinaadid kasutades teeki `Geocoder`[5]. Ka asukohad ja koordinaadid salvestatakse rakenduse andmebaasi – kui kasutaja otsib mitu korda sama asukohta, siis võetakse koordinaadid rakenduse andmebaasist. Siis teostatakse päring serverile andes parameetritena kaasa koordinaadid, kuupäeva ja kellaaja. Saadud vastusest kuvatakse samuti ainult neid sündmusi, mis kasutajat huvitavad. Ka otsingu tulemusi on võimalik vaadata nii nimekirjas, detailvaates kui ka kaardilt.

## 4.4 Androidi rakenduse klassidiagramm



### Joonis 9 Rakenduse klassidiagramm

Klassidiagrammidelt (joonis 10) on näha, milliseid Androidi klasse olen kasutanud – need on eristatud helesinise taustaga. Samuti võib klassidiagrammidelt saada ülevaate rakenduse ülesehitusest. Peamine tegevus toimub `MainActivity`'s, see klass vastutab nii asukoha kindlakstegemise kui ka kasutajale peamiste vaadete kuvamise eest. Peamised vaated on

sündmuste, otsingu ja profiili vaade, igäihe kuvamiseks on oma `Fragment` (`EventsFragment`, `SearchFragment`, `ProfileFragment`). Vaadete vahel navigeerimiseks saab kasutada *tab*'e (sakke) või *swipe*'i, vaadete kuvamist ja nende vahel navigeerimise teostab `AppSectionsPagerAdapter`. Selle realiseerimiseks olen kasutanud Androidi poolt pakutavaid võimalusi ja vastavat juhendit [10].

Sündmuse kuvatakse vaadetes nimekirjana. Seda, kuidas sündmuse andmed nimekirja vaatesse paigutada, teostab `EventsArrayAdapter`. Samamoodi vastutab `ProfileArrayAdapter` selle eest, kuidas erinevaid sündmuse tüüpe nimekirja paigutada. Mõne sündmuse peal klikates avaneb selle detailvaade, kus saab *swipe*'iga liikuda eelmise või järgmise sündmuse juurde. Seda võimaldab `CollectionPagerAdapter`, mille realiseerimisel olen samuti lähtunud eelpoolmainitud juhendist. Sündmuse detailvaadet kuvab `EventFragment`.

Kaardi kuvamine toimub `MapActivity`'s.

## 4.5 Edasiarendused

Rakenduses võiks olla realiseeritud sisselogimine ja kasutajakontod, mis võiks olla seotud kasutaja sotsiaalmeedia kontodega. Siis saaks pakkuda kasutajale võimalust anda oma sõpradele teada, millistest sündmustest ta kavatseb osa võtta, ja näha, millistest sündmustest tema sõbrad osa võtavad.

Kasutajale võiks pakkuda võimalust märkida ära, millistest sündmustest ta kavatseb osa võtta, ning siis enne selle sündmuse toimumist teda teavitada. Kui kasutaja on mõnda sündmust juba külastanud, siis võiks talle selle sündmuse järgmisi toimumisi enam mitte välja näidata. Näiteks kui kasutaja on mõnda filmi juba näinud, siis selle filmi tulevase seansse talle enam mitte välja näidata.

Rakenduses võiks olla realiseeritud teavitamine – rakendus võiks kasutajat aeg-ajalt teavitada tema läheduses toimuvatest huvitavatest sündmustest. Erinevaid sündmuse kategooriaid võiks olla rohkem ning iga kategooria võiks olla jagatud alamkategooriateks. Nii saaks kasutaja eelistused täpsemalt paika panna ning pakkuda talle täpselt tema huvidele vastavaid sündmusi. Kasutaja kohta võiks koguda statistikat – näiteks jätta meelde, milliste sündmuste peal ta klikkab. Selle põhjal saaks järeldada, mis kasutajat huvitab.

## 5. Kokkuvõte

Lõputöö eesmärgiks oli teostada teenus, mis võimaldaks kuvada kasutajale tema läheduses toimuvaid sündmusi ning otsida sündmusi muul ajal ja asukohas. Lõputöös võrreldakse olemasolevaid rakendusi, mis kasutajale sündmusi soovivad. Lõputöö käigus teostati nii serveris andmete kogumine ja väljastamine kui ka Androidi rakendus. Andmete kogumiseks kasutati meetodit, mille puhul laetakse alla veebilehekülje lähtekood ja eraldatakse sellest vajalikud andmed. Androidi rakendus kuvab kasutajale sündmusi, mis toimuvad tema läheduses antud ajahetkel. Kasutajal on võimalik määrata, millised sündmuste kategooriad teda huvitavad, ning talle kuvatakse ainult vastavaid sündmusi. Kasutaja saab ka otsida sündmusi muul ajal ja asukohas. Rakenduses on teostatud ka sündmuste andmete puhverdamine.

Teostatud lahendust on võimalik mitmel viisil täiendada. Näiteks võiks andmeid koguda mitmest erinevast kohast, serveris teostada autentimine ja Androidi rakenduses kasutajakontod.

## Summary

The objective of this thesis was to implement a service that enables to display to the user events taking place nearby and to search for events at other times and locations. Part of this thesis is a comparison of existing applications that recommend events to the user. As part of this thesis was implemented the gathering and delivering of the events data on a server and an Android application. The data were gathered using a method that involves downloading the source of a web page and extracting the necessary data. The Android application displays to the user events currently taking place nearby. The user can indicate what events categories he is interested in and he is shown only corresponding events. The user can search for events taking place at other times and locations. Caching of the events data is implemented in the application.

There are several ways to improve the implementation. For example, to gather the data from different locations, implement authentication on the server and user accounts on the Android application.

## Kasutatud kirjandus

1. All Events in City. [WWW] <http://allevents.in/> (24.05.2015)
2. Common Gateway Interface. [WWW] [http://en.wikipedia.org/wiki/Common\\_Gateway\\_Interface](http://en.wikipedia.org/wiki/Common_Gateway_Interface) (24.05.2015)
3. Eventbrite. [WWW] <https://www.eventbrite.com/> (24.05.2015)
4. Eventful. [WWW] <http://eventful.com/tallinn/events> (24.05.2015)
5. Geocoder. [WWW] <http://developer.android.com/reference/android/location/Geocoder.html> (24.05.2015)
6. GeoPy. [WWW] <https://geopy.readthedocs.org/en/1.10.0/> (06.05.2015)
7. Google Maps Android API v2. [WWW] <https://developers.google.com/maps/documentation/android/> (06.05.2015)
8. Google Play. [WWW] <https://play.google.com/store?hl=en> (24.05.2015)
9. Hypertext Transfer Protocol. [WWW] [http://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) (24.05.2015)
10. Implementing Effective Navigation. [WWW] <https://developer.android.com/training/implementing-navigation/index.html> (06.05.2015)
11. JSON. [WWW] <http://en.wikipedia.org/wiki/JSON> (24.05.2015)
12. Kultuurikava. [WWW] <http://www.kultuurikava.ee/> (24.05.2015)
13. Location-Based Services. [WWW] <http://www.pewinternet.org/2013/09/12/location-based-services/> (08.05.2015)
14. Location Strategies. [WWW] <http://developer.android.com/guide/topics/location/strategies.html> (06.05.2015)
15. Nearify. [WWW] <http://www.nearify.com/> (24.05.2015)

16. Python. [WWW] <https://www.python.org/> (26.05.2015)
17. PostgreSQL. [WWW] <http://www.postgresql.org/> (26.05.2015)
18. Scrapy. [WWW] <http://scrapy.org/> (06.05.2015)
19. SharedPreferences. [WWW]  
<http://developer.android.com/reference/android/content/SharedPreferences.html>  
(24.05.2015)
20. SQLAlchemy. [WWW] <http://www.sqlalchemy.org/> (06.05.2015)
21. SQLite. [WWW] <https://www.sqlite.org/> (26.05.2015)
22. Web scraping. [WWW] [http://en.wikipedia.org/wiki/Web\\_scraping](http://en.wikipedia.org/wiki/Web_scraping) (24.05.2015)
23. XPath. [WWW] <http://en.wikipedia.org/wiki/XPath> (24.05.2015)

## Lisa 1

```
{"events": [{"description": "\r\n Kakukivi lossis elutses igiammustest aegadest peale v\u00e4ike tont. P\u00e4eviti ta magas, aga \u00f6\u00f6sel, kui lossi jalamil asuva linnakese raekoja tornikell kesk\u00f6\u00f6tundi l\u00f6si, \u00e4rkas v\u00e4ike tont \u00f6les. Ometi unistas ta kogu aeg, et v\u00f6iks terve p\u00e4eva \u00f6leval olla ja inimestega p\u00e4evalguses kohtuda. \u00dchel p\u00e4eval saabki see unistus teoks ning sellest s\u00f6nib v\u00e4gevaid seiklusi nii v\u00e4ikese tondi enda kui Kakum\u00e4e linnakese laste jaoks. Film on kuulsa Saksa lastekirjaniku Otfried Preussleri raamatu ekraniseering. Dubleeritud Wavelength Studios filmistuudio Estinilm tellimusel.\r\nTeksti loevad Maiken Shmidt ja Karol Kuntzel\r\nT\u00f6lkija Ellen Suits\r\nDublaazirezhiss\u00f6\u00f6r Argo Heinmaa\r\nDublaazhiprodutsent\u00a0 Mati Sepping \u00a0Tootjamaa: Saksamaa 2013\r\nRezhiss\u00f6\u00f6r Alain Gsponer\r\nFilmi pikkus:\u00a0 88 min\r\nTreiler \r\n \r\n ", "venue": "Kuressaare Linnateater", "longitude": "22.4896359", "name": "V\u00e4ike tont", "latitude": "58.2541207", "time": "13:00:00", "date": "2015-05-06", "event_type": "kino"}, {"description": "\r\n Avatud: E-R 8-18; L 8-15\r\n ", "venue": "Ajamaja galerii", "longitude": "22.488969", "name": "Taave Tuutma ja Vana-Kalamaja kommuun 'Meie maja'", "latitude": "58.253929", "time": "00:00:00", "date": "2015-05-06", "event_type": "naitus"}, {"description": "\r\n Avamine 10.04.15 kell 16.30\r\n Avatud: E-R 10.00-19.00, L 10.00-16.00\u00a0\r\n ", "venue": "Kuressaare kultuurikeskus", "longitude": "22.4868894", "name": "Pusa (Piret Bergmann-Torim) maalin\u00e4itus \u201c\u00d5itsemise aegu\u201d", "latitude": "58.2529014", "time": "00:00:00", "date": "2015-05-06", "event_type": "naitus"}]}
```