

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Liisi Taimre 175754IDDR

**MUSEAALIDE PÄRITOLUKAART:
MUUSEUMITE INFOSÜSTEEMI
AVAANDMEID KASUTAV
KAARDIRAKENDUS**

bakalaureusetöö

Juhendaja: Riina Maigre
PhD

Tallinn 2020

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.

Autor: Liisi Taimre

14.05.2020

Annotatsioon

Lõputöö eesmärk oli luua museaalide päritolukaart, mis annab ülevaate Eesti muuseumides olevate esemete päritolust. Museaalide päritoluandmeid on võimalik filtreerida eseme tüübi ja valmistamise aja järgi. Veebirakendus kasutab Eesti Muuseumide Infosüsteemi avaandmeid.

Rakenduse arendamise käigus analüüsiti sarnaseid veebirakendusi ja Eesti Muuseumide Infosüsteemi avaandmete struktuuri.

Lõputöö tulemina valmisid toimiv veebirakendus ja soovitused Eesti Muuseumide Infosüsteemi avaandmete portaali täiustamiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 23 leheküljel, 6 peatükki, 10 joonist, 1 tabelit.

Abstract

Origin Map of Museum Items: Map Application Based on Open Data of Museums Public Portal

The aim of the thesis was to create a map application based on open data of Museums Public Portal, which provides an overview of the origin of museum items in Estonian museums. The origin data of museum items can be filtered by type and time of manufacture. The web application uses the open data of the Estonian Museums Public Portal.

During the development of the application, similar web applications and the structure of the open data of the Estonian Museums Public Portal were analyzed.

The main result of the thesis was a functioning web application. Also recommendations for improving the open data of the Estonian Museums Public Portal were provided.

The thesis is in Estonian and contains 23 pages of text, 6 chapters, 10 figures, 1 table.

Lühendite ja mõistete sõnastik

API	<i>Application Program Interface</i> , rakendusliides, mille ülesanne on lihtsustada tarkvara arendamist
GEOJSON	Andmevahetusvorming geograafiliste omaduste kujutamiseks. Põhineb JSON-formaadil.
JSON	<i>JavaScript Object Notation</i> , andmevahetusvorming, mis põhineb JavaScripti programmeerimiskeelel
MuIS	Eesti Muuseumite Infosüsteem
URI	<i>Uniform Resource Identifier</i> , protokoll infoallika üheseks määramiseks veebis

Sisukord

1 Sissejuhatus	9
1.1 Taust ja probleem	9
1.2 Ülesande püstitus	9
1.3 Metoodika	10
1.4 Ülevaade tööst.....	10
2 Avaandmed	12
2.1 Avaandmed Eestis.....	12
2.2 RDF.....	13
3 Sarnaste veebirakenduste analüüs	14
3.1 Ajapaik	14
3.2 Britain on Film andmebaasi kaardivaade.....	16
3.3 Koroonakaart.....	17
3.4 Järeldused	18
4 Veebirakenduse arendus	20
4.1 Funktsionaalsed nõuded	20
4.2 Muuseumide Infosüsteemi avaandmete struktuuri analüüs	21
4.2.1 Muuseumi kogusse kuuluvad museaalid	22
4.2.2 Üksiku museaali andmed	22
4.2.3 Museaal → Olemus.....	22
4.2.4 Museaal → Sündmus	22
4.3 Tehnoloogiad ja tööriistad.....	23
4.3.1 Node.js ja Express.....	23
4.3.2 React	24
4.3.3 Bootstrap raamistik	24
4.4 Realisatsiooni kirjeldus.....	24
4.4.1 Andmekiht	24
4.4.2 Kasutajaliides.....	27
5 Valminud rakenduse analüüs	30
5.1 Vastavus esialgsetele eesmärkidele	30
5.2 Edasiarendusvõimalused	30
5.3 Ettepanekud MuIS-i avaandmete portaali täiustamiseks	31
6 Kokkuvõte	32
Kasutatud kirjandus	33

Jooniste loetelu

Joonis 1. RDF-i kolmik	13
Joonis 2. Veebirakenduse Ajapaik kaardivaade	16
Joonis 3. Andmebaasi <i>Britain on Film</i> kaardivaade	17
Joonis 4. Veebirakenduse Koroonakaart avavaade	18
Joonis 5. Museaalide päritolukaardi jaoks vajalike andmete struktuur MuIS-i avaandmete portaalis	21
Joonis 6. Üksiku museaali objekt JSON-failis	25
Joonis 7. Eesti maakondade nimed ja maakondade koodid JSON kujul.....	25
Joonis 8. Kaardi loomiseks vajalike väljadega objekti tagastav funktsioon.....	26
Joonis 9. URL-ile <i>filteredItems/:type/:period</i> päringu tegemisel käivitatav kood	26
Joonis 10. Veebirakenduse Museaalide päritolukaart kasutajaliides.....	27

Tabelite loetelu

Tabel 1. Rakenduse funktsionaalsed nõuded.....	20
--	----

1 Sissejuhatus

Avaandmete mitmekülgne kasutamine on oluline suund nii riigi- kui avaliku sektori juhtimisotsuste planeerimisel. Andmete ümberstruktureerimine ja omavahel kombineerimine võib pakkuda rohkelt kasulikku informatsiooni, mis muidu võiks märkamatuks jääda.

Andmete taaskasutamine on väga levinud ka kultuuripärandi valdkonnas. Seoses muuseumites, arhiivides ja raamatukogudes leiduva materjali digiteerimisega on tehtud kättesaadavaks rohkelt metainfot ajalooliste fotode, esemete ja dokumentide kohta. Eesti kontekstis on üks suuremaid ja olulisemaid kultuuripärandi valdkonnaga seotud andmehulki Eesti Muuseumide Infosüsteemi (MuIS) avaandmed [1].

1.1 Taust ja probleem

MuIS-i kasutaja saab teha päringuid parameetrite kaupa, sirvida ühe muuseumi kogusid ja näha museaali infot detailvaates. Puudub aga võimalus saada üldine ülevaade sellest, kust on erinevad Eesti muuseumites asuvad esemed pärit ja kui palju ning millist tüüpi esemeid on ühest või teisest piirkonnast kogutud.

Selline info on vajalik Eesti muuseumite kogudes leiduva materjali analüüsiks ja muuseumite kogumispoliitika mugavamaks planeerimiseks.

Käesoleva töö teema on välja kasvanud autori tööalasest kokkupuutest MuIS-iga – nii muuseumi koguhoidjana kui ka tavakasutajana.

1.2 Ülesande püstitus

Käesoleva bakalaureusetöö peamine eesmärk on luua veebirakendus kaardirakendusena, mis annab ülevaate Eesti muuseumites olevate esemete päritolust. Veebirakendus kasutab MuIS-i avandmeid [1].

Kaardil on maakonniti näha, kust pärinevad Eesti muuseumide kogudes olevad museaalid. Museaalide päritoluandmeid on võimalik sorteerida eseme tüübi ja dateeringu järgi.

Rakenduse peamine sihtgrupp on Eesti muuseumide peavarahoidjad, Kultuuriministeerium ja Muinsuskaitseamet. Samuti võib see huvi pakkuda kodukoha ajaloo-huvilistele.

Töö lisaeesmärkideks on anda ülevaade rakenduse arenduskäigust ja võimalikest edasiarendustest.

1.3 Metoodika

Käesoleva bakalaureusetöö peamiseks tulemiks on toimiv veebirakendus, mis kasutab MuIS-i avaandmeid.

Töö kõige esimeseks etapiks on sarnaste veebirakenduste analüüs, et välja selgitada, millised on andmete esitamiseks kaardilahendusi kasutavate veebirakenduste olulised komponendid ja funktsionaalsused.

Rakenduse arenduse väga oluliseks osaks on välja selgitada funktsionaalsed nõuded, ning see, milliseid andmeid rakendus vajab ja viimasest lähtuvalt analüüsida MuIS-i avaandmete ülesehitust.

Sellele järgneb vajalike tehnoloogiate analüüs ja kõige sobivamate tööriistade väljavalimine.

Kõige lõpuks analüüsitakse valminud rakendust ja selle vastavust algsetele eesmärkidele ning pakutakse välja edasisi arendusvõimalusi.

1.4 Ülevaade tööst

Töö teises peatükis antakse teoreetiline ülevaade sellest, mis on avaandmed. Eraldi alapeatükid on pühendatud avaandmetele Eestis ja kõige levinumale avaandmete esitamiseks kasutatud andmemudelile – RDF-ile.

Kolmandas peatükis analüüsitakse sarnaseid veebirakendusi. Valikusse on võetud kaks näidet kultuuripärandi valdkonnast – Ajapaik [2] ja Briti Filmi Instituudi kaardirakendus Britain on Film [3] ning üks andmeid maakondade kaupa kuvav rakendus – Koroonakaart [4].

Töö neljas peatükk on pühendatud rakenduse arendamisele. Selle alapeatükkides kirjeldatakse rakenduse funktsionaalseid nõudeid ja MuIS-i avaandmete struktuuri. Samuti antakse ülevaade arendusel kasutatavatest tehnoloogiatest ja tööriistadest ning kirjeldatakse rakenduse realiseerimise ja selle käigus esile kerkinud suuremaid probleeme.

Viimases, viiendas peatükis analüüsitakse valminud rakendust ja selle vastavust algselt püstitatud eesmärkidele.

2 Avaandmed

Avaandmete käsiraamat defineerib avaandmeid kui andmeid, mida on kõigil võimalik vabalt kasutada, taaskasutada ja edasi jagada – kasutamise ainsateks piiranguteks võivad olla nõue viidata autorile ja edasijagamine originaallikatega samadel tingimustel [5].

Vastavalt avatud teadmuse definitsioonile peavad avaandmed olema:

1. Avalikus omandis või tehtud kättesaadavaks avatud litsentsi alusel
2. Tervikuna kättesaadavad
3. Masinloetavad
4. Avatud vormingus ehk täielikult töödeldavad vähemalt ühel vabal ja avatud lähtekoodil põhineva tööriistaga [6].

2.1 Avaandmed Eestis

Avaandmete temaatikaga tegelemine riiklikul tasemel ulatub Eestis tagasi 21. sajandi esimestesse aastatesse. Avaliku teabe seaduse 15.11.2000. aastal vastu võetud versioonis sätestati, et 2015. aasta 1. jaanuariks peab olema tagatud tasuta juurdepääs andmekogude avaandmetele masinloetaval kujul [7].

2018. aasta veebruaris käivitas MTÜ Open Knowledge Estonia koostöös Majandus- ja Kommunikatsiooniministeeriumiga projekti „Avaandmete kasutamise edendamine“, mille eesmärk on edendada Eestis avaandmete avaldamist ja kasutamist [8].

Projekti üheks tulemiks on Eesti Avaandmete Portaal, mis koondab endas nii avalikustatud andmehulki, kui ka nende põhjal kirjutatud rakendusi. 3. aprilli seisuga oli Eesti Avaandmete Portaalil vabalt kättesaadavad 384 erinevat andmehulka[9] ja nende põhjal valminud 36 rakendust [10].

MuIS-i avaandmed, mis on käesoleva lõputöö alusandmestikuks, on samuti avaldatud Eesti Avaandmete Portaalil.

2.2 RDF

Üks levinumaid andmemudeleid avaandmete esitamiseks on RDF. RDF-i kõige olulisemaks omaduseks on võimalus defineerida seos kahe URI (*Uniform Resource Identifier*, protokoll infoallika üheseks määramiseks veebis) vahel. Kaks ressursi ja nende vaheline seos moodustavad RDF-i kolmiku – subjekt, predikaat ja objekt. Joonisel 1 on kujutatud RDF-i kolmikut. Oma olemuselt on see struktuur suunatud siltidega graaf, mille servad esitavad seost kahe ressursi vahel. [11] RDF võimaldab poolenisti struktureeritud andmed lihtsalt omavahel kombineerida.



Joonis 1. RDF-i kolmik

RDF-i andmemudelite abil kirjeldatud andmete serialiseerimiseks on mitmeid erinevaid formaate. Neist kõige vanem on RDF/XML, mis on XML-il põhinev süntaks RDF-i serialiseerimiseks [12].

3 Sarnaste veebirakenduste analüüs

Andmeid visualiseerivate rakenduste üheks oluliseks eesmärgiks on esitletavast infost võimalikult lihtsalt ja kiiresti ülevaate andmine. Samuti on oluline, et kasutaja saaks mugavalt välja filtreerida just talle huvi pakkuva info.

Käesolevas peatükis analüüsitakse, kuidas sarnased veebirakendused infot kuvavad ja milliseid võimalusi info sorteerimiseks kasutajale pakuvad. Valikusse on võetud kaks näidet kultuuripärandi valdkonnast – Eestis rendatav Ajapaik ja Briti Filmi Instituudi kaardirakendus Britain on Film. Samuti vaadeldakse rakendust Koroonakaart, mis kuvab andmeid maakondade kaupa.

Rakenduste analüüsil on eelkõige arvesse võetud info lihtsat kättesaadavust ja filtreerimisvõimalusi. Kuna Ajapaik ja Koroonakaart põhinevad vähemalt osaliselt avaandmetel, siis on vaadeldud mõlema rakenduse puhul ka seda, kuidas ja milliseid avaandmeid nad kasutavad.

3.1 Ajapaik

Ajapaik keskkonna peamiseks funktsionaalsusteks on võimalus märkida kaardile ajalooliste fotode täpne pildistamiskoht ja -suund ning soovi korral pildistada ka vanad ajaloolised vaated üle. Sel viisil täieneb vabatahtlike kasutajate abiga muuseumide kogudes olevate digiteeritud fotode metainfo ning on võimalik näha ka kohtade muutumist ajas. Joonisel 2 on näha veebirakenduse Ajapaik kaardivaadet.

Käesoleva töö kontekstis on Ajapaik rakenduse juures eelkõige huvipakkuv see, kuidas on pildid kaardil kuvatud ja kuidas on kaardivaates võimalik fotosid sortida. Samuti see, kuidas ja milliseid avaandmeid rakendus kasutab.

Positiivsed küljed

Kui kasutaja lubab rakendusel kasutada oma asukohainfot, siis kuvab kaardirakendus kohe tema ümbruses asuvaid fotosid. Nii ei pea kasutaja asukoha läheduses pildistatud ajalooliste fotode nägemiseks eraldi otsingut tegema.

Kaarti suurendades on visuaalselt näha, palju ühes või teises paigas on fotosid kaardile märgitud. See on eriti mugav nende objektide puhul, mille kohta eksisteerivad suured

fotokogumid. Lisaks numbrilisele väärtusele on eri suurusega fotokogumite eristamiseks kasutatud erinevaid värve.

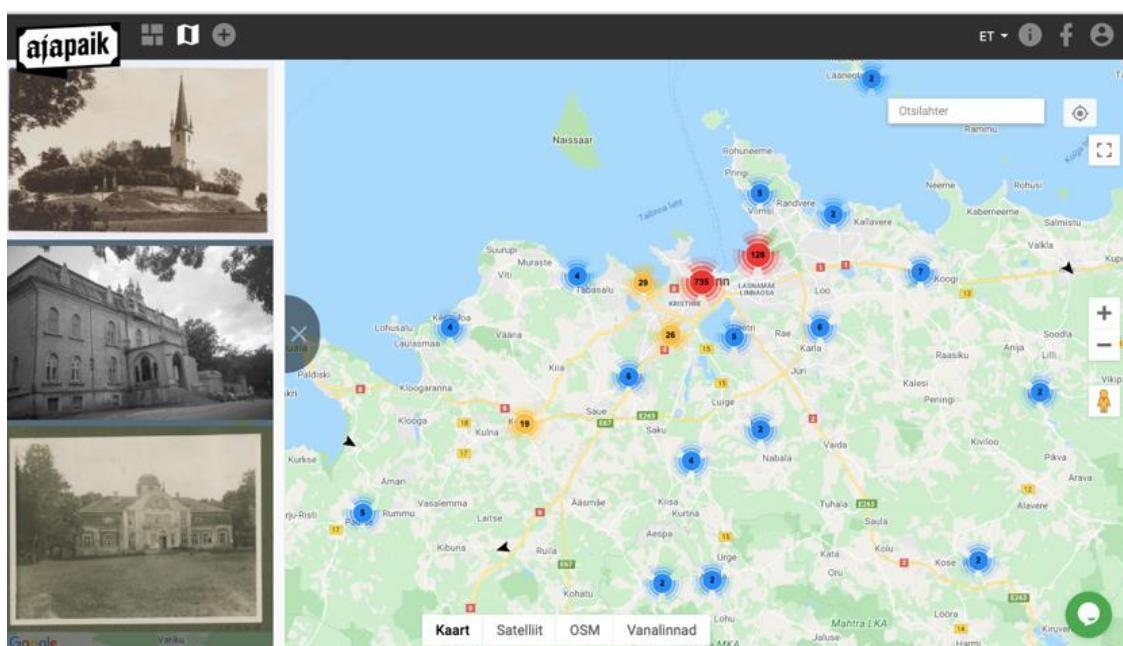
Negatiivsed küljed

Ajapaik on eelkõige mõeldud kasutajale, kes soovib ise piltidele infot juurde lisada neid kaardistades või muul moel märgendades. Rakenduse kaardivaates on võimalik vaadata kas üksnes mõnda albumisse kuuluvaid või kõiki kaardistatud pilte, muid täiendava filtreerimise võimalusi hetkel pole. Otsing ja erinevad sorteerimisvõimalused on võimalikud vaid galeriivaates.

Avaandmed

Kasutajal on võimalik fotosid Ajapaika lisada Rahvusrhiivi fotode andmebaasist Fotis, Soome Rahvusrhiivi andmebaasist Finna, Wikimedia Commonsi, Euroopa digitaalraamatukogust Europeana ja Flickr Commonsi [13].

Ajapaigal on eksisteerinud liidestus ka MuIS-iga OAI (*Open Archives Initiative*, protokoll metaandmete kogumiseks ja koostalitusvõime tekitamiseks) liidese kaudu. Ajapaiga idee autori ja projektijuhi Vahur Puigi sõnul laeti kahel korral kuus andmed OAI-liidese abil Ajapaiga serverisse [14]. Ajapaiga kasutajaliidesesse jõudsid kogutud andmed eraldi Ajapaiga API kaudu, mis väljastas andmeid JSON-formaadis [15]. Hetkel ajutiselt liidestus ei toimi [16].



3.2 *Britain on Film* andmebaasi kaardivaade

Briti Filmi Instituudi (BFI) rakenduse *Britain on Film* peamiseks eesmärgiks on pakkuda paremat ligipääsu nende filmipärandile. BFI-s on hoiul tuhandeid ajaloolisi videoklippe, mis pärinevad erinevatest Suurbritannia osadest. Keskkonda tutvustavaks lauseks on: Vaata filme kohtadest, mis on sinu jaoks olulised. Seetõttu on *Britain on Film* keskkonna oluliseks osaks võimalus kuvada ajaloolisi filme kaardil.

Käesoleva töö kontekstis on eelkõige huvipakkuv see, kui lihtne ja mugav on saada rakenduse kaardivaates üldine ülevaade BFI kogudes leiduvast materjalist.

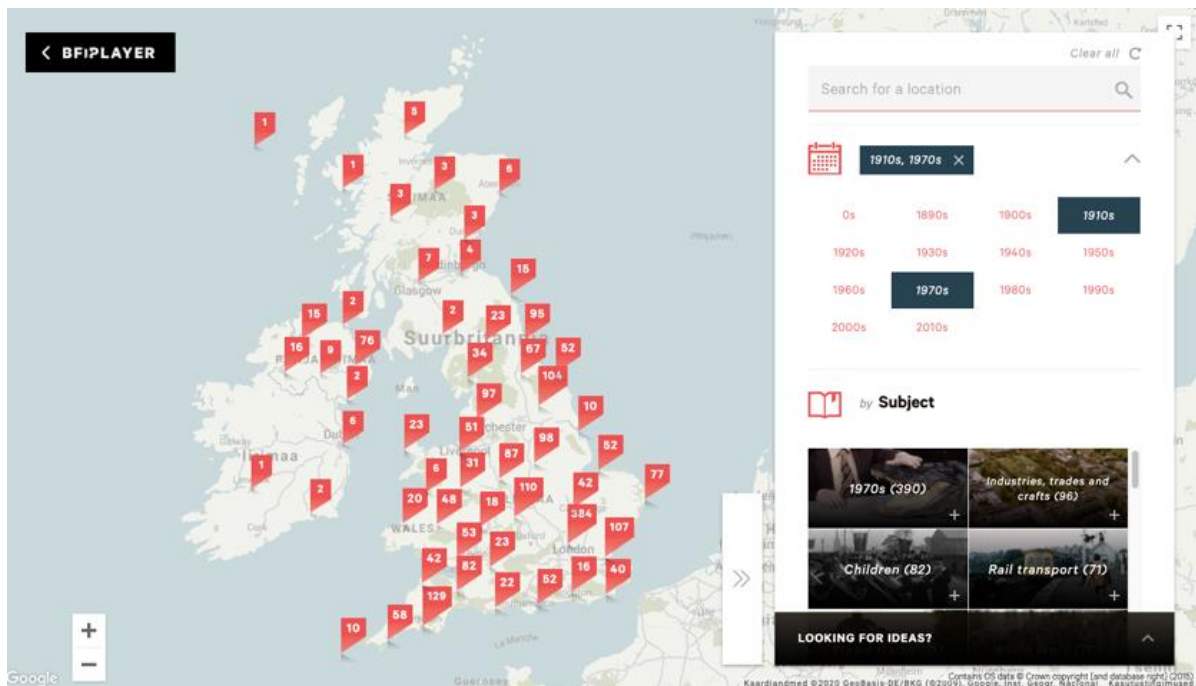
Positiivsed küljed

Kuna *Britain on Film* on oma olemuselt andmebaas, siis väga palju on rõhku pandud filtreerimisvõimalustele. Kaardil kuvatavaid filmiklippe on võimalik filtreerida aastakümne ja teema kaupa. Samuti on väga lihtne erinevaid filtreid omavahel kombineerida.

Sarnaselt Ajapaigale on kaardil numbriliste markerite abil näha, palju leidub ühe või teise koha kohta ajaloolisi filmiklippe. Joonisel 3 on näha *Britain on Film* avavaadet.

Negatiivsed küljed

Britain on Film andmebaasi kaardivaates on ühes piirkonnas filmitud videoklippide arv märgitud numbriliselt, kuid visuaalselt pole võimalik eristada, kas piirkonna kohta on ajaloolisi kaadreid 1 või 1000 – markerite suurus ja värv on mõlemal juhul täpselt samasugune.



Joonis 3. Andmebaasi *Britain on Film* kaardivaade

3.3 Koroonakaart

Koroonakaart on häkatoni raames valminud veebirakendus, mis koondab infot SARS-CoV-2 viiruse leviku kohta Eestis. Käesoleva töö kontekstis on Koroonakaart huvipakkuv eelkõige seetõttu, et ta kuvab kaardil infot kinnitatud haigusejuhtude kohta maakondade kaupa. Seetõttu on vaatluse alla võetud eelkõige see, kuivõrd lihtne on saada kaardilt infot üksikute maakondade kohta. Joonisel 4 on näha Koroonakaardi rakenduse avavaade.

Positiivsed küljed

Visuaalse ülevaate saamiseks aitab kaasa *heatmapi* põhimõte – värvitooni järgi näha, millistes maakondades on viirus rohkem levinud ja millistes vähem.

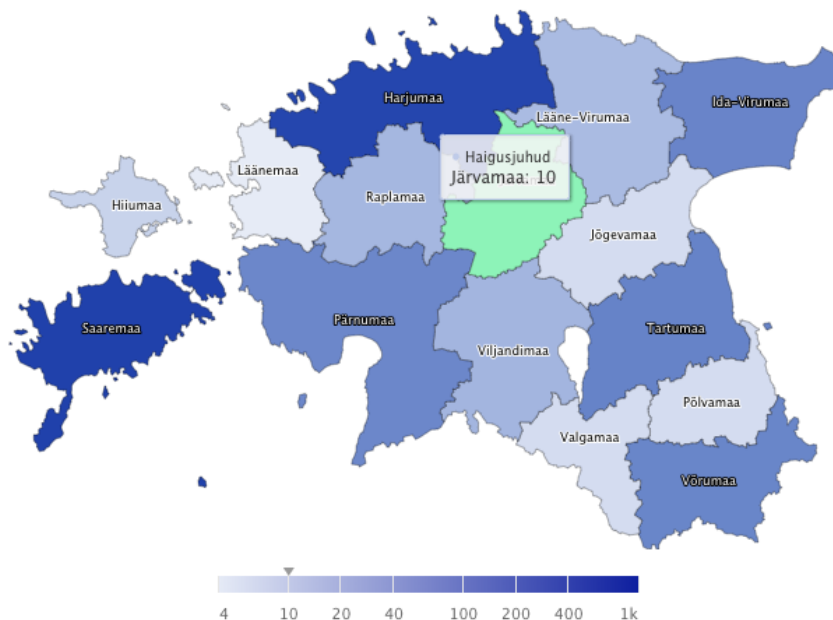
Kui on soovi detailsema info järele, siis hiirega maakonna peale liikudes kuvatakse ka konkreetne number, mitu kinnitatud haigusjuhtu maakonnas on.

Negatiivsed küljed

Rohkem edasiarendusidee kui negatiivse küljena võib välja tuua, et huvitav oleks kaardil maakondade kaupa näha ka andmeid testide arvu ja testide/kinnitatud haigusjuhtude suhtarvu kohta.

Avaandmed

Kõik Koroonakaardi kasutatavad andmed pärinevad Tervise ja Heaolu Infosüsteemide keskusest [17]. SARS-CoV-2 testide andmeid väljastatakse JSON kujul ja CSV vormingus [18].



Joonis 4. Veebirakenduse Koroonakaart avavaade

3.4 Järeldused

Lähtudes sarnaste veebirakenduste analüüsist võib välja tuua kolm olulist funktsionaalsust, mis võiksid andmeid visuaalselt kaardil kuvaval veebirakendusel olla.

Võimalus piirkondi visuaalselt eristada vaadeldavate väärtuste kaupa. Koroonakaart kasutas selleks *heatmapi* põhimõtet, Ajapaiga rakenduses olid suuremaid fotokogumeid tähistavad markerid eristatavad lisaks numbrilise väärtusele ka erinevate värvide abil.

Võimalus lihtsalt jõuda detailsema infoni. Hea näide üldise ja detailse info kombineerimisest oli Koroonakaardi funktsionaalsus, mis kuvas hiirega maakonna peale liikudes kinnitatud haigusjuhtude arvu maakonnas.

Võimalus andmeid filtreerida. Kui kaardil olevatel andmetel on veel metaandmeid, siis annab palju lisandväärtust võimalus andmeid veel omakorda filtreerida. Analüüsitud

rakendustest oli filtreerimisvõimalus kõige paremini realiseeritud Briti Filmi Instituudi andmebaasi *Britain on Film* kaardivaates.

4 Veebirakenduse arendus

Kaardirakendusena realiseeritud veebirakenduse peamiseks eesmärgiks on MuIS-i avaandmetele toetudes anda ülevaade Eesti muuseumites olevate esemete päritolust.

Veebirakenduse funktsionaalsed nõuded töötati välja toetudes sarnaste veebirakenduste analüüsile ja konsultatsioonidele Muinsuskaitseameti ekspertidega.

4.1 Funktsionaalsed nõuded

Tabelis 1 on kirjeldatud arendatava veebirakenduse peamised funktsionaalsed nõuded. Iga nõude juures on välja toodud ka andmed, mis on vajalikud selle nõude realiseerimiseks.

Tabel 1. Rakenduse funktsionaalsed nõuded

Identifikaator	Nõue	Vajalikud andmed
FN_1	Museaalide arv kuvatakse kaardil maakondade kaupa, arvestades museaali päritolu.	Museaalid, mille päritolu on kirjeldatud vähemalt maakonna tasemel.
FN_2	Kaardil kuvatavaid andmeid on võimalik filtreerida valmistamise aja järgi.	Museaalid, mille päritolu on kirjeldatud vähemalt sajandi täpsusega ja vähemalt maakonna tasemel
FN_3	Kaardil kuvatavaid andmeid on võimalik filtreerida museaali tüübi järgi.	Museaalid, mille tüüp on kirjeldatud kui foto, dokument või ese ja mille päritolu on kirjeldatud vähemalt maakonna tasemel

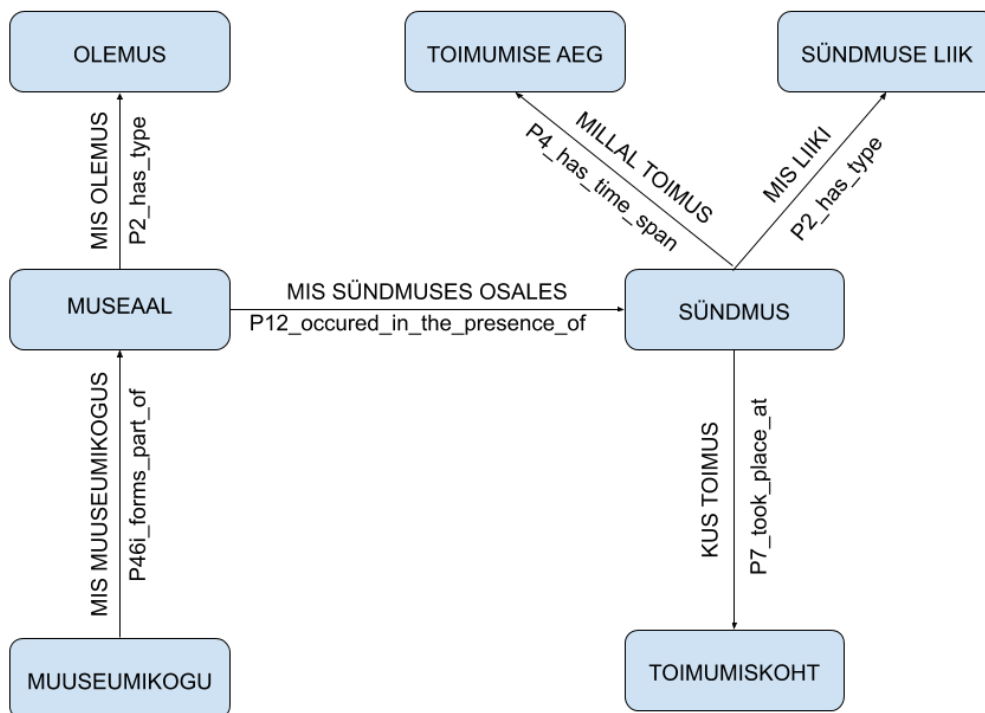
4.2 Muuseumide Infosüsteemi avaandmete struktuuri analüüs

MuIS-i avandmete liides väljastab andmeid RDF/XML formaadis ning andmetes olevaid objekte identifitseeritakse URI-dega.

Muuseumide infosüsteemi avaandmed põhinevad suures osas CIDOC (*International Committee for Documentation*, Rahvusvahelise Muuseumikomitee alamüksus, mis vastutab dokumenteerimise eest) nimeruumil. CIDOC-i kontseptuaalne viitamismudel on mudel, mille abil on nii teoorias kui ka praktikas võimalik kirjeldada kultuuripärandi spetsiifiliste objektide ja kontseptsioonide omavahelisi suhteid [19].

MuIS-i avandmed sisaldavad kõiki andmeid, mis on avalikud ka MuIS-is. Siiski on andmete kättesaamine suuremahuliseks analüüsiks pigem keeruline. Masspäringuna on võimalik pärida ainult muuseumi ja muuseumikogu museaalide nimekirja.

Joonis 5 esitab Museaalide päritolukaardi jaoks vajalike andmete struktuuri MuIS-i avaandmete portaalis.



Joonis 5. Museaalide päritolukaardi jaoks vajalike andmete struktuur MuIS-i avaandmete portaalis

4.2.1 Muuseumi kogusse kuuluvad museaalid

Masspäring konkreetse muuseumi kogusse kuuluvate museaalide kohta on võimalik teha kujul http://www.muis.ee/rdf/objects-by-museum/{muuseumi_id}, mis väljastab kõikide muuseumis kirjeldatud museaalide URI-d.

4.2.2 Üksiku museaali andmed

Üksiku museaali andmeid on võimalik pärida kujul http://opendata.muis.ee/object/{museaali_id}.

Üksiku museaali RDF-fail sisaldab kõiki andmeid, mida selle konkreetse museaali kohta on sisestatud.

Käesoleva veebirakenduse kontekstis on üksiku museaali andmetes olulised kaks väärtust:

Olemus (*P2_has_type*), mis võimaldab filtreerida andmeid museaali tüübi järgi.

Sündmus (*P2_occured_in_the_presence_of*), mis võimaldab saada infot museaali päritolu ja vanuse kohta.

4.2.3 Museaal → Olemus

Olemus on üksiku museaaliga seotud predikaadi *P2_has_type* kaudu. Olemust kirjeldatakse eraldi sõnastiku abiga, mis on osaliselt struktureeritud.

Erinevate tasemete vahel on võimalik liikuda predikaatide *P127i_has_narrower_term* ja *P127_has_broader_term* abil.

4.2.4 Museaal → Sündmus

Sündmus on üksiku museaaliga seotud predikaadi *P12_occurred_in_the_presence_of* kaudu.

Käesoleva rakenduse kontekstis on huvipakkuvad museaaliga seotud sündmused, mille tüüp (*P2_has_type*) on valmistamine ja mille puhul on kirjeldatud toimumise koht (*P7_took_place_at*) vähemalt maakonna tasemel.

Nõude „FN_2. Kaardil kuvatavaid andmeid on võimalik filtreerida valmistamise aja järgi“ täitmiseks on vajalik teada lisaks eelmises lõigus nimetatud andmetele ka sündmuse toimumise aega (*P4_has_time_span*).

Sündmuste tüübid on kirjeldatud sõnastiku abiga, mis on liigendatud hierarhiliselt. Erinevate tasemete vahel on võimalik liikuda predikaatide *P127i_has_narrower_term* ja *P127_has_broader_term* abil.

Toimumise kohad on liigendatud hierarhiliselt. Erinevate tasemete vahel on võimalik liikuda predikaatide *P89i_contains* ja *P89_falls_within* abil.

4.3 Tehnoloogiad ja tööriistad

Arenduse käigus selgus, et rakendusel pole võimalik otse MuIS-i avaandmete portaalist andmeid küsida ning mõistlik on rakendus ehitada üles kahekihiliseks (vt täpsemalt alapeatükk 4.4.). Seetõttu tuli valida sobivad tehnoloogiad nii andmetöötluskihi kui ka kasutajaliidese jaoks.

Valikute tegemisel peeti silmas, et tehnoloogia oleks sobiv veebirakenduse arendamiseks. Kuna rakenduse kasutatavad andmed olid JSON kujul (*JavaScript Object Notation*, andmevahetusvorming, mis põhineb JavaScripti programmeerimiskeelel), siis olid JavaScriptil põhinevad tehnoloogiad siinkohal sobivad.

Oluliseks kriteeriumiks olid ka autori varasemad kogemused vastava tehnoloogia kasutamisel.

4.3.1 Node.js ja Express

Node.js on JavaScriptil käitussüsteem, millega on võimalik kirjutada lihtsalt skaleeritavaid veebirakendusi [20]. Express on Node.js-l põhinev raamistik, mille üheks oluliseks omaduseks on võimalus kiirelt ja vähese vaevaga luua API-sid (*Application Program Interface*, rakendusliides, mille ülesanne on lihtsustada tarkvara arendamist) [21].

4.3.2 React

React on JavaScripti raamistik, mis on mõeldud eelkõige kasutajaliideste loomiseks. Reacti toimimisloogika põhineb komponentidel, mis lasevad kasutajaliidese jagada iseseisvateks ja taaskasutatavateks osadeks [22].

4.3.3 Bootstrap raamistik

Vaadete kujundamiseks valiti Bootstrapi [23] raamistik, kuna see võimaldas kiiresti luua kasutajaliidesele algelise kujunduse. Samuti oli oluline, et tal oli olemas Reacti tugi.

4.4 Realisatsiooni kirjeldus

Rakenduse arendamise käigus selgus, et MuIS-i avaandmete portaalis on andmed väga sügavalt struktureeritud ja vajaliku info kättesaamiseks on vaja teha väga suur hulk päringuid. RDF/XML formaat, milles andmeid väljastati, ei võimaldanud ka päringuid kuidagi optimeerida. Nii võttis isegi väikeste muuseumikogude (kuni 700 museaali) analüüsiks vajalike andmete laadimine ebamõistlikult kaua aega ja oli suureks koormuseks MuIS-i avaandmete portaali serverile.

Kokkuleppel MuISi administraatoriga jõuti lahenduseni, kus rakendus ei suhtle MuISi avaandmete portaaliga otse, vaid kasutab MuIS-i andmebaasist pärinevaid andmeid, mille on käsitsi väljastanud MuIS-i administraator. Andmed on JSON kujul. Lahenduse testimiseks otsustati piirduda vaid Eesti Rahva Muuseumis olevate museaalidega.

Kuna otsepäringutest MuIS-i avaandmete portaali tuli loobuda, osutus otstarbekaks rakendus realiseerida kahekihilisena. Andmekiht, mis vastutab JSON-failist vajalike andmete kättesaamise ning sobival kujul väljastamise eest ja kasutajaliides mis tegeleb andmete kaardikujul eksponeerimise ja kasutaja päringutega.

4.4.1 Andmekiht

Andmekiht väljastab andmeid API abil kahe URL-i kaudu. Üks väljastab valmistamise asukohaandmeid kõikide museaalide kohta ja teine asukohaandmeid kõikide museaalide kohta vastavalt kasutaja valitud filtritele.

Andmekiht saab andmeid kahest JSON-failist. Museaalide asukohaandmeid sisaldav fail sisaldab iga museaali kohta eraldi objekti, kus on kirjeldatud museaali ID, muuseum, museaali olemus, valmistamise alatüüp, maakond ja valmistamise aasta. Joonisel 6 on näha üksiku museaali objekti JSON-failis.

```
{
  "MUSEAAL_ID": 1347746,
  "MUUSEUM": "Eesti Rahva Muuseum",
  "OLEMUS": "foto",
  "SYNDMUS_TYYP": "pildistamine",
  "Maakond": "Harju",
  "ALGUSA": 2006
}
```

Joonis 6. Üksiku museaali objekt JSON-failis

Teine JSON-fail, mida andmekiht kasutab, on autori käsitsi koostatud ja sisaldab ühte objekti, kus objekti võtmeteks on Eesti maakondade nimed sellisel kujul nagu neid oli kasutatud museaalide asukohaandmeid sisaldavas JSON-failis. Väärtuseks on vastavate maakondade koodid. Maakondade koodid pärinevad Maa-Ameti avaandmete põhjal loodud maakonnapiiride GEOJSON-failist, mida kasutab rakenduse kliendipool. JSON kujul andmestik, mis viib kokku Eesti maakondade nimed ja maakondade koodid, on näha joonisel 7.

```
{
  "Harju": "0037",
  "Hiiu": "0039",
  "Ida": "0045",
  "Jõgeva": "0050",
  "Järva": "0052",
  "Lääne": "0056",
  "Põlva": "0064",
  "Pärnu": "0068",
  "Rapla": "0071",
  "Saare": "0074",
  "Tartumaa": "0079",
  "Valga": "0081",
  "Viljandi": "0084",
  "Võru": "0087"
}
```

Joonis 7. Eesti maakondade nimed ja maakondade koodid JSON kujul

URL /allItems

Sellele URL-ile tulnud päringu puhul filtreeritakse museaalide asukohaandmed maakonna kaupa ja tagastatakse objekt. Objekti võtme *total* väärtuseks on kõikide valikus olevate museaalide arv. Võtme *mapData* väärtuseks on objekt, mille võtmed on maakondade koodid ja väärtusteks vastavast maakonnast pärit olevate museaalide arv.

Andmed väljastatakse JSON-formaadis. Joonisel 7 on näha funktsiooni, mis tagastab kaardi loomiseks vajalike väljadega objekti.

```
function returnMapObject(dataToMap, mapKeys) {
  let returnObject = {};
  returnObject.total = dataToMap.length;
  let mapObject = {};
  for (let [key, value] of Object.entries(mapKeys)) {
    let posts = dataToMap.filter(item => (item.Maakond == key));
    mapObject[value] = posts.length;
  }
  returnObject.mapData = mapObject;
  return returnObject;
}
```

Joonis 8. Kaardi loomiseks vajalike väljadega objekti tagastav funktsioon

URL /filteredItems/:type/:period

Sellele URL-ile tulnud päringu puhul filtreeritakse kõigepealt vastavalt parameetrites olevatele väärtusele välja museaalid, mis vastavad päringu tingimustele. Seejärel filtreeritakse saadud andmehulk maakondade kaupa ja väljastatakse saadud objekt JSON-formaadis. Joonisel 8 on näha URL-ile `filteredItems/:type/:period` päringu tegemisel käivitavat koodi.

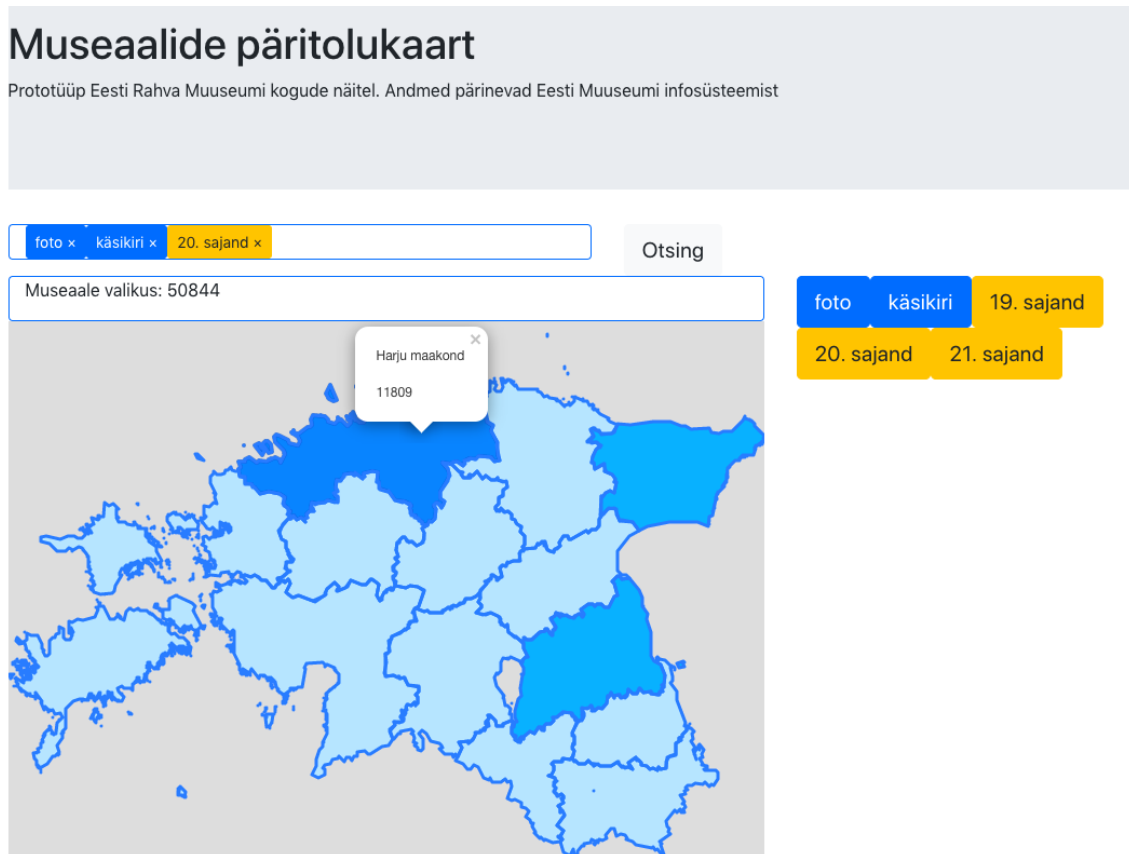
```
app.get('/filteredItems/:type/:period', function (req, res, next) {
  const typeArray = req.params.type !== "null" ? req.params.type.split(",")
  : [];
  const periodArray = req.params.period !== "null" ?
  req.params.period.split(",").map(item => parseInt(`${item}00`)) : [];

  const filtered_museum_data = muuseum_data.filter(item =>
  ((checkValue(item.OLEMUS, typeArray) && (checkCentury(item.ALGUSA,
  periodArray)))));
  const mapData = returnMapObject(filtered_museum_data, maakond_data);
  res.json(mapData);
});
```

Joonis 9. URL-ile `filteredItems/:type/:period` päringu tegemisel käivitav kood

4.4.2 Kasutajaliides

Kasutajaliides on kirjutatud Reacti raamistikus. Lisaks *App* komponendile, mis juhib kogu rakenduse toimimist, on rakendusel veel neli alamkomponenti: *MuseumMap*, *InfoField*, *FilterList* ja *InputField*. Joonisel 9 on näha veebirakenduse Museaalide päritolukaart vaadet.



Joonis 10. Veebirakenduse Museaalide päritolukaart kasutajaliides

MuseumMAP

Komponent võtab parameetritena sisse kaks väärtust – *query*, mis sisaldab URL-i API-st päringu tegemiseks ja funktsiooni *setInfo*, mis annab *App*-i komponendi staatusele (state) *totalCount* uue väärtuse.

Komponent on vastutav rakenduses kaardi kuvamise eest. Kaart luuakse Leaflet teegi [24] abil. Maakonnapiirid genereeritakse kaardile maakonnapiiride andmete põhjal, mis pärinevad Eesti Maa-Ameti Geoportaali haldus ja asustusjaotuse alalehelt [25]. Seda infot hoitakse staatiliselt GEOJSON-failis.

Museaalide asukohainfo visualiseerimiseks kasutatakse API-st päringu vastusena saadud objekti, mis on struktureeritud GEOJSON-i kasutatavate maakonnakoodide kaupa ja selle abil lisatakse GEOJSON-ist pärinevale maakonnainfole väärtus *count*.

Väärtuse *count* põhjal genereeritakse ka kaardile kaks kihti. Esimene kiht määrab vastavalt väärtuse *count* suurusele maakonnale värvi-stiili. Teine kiht vastutab kaardil kuvatavate *pop-up* infoakende eest, mis näitavad maakonnast pärit olevate museaalide täpset arvu.

Kui *query* parameeter muutub (ehk kasutaja teeb uue päringu), siis funktsioon *componentDidUpdate* uuendab kaardiinfot, eemaldab eelmised kaardikihid ja *setInfo* funktsiooni kaudu saadab *App* komponendile uue *totalCount* väärtuse.

InfoField

InfoField on ilma staatusteta (stateless) komponent, mis võtab ainsa parameetrina sisse kaardil kuvatavate museaalide koguarvu.

Komponendi väljakutsumisel on parameetri väärtuseks *App* komponendi staatus *totalCount*. Komponenti *InfoField* kuvatakse ainult siis, kui *App* komponendi staatus *totalCount* on suurem kui null.

FilterList

FilterList on ilma staatusteta komponent, mis võtab parameetritena sisse kaks väärtust – *choices*, mis sisaldab nimekirja võimalikest filtri valikutest ja funktsiooni *addItem*, mis annab *App* komponendi staatusele *queryList* uue väärtuse.

Komponent vastutab võimalike filtreerimisvalikute kuvamise eest. Vastavalt filtri tüübile (museaali tüüp või periood), kuvatakse neid kasutajale erinevates värvitoonides.

Konkreetselt filtri väärtusele vajutades (*onClick*) lisatakse see väärtus *addItem* funktsiooni abil *App* komponendi staatusele *queryList* nimekirja.

InputField

InputField on ilma staatusteta komponent, mis võtab parameetritena sisse kaks väärtust – *choices*, mis sisaldab nimekirja kasutaja valitud filtritest ja funktsiooni *removeItem*, mis annab *App* komponendi staatusele *queryList* uue väärtuse.

Komponent vastutab kasutaja valitud filtrite kuvamise eest. Vastavalt filtri tüübile (museaali tüüp või periood) kuvatakse neid kasutajale erinevates värvitoonides.

Juba valitud filtreid on kasutajal ka võimalik eemaldada. Ristimärgile vajutamisel eemaldatakse filter *removeItem* funktsiooni abil App komponendi staatuse *queryList* nimekirjast.

5 Valminud rakenduse analüüs

5.1 Vastavus esialgsetele eesmärkidele

Sarnaste veebirakenduste analüüsi põhjal sõnastati rakendusele 3 funktsionaalset nõuet. Kõikidele kolmele nõudele valminud rakendus ka vastab. Museaalide arv kuvatakse kaardil maakondade kaupa, arvestades museaali päritolu (funktsionaalne nõue FN_1). Kaardil kuvatavaid andmeid on võimalik filtreerida museaali valmistamise aja (funktsionaalne nõue FN_2) ja museaali tüübi järgi (funktsionaalne nõue FN_3).

Ainus muudatus tuli sisse viia seoses rakenduse kasutatavate algandmetega. Algselt oli planeeritud pärida andmeid otse MuISi avandmete portaalist. Peale MuISi avaandmete struktuuri analüüsi selgus, et MuISi avaandmete portaali väljastatava RDF/XML-i formaat on sobimatu nii mitmel tasandil asuvate andmete optimaalseks kättesaamiseks.

Seetõttu tuli kasutada algandmetena MuIS-i administraatori käest saadud JSON-faili ja realiseerida rakendus kahekihilisena (andmekiht ja kasutajaliides).

5.2 Edasiarendusvõimalused

Rakenduse edasiarendusvõimalused võib jagada kahte peamisse suunda – filtreerimisvõimaluste suurendamine ja funktsionaalsuste juurdetekitamine.

Lisaks olemasolevatele filtreerimisvõimalustele võiks rakenduses tulevikus olla võimalik filtreerida museaalide asukohaandmeid ka alatüüpide järgi, pakkuda võimalust teha perioodi päringuid kümnendite kaupa ja lisada juurde ka täiesti uusi filtreerimiskategooriaid. Arvestades ajaloolist konteksti oleks lisaks maakonna tasemele huvipakkuv filtreerida museaalide päritoluinfot ka kihelkonna tasemel.

Hetkel on tegu eelkõige analüüsitööriistaga muuseumite peavarahoidjatele, Kultuuriministeeriumile ja Muinsuskaitseametile, mille peaesmärgiks on anda ülevaade sellest, kui palju ühest või teisest maakonnast museaale pärineb. Mõeldes koha-ajaloo uurijatele, oleks huvitav pakkuda ka iga maakonna/kihelkonna tasemel nimekirja konkreetsetest museaalidest koos linkidega, mis juhataksid tagasi MuIS-i kasutajakeskkonda.

Kõikide edasiarendusideede realiseerimise üheks oluliseks eelduseks on aga MuIS-i avaandmete portaali täiustamine.

5.3 Ettepanekud MuIS-i avaandmete portaali täiustamiseks

Pikas perspektiivis ei ole rakenduse jaoks algandmete küsimine MuIS-i administraatorilt jätkusuutlik. Samuti oleks edasiarenduste realiseerimiseks vaja saada MuIS-i avaandmete portaalist rohkem ja täpsemaid andmeid. Mõeldes käesoleva rakenduse edasiarendustele ja ka teistele MuIS-i avaandmeid kasutatavatele rakendustele, oleksid vajalikud järgnevad täiendused MuIS-i avaandmete portaalis.

Võimalus ühe päringuga kätte saada rohkem andmeid. Hetkel on ühe päringuga võimalik kätte saada vaid ühe struktuuritaseme andmed. Kuna kohati on MuISi andmed sügavalt struktureeritud, siis päringute arvu vähendamise peamiseks eelduseks oleks ühe päringuga rohkemate andmete kätte saamine.

Täiustada andmete väljastamise formaati. RDF/XML on kõige vanem RDF-i andmemudelite abil kirjeldatud andmete serialiseerimise formaat, mistõttu on tal mitmed piirangud. Vähemalt JavaScripti rakenduste kontekstis oleks otstarbekas, kui lisaks RDF/XML vormingule väljastaks MuISi avaandmete portaal andmeid ka JSON kujul.

Võimalus teha otsepäringuid museaale kirjeldavate sõnastike kohta. Museaalid, nende osad ja nendega seotud sündmused on MuISis kirjeldatud erinevate sõnastike abiga, millest osad on hierarhilise ülesehitusega. Detailsete päringute tegemiseks on vajalik, et lisaks museaale puudutavale päringutele oleks võimalik teha otsepäringuid ka neid kirjeldavate sõnastike kohta.

6 Kokkuvõte

Käesoleva bakalaureusetöö peamine eesmärk oli luua veebirakendus kaardirakendusena, mis annab ülevaate Eesti muuseumites olevate esemete päritolust.

Rakenduse realiseerimisele eelnes sarnaste rakenduste analüüs, mis oli vajalik funktsionaalsete nõuete sõnastamiseks. Samuti anti töös ülevaade MuIS-i avaandmete struktuurist, millele toetudes oli võimalik defineerida, millised andmed on rakenduse toimimiseks vajalikud.

Esialgselt eesmärgist küsida andmeid otse MuISi avaandmete portaalist tuli loobuda ja kasutada algandmetena MuISi administraatori käest saadud JSON-faili. Seetõttu realiseeriti rakendus kahekihiliseks – andmekiht, mis väljastab andmeid API kaudu ning kasutajaliides. Tänu rakenduse kihilisele struktuurile on tulevikus võimalik vastavalt vajadustele ja võimalustele üks või teine rakenduse kiht välja vahetada.

Positiivse poole pealt võib aga välja tuua, et tänu selgunud kitsaskohale MuISi avaandmete portaalist andmete kättesaamisel, loob käesolev lõputöö lisaks algsele püstitatud eesmärgile lisandväärtust pakkudes välja täiendustepanekuid MuIS-i avaandmete portaalile.

Kokkuvõtvalt võib öelda, et käesoleva lõputöö raames valminud rakendus vastab esialgselt seatud eesmärkidele. Rakenduse realiseerimisele eelnenud analüüs pakub aga veel rohkelt ideid rakenduse edasiarenduseks.

Kasutatud kirjandus

- [1] „MuIS püsiviited ja avaandmete liides“. <https://opendata.muis.ee/> (vaadatud 09.05.2020).
- [2] „Ajapaik“. <https://ajapaik.ee/?page=1> (vaadatud 09.05.2020).
- [3] „Britain on film map“, *BFI Player*. <https://player.bfi.org.uk/britain-on-film/map> (09.05.2020).
- [4] „Koroonakaart - Statistika koroonaviiruse leviku kohta Eestis“. <https://www.koroonakaart.ee/et> (vaadatud 09.05.2020).
- [5] „What is Open Data?“ <http://opendatahandbook.org/guide/en/what-is-open-data/> (vaadatud 28.03.2020).
- [6] „Avatud teadmuse definitsioon 2.1 - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge“. <http://opendefinition.org/od/2.1/et/> (vaadatud 03.04.2020).
- [7] „Avaliku teabe seadus – Riigi Teataja“ (15.11.2000). <https://www.riigiteataja.ee/akt/119122012005>.
- [8] „Algas Suurprojekt Avaandmete Edendamiseks“, *Eesti Avaandmete Portaal*. <https://opendata.riik.ee/et/blogi/algas-suurprojekt-avaandmete-edendamiseks/> (vaadatud 03.04.2020).
- [9] „Andmehulgad“, *Eesti Avaandmete Portaal*. <https://opendata.riik.ee/andmehulgad/> (vaadatud 03.04.2020).
- [10] „Rakendused“, *Eesti Avaandmete Portaal*. <https://opendata.riik.ee/rakendused/> (vaadatud 03.04.2020).
- [11] „RDF - Semantic Web Standards“ (W3C soovitus 25.02.2014). <https://www.w3.org/RDF/>.
- [12] „RDF 1.1 XML Syntax“ (W3C soovitus 5.02.2014). <https://www.w3.org/TR/rdf-syntax-grammar/#section-Introduction>.
- [13] „Ajapaik“. <https://ajapaik.ee/curator/> (vaadatud 09.05.2020).
- [14] V. Puik, *Vestlus Slacki keskkonnas*, 05.05.2020.
- [15] „Ajapaiga valimimoodul“. <https://valimimoodul.ajapaik.ee/ajapaik-ui/> (vaadatud 09.05.2020).
- [16] *Ajapaik/ajapaik-harvester*. Ajapaik.ee, 2020.
- [17] „Koroonakaart - Statistika koroonaviiruse leviku kohta Eestis“. <https://www.koroonakaart.ee/et/faq> (vaadatud 09.05.2020).
- [18] „Avaandmed | Terviseamet“. <https://www.terviseamet.ee/et/koroonaviirus/avaandmed> (vaadatud 09.05.2020).
- [19] „Home | CIDOC CRM“. <http://www.cidoc-crm.org/> (vaadatud 04.04.2020).
- [20] Node.js, „About“, *Node.js*. <https://nodejs.org/en/about/> (vaadatud 25.04.2020).
- [21] „Express - Node.js web application framework“. <https://expressjs.com/> (vaadatud 25.04.2020).
- [22] „Components and Props – React“. <https://reactjs.org/docs/components-and-props.html> (vaadatud 11.04.2020).
- [23] „Bootstrap“. <https://getbootstrap.com/> (vaadatud 09.05.2020).
- [24] „Leaflet - a JavaScript library for interactive maps“. <https://leafletjs.com/> (vaadatud 09.05.2020).

[25] Maa-amet, „Haldus- ja asustusjaotus“. index.php?lang_id=1&page_id=119
(vaadatud 09.05.2020).

