

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Joosep Raudväli 185562IAIB
Stiina Salumets 193642IAIB
Triin Sarapuu 194059IAIB

**EESTI KIRJANDUSMUSEUMI RAHVAVIISIDE
INFOSÜSTEEMI TARKVARA EDASIARENDAMINE**

Bakalaureusetöö

Juhendaja: Erki Eessaar
PhD

Tallinn 2023

Autorideklaratsioon

Kinnitame, et oleme 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.

Autorid: Joosep Raudväli, Stiina Salumets, Triin Sarapuu

22.05.2023

Annotatsioon

Eesti Kirjandusmuuseumi rahvaviiside infosüsteemi tarkvara edasiarendamine

Lõputöö eesmärgiks on täiendada Eesti Kirjandusmuuseumi Eesti Rahvaluule Arhiivi rahvaviiside infosüsteemi tarkvara. Täienduste sisuks on luua hästi kasutatav ja ligipääsetav veebipõhine kasutajaliides, sh otsingusüsteem, mis hõlmab rahvaviiside otsingut metaandmete ja meloodia järgi. Samuti on vaja silbitada laulutekst ning paigutada see nootide alla. Rahvaviiside haldusliides on mõeldud arhiivi töötajatele ning otsingusüsteem nii töötajatele kui ka avalikkusele. Töös arendatakse edasi süsteemi, mis loodi kahe varem Tallina Tehnikaülikoolis valminud lõputöö tulemusena.

Töö kõige olulisemaks ülesandeks on otsingusüsteemi koostamine, mis võimaldab nii töötajatel kui ka avalikkuse esindajatel saada hõlpsasti infot rahvaviiside kohta. Eelnevate lõputööde tulemusena valminud otsingusüsteem võimaldas otsida ainult nelja rahvaviisi küljes oleva viite järgi, kuid käesoleva töö tulemusena lisandub võimalus otsida meloodia ja suurema hulga metaandmete järgi. Otsingusüsteemi loomisel järgitakse veebipõhiste otsingusüsteemide parimaid tavasid.

Töö tulemusena valmib kasutajaliides, mis järgib valitud WCAG [1] taseme AA nõudeid, tagades veebilehe kasutatavuse ja ligipääsetavuse.

Valminud töö lähtekood antakse üle Eesti Kirjandusmuuseumile. Töö tegemiseks vajalikud tehnoloogiad valiti Kirjandusmuuseumi töötajaid silmas pidades, et neil oleks võimalik jätkata süsteemi edasiarendamist.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 44 leheküljel, 8 peatükki, 46 joonist, 4 tabelit.

Abstract

Further Development of the Folk Tunes Information System Software of the Estonian Literary Museum

The aim of the thesis is to improve and extend the software of the folk tunes information system of the Estonian Folklore Archives of the Estonian Literary Museum by creating an easy-to-use and accessible web-based user interface, including a search system. The search system allows users to search folk tunes by metadata and by melody. The system is able to syllabify the song text and place it under notes. The user interface for managing metadata about the tunes and the search system is meant for the archive staff. Moreover, the search system will also be available to the general public. The work further develops the system that was created as the result of two theses at Tallinn University of Technology.

The most important task of the thesis is to create a web-based search system that will allow both staff and users to easily find information about folk tunes. As a result of the previous theses, the search system only allowed searching by four references (text, tune, sound, and video references). However, this thesis will add the possibility to search folk tunes by the tune and by a larger set of metadata. The search system will be built following the best practices of web-based search systems.

The result of this work will be a user interface that complies with the selected WCAG [1] requirements, ensuring the usability and accessibility of the website.

The source code of the completed work will be handed over to the Estonian Literary Museum. The technologies required to complete the work were chosen with the Literary Museum staff in mind, allowing them to continue to develop the system. The application is created in the PHP programming language. The database management system is PostgreSQL.

During the course of the work, information on best practices in search systems was collected, and existing sound search systems were analyzed. By creating a more accessible and efficient search system, this work contributes to the preservation of Estonian folk music heritage.

The thesis is in Estonian and contains 44 pages of text, 8 chapters, 46 figures, 4 tables.

Lühendite ja mõistete sõnastik

ABC notatsioon	Süsteem muusika noodistamiseks lihttekstina. [2]
API	<i>Application Programming Interface</i> . Rakendusliides, programmiliides. Arvuti operatsioonisüsteemiga või rakendusprogrammiga määratud reeglistik, mille alusel rakendusprogramm kasutab operatsioonisüsteemi või teise rakendusprogrammi teenuseid. [3]
CSS	<i>Cascading Style Sheets</i> . Kaskaadlaadistik. Veebilehtede valmistajatele ja kasutajatele mõeldud laadistik, mis koosneb reeglitest, mis teatavad veebilehitsejale, kuidas dokumenti kuvada. [3]
EKM	Eesti Kirjandusmuuseum.
ERA	Eesti Rahvaluule Arhiiv.
HTML	<i>HyperText Markup Language</i> . Hüpertekst-märgistuskeel. Enimlevinud kodeerimissüsteem (tekstivorming) veebidokumentide loomiseks. [3]
Metaandmed	Andmeid kirjeldavad andmed ehk andmed andmete kohta. [4]
PDO	<i>PHP Data Objects</i> . PHP andmeobjektid.
SQL	<i>Structured Query Language</i> . Relatsioonilise andmemudeli alusel loodud andmebaasikeel andmete-, andmebaasiobjektide-, andmebaasi kasutamise õiguste- ja andmebaasis toimuvate tehingute haldamiseks.
URL	<i>Uniform Resource Locator, Universal Resource Locator</i> . Internetiaadress, võrguaadress. [5]
Viide	ERA arhivaaride poolt viisidele määratud identifikaatorid, mida on nelja tüüpi: viisiviide, tekstiviide, heliviide ja videoviide.
WCAG	<i>Web Content Accessibility Guidelines</i> . Veebisisu ligipääsetavuse suunised.
XSS	<i>Cross-Site Scripting</i> . Saidiülene skriptimisrännak.
ZIP	Failide ja kataloogide arhiveerimiseks mõeldud kadudeta andmetihendusvorming.

Sisukord

1	Sissejuhatus	12
1.1	Taust ja probleem	12
1.2	Töö kirjeldus	12
1.3	Töö struktuur	13
2	Arendusmetoodika	14
2.1	Ülevaade objektist	14
2.2	Ülevaade töö protsessist	15
2.3	Tööriistade kirjeldus	16
2.4	Üleminek PHPle	17
3	Otsingusüsteemide parimad tavad	19
3.1	Täiendavate filtrite kuvamine	19
3.1.1	Filtrite valik peale otsingut	19
3.1.2	Lihvitud otsing	20
3.2	Otsingu käivitamine	20
3.3	Otsingutulemuste kuvamine	21
3.3.1	Otsingutulemused samal või uuel lehel	21
3.3.2	Lehekülgedeks jagamine	21
3.4	Ligipääsetavus	22
4	Olemasolevate otsingusüsteemide analüüs	24
4.1	FindSounds	24
4.1.1	Materjalide maht	25
4.1.2	Otsingu funktsionaalsus	25
4.1.3	Kasutajaliidese disain	25
4.1.4	Tehniline jõudlus	25
4.1.5	Ligipääsetavus	26
4.2	Audioburst	26
4.2.1	Materjalide maht	26
4.2.2	Otsingu funktsionaalsus	26
4.2.3	Kasutajaliidese disain	26
4.2.4	Tehniline jõudlus	27
4.2.5	Ligipääsetavus	27
4.3	Shazam	27

4.3.1	Materjalide maht	27
4.3.2	Otsingu funktsionaalsus	27
4.3.3	Kasutajaliidese disain	28
4.3.4	Tehniline jõudlus	28
4.3.5	Ligipääsetavus	28
4.4	Audio Archive	28
4.4.1	Materjalide maht	28
4.4.2	Otsingu funktsionaalsus	29
4.4.3	Kasutajaliidese disain	29
4.4.4	Tehniline jõudlus	29
4.4.5	Ligipääsetavus	29
4.5	Folk tune finder	29
4.5.1	Materjalide maht	29
4.5.2	Otsingu funktsionaalsus	30
4.5.3	Kasutajaliidese disain	30
4.5.4	Tehniline jõudlus	30
4.5.5	Ligipääsetavus	30
5	Tulemused	31
5.1	Nõuded	31
5.1.1	Funktsionaalsed nõuded	31
5.1.2	Mittefunktsionaalsed nõuded	31
5.2	Kasutajaliidese disain	33
5.3	Otsingusüsteem	33
5.3.1	Vabaotsing	34
5.3.2	Viisotsing	34
5.3.3	Kombineeritud otsing	35
5.4	Tagarakenduse disain	40
5.4.1	Otsingusüsteem	40
5.5	Andmebaasi disain	41
5.6	Silbitamine	43
5.7	Turvalisuse tagamine	44
5.7.1	SQL süstimine	45
5.7.2	Saidiülene skriptimisrännak	45
5.8	Koodi testimine	46
5.8.1	Ühiktestimine	46
5.8.2	Päringute testimine	46
5.8.3	Kasutajaliidese testimine	47
5.9	Juurutamine	47

6	Tulemuste analüüs	48
6.1	Kasutajatepoolne testimine	48
6.2	Otsingutulemuste kuvamise kiirus	48
6.3	Võrdlus olemasolevate süsteemidega	49
6.3.1	Materjalide maht	49
6.3.2	Otsingu funktsionaalsus	50
6.3.3	Kasutajaliidese disain	50
6.3.4	Tehniline jõudlus	50
6.3.5	Ligipääsetavus	51
6.4	Võrdlus parimate praktikatega	51
7	Arendusvaade	53
7.1	Heli tuvastamine	53
7.2	Visualiseeritud otsing	53
7.3	Kõnesüntesaator	53
7.4	Tarkvara ühtlustamine	53
7.5	Mobiilivaade	54
8	Kokkuvõte	55
	Kasutatud kirjandus	56
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	61
	Lisa 2 – Andmebaasi indekse loomise laused	62
	Lisa 3 – Andmebaasipäringute näiteid	63
	Lisa 4 – Kavandi kuvatõmmised	67
	Lisa 5 – Figma prototüüp	69
	Lisa 6 – Kasutajaliidese kuvatõmmised	72
	Lisa 7 – Vabaotsingu koodinäide	83
	Lisa 8 – Algne projektiplaan	84
	Lisa 9 – Lõplik projektiplaan	86

Jooniste loetelu

1	INTERSECT operaatoriga päringu näide.	36
2	Liitpäringu näide.	37
3	Päring viisi otsimiseks esitaja nime järgi.	38
4	Ebasobiv päring viisi otsimiseks esitaja ja koguja nime järgi.	39
5	Sobiv päring viisi otsimiseks esitaja ja koguja nime järgi.	40
6	Tabelid maakondade ja kihelkondade kohta andmete hoidmiseks.	41
7	Näide ettevalmistatud päringust.	45
8	Näide turvalisest küpsisest.	46
9	Välisvõtmete veergudele lisatavad indeksid.	62
10	Funktsioonil põhinev indeks.	62
11	Otsing esitaja järgi.	63
12	Otsing esitaja ja koguja järgi.	64
13	Otsing esitaja ja laululiigi järgi.	65
14	Otsing kahe esitaja järgi.	66
15	Kavandi avaleht.	67
16	Kavandi otsinguleht.	67
17	Kavandi otsingutulemuste leht.	68
18	Kavandi viisiinfo leht.	68
19	Prototüübi avaleht.	69
20	Prototüübi otsinguleht.	70
21	Prototüübi otsingu abi leht.	70
22	Prototüübi otsingutulemuste leht.	71
23	Prototüübi sisselogimise leht.	71
24	Rakenduse avaleht.	72
25	Rakenduse otsinguleht, sisselogimata.	72
26	Näide otsingust kihelkonna ja kogumise algusaasta järgi.	73
27	Rakenduse otsinguleht peidetud lisafiltritega.	73
28	Rakenduse detailne viisivaade.	74
29	Rakenduse viisivaade lauluteksti, helifaili ja noodistuse andmetega.	74
30	Rakenduse viisivaade, täpsema info kuvamine.	75
31	Rakenduse viisivaade, täpsema info kuvamine.	75
32	Rakenduse isikute vaade.	76
33	Rakenduse ühe isiku vaade.	76
34	Rakenduse sisselogimise leht.	77

35	Rakenduse avaleht, sisselogitud.	77
36	Rakenduse otsinguleht, sisselogitud.	78
37	Rakenduse viisivaade, sisselogitud.	78
38	Rakenduse viisivaade, täpsem info, sisselogitud.	79
39	Rakenduse viisivaade, täpsem info, sisselogitud.	79
40	Rakenduse viisivaade, täpsem info, sisselogitud.	80
41	Rakenduse viisivaade, täpsem info, sisselogitud.	80
42	Rakenduse viisivaade, täpsem info, sisselogitud.	81
43	Rakenduse viisivaade, täpsem info, sisselogitud.	81
44	Rakenduse klassifikaatorite vaade.	82
45	Rakenduse klassifikaatorite vaade, kindla klassifikaatori vaade.	82
46	Materialiseeritud vaate loomine.	83

Tabelite loetelu

1	Mittefunktsionaalsed nõuded.	32
2	Viisi meloodias sisalduvate nootide vahede leidmise sammud.	35
3	Andmebaasi lisatud tabelite <i>Counties</i> ja <i>County_parishes</i> veergude sõn- nalised kirjeldused.	41
4	Tabelisse <i>Tune_melodies</i> lisatud veeru sõnaline kirjeldus.	43
5	Andmebaasipäringute käsitsi testimise testjuhud.	46
6	Kasutajaliidese integratsioonitestid.	47
7	Otsingutulemuste leidmise kiiruse võrdlus andmebaasisüsteemi ja kasuta- jaliidese vahel.	49
8	Otsingusüsteemide tehnilise jõudluse võrdlus.	50

1. Sissejuhatus

Selles jaotises kirjeldatakse töö tausta, probleemi, eesmärgi ja tuuakse välja töö struktuur.

1.1 Taust ja probleem

Eesti rahvaviiside uurimine ja säilitamine on oluline kultuuripärandi ja rahvusliku identiteedi jaoks, kuna see aitab säilitada traditsioone, kultuurilisi väärtusi ja keelt. Eesti Kirjandusmuuseumi (edaspidi EKM) Eesti Rahvaluule Arhiiv (edaspidi ERA) on riigi üks peamisi asutusi, mis kogub, talletab ja uurib rahvaviise. ERA rahvaviiside infosüsteem on üks selle eesmärgi saavutamise abivahend.

Seda infosüsteemi on varasemalt arendatud kahe Tallinna Tehnikaülikooli lõputöö raames: 2020. aastal loodi rahvaviiside andmebaas ning andmebaasirakenduse prototüüp [6]. 2021. aastal arendati seda edasi [7] ühtse nootide kodeerimise standardi juurutamise, rakenduse uuel platvormil realiseerimise ja sinna uute funktsionaalsuste lisamise näol.

Süsteemi suurimad puudujäägid 2023. aasta alguse seisuga on kasutajaliidese ligipääsetavus ja otsingusüsteemi võimalused. Enne käesoleva töö tegemise algust eksisteeriv otsingusüsteem võimaldab rahvaviise otsida viidete järgi, aga muude metaandmete järgi mitte. Näiteks ei saa kasutajad otsida viise esitaja või kogumise kuupäeva järgi.

Olemasolev kasutajaliides on loodud eelkõige EKM-i töötajate tarbeks ning avalikkusel puudub sellele ligipääs. Lisaks kasutab kasutajaliides tehnoloogiaid, mille hooldamiseks ja edasiarendamiseks vajalikke oskuseid EKM-i IT-osakond ei oma.

1.2 Töö kirjeldus

Käesoleva lõputöö eesmärk on täiendada ERA rahvaviiside infosüsteemi tarkvara, luues veebipõhise avalikkusele suunatud kasutajaliidese rahvaviiside kohta info otsimiseks ning töötajatele suunatud haldusliidese viisiandmete haldamiseks.

Töös keskendutakse järgmiste ülesannete täitmisele.

- Avalikkusele mõeldud veebipõhise kasutajaliidese loomine viiside otsimiseks ja nende kohta käiva informatsiooniga tutvumiseks.

- Töötajatele mõeldud veebipõhise haldusliidese loomine, mis võimaldab viiside andmeid hallata.
- Valmivad kasutajaliidesed peavad olema hästi kasutatavad ja ligipääsetavad ning peavad täitma WCAG taseme AA nõudeid.
- Otsingusüsteem peab võimaldama viiside otsingut metaandmete ja meloodia järgi. Viisi kohta otsingut tehes kuvatakse laulu elemendid, milleks on lauluviis (koos kuulamise võimalusega), tekst, originaali helisalvestus, peamised metaandmed ja võimalus valikuliselt vaadata muid andmeid.
- Lauluteksti silbitamine ja selle paigutamine nootide alla. Varasemalt loodud kasutajaliideses kuvatakse teksti viisi all silbitamata kujul. Eesmärk on kuvada silp vastava noodi all. See tulemus kuvatakse otsingu tulemusena saadud viisi elemendina.

Töö tulemusena soovivad autorid luua süsteemi, mis võimaldab ERA-l paremini toetada rahvaviiside uurimist ja säilitamist, pakkudes kultuurihuvilistele veebilehte, kus nad saavad viiside kohta infot. Kasutajaliidese kavandamisel järgitakse disaini parimaid tavasid [8], mis aitavad saavutada meeldivat kasutajakogemust.

1.3 Töö struktuur

Teises peatükis kirjeldatakse arendusmetoodikat, sh töö tegemise protsessi ning kasutatud töövahendeid. Kuna süsteemi täienduse oluliseks osaks on otsingusüsteem, siis annab kolmas peatükk ülevaate otsingusüsteemide parimatest tavadest ja neljas peatükk analüüsib mõningaid olemasolevaid viiside otsingusüsteeme. Viiendas peatükis kirjutatakse töö tulemustest, sh esitatakse nõuded ja kirjeldatakse realisatsiooni. Kuuendas peatükis neid tulemusi analüüsitakse. Kirjutatakse kasutajate poolsest testimisest, otsingute töökiirusest ning loodud süsteemi võrreldakse olemasolevate süsteemide ja parimate praktikatega. Seitsmes peatükk kirjutab sellest, kuidas süsteemi edasi arendada.

2. Arendusmetoodika

Käesolevas peatükis antakse ülevaade objektist, töö protsessist ning kasutatud tööriistadest.

2.1 Ülevaade objektist

EKM on teadus- ja arendusasutus, mis haldab eesti kultuuriloo arhiive, mille hulka kuulub ka ERA. ERA on 1927. aastal asutatud folklooriarhiiv, mille ülesandeks on rahvaluule kogumine, säilitamine ja laiemale üldsusele kättesaadavaks tegemine. [9]

Arhiivid on dokumentaalne kõrvalsaadus inimeste tegevusest, mida säilitatakse nende pikaajalise väärtuse pärast. Erasisikute ja organisatsioonide poolt nende tegevuse käigus loodud dokumendid annavad võimaluse tulevastel põlvedel vaadata minevikku. [10]

ERA haldab ligi kahe sajandi vältel kogutud folkloorimaterjale. Sinna hulka kuuluvad poolteist miljonit lehekülge käsikirju, tuhanded fotod ning heli- ja videosalvestused. Lisaks eesti rahvaluulele sisaldavad arhiivi kogud ka Eesti rahvusvähemuste folkloorikogusid ja materjale, mis on kogutud sugulasrahvastelt. [11]

Käesolev lõputöö on jätk kahele varasemale lõputööle, mida kaitsti Tallinna Tehnikaülikoolis vastavalt 2020. ja 2021. aastal. [6, 7] Tööde tulemusena on valminud rahvaviiside andmete haldamise infosüsteem, mis vahendab muusikapärimust ERA kogudest. Võimalik on vaadata viiside noodistusi ning kuulata helisalvestusi. Viiside juurde on lisatud üldandmed ja muusikalised tunnused. Andmebaas sisaldab üle 4000 kaherealise vormiga regiviisi. Kaugemaks eesmärgiks on sinna lisada kõik kogutud eesti rahvaviisid, mida on hinnanguliselt üle poolesaja tuhande.

Paraku ei ole praegune infosüsteem avalikkusele ligipääsetav ning seda kasutavad oma igapäevatoos ainult ERA töötajad.

Lõputöö [6] tulemusena loodi viiside haldusliidese prototüüp kasutades PHP keelt ja CakePHP raamistikku.

Lõputöö [7] tulemusel realiseeriti eelmises infosüsteemi tarkvara versioonis realiseeritud funktsionaalsus uuel platvormil (Node.js, Loopback 4 ja React). Sellele tehnoloogiale üleminek toimus tollaste lõputöö tegijate soovitusel. Seega enne käesoleva töö algust oli

realiseeritud haldusliides, kus tagarakendus oli tehtud Node.js ja eesrakendus Reacti abil. Enne käesoleva töö algust oli realiseeritud otsingusüsteem, kus rahvaviise sai otsida nelja viite (heli-, teksti-, video- ja viisiviide) alusel. Ka see oli realiseeritud viisil, et tagarakendus oli tehtud Node.js'is ja eesrakendus Reactis.

EKM sooviks rakenduse puhul tagasi minna PHP kasutamisele, sest sellist tarkvara on neil lihtsam hallata ja edasi arendada võrreldes rakendusega, mis kasutab Node.js'i ja Reacti.

Tagarakenduse, mis on tehtud Node.js abil, ümbertegemine oleks väga mahukas töö. Seega jõudsid töö autorid tellijaga kokkuleppele, et tagarakendus jääb Node.js'i, kuid eesrakendus realiseeritakse PHP's. See tähendab, et üheks töö ülesandeks oli varem Reactis realiseeritud haldusliidese ümbertegemine PHP-le. Otsingusüsteem realiseeritakse nii ees- kui tagarakenduse osas PHPs.

2.2 Ülevaade töö protsessist

Projekti tellijaks on EKM ning selle aluseks on olemasolev rahvaviiside infosüsteem. Koostöös tellijaga pandi paika töö skoop ning planeeritav funktsionaalsus.

Projekti elluviimiseks loodi käesoleva töö autoritest koosnev meeskond. Liikmete rollid ja tööjaotus kujunesid nende soovide ning oskuste põhjal. Stiina Salumetsa vastutusala oli otsingusüsteemi kasutajaliidese prototüübi loomine ja realisatsioon. Triin Sarapuu ülesandeks oli otsingusüsteemi funktsionaalsus. Joosep Raudväli tegeles peamiselt rakenduse haldusliideselega. Iga meeskonnaliige vastutas oma töö testimise ja dokumenteerimise eest.

Meeskonnasisese infovahetuse edendamiseks toimusid kolm korda nädalas tiimikoosolekud, kus iga liige andis ülevaate tehtud tööst. Samuti kasutati seda aega, et arutada tekkinud probleemide üle ning jooksvalt kohandada projektiplaani. Algne projektiplaani on esitatud lisa 8 ja lõplik projektiplaani lisa 9.

Tellijaga peeti iganädalasi videokoosolekuid, mille käigus esitleti tehtud tööd ning saadi tagasisidet. Lisaks toimus tellijaga tihe meilivahetus ajakriitilisemate küsimuste lahendamiseks.

Lõputöö tulemus on kombinatsioon tellija soovidest ning autorite endi ideedest, mida erialastele teadmistele tuginedes tellijale välja pakuti.

Töö valmis järgides Scrum metoodikat, mida töö autorid vastavalt oma vajadustele muundasid. Järgnevalt on välja toodud mõned olulisemad Scrumi praktikad, mida lõputöö

tegemisel kasutati. [12, 13, 14]

1. **Iseorganiseeruv meeskond:** Meeskonnas puudus kindel hierarhia, liikmed planeerisid oma tegevust iseseisvalt ja vastutasid selle eest.
2. **Backlog:** Tegevused jaotati alamülesanneteks.
3. **Ajaga piiratud arendusetapid:** Töö jagati kahenädalasteks iteratsioonideks ehk sprintideks ning määrati tegevused, mis iga sprindi lõpuks valmima peaksid.
4. **Empiiriline protsessijuhtimine:** Scrum kasutab iteratiivset ja järkjärgulist lähenemist, mis võimaldab projektiplaani täitmise sagedast kontrollimist ning selle vajaduspõhist kohandamist. Kui töö käigus ilmnes, et mingi ülesanne enne vastava sprindi lõppu valmis ei saa, siis lükkus selle täitmine edasi järgmisesse sprinti (vt algne projektiplaani lisas 8 ja lõplik projektiplaani lisas 9).

Töö kirjutamisel kasutati allikana Tallinna Tehnikaülikooli õppeaine "Andmebaasid II" näiteprojekti "Ülikooli infosüsteemi vastuvõtuaegade allsüsteem". [15]

2.3 Tööriistade kirjeldus

Omavahelise kiirsuhtluse jaoks oli kasutusel Facebook Messenger. Koosolekuid nii meeskonna, tellija kui juhendajaga peeti Microsoft Teamsi vahendusel.

Failide hoidmiseks valisid töö autorid Google Drive pilveteenuse, mis võimaldas tööprotsessi jooksul tekkivaid märkmeid mugavalt kirja panna ning omavahel jagada. Töö lõplik dokument valmis LaTeXi süsteemis ning seda kirjutati Overleafi keskkonnas.

Süsteemi edasiarendamiseks oli vaja arenduskeskkonnas luua andmebaasi koopia ja laadida sellesse andmed. Käesoleva töö autorid kasutasid *docker-compose* faili, et Dockerit kasutades luua andmebaasi koopia. Docker valiti seetõttu, et sellega oli töö autoritel varasem kogemus ja eelnevate tiimidega suheldes soovitati just seda varianti.

Lähtekoodi lokaalseks käivitamiseks oli vajalik PostgreSQL andmebaasisüsteemi, Dockeri ja Node.js olemasolu autorite arvutites. Tarkvaraarenduseks kasutati IntelliJ IDEA integreeritud arenduskeskkonda, millega töö autorid on varasemast tuttavad. Andmebaasiga töötamiseks oli kasutusel graafilist kasutajaliidest pakkuv pgAdmin 4 programm.

Rakenduse kasutajaliidese prototüüp loodi Figma abil. Lõplik kasutajaliides valmis peamiselt PHP, JavaScripti, HTML ja CSS põhjal. Stiili ja skaleeruvuse parandamiseks kasutati lisaks Bootstrap 5 raamistikku.

Koodi testimiseks valiti PHPUnit ja MochaJS raamistikud. Need raamistikud on ühed populaarseimad valikud vastavalt PHP ja JavaScripti testimiseks, samuti on töö autoritel eelnev kogemus nende kasutamisel.

Töö käigus valminud koodi hoiti GitLabi repositooriumis. Projekti halduseks valiti Jira, mille abil jagati tööülesandeid ja jälgiti projekti kulgu. Jira valiti seetõttu, et tellija avaldas soovi saada meie ülesannetele ligipääs, kuid Tallinna Tehnikaülikooliga seotud GitLabi repositooriumile ei saa kooliväliste isikutele ligipääsu võimaldada.

Käesolevas dokumendis esitatud andmebaasidiagrammid loodi veebirakendusega dbdiagram. [16]

Mittefunktsionaalsetele nõutele vastamise kontrollimiseks kasutati erinevaid pistikprogramme:

1. Wave [17] vaegnägijate ligipääsetavuse hindamiseks
2. Colorblindly [18] värvide kontrastsuse hindamiseks
3. Page load time [19] lehekülje laadimisaja hindamiseks
4. Open Chrome DevTools [20] samuti laadimisaja hindamiseks

2.4 Üleminek PHPl

Varasemalt oli rakendus kirjutatud Node.js ja Reacti abil. See üleminek toimus eelmise lõputöö [7] tulemusel ning selle tegijate soovitusel. EKM-i üks soovide oli, et käesolevas töös valmiv tarkvara kirjutatakse PHP-s, sest EKM-i IT osakond ei ole Node.js ja Reacti kasutamisel piisavalt pädev. Seetõttu on olemasoleva süsteemi haldamine olnud nende jaoks keeruline. Töö autorid nõustusid PHP-d kasutama, kuna kõigil töö tegijatel oli sellega varasem kogemus olemas, samas Node.js ja Reactiga oli kokkupuude minimaalne.

PHP pakub mugavat võimalust integreerida loodav kasutajaliides rahvaviiside andmebaasidega tänu PHP heale ühilduvusele. [21] Samuti on PHP-l laialdane kogukond ja toetus erinevatele raamistikele ja tööriistadele.

PHP miinuseks on selle vanus, mistõttu on aeglustunud uute raamistike väljatöötamine. Samuti on suurem turvariskide oht, mida töö autorid meeles peavad.

Käesolevas töös asendatakse esirakendus täielikult PHP ja JavaScriptiga ning tagarakendus osaliselt. Kuna varasemalt Node.js-is loodud tagarakenduse suhtlus esirakendusega käib peamiselt API-de kaudu, saab olemasolevat tagarakendust mugavalt kasutada nii

avalikkusele mõeldud kasutajaliidese kui ka haldusliidese puhul.

3. Otsingusüsteemide parimad tavad

Enamik otsingusüsteemide parimatest tavadest on kirjutatud ühe filtri järgi otsingut pakkuvate süsteemide alusel, kus on vaid vabaotsing ja täpsem otsing on heal juhul lisavalik. Selles töös loodava süsteemi eesmärk on võimaldada otsingut paljude erinevate filtrite alusel. Seega tuli leida parimaid tavaid täpsemate otsingute kohta.

Antud peatükis analüüsitakse täpsema otsingu häid tavaid [8] ning nende sobivust arendatavasse süsteemi.

3.1 Täiendavate filtrite kuvamine

Täiendavate filtrite kuvamisel on kasutajaliidese heade tavade mõttes kaks valikut.

3.1.1 Filtrite valik peale otsingut

Otsingusüsteemide puhul, kus kasutaja üldjuhul kasutab vaid tavaotsingut, on parim lahendus kuvada esmalt kasutajale vaid tavaotsingu võimalus ning minimaalne arv lisafilteid. Pärast otsingutulemuste kuvamist saab anda kasutajale ka võimaluse otsingutulemusi täiendavate filtrite alusel kitsendada.

Sellel lahendusel on järgnevad plussid.

- + Lihtsus. Vähemate filtritega otsingut on kergem tundma õppida ja kasutaja kohaneb sellega kiiremini. Kõigi valikute kohesel kuvamisel tõuseb otsingusüsteemi keerukus, mis võib kasutajaid heidutada. [22]
- + Vajadus eelteadmiste järgi on väiksem. Uued kasutajad ei pruugi teada, milliseid filtreid nad soovivad kasutada. Otsingusüsteemi algvalikuid piirates tõuseb kasutajamugavus vähemkogenenud kasutajate jaoks.

Sellel lahendusel on järgnevad miinused.

- Kasutaja ei saa kohe soovitud otsingut teha. Kogenenud kasutaja soovib kohe sooritada rohkemate filtritega otsingut. Sammude lisamine soovitud otsingutulemuse saavutamiseks võib tekitada kasutajates ärritust või tüdimust.
- Täiendavate filtrite peitmine esimese otsingu taha aeglustab soovitud tulemuse

saamist. See on oluline EKM-i töötajate jaoks, kes on süsteemi ühed põhilised kasutajad. Taoline otsingusüsteem aeglustaks tunduvalt nende tööd.

3.1.2 Lihvitud otsing

Lihvitud otsing on üks otsingusüsteemi tüüpe, mis võimaldab kasutajatel oma otsingutulemusi kitsendada, jagades otsingufiltrid eri kategooriate alla. [23] Lihvitud otsing on kasulik suuremate andmehulkade korral, kuna rohkem andmeid tähendab ka tõenäoliselt suuremat hulka filtreid, mille jaoks saab pakkuda kasutajale mugavat lahendust.

Sellel lahendusel on järgnevad plussid.

- + Kasutajal on võimalus kohe spetsiifiline otsing sooritada. Otsingutulemused on sellisel juhul asjakohasemad ja süsteemi paindlikkus on suurem, sest jääb ära täiendav samm võrreldes filtrite kuvamisega pärast esmast otsingut. Kasutajatele, kes teavad, mida nad otsida soovivad, ei näidata ebavajalikke tulemusi.
- + Soovitud otsingutulemuseni jõudmine muutub kiiremaks. Kiiremini kasutatav süsteem on kasutaja jaoks mugavam. EKM-i töötajate jaoks on kiirus oluline, kuna päevas sadu otsinguid tehes on ajavõit tuntav.

Sellel lahendusel on järgnevad miinused.

- Otsinguvorm muutub keerukamaks. Vähemkogenenud kasutajale võib see tekitada segadust ja teha orienteerumise raskemaks.
- Lihvitud otsingut on keerulisem teostada. Rohkem filtreid muudab otsingu teostamise süsteemile kulukamaks, mistõttu võib otsingutulemuste kuvamine kauem aega võtta.

Keeruka otsingu puhul on see variant eelistatud, kuna soov on tagada kasutajale võimalikult sujuv kasutajakogemus. Sellisel juhul soovib kasutaja enne otsingu sooritamist enamasti panna kirja mitu erinevat kitsendust ning oleks häiritud, kui ta peab selleks avama uue vaate. Samuti tekitaks kasutajas segadust see, et kasutajaliideses pole kohe välja toodud kõiki otsinguvõimalusi.

3.2 Otsingu käivitamine

Olenevalt andmemahust ning keskmisest rakendatavate filtrite arvust tuleb otsustada, kas otsingutulemused kuvatakse jooksvalt või pärast sisestusklahvi vajutamist.

Jooksvalt kuvatavate otsingutulemuste eeliseks on, et kasutajad saavad otsingutulemusi kiiremini näha ning saavad kohandada oma päringuid vastavalt saadud tagasisidele. See sobib väiksemate andmemahtude puhul, kui otsingutulemuste kuvamine ei võta kaua aega.

Sisestusklahvi vajutamise nõude puhul on otsinguprotsessi käivitamine kasutajale selgem. See vähendab süsteemi koormust, kuna päringuid tehakse vaid siis, kui kasutaja on valmis tulemusi saada. Mitu otsingut järjest tehes võib kasutaja soovida, et otsingufiltreid valides ei muutuks juba kuvatud andmed. Sisestusklahvi vajutamise variant on suure otsingu puhul kasutajasõbralikum, sest pärast iga otsingutingimuse sisestamist ei pea kasutaja ootama andmete laadimist, et otsingutulemuste kitsendamisega jätkata, vaid saab otsingu sooritada ühe korraga. Suuremate andmemahtude puhul võib otsingutulemuste kuvamine kauem aega võtta ja sage otsingute ootamine jätab kasutajale süsteemist kohmaka mulje.

3.3 Otsingutulemuste kuvamine

Pärast filtrite täpsustamist on kasutajakogemuse mõttes oluline otsingutulemuste kuvamine. Tähtis on kasutajale vastused selgelt kuvada, et nad saaksid võimalikult kiiresti soovitud tulemuseni jõuda.

3.3.1 Otsingutulemused samal või uuel lehel

Otsingutulemused saab kuvada kas samal lehel, kus otsing sooritati, või suunata kasutaja edasi eraldi otsingutulemuste lehele. Suunates kasutaja uuele lehele, peab kasutaja filtrite muutmise soovi korral pöörduma tagasi eelmisele lehele, mille tulemusena eelmiseid otsingutulemusi enam ei kuvata. See oleks kasutajale ebamugav ja aeglustaks tööprotsessi. Samal lehel otsingutulemuste kuvamise korral ta seda tegema ei pea.

Kui otsingutulemused on samal lehel koos filtrite valikuga, siis tuleb rohkem arvestada elementide paigutusega leheküljel. Nii tulemused kui filtrid peavad ühele lehele ära mahtuma, mis muudab kasutajaliidese kavandamise keerulisemaks. Tulemuste kuvamine eraldi lehel jätab kasutajaliidese elementidele rohkem ruumi, mis annab võimaluse andmeid selgemalt välja tuua.

3.3.2 Lehekülgedeks jagamine

Suure tulemuste mahu korral tuleb otsustada, kas need kuvatakse keritava järjendina ühekorraga või nummerdatud lehekülgede kaupa.

Tulemuste korraga kuvamise eelis on, et kõik andmed on ühel lehel näha. Kasutajale, kes seda varianti eelistab, annab see parema võimaluse kasutada veebilehitseja "Otsi lehel" funktsiooni. Tulemuste lehekülgede kaupa kuvamise puhul oleks seda tülikam ja aeganõudvam kasutada, sest otsing tuleb igal leheküljel uuesti teha.

Tulemuste jagamine eraldi lehekülgedele pakub suuremat paindlikkust. Suurte andmemah-tude korral ei muutu tulemuste kuvamise leht liiga pikaks. Kasutajale saab anda valiku, kui mitut otsingutulemust ta korraga ühel lehel näha soovib. Nii saavad kasutajad, kes eelistavad näha korraga palju andmeid, vastavalt vajadusele seda muuta.

3.4 Ligipääsetavus

Ligipääsetavus on oluline aspekt igas tarkvarasüsteemis, sealhulgas otsingusüsteemides. See tagab, et kõik kasutajad, sealhulgas erivajadustega kasutajad, saavad süsteemi mugavalt ja tõhusalt kasutada.

Otsingusüsteemi disain peaks järgima veebistandardeid nagu WCAG, et tagada süsteemi lai ühilduvus erinevate veebilehitsejate, seadmete ja tehnoloogiatega. Nende juhiste järgimine muudab süsteemi kasutatavaks vaegnägijatele, kes kasutavad veebisisule ligi pääsemiseks abivahendeid nagu ekraanilugejad ja suurendusfunktsioonid.

Visuaalse kujunduse juures on oluline värvikontrastsus teksti ja tausta vahel, et tekst oleks nägemispuudega inimestele võimalikult loetav. [24] Fondi valiku koha pealt on tähtis, et font oleks lugemisraskuste vältimiseks selge ja loetav.

Kasutajaliidese disain peab olema loogiline. Elementide paigutus peab järgima järjepidevuse põhimõtet, et samad elemendid on erinevatel veebilehtedel samal kohal. [25] Sel juhul saab kasutaja eeldada, kust soovitud elementi otsida. Elementide paigutamine ebatavalisse kohta raskendab vaegnägijatel nende leidmist.

Vaegnägijatele on oluline ka heli. Teksti-sõnumi funktsioonid, mis loevad otsingutulemusi ja muud olulist teavet ette, võimaldavad vaegnägijatel süsteemi kasutada ilma eraldi abiva-henditeta. Lisaks võib helikujundus, näiteks heliefektid nuppude vajutamisel või erinevate otsingufunktsioonide aktiveerimisel, anda vaegnägijatele lisainformatsiooni süsteemi kasu-tamise kohta ja muuta kasutajakogemuse intuiitsemaks.

Ligipääsetavus on otsingusüsteemi kavandamisel oluline, et tagada süsteemi kasutatavus kõigile kasutajatele, sõltumata nende võimetest ja erivajadustest. Selle jaoks tuleb arvestada erinevate kasutajate vajadusi, tagades selge ja intuiitvise kasutajaliidese, värvikontrasti,

fondi valiku, loogilise elementide paigutuse ja helikujunduse.

4. Olemasolevate otsingusüsteemide analüüs

Maailmas leidub juba mitmeid erineva otstarbega viisiotsingute võimalust pakkuvaid süsteeme. Antud peatüki eesmärk on analüüsida viit erinevat otsingusüsteemi ja tuua välja nende puudused ja eelised. Iga süsteemi kohta vastatakse järgnevalt välja toodud küsimustele. Loodavat süsteemi võrreldakse tulemuste analüüsi käigus siin väljatoodud süsteemidega nende küsimuste vaates.

- Milline on otsinguga kaetud materjalide maht?
 - Leitakse võimalike otsingutulemuste arv ning võrreldakse seda antud andmeliigis olevate andmete koguarvuga, et hinnata, kui suur võimalus on kasutajal soovitud tulemus leida. Süsteemi töödeldavate andmete maht mõjutab ka süsteemi jõudlust.
- Milline on otsingu funktsionaalsus?
 - Hinnatakse, kui lihtne ja efektiivne on otsingufunktsioonide ja veebilehe kasutamine.
- Kui hea on kasutajaliidese disain?
 - Vaadatakse, kui lihtne on navigeerida, kas järgitakse häid disainipraktikaid, kui palju kasutatakse visualiseerimist ja kui kooskõlaline on kasutajaliides.
- Milline on süsteemi tehniline jõudlus?
 - Google Chrome pistikprogrammiga *Page load time* [19] hinnatakse lehekülgede laadimisaega. Arvestatades, et 40% kasutajatest lahkub veebilehelt, kui laadimine võtab kauem kui kolm sekundit [26], siis hinnatakse, kas veebileht laeb piisavalt kiiresti. Laadimisajad on leitud ühes ja samas Wi-Fi võrgus, et võimaldada täpsemat võrdlust lehekülgede vahel.
- Kas süsteemi ligipääsetavus on piisavalt hea?
 - Hinnatakse, kas süsteem on vaegnägijatele ja vaegkuuljatele [27] kättesaadav. Ligipääsetavust vaegnägijatele hinnatakse kahe Google Chrome pistikprogrammi abil: *Wave* [17] üldlase ning *Colorblindly* [18] erinevate värvipimeduse sümptomite all kannatajate ligipääsetavuse hindamiseks.

4.1 FindSounds

Findsounds [28] on erinevate heliefektide leidmiseks mõeldud otsingusüsteem. Otsingut on võimalik sooritada märksõna abil otsingukastis või läbi kategooriate hüperlinkide abil.

4.1.1 Materjalide maht

Veebilehel puudub ülevaade heliefektide arvust, mida on võimalik otsida, kuid uurides erinevate kategooriate sisu ja kategooriate arvu, saab teha umbkaudse arvutuse: lehel on välja toodud 22 kategooriat, mille all on omakorda keskmiselt 25 alamkategooriat ja iga alamkategooria all keskmiselt 50 heliefekti. Seega on ligikaudne efektide arv 27 500, mis on piisav väiksemate projektide jaoks. Samas, arvestades, et näiteks "Star Wars: Episode IV - A New Hope" film kasutab üle tuhande heliefekti [29], ei pruugi seal suurema projekti jaoks piisavalt palju spetsiifilisi heliefekte leiduda.

4.1.2 Otsingu funktsionaalsus

Otsingu realisatsioon on lihtne ning märksõna järgi saab tulemuse kätte väga kiiresti. Otsingutulemused on 10 kaupa lehekülgedeks jaotatud ning iga leitud efekti kohta on esmaselt vajalik info kohe kuvatud. Heli on võimalik mängida ja alla laadida, lisaks on antud ka helifaili päritoluleht. Väga kasulik funktsionaalsus on sarnaste helide leidmine, mida saab käivitada vastava nupuga iga otsingutulemuse kohta. Negatiivse poolena on otsingutulemuste kohta vähe infot, samuti puudub otsingutulemuste täiendava filtreerimise ja sorteerimise võimalus.

Kokku on võimalik otsida kolme erineva tunnuse järgi.

4.1.3 Kasutajaliidese disain

Kasutajaliidese poolest on veebileht aegunud disainiga, sest ei järgita kaasaegseid disainipraktikaid. Mõned näited nõrkadest kohtadest: menüü asub veebilehe alaservas, logo ikoonile vajutades ei liiguta tagasi avalehele, puudub skaleeritavus, otsingutulemusi ei saa täiendavalt filtreerida. Sellest hoolimata on veebileht lihtsasti navigeeritav ning otsingutulemused on lihtsasti kättesaadavad ning hästi visualiseeritud.

4.1.4 Tehniline jõudlus

Jõudluse poolest toimib leht väga hästi, sest otsingut sooritades kuvatakse tulemused kiiresti ning helifaili avamine on samuti kiire ning lihtne. Tehes veebilehel erinevaid otsinguid ja jälgides laadimisaega, ei läinud ühegi otsinguga aega kauem kui 0.7 sekundit.

4.1.5 Ligipääsetavus

Veebilehe ligipääsetavus on kehv, sest kasutatakse värve, mis pole kontrastsed ja pole seega näiteks värvipimedatele inimestele hästi eristatavad. Ekraanilugeri tööd takistavad otsinguväljadel puuduvad sildid, veebilehel määramata keel, struktuuri puudumine ning HTML tabelite kasutamine. Vaegkuuljate ligipääsetavuseks on iga tulemuse kohta värviline heli lainekuju, mille järgi saab hinnata heli kõrgust, tugevust ja amplituudi [30], puuduvad aga subtiitrid, mis võimaldaks vaegkuuljatel näha ka heli tekstilist sisu.

4.2 Audioburst

Audioburst [31] on platvorm heli otsimiseks [32] ja edastamiseks. See on maailma esimene tehisintellektil põhinev heliotsingu platvorm.

4.2.1 Materjalide maht

Veebilehe andmetel lisandub süsteemi päevas 8500 uut helifaili [33], mis tähendab, et ühes aastas lisanduvate helifailide arv on ligikaudu kolm miljonit. See on märgatavalt väiksem võrreldes näiteks ülemaailmse taskuhäälingute hulgaga, mida ilmub aastas ligikaudu 28 miljonit [34] episoodi. Arvestades, et taskuhääling on vaid üks osa *Audiobursti* heliliikidest, on erinevate allikate maht lehel üpris väike ning kasutaja ei pruugi soovitud tulemust leida.

4.2.2 Otsingu funktsionaalsus

Tänu tehisintellektile toimib märksõna järgi otsing väga hästi ning kuvatakse isegi positiivset, mis on vaid mõned minutid vanad. Lisaks on võimalus kuvada tulemusi kindlate kategooriate kaupa, valides avalehel soovitud variandi.

Kokku on võimalik otsida kolme erineva tunnuse järgi.

4.2.3 Kasutajaliidese disain

Üldiselt on veebilehe disain puhas ja kaasaegne, otsinguriba on lehe ülaosas ja kasutajatele koheselt kättesaadav. Lisaks on otsinguribas olemas abistav näidistekst. Täpsema otsingu filtrid asuvad otsinguriba all, otsingutulemuste kohta on lühike info olemas ning peale vajutades saab nende kohta edasi lugeda.

4.2.4 Tehniline jõudlus

Otsingu tehniline jõudlus on veidi aeglane, kuid jääb soovitatava ajalimiidi sisse - enamikuga otsingutest läks aega kolm sekundit. Otsingutulemuse peale vajutades kulub tulemuse laadimiseks keskmiselt kaks sekundit. Kasutajana tundus laadimiskiirus piisavalt kiire ja ei häirinud veebilehe kasutamist, kuid arvestades, et aeglasema internetiühendusega võib laadimisaeg pikeneda, võib see muutuda probleemiks.

4.2.5 Ligipääsetavus

Audiobursti ligipääsetavus vaegkuuljatele on väga hea, sest iga heli kohta on olemas transkriptsioon, mis võimaldab vaegkuuljatel pääseda ligi heli sisule. Probleemina võib välja tuua aga selle, et heli hakkab automaatselt mängima, mistõttu võib tekkida olukord, kus vaegkuulja külastab veebilehte avalikus kohas ja saab tahtmatut tähelepanu, millest ta ise teadlik pole. Vaegnägijate ligipääsetavus on aga halb: värvidel puudub kontrast, nuppudel ja linkidel puuduvad kirjeldused ja sildid ning leheküljel kasutatakse *noscript* elemente, mis pole ekraanilugeritele kättesaadavad.

4.3 Shazam

Shazam [35] on kõige laialdasemalt kasutatav laulutuvastuse rakendus, mis võimaldab lihtsasti leida laulu, mida kasutaja ümbruses mängitakse. Rakendus kuulab ümbritsevat heli ning tagastab laulu, juhul kui see on nende andmebaasis olemas.

4.3.1 Materjalide maht

Süsteemi hallatavate andmete maht on päris suur. 2021. aasta andmetel oli nende andmebaasis üle 15 miljoni laulu [36] ning neid lisandub pidevalt juurde läbi muusikafirmade, mis lisavad uute laulude ilmumisel need *Shazami* süsteemi [37]. Hinnates, et maailmas on üle 82 miljoni laulu [38], on *Shazamis* neid siiski vähe. Seega vähemtuntud laule ei pruugi rakendus üles leida. Arvestades aga sellega, et suurimad muusikafirmad teevad *Shazamiga* koostööd, leiab rakendus suurema osa populaarsematest teostest siiski üles.

4.3.2 Otsingu funktsionaalsus

Heli otsimise funktsioon on intuitiivne ning tulemuseni jõuab ilma suurema vaevata. *Shazami* algoritmi [36] miinuseks on, et see tuvastab laule ainult nende originaalfaili järgi, kuna otsing töötab läbi laulu sõrmejäljest kõrgemate helisageduste otsimise, mis ei pruugi

kattuda ümisedes või otse esitatud lauluga.

Kokku on võimalik otsida ühe tunnuse järgi.

4.3.3 Kasutajaliidese disain

On näha, et *Shazami* otsingu funktsionaalsus on esmatähtis. Otsing on lihtne ja kasutajal ei ole vaja teha muud kui vajutada nuppu. Laulu leides kuvatakse esmane info ja sellega seotud lingid. Kokkuvõttes on otsingusüsteemi kasutajaliides väga lihtne ja kasutajasõbralik.

4.3.4 Tehniline jõudlus

Antud rakenduses ei olnud võimalik pistikprogrammi kasutada, sest pistikprogramm ei tuvastanud laadimisaega ning näitas seda ainult veebilehe värskendamisel. Seega analüüsi laadimisaega kasutades *Open Chrome DevTools* [20] võimalust. Katsetades erinevaid laule, läks otsingutulemuse leidmiseks aega 7-9 sekundit, mis on kindlasti kauem kui soovitatav aeg.

4.3.5 Ligipääsetavus

Vaegnägijate perspektiivist puuduvad kontrastsed värvid, pilte kirjeldavad tekstid ja nuppude kirjeldused. Lisaks kasutatakse nuppudena *aria* elemente, mis tähendab, et nuppude kasutamiseks pole võimalik kasutada klaviatuuri. Vaegkuuljate jaoks on ligipääsetavus hea, sest tulemuses kuvatakse laulu sõnad ja laul ei hakka kohe mängima.

4.4 Audio Archive

Audio Archive [39] on digitaalse raamatukogu *Internet Archive* [40] helisalvestuste osakond.

4.4.1 Materjalide maht

Helisalvestusi on süsteemis ligi 15 miljonit, mille hulka kuuluvad kontsertide salvestused, raadiosaated, audioraamatud ja palju muud. Arhiivina on neil ka suures mahus vanu helifaile.

4.4.2 Otsingu funktsionaalsus

Kui mitte arvestada veebilehe tehnilist jõudlust, millest räägitakse alajaotises Tehniline jõudlus 4.4.4, on otsing kasutajasõbralik ning intuiitiivne.

Kokku on võimalik otsida kaheksa erineva tunnuse järgi.

4.4.3 Kasutajaliidese disain

Kasutajaliidese disain on veebilehel lihtne ja üldjoontes parimaid tavaid järgiv: leht on visuaalselt puhas, on olemas täpsema otsingu võimalus ja lehel on lihtne navigeerida. Negatiivsena võib välja tuua, et täpsema otsingu valikuid on liiga palju ja otsinguriba ei ole lehel kõige paremini välja toodud ning võib seega kasutajale kahe silma vahele jääda.

4.4.4 Tehniline jõudlus

Audio Archive tulemuste laadimiseks kulus erinevate otsingutega 6-8 sekundit, millele lisandus veel kaks sekundit, mis läks filtrite laadimisele. Antud laadimisaeg loob ebameeldiva kasutajakogemuse ning kindlasti põhjustab kasutajate arvu langust.

4.4.5 Ligipääsetavus

Antud otsingusüsteemi ligipääsetavusel on palju probleeme: värvid pole piisavalt kontrastsed, paljudel pildidel puuduvad kirjeldavad tekstid, otsinguväljadel puuduvad sildid, piltide kirjeldused kattuvad, suur osa tekstist on liiga väike ja kasutusel on ka *noscript* element. Vaegkuuljate jaoks on ainuke ligipääsetav info heli kohta lainekuju, kuid ei ole olemas subtiitreid. Positiivsena saab välja tuua ka selle, et heli ei lähe automaatselt käima.

4.5 Folk tune finder

Folk tune finder [41] on rahvaviiside otsingusüsteem, mis võimaldab otsida erinevat päritolu rahvaviise.

4.5.1 Materjalide maht

Veebilehel on vastust võimalik leida 190 000 [41] erineva rahvaviisi seast. Rahvaviisid pärinevad ABC notatsioonis [2] kodeeritud viisidest, mis on süsteemi üles laetud ning läbi

ABC notatsiooni veebilehe omakorda *Folk tune finder* otsingusüsteemi lisatud. Kahjuks ei ole võimalik rahvaviiside täpset arvu ennustada, kuna antud muusikavorm kandub edasi suuliselt ning on alles tänapäeval kirjalikule kujule kantud. Leheküljel ringivaatamise järel võib järeldada, et enamik viisidest on inglise- või saksakeelsed. Lisaks leiti ka mõned Skandinaavia keeltes esitatud viisid. Seega võib järeldada, et süsteemis on leitav suur hulk inglise- ja saksa päritoluga rahvaviise, kuid teiste rahvaviiside hulk on vähene.

4.5.2 Otsingu funktsionaalsus

Otsing on lihte ja intuitiivne, kuna viisiotsingu tarbeks on loodud võimalus viis sisestada visuaalse klaveri klaviatuuriga. Selle lähenemise miinus on see, et kasutajalt eeldatakse eelnevaid muusikalisi teadmisi.

Kokku on võimalik otsida 11 tunnuse järgi.

4.5.3 Kasutajaliidese disain

Hea kasutajakogemuse jaoks on otsinguriba asetatud kergesti leitavasse kohta, otsingutulemused on selged ja leheküljel on lihte navigeerida. Negatiivsena võib tuua välja, et täpsema otsingu parameetreid on ebamugav kasutada.

4.5.4 Tehniline jõudlus

Folk tune finder otsingukiirus on parem kui teistel võrreldavatel süsteemidel, keskmiselt läks otsingule 1.5 sekundit. Kõigi katsetatud otsingute korral jäi tulemuste leidmine alla kahe sekundi. Antud tulemus on väga hea ja aitab kindlasti kaasa meeldivale kasutajakogemusele.

4.5.5 Ligipääsetavus

Võrreldes teiste vaadeldud otsingusüsteemidega on veebilehe ligipääsetavus hea, sest värvid on kontrastsed, kirjeldus puudub vaid ühel lingil ja ühel pildil ning leht on hästi struktureeritud. Vaegkuuljatel on võimalik viisi kohta infot saada noodistikust ning subtiitrid pole tarvis, kuna viisidel puuduvad sõnad. Samuti ei hakka heli tulemuse peale vajutades automaatselt kõlama.

5. Tulemused

Selles peatükis tuuakse välja töö tulemused.

5.1 Nõuded

Selles jaotises tuuakse välja nõuded süsteemile. Esitatakse funktsionaalsed nõuded, mis väljendavad soove süsteemi käitumise osas ja mittefunktsionaalsed nõuded, mis väljendavad soove süsteemi omaduste osas.

5.1.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded süsteemile esitatakse kasutuslugudena, mis järgivad [42] välja pakutud formaati. Nende abil tuuakse välja nõuded otsingusüsteemile. Haldusliidest luues saab funktsionaalsete nõuete mõttes täielikult toetuda varem valminud töödele [6, 7] ja neid nõudeid siinkohal ei korrata.

1. Mina kui rahvamuusika entusiast, tahan otsida viise nime või sõnade järgi, et leida kindlat viisi.
2. Mina kui muusikaajaloo õpilane, tahan otsida viise regiooni järgi, et uurida erinevate regioonide muusikat.
3. Mina kui muusikaajaloo õpilane, tahan otsida viisi kogumisaasta järgi, et saada parem ajalooline ülevaade kogutud viisidest.
4. Mina kui muusik, tahan näha noodikirja, et saaksin õppida viisi ka ise mängima.
5. Mina kui Eesti Kirjandusmuuseumi töötaja, tahan otsida viise metaandmete järgi, et leida spetsiifilisi viise, mida mul on töö jaoks vaja.

Kõikvõimalikud otsingufiltrid peavad olema kättesaadavad kõigile otsingusüsteemi kasutajatele.

5.1.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded süsteemile esitatakse tabelis 1.

Tabel 1. Mittefunktsionaalsed nõuded.

Nõue	Nõude kirjeldus
WCAG tase	<p>Süsteem peab järgima järgmisi WCAG nõudeid [1]:</p> <ul style="list-style-type: none"> ■ Veebisaidi struktuur on loogiline. ■ Kogu veebilehe teksti peab olema võimalik muuta (nii suurendada kui vähendada) kuni kaks korda originaalsest suurusest, ilma, et veebilehe sisu loetavus või funktsionaalsus väheneks. ■ Veebilehel ei tohi midagi vilkuda rohkem kui kolm korda sekundis. ■ Kõik interaktiivsed elemendid peavad võtma vähemalt 24×24 CSS-pikslit ruumi. See võib sisaldada ka valget ruumi sihtmärgi ümber. ■ Veebisaidil ei tohi sisu automaatselt muutuda ainult seetõttu, et kasutaja sisestab teksti, märgistab märkeruudu või valib midagi rippmenüüst. ■ Sisestusvea automaatsel tuvastamisel hoiatatakse kasutajat, kirjeldatakse viga ja antakse juhiseid selle parandamiseks.
Päringutele vastamise kiirus	<p>Otsingusüsteem peab töötleva päringuid võimalikult kiiresti ja efektiivselt, et tagada sujuv toimimine. See tähendab, et süsteem peab olema optimeeritud suure hulga päringute töötlemiseks lühikese aja jooksul.</p>
Turvalisus	<p>Süsteem peab olema kaitstud erinevate turvariskide realiseerumise eest. See hõlmab nii andmebaasi turvalisust, kasutajate teabe kaitsmist kui ka võimet seista vastu erinevatele küberohtudele. Tähelepanu pööratakse eriti SQL süstimise ja saidiülele skriptimisrännaku ennetamisele, mida kirjeldatakse jaotises 5.7. Andmete tervikluse ja konfidentsiaalsuse puhul on nõudeks saavutada järgnev tase [43]:</p> <ul style="list-style-type: none"> ■ Andmete tervikluse tase T1: info muutmise ja hävitamise fakt peavad olema tuvastatavad; info õigsuse, täielikkuse ja ajakohasuse kontroll erijuhtudel ja vastavalt vajadusele. ■ Andmete konfidentsiaalsuse tase S0: avalik info, juurdepääsu teabele ei piirata

Nõue	Nõude kirjeldus
Töökindlus	<p>Süsteem peab olema töökindel, et tagada kasutatavus kõikidele kasutajatele. Veajuhtumite korral, näiteks andmebaasiga ühenduse katkemise korral, peab süsteem kuvama ettenähtud veateadet. Töökindluse puhul on nõudeks saavutada järgnev tase [43]:</p> <ul style="list-style-type: none"> ■ Töökindluse tase K1: lubatud summaarne seisak nädalas ööpäev); lubatav nõutava reaktsioonaja kasv tippkoormusel – tunnid (1÷10).
Hooldatavus	<p>Süsteem peab olema kergesti mõistetav, muudetav ja testitav. See tähendab, et kood peaks olema korralikult dokumenteeritud ja järgima puhta koodi põhimõtteid. Süsteem peab olema loodud nii, et seda saaks tulevikus uuendada ja laiendada. [44]</p>

5.2 Kasutajaliidese disain

Töö sisendina said autorid tellija poolt koostatud joonised nende visiooniga tulevases kasutajaliidese. Kuvatõmmised sellest kavandist on esitatud lisan 4.

Töö käigus koostasid autorid kõigepealt Figma kasutajaliidese prototüübi. Seda näidati tellijale ja saadud tagasiside alusel tehti parandusi ning näidati tulemust uuesti tellijale. Kui tellija jäi rahule, siis asuti kasutajaliidest realiseerima. Kuvatõmmised prototüübist on esitatud lisan 5.

Kasutajaliidest realiseerides täiendati seda vastavalt EKM-i töötajate tagasisidele ja autorite enda hinnangutele. Kuvatõmmised valminud kasutajaliidese mõnedest osadest on esitatud lisan 6.

5.3 Otsingusüsteem

Käesolevas jaotises tutvustatakse loodud otsingusüsteemi olulisemaid komponente.

5.3.1 Vabaotsing

Vabaotsing on realiseeritud kasutades PostgreSQL täistekstiotsingu funktsionaalsust [45]. Selleks loodi PostgreSQL andmebaasis materialiseeritud vaade e hetktõmmis, milles on kaks veergu. Esimeses veerus on viisi unikaalne identifikaator. Teises veerus on tekstivektor, mille väärtus on leitud viisi kohta käivate erinevate tekstiliste väärtuste konkatenatsioonina tulemusena moodustatud stringi e sõne põhjal. Selle konkatenatsiooni tulemus on dokument ning tekstivektor on selle dokumendi kompaktne esitus, kus on sorteeritud unikaalsete lekseemide nimekiri. See nimekiri on normaliseeritud, st sama sõna erinevad vormid on kokku pandud üheks sõnaks. Töös kasutatakse lekseemide leidmiseks inglise keele jaoks mõeldud tekstiotsingu konfiguratsiooni, mis ei ole ideaalne lahendus. [46] Erinevaid viisi andmevälju käsitletakse erineva kaaluga. Kõige olulisemad andmeväljad on viisiviide, tekstiviide, heliviide, videoviide, kataloog, publikatsioonid ja märkused. Hetktõmmise värskendamiseks kasutatakse lauset REFRESH MATERIALIZED VIEW. [47] See tähendab, et hetktõmmise väärtus arvutatakse nullist uuesti. Hetktõmmise värskendamiseks kulub andmebaasis olevate üle 5000 viisi korral umbes 550 ms. Kuna see ei ole palju, siis hetktõmmist värskendatakse iga kord, kui viiside tabeli põhjal täidetakse rea lisamise, muutmise või kustutamise lause. Selleks kasutatakse tabeliga seotud lause taseme trigerit. Alternatiiviks oleks kasutada Cron plaanurit[48], mis võimaldab tegevusi regulaarselt käivitada. Selle abil saaks käivitada hetktõmmise värskendamise näiteks korra päevas. Sellise lähenemise probleemiks on, et viimase 24 tunni jooksul viiside andmetes tehtud muudatused ei kajastu vaba teksti otsingu tulemusel. Hetktõmmise loomise lause on esitatud lisas 7. [8]

5.3.2 Viisiotsing

Andmebaasis on rahvaviiside noodistused kodeeritud ABC notatsioonis, mis seab iga heli absoluutkõrguse vastavusse tähtnimega C, D, E, F, G, A või B [2]. Klassikalises muusikas on viisi helidel püsiv absoluutkõrgus, kust neid alati lauldakse. Selline viis on absoluutsete helikõrguste püsiv järgnevus. Rahvaviis on seevastu suhteliste helikõrguste püsiv järgnevus ning ühe rahvaviisi noodistamisel võib olla mitu erinevat tulemust. See tähendab, et rahvaviisi leidmiseks ei piisa viisikodeeringute üks-ühele võrdlemisest.

Susi, Lymar ja Rajur kirjutasid oma magistritöös [7], et võimaldamaks viiside otsimist struktuuri alusel on vaja rakendada mingit meetodit, mille abil võrrelda otsingu aluseks olevat noodistust andmebaasis oleva noodistusega. Sellist tüüpi otsingu tegemisel on tähtis, et otsing oskaks leida ka sarnaseid viise.

Realiseeritav otsingusüsteem peab suutma leida sama viisi eri kõrgustelt. Selle saavutamiseks on vaja muusikalised noodid teisendada arvilisele kujule. Arvud tähistavad erinevaid helikõrguseid ning seeläbi on võimalik välja arvutada meloodias sisalduvate nootide kõrguste vahed.

Töös kasutati nootide teisendamiseks numbrilise muusika märgistuse süsteemi Jianpu [49], mida täiendati vastavalt ABC notatsiooni eripäradele. Meloodiate kodeeringud salvestati andmebaasist eraldi faili ning teisenduse teostamiseks kirjutati PHP skript. Saadud tulemuste põhjal sai iga meloodia kohta välja arvutada selles sisalduvate nootide vahed, mis liideti kokku sõneks. Et sõnesid oleks võimalik otsingu teostamiseks kasutada, lisati need andmebaasi. Tabel 2 toob näite viisikodeeringu teisendamise sammudest. Andmebaasis salvestatakse lisaks ABC notatsioonis kodeeringule ka nootide vahed.

Tabel 2. Viisi meloodias sisalduvate nootide vahede leidmise sammud.

Kodeering ABC notatsioonis	Noodid numbrilisel kujul	Nootide vahed
EF GA BA GE GF FB, EE	3 4 5 6 7 6 5 3 5 4 4 0 3 3 3 3	1 1 1 1 -1 -1 -2 2 -1 0 -4
EE		3 0 0 0

Eelmainitud skripti kasutatakse ka sisendi teisendamiseks juhul, kui kasutaja teostab otsingut kodeeritud viisi järgi. Samuti kasutatakse seda haldusliideses uue viisi meloodia lisamisel või olemasoleva viisi meloodia muutmisel nootide vahede väljaarvutamiseks.

5.3.3 Kombineeritud otsing

Otsingusüsteem peab võimaldama otsida rahvaviise korruga mitut filtrit kasutades. Selle saavutamiseks oli vaja esmalt otsustada, kuidas lahendada päringute kombineerimine.

Üks võimalus on luua iga filtri kohta eraldi päring ja leida nende tulemustest INTERSECT operaatoriga ühisosa. Joonis 1 esitab näitena sellisel viisil tehtud päringu viisi leidmiseks heliulatuse järgi tingimusel, et sellel viisil leidub ka helifail.


```

SELECT DISTINCT tune_reference , text_reference ,
COALESCE(sound_reference , video_reference)
AS sound_video_reference ,
parishes.title AS parish , villages.title AS village
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS places
ON tunes.id=places.tune_id
LEFT JOIN folk_tune.parishes AS parishes
ON places.parish_id=parishes.id
LEFT JOIN folk_tune.villages AS villages
ON places.village_id=villages.id
INNER JOIN folk_tune.musical_characteristics AS MC
ON tunes.id=MC.tune_id
WHERE MC.sound_range_id=8
INTERSECT
SELECT DISTINCT tune_reference , text_reference ,
COALESCE(sound_reference , video_reference)
AS sound_video_reference ,
parishes.title AS parish , villages.title AS village
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS places
ON tunes.id=places.tune_id
LEFT JOIN folk_tune.parishes AS parishes
ON places.parish_id=parishes.id
LEFT JOIN folk_tune.villages AS villages
ON places.village_id=villages.id
INNER JOIN folk_tune.tune_transcriptions AS transcriptions
ON tunes.id=transcriptions.tune_id
WHERE EXISTS (SELECT 1 FROM
folk_tune.transcription_sources AS sources
WHERE transcription_source_id=1);

```

Joonis 1. INTERSECT operaatoriga päringu näide.

Teine võimalus on leida kõigile tingimustele vastavad viisid ühe liittingimusega päringuga. Eelmisele näitele vastava liittingimusega päringu esitab joonis 2.

```

SELECT DISTINCT tune_reference , text_reference ,
COALESCE(sound_reference , video_reference )
AS sound_video_reference ,
parishes.title AS parish , villages.title AS village
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS places
ON tunes.id=places.tune_id
LEFT JOIN folk_tune.parishes AS parishes
ON places.parish_id=parishes.id
LEFT JOIN folk_tune.villages AS villages
ON places.village_id=villages.id
INNER JOIN folk_tune.musical_characteristics AS MC
ON tunes.id=MC.tune_id
INNER JOIN folk_tune.tune_transcriptions AS transcriptions
ON tunes.id=transcriptions.tune_id
WHERE MC.sound_range_id=8
AND EXISTS (SELECT 1 FROM
folk_tune.transcription_sources AS sources
WHERE transcription_source_id=1);

```

Joonis 2. Liitpäringu näide.

Otsus langetati liitringimusega päringu kasuks järgmistel kaalutlustel. [50, 51, 52]

1. **Tõhusus:** Andmebaasiserver peab täitma vaid ühe päringu. Kuna võimalikke otsinguparameetreid on palju, siis igaihe jaoks eraldi päringu käivitamine tähendaks märkimisväärset ressursi- ja ajakulu.
2. **Täpsus:** Ühe päringuga saab tagada, et arvesse võetakse kõigi otsinguparameetreid. Mitme päringu tegemine võib põhjustada puudulikke või ebakorrekseid tulemusi, kui mõnes päringus jääb mõni parameeter arvestamata.
3. **Hallatavus:** Ühte päringut võib olla lihtsam hallata ja muuta kui mitut päringut, eriti juhul, kui otsinguparameetreid on palju või kui need aja jooksul muutuvad.

Päringus ja seega ka kasutajale pakutavas otsingus on tingimused ühendatud AND loogikaoperaatoriga. Teatud otsinguparameetrite puhul on lisaks tarvis kasutada alampäringuid, näiteks olukorras, kus otsingut teostatakse viisiga seotud isikute põhjal.

Viiside tabel on isikute tabeliga seotud tunes_persons_roles tabeli kaudu, kus tune_person_role_type_id määrab isiku rolli. Rollid "viisikoguja", "salvestaja" ja "tekstikoguja", mille id on vastavalt 1, 2 ja 3, on otsingus ühtlustatud "koguja" rolliks. Rollile "esitaja" vastab id 4. Seega esitaja nime järgi saab viisi otsida kasutades päringut joonisel 3.

```

SELECT DISTINCT tune_reference , text_reference ,
COALESCE(sound_reference , video_reference)
AS sound_video_reference ,
parishes.title AS parish , villages.title AS village
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS places
ON tunes.id=places.tune_id
LEFT JOIN folk_tune.parishes AS parishes
ON places.parish_id=parishes.id
LEFT JOIN folk_tune.villages AS villages
ON places.village_id=villages.id
INNER JOIN folk_tune.tunes_persons_roles AS TPR
ON tunes.id=TPR.tune_id
INNER JOIN folk_tune.persons AS persons
ON TPR.person_id=persons.id
WHERE tune_person_role_type_id=4
AND TRIM(COALESCE(given_name , '' ) || ' ' ||
COALESCE(surname , '' )) ILIKE 'ida aab ');

```

Joonis 3. Päring viisi otsimiseks esitaja nime järgi.

Kui aga tahta otsida korraga nii esitaja kui koguja järgi, siis oleks esialgse päringute genereerimise eeskirja järgi tulemuseks päring joonisel 4. Päringus on liittingimus, mille alamtingimused on seotud AND operaatoriga.

```

SELECT DISTINCT tune_reference , text_reference ,
COALESCE(sound_reference , video_reference)
AS sound_video_reference ,
parishes.title AS parish , villages.title AS village
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS places
ON tunes.id=places.tune_id
LEFT JOIN folk_tune.parishes AS parishes
ON places.parish_id=parishes.id
LEFT JOIN folk_tune.villages AS villages
ON places.village_id=villages.id
INNER JOIN folk_tune.tunes_persons_roles AS TPR
ON tunes.id=TPR.tune_id
INNER JOIN folk_tune.persons AS persons
ON TPR.person_id=persons.id
WHERE tune_person_role_type_id=4
AND TRIM(COALESCE(given_name , '' ) || ' ' ||
COALESCE(surname , '')) ILIKE 'ida aab')
AND tune_person_role_type_id IN (1,2,3)
AND TRIM(COALESCE(given_name , '' ) || ' ' ||
COALESCE(surname , '')) ILIKE 'helgi sirmais');

```

Joonis 4. Ebasobiv päring viisi otsimiseks esitaja ja koguja nime järgi.

Sellise päringu tulemuseks oleks alati null rida, sest tingimused lähevad omavahel vastuollu – tune_person_role_type_id ei saa kunagi olla vahemikus 1-3 ning samal ajal võrduda neljaga. Olukorra lahendamiseks kasutatakse EXISTS predikaati koos korreleeruva alam-päringuga, mille näide on toodud joonisel 5.

```

SELECT DISTINCT tune_reference , text_reference ,
COALESCE(sound_reference , video_reference)
AS sound_video_reference ,
parishes.title AS parish , villages.title AS village
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS places
ON tunes.id=places.tune_id
LEFT JOIN folk_tune.parishes AS parishes
ON places.parish_id=parishes.id
LEFT JOIN folk_tune.villages AS villages
ON places.village_id=villages.id
WHERE EXISTS (SELECT 1
FROM folk_tune.tunes_persons_roles AS TPR
INNER JOIN folk_tune.persons AS persons
ON TPR.person_id=persons.id
WHERE TPR.tune_id=tunes.id
AND tune_person_role_type_id=4
AND TRIM(COALESCE(given_name , '') || ' ' ||
COALESCE(surname , '')) ILIKE 'ida aab')
AND EXISTS (SELECT 1
FROM folk_tune.tunes_persons_roles AS TPR
INNER JOIN folk_tune.persons AS persons
ON TPR.person_id=persons.id
WHERE TPR.tune_id=tunes.id
AND tune_person_role_type_id IN (1, 2, 3)
AND TRIM(COALESCE(given_name , '') || ' ' ||
COALESCE(surname , '')) ILIKE 'helgi sirmais');

```

Joonis 5. Sobiv päring viisi otsimiseks esitaja ja koguja nime järgi.

5.4 Tagarakenduse disain

Käesolev jaotis käsitleb tagarakenduse ülesehitust.

5.4.1 Otsingusüsteem

Tagarakenduse loomisel oli osaliselt võimalik toetuda olemasolevale tagarakendusele. Haldusliidese puhul oli varasemast olemas toimiv tagarakendus ning ei olnud põhjust seda antud töö raames muuta. Otsingusüsteemi loomisel oli vaja tagarakendus nullist luua.

Otsingusüsteemi tagarakenduse osa loodi kasutades protseduurilist lähenemist ning see hõlmab otsingupäringute töötlemist ja andmebaasiga suhtlemist. Kuigi eelnevalt loodud tagarakendus kasutas Node.js-i, otsustasid töö autorid koostöös EKM-iga kasutada otsin-

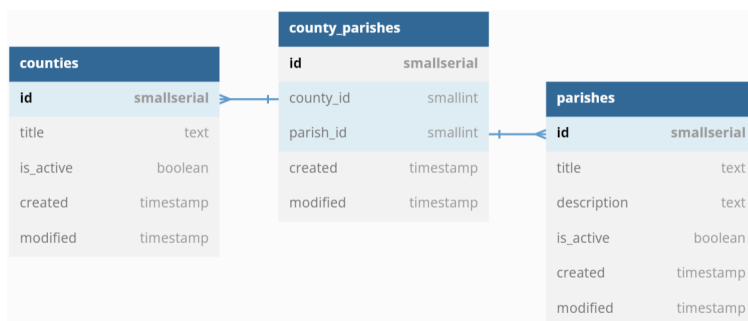
gusüsteemi jaoks PHP programmeerimiskeelt. PHP valiti, kuna seda valdavad nii töö autorid kui ka EKM-i IT-osakond. Node.js-ga on autoritel vähem kogemust ja see ei ole sobilik tehnoloogia ka EKM-i jaoks. Üleminekust PHPle kirjutatakse jaotises 2.4.

PHP abil käivitatakse otsingupäringute tulemuste saamiseks vajalikud SQL laused, et võimaldada kasutajatel teostada erinevaid otsinguid andmebaasis olevate andmete põhjal.

5.5 Andmebaasi disain

Loodav otsingusüsteem pakub võimalust teha otsingut maakondade järgi, millega on rahvaviis seotud. Enne töö alustamist valminud andmebaas ei sisaldanud teavet maakondade kohta. Seega tekkis vajadus see sinna lisada. Kuna siinkohal on tegemist kümne ajaloolise maakonnaga 1945. aasta haldusjaotusest [53], siis ei olnud võimalik kasutada tänapäevast Eesti haldus- ja asustusjaotuse klassifikaatorit [54], vaid tuli luua uus.

Maakondade andmete hoidmiseks lisati andmebaasi tabel *Counties*. Kihelkondade maakondadesse jagunemise esitamiseks loodi tabel *County_parishes*. Uute tabelite omavaheline seos ning seos andmebaasis juba eelnevalt loodud kihelkondade tabeliga *Parishes* on välja toodud joonisel 6.



Joonis 6. Tabelid maakondade ja kihelkondade kohta andmete hoidmiseks.

Tabel 3 esitab uute tabelite veergude sõnalised kirjeldused.

Tabel 3. Andmebaasi lisatud tabelite *Counties* ja *County_parishes* veergude sõnalised kirjeldused.

Tabel	Veerg	Veeru definitsioon	Näiteväärtus
<i>Counties</i>	<i>title</i>	Maakonna nimi	Harjumaa

Tabel	Veerg	Veeru definitsioon	Näiteväärtus
<i>Counties</i>	<i>is_active</i>	Väärtus, mis kirjeldab, kas maakond on aktiivne (TRUE) või mitte (FALSE). Mitteaktiivseid maakondi kasutajaliideses ei kuvata ning nende põhjal otsingut teostada ei saa. {Registreerimine on kohustuslik, vaikimisi väärtus on TRUE.}	TRUE
<i>Counties</i>	<i>created</i>	Maakonna andmebaasi lisamise kuupäev ja kellaaeg sekundi täpsusega, ilma ajavööndi ning sekundi murdosadeta. Süsteem määrab selle automaatselt. {Registreerimine on kohustuslik.}	2023-04-19 14:18:35
<i>Counties</i>	<i>modified</i>	Maakonna andmete viimase muutmise kuupäev ja kellaaeg sekundi täpsusega, ilma ajavööndi ning sekundi murdosadeta. Süsteem määrab selle automaatselt. {Registreerimine on kohustuslik.}	2023-04-19 14:18:35
<i>County_parishes</i>	<i>created</i>	Maakonna ja kihelkonna seose andmebaasi lisamise kuupäev ja kellaaeg sekundi täpsusega, ilma ajavööndi ning sekundi murdosadeta. Süsteem määrab selle automaatselt. {Registreerimine on kohustuslik.}	2023-04-19 14:18:35

Tabel	Veerg	Veeru definitsioon	Näiteväärtus
<i>County_parishes</i>	<i>modified</i>	Maakonna ja kihelkonna seose andmete viimase muutmise kuupäev ja kellaeg sekundi täpsusega, ilma ajavööndi ning sekundi murdosadeta. Süsteem määrab selle automaatselt. {Registreerimine on kohustuslik.}	2023-04-19 14:18:35

Uutele tabelitele indeksite loomise laused on esitatud lisas 2. Olemasolevad tabelid on varasemast indekseeritud ning neile täiendavaid indekseid ei loodud.

Võimaldamaks rahvaviise otsida viisikodeeringute järgi, oli vaja olemasolevat tabelit *Tune_melodies* täiendada uue veeruga *note_differences*.

Uue veeru sõnaline kirjeldus on esitatud tabelis 4.

Tabel 4. Tabelisse *Tune_melodies* lisatud veeru sõnaline kirjeldus.

Veerg	Veeru definitsioon	Näiteväärtus
<i>note_differences</i>	Tühikutega eraldatud arvudest koosnev sõne, kus iga arv tähistab viisi meloodias sisalduva noodi kõrguse muutust võrreldes eelmise noodiga.	0 2 0 -1 -2 0 0 2 -4 -1 2 -1 0 0 0

5.6 Silbitamine

Üks töös püstitatud eesmärkidest oli lauluteksti silbitamine ja silpide paigutamine vastavate nootide alla. Kui kasutaja on teinud otsingu, siis otsingutulemuse (rahvaviisi) peale klikkides kuvatakse selle metaandmed, sealhulgas ka rahvaviisi esimene värss. Värsi puhul kuvatakse selle tekst ja sellele vastav meloodia noodikirjas. Eelnevate tegijate loodud kasutajaliideses kuvati tekst noodistiku all silbitamata kujul.

EKM-il oli olemas väga vana programm silbitamise jaoks. See ei sobinud uues süsteemis kasutamiseks, sest oli kirjutatud vanemas programmeerimiskeeles, mida poleks saanud antud rakenduse puhul kasutada.

Silbitamise jaoks kirjutati JavaScriptis programm, mis saab sisendiks värssi ja tagastab selle silbitatud kujul, lisades silpide vahele sidekriipsu. Programm rakendab silbitamise jaoks järgmisi reegleid, kus "C" tähistab kaashäälikut ja "V" täishäälikut. Need reeglid anti ette EKM-i poolt.

- "VVV" -> "VV-V"
- "VCVC" -> "V-CVC"
- "VCV" -> "V-CV"
- "VCCV" -> "VC-CV"
- "VCCCV" -> "VCC-CV"
- "VCCCCV" -> "VCCC-CV"
- "CVVCV" -> "CVV-CV"
- "CVCV" -> "CV-CV"

Näiteks saab programm sisendiks "lähme kiigele kiikumaie, kiigele" ja väljastab "läh-me kii-ge-le kii-ku-mai-e, kii-ge-le".

Lisaks võivad värssis olla erisümbolid "" ja "+".

- "" sümbol tähistab silbitamiskohta ning seda on vaja kasutada juhul, kui värssi laudes ei lähe rõhukohad silpidega kokku. Näiteks sõna "kauge'elle" tuleb silbitada "kau-ge-el-le". Ilma selle reeglita oleks väljundiks "kau-geel-le".
- "+" sümbol tähistab liitsõna liitekohta. Liitsõnal peab olema silbituskoht liitekohal. See väldib antud programmi puhul ebakorrektsed silbitustulemused. Näiteks sõna "karja+ajad" tuleb silbitada "kar-ja-a-jad". Ilma selle reeglita oleks väljundiks "kar-jaa-jad".

Kui andmebaasis on esimene värss kujul "karja+ajad kauge'elle", siis kasutajale kuvatakse see kujul "kar-ja-a-jad kau-ge-el-le".

5.7 Turvalisuse tagamine

Turvalisuse tagamine on üks olulisemaid aspekte tarkvarasüsteemi arendamisel. Käesolevas jaotises käsitletakse kahte kriitilist turvaohu: SQL süstimist ja saidiülest skriptimisrünakat (XSS). Samuti tuuakse välja nende vältimise meetodeid.

5.7.1 SQL süstimine

SQL süstimine on levinud rünnakutüüp, mille käigus ründaja manipuleerib süsteemile antavat sisendit, et see hakkaks käivitama kahjulikku SQL koodi, et võtta kontrolli andmebaasi üle või varastada andmeid. [55]

Rakenduse loomiseks kasutatakse PHP programmeerimiskeelt, mille laiendus PDO on muuhulgas kasulik SQL rünnakute vältimiseks. PDO on andmebaasipäringute abstraktsioonikiht, mis võimaldab kasutada "ettevalmistatud päringuid". See tähendab, et kasutaja sisendit ei panda otse päringusse, vaid see puhastatakse enne päringu käivitamist. [56] Seda lahendust kasutatakse antud rakenduses kõigi andmebaasipäringute puhul. Joonis 46 esitab koodinäite ettevalmistatud päringust.

```
$query = 'SELECT DISTINCT tune_reference , text_reference ,
COALESCE(sound_reference , video_reference )
AS sound_video_reference ,
parishes.title AS parish , villages.title AS village
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS places
ON tunes.id=places.tune_id
LEFT JOIN folk_tune.parishes AS parishes
ON places.parish_id=parishes.id
LEFT JOIN folk_tune.villages AS villages
ON places.village_id=villages.id
WHERE places.parish_id=1;';
$stmt = $db->prepare($query);
$stmt->execute();
return $stmt->fetchAll(PDO::FETCH_ASSOC);
```

Joonis 7. Näide ettevalmistatud päringust.

5.7.2 Saidiülene skriptimisrünnak

Saidiülene skriptimisrünnak (XSS) on teine levinud rünnakutüüp, kus ründaja sisestab veebilehele kahjuliku skripti, mida teised kasutajad hiljem oma veebilehitsejas käivitavad. [57] XSS rünnakud võivad põhjustada andmete vargust, kasutajakontode ülevõtmist või muude turvaprobleemide tekkimist. XSS rünnakute ennetamiseks kasutatakse järgmisi meetodeid.

- **Andmete kodeerimine.** Kõik kasutaja sisestatud andmed kodeeritakse enne nende kuvamist veebilehel, et vältida kahjulike skriptide käivitamist.

- **Sisendandmete valideerimine.** Kasutaja sisendit kontrollitakse enne töötlemist, veendumaks selle sobivuses.
- **Küpsiste turvalisuse suurendamine.** Küpsised salvestatakse HttpOnly atribuudiga, et vältida küpsisele ligipääsu JavaScripti kaudu. Joonis 8 esitab sellest koodinäite.

```
ini_set( 'session.cookie_httponly', 1 );
```

Joonis 8. Näide turvalisest küpsisest.

5.8 Koodi testimine

Käesolev jaotis käsitleb rakenduse testimist ja selleks kasutatud meetodeid.

5.8.1 Ühiktestimine

Ühiktestimise eesmärgiks oli katta vähemalt 70% koodist. PHP koodi testimiseks kasutatakse PHPUnit raamistikku [58] ja JavaScripti koodi testimiseks Jest raamistikku [59].

5.8.2 Päringute testimine

Koodis genereeritavate SQL päringute süntaktilises korrektsuses veendumiseks kasutati käsitsi testimist testjuhtudel, mis on välja toodud tabelis 5.

Tabel 5. Andmebaasipäringute käsitsi testimise testjuhud.

Testjuht	Näide
Ühe parameetri põhjal	Otsing esitaja järgi
Mitme erineva parameetri põhjal, mille info pärineb ühisest tabelist	Otsing esitaja ja koguja järgi
Mitme erineva parameetri põhjal, mille info pärineb erinevatest tabelitest	Otsing esitaja ja laululiigi järgi
Mitme samasuguse parameetri põhjal	Otsing kahe esitaja järgi

Testimiseks moodustati erinevaid liitpäringuid, samuti testiti individuaalselt kõiki ühe parameetriga päringuid. Lisas 3 esitatakse näiteid testjuhtudele vastavatest päringutest.

5.8.3 Kasutajaliidese testimine

Kasutajaliidese testimiseks kasutatavad integratsioonitested on esitatud tabelis 6.

Tabel 6. Kasutajaliidese integratsioonitestedid.

Testjuht	Oodatav tulemus
Kasutaja proovib sisse logida, sisestades korrektse emaili ja parooli.	Sisselogimine on edukas, kasutaja suunatakse avalehele ja talle kuvatakse sisselogitud kasutajale mõeldud menüüriba.
Kasutaja proovib sisse logida, sisestades vigase emaili või parooli.	Kasutaja jääb sisselogimislehele ja talle kuvatakse sõnum, et sisselogimine ebaõnnestus.
Kasutaja läheb otsingulehele, sooritab otsingu, klikib otsingutulemuse peale.	Otsingutulemuse peale klikates avaneb uus aken, kus on rahvaviisi metaandmed.
Sisselogitud kasutaja muudab avalehel teksti ja salvestab selle.	Avalehel kuvatakse kasutaja poolt tehtud muudatustega tekst.
Sisselogitud kasutaja läheb klassifikaatorite lehele, klõpsab klassifikaatori peale.	Kasutaja suunatakse uuele lehele, kus kuvatakse info klikitud klassifikaatorite kohta.

5.9 Juurutamine

Projekti lähtekood antakse EKM-le üle ZIP-failina. Lisaks lähtekoodile sisaldab pakett ka dokumentatsiooni, mis annab üksikasjalikud juhised rakenduse käivitamiseks. Selleks, et tagada sujuv üleminek uuele koodile, on projekti autorid valmis selgitama EKM-i töötajatele koodi struktuuri ja tehnoloogilisi valikuid, et muuta projekti mõistmine ja koodiga tutvumine arusaadavamaks.

Integreerimisprotsessi osas on peamine vastutus EKM-i töötajatel, kes tegelevad uue koodi süsteemi integreerimisega. Projekti autorid valmis pakkuma vajadusel tuge, et tagada integreerimisprotsessi toimimine.

6. Tulemuste analüüs

Selles peatükis analüüsitakse tulemusi, mida kirjeldati eelnevas peatükis.

6.1 Kasutajatepoolne testimine

Kasutajatepoolse testimise eesmärk on hinnata otsingusüsteemi kasutatavust ja kasutajakogemust, et saada parem arusaam süsteemi tugevustest ja nõrkustest.

Kasutajaliidese testimiseks valiti kaastudengid ja tuttavad. Kaastudengitel on olemas IT taust, mistõttu oskavad nad märgata tavainimesele märkamatuks jäävaid asju, olles samas süsteemi esmakordsed kasutajad, kellel eelnevad teadmised rahvaviisidest puuduvad.

Tavakasutajate tagasiside põhjal selgus, et kasutajaliides meenutas tavalist veebilehte - see oli tuttav ja arusaadav ning aitas neil süsteemiga kiiresti kohaneda. Mõningate elementide funktsionaalsus oli kohati ebaselge. Näiteks kasutajatele antud versioonis hiirt tabelireale viies ei tõstetud seda tabelirida esile. Samuti ei töötanud mõned kasutajaliidese nupud ootuspäraselt - mõnda nuppu tuli soovitud tulemuse saavutamiseks kaks korda vajutada.

Kaastudengid andsid konstruktiivset tagasisidet ning tänu nende IT taustale suutsid nad märgata süsteemi nõrkusi ja aidata kaasa nende parandamisele. Tagasisides toodi välja, et kasutajaliides on lihtne, intuitiivne ja puhas. Kohati isegi liiga puhas, osadel lehtedel oleks võinud olla rohkem elemente.

Tänu saadud tagasisidele oli võimalik süsteemi täiustada ning tagada kasutajatele parem kogemus. Tagasiside alusel muudeti näiteks elementide paigutust ja järjekorda lehel ning teksti suurust.

6.2 Otsingutulemuste kuvamise kiirus

Omavahel võrreldakse päringute sooritamise kiirust andmebaasisüsteemis ning tulemuste kuvamise kiirust kasutajaliideses. Nii on võimalik hinnata, kas loodud kasutajaliides on programmeeritud optimaalselt. Iga võrdluse jaoks kasutatud päringut käivitati kolm korda ning leiti keskmine päringu sooritamiseks kulunud aeg millisekundites. Tulemused on välja toodud tabelis 7.

Tabel 7. Otsingutulemuste leidmise kiiruse võrdlus andmebaasisüsteemi ja kasutajaliidese vahel.

	Kiirus andmebaasisüsteemis (ms)	Kiirus kasutajaliidese (ms)
Ühe parameetriga päring	90	559
Kahe parameetriga päring	126	605
Nelja parameetriga päring	228	673
Kaheksa parameetriga päring	268	691
12 parameetriga päring	283	699
16 parameetriga päring	298	729
24 parameetriga päring	366	898
Kõikide viiside leidmine	154	570

Võrreldes andmebaasisüsteemiga kuvab kasutajaliides otsingutulemusi keskmiselt 3.5 korda aeglasemalt. Sellegipoolest ei ole kasutajaliidese otsingu sooritamisele kuluv aeg häirivalt pikk, sest see on peaaegu alati alla sekundi. Võib järeldada, et praeguse andmemahu juures on rakenduse tehniline jõudlus rahuldav.

6.3 Võrdlus olemasolevate süsteemidega

Käesoleva süsteemi koostamiseks sisendi saamiseks analüüsiti erinevaid olemasolevaid otsingusüsteeme (vt peatükk 4). Saadud tulemuste valideerimiseks vastatakse selle süsteemi puhul samadele küsimustele kui võrreldavate süsteemide korral ja tuuakse vajadusel välja ka võrdlus olemasolevate süsteemidega.

6.3.1 Materjalide maht

Antud süsteemis on töö kirjutamise hetkel 4 559 rahvaviisi. Materjalid on pärit EKM-i andmebaasist Kivike [60], mis hoiab Eesti rahvaluule, kirjakultuuri ja kultuurilugude kogusid. Kivikeses on erinevaid helisid ligikaudu 12 000. Eesti rahvaluule kogumine on EKM-i töö ja seega võib arvestada, et enamik kogutud rahvaluulest on EKM-i erinevates andmekogudes olemas. Antud töös valminud süsteemis on seega ligikaudu kolmandik kogutud andmetest ning ülejäänud on plaanis sinna tulevikus sisse kanda. Seega võib öelda, et andmemahu mõttes ei ole käesolev süsteem väga suur ning võrreldavate süsteemide hulgas on see süsteem viiside mahult üks väiksemaid (vt tabel 8).

6.3.2 Otsingu funktsionaalsus

Otsingu funktsionaalsus on mõistlik, kuna väljad, mida kasutajatele kuvatakse, on põhilised väljad, mille järgi soovitakse otsida. Peamiselt EKM-i töötajate jaoks vajalik metaandmete otsing pole kasutajale koheselt täies mahus nähtav. See tähendab, et "+" nupu all asuvad otsinguväljad on kasutajale nähtavad alles peale "+" nupule vajutamist.

Kokku on võimalik otsida 40 erineva tunnuse järgi, mis on tunduvalt rohkem kui teistel võrreldavatel otsingusüsteemidel. Võrreldavate süsteemide hulgast on kõige rohkem otsingutunnuseid *Folk tune finder* süsteemil, kus saab otsida 11 erineva tunnuse järgi. Otsingutulemus jaotatakse vaikimisi 10 viisi kaupa lehekülgedeks. Kasutaja saab soovi korral tulemusi kuvada ka 25, 50 või 100 kaupa.

6.3.3 Kasutajaliidese disain

Hea kasutajakogemuse jaoks on vabaotsing ja kindla failitüübi otsing asetatud ülariibasse ning metaandmed vasakule kõrvalriibasse, mida on võimalik peita, kui kasutaja soovib tulemusi paremini näha. Kasutajale mõeldes on otsing ja tulemused samal lehel ning on võimalus otsingut peale selle sooritamist täpsustada, kuna otsinguväljadesse jääb eelmises otsingus sisestatud info.

Otsingusüsteemi arendades võeti arvesse eelnevalt tehtud analüüs teiste süsteemide kohta. Võeti arvesse näiteks tähelepanekut, et *Audio Archive* otsingusüsteemil oli palju lisafiltreid ja nende hulgas oli raske navigeerida. Käesolevas töös võetigi seega kasutusele lahendus, kus metaandmete otsingu väljad ei ole koheselt nähtavad.

6.3.4 Tehniline jõudlus

Katsetades otsingut samas interneti võrgus kui teisi otsingusüsteeme, tulid vastused kahe kuni kolme sekundiga, mis mahub hea kasutajaliidese soovitude sisse. Tehti ka võrdlus teiste otsingusüsteemidega (vt tabel 8).

Tabel 8. Otsingusüsteemide tehnilise jõudluse võrdlus.

Otsingusüsteemi nimi	Viiside arv	Aeglaseima otsingu kiirus (s)	Otsingutunnuseid
FindSounds	27 500	0.7	3
Audioburst	8 500	2	3
Shazam	15 miljonit	9	1

Otsingusüsteemi nimi	Viiside arv	Aeglaseima otsingu kiirus (s)	Otsingutunnuseid
Audio Archive	15 miljonit	10	8
Folk Tune Finder	190 000	1.5	11
Antud lõputöö	4 559	3	40

Võrreldes materjalide mahtu ja otsingu kiirust, on näha, et antud töö tulemusena valminud süsteem on teistest süsteemidest aeglasem. Töö autorid peavad põhjuseks seda, et antud töös laetakse kõik otsingutulemused ühe korraga ning sorteerimiseks või järgmisele otsingutulemuste lehele minemiseks ei ole tarvis teha uut päringut, sest vajalik töötlemine on tulemustele juba tehtud. Selline lahendus valiti, kuna nähti suurt vajadust sorteerimise järele. Teistel võrreldavatel süsteemidel tehakse sorteerimisel ja järgmisele otsingutulemuste lehele minnes uus päring, mis võtab omakorda laadimisaega. Seega võib küll öelda, et antud süsteemis on tulemuste laadimisaeg suur, kuid sellevõrra on otsingutulemusi kiirem sorteerida ning läbi käia. Samuti saab aeglase otsingu põhjusena välja tuua selle, et antud töös on tulemusi võimalik otsida suurema arvu tunnuste järgi ja keerukam päring tingib ka aeglasema otsingu täitmise.

6.3.5 Ligipääsetavus

Ligipääsetavust vaegnägijatele on hea katsetada *Wave* [17] pistikprogrammiga. Selle abil leiti paar viga puuduva kirjelduse kohta, aga värvide kontrastus oli hea ja struktuur oli paigas. Vaegkuuljatel on võimalik näha viisi sõnu ja noodistikku. Kuna noodistik on olemas, siis näiteks lainekuju, mida kuvab *Audio Archive*, ei ole vaja esitada.

Süsteemi luues on pööratud tähelepanu ligipääsetavusele ja seega on valminud rakendus võrreldes teiste otsingusüsteemidega ligipääsetavam. Vaegkuuljatele ja -nägijatele pöörati tähelepanu ka *Folk tune finder* [41] otsingusüsteemis. See leht hinnati autorite poolt samuti ligipääsetavaks, kuid kõikidel teistel otsingusüsteemidel leiti ligipääsetavuse osas probleeme.

6.4 Võrdlus parimate praktikatega

Töö autorid on otsingusüsteemi kavandamisel püüdnud järgida parimaid praktikaid, et tagada süsteemi optimaalne kasutatavus ja ligipääsetavus.

Täiendavate filtrite kuvamiseks valiti lihvitud otsing. Võrreldes seda filtrite kuvamisega peale esmase otsingu sooritamist, pakub lihvitud otsing kasutajale võimaluse teha koheselt

enda soovitud otsing. See muudab soovitud tulemuseni jõudmise võimalikult kiireks. See on oluline just EKM-i töötajate jaoks, kes eelistasid seda lahendust, sest võivad päevas teha sadu otsinguid. Lihvitud otsing on küll keerulisem kui filtrite kuvamine peale esmast otsingut, kuid tõenäoliselt on keskmine kasutaja muusikakogemusega ning teab, mida ta otsida tahab.

Otsingu käivitamisel eelistati varianti, kus otsing sooritatakse peale sisestusklahvi vajutamist. Selline otsinguprotsess on kasutajale selgem ja väldib olukorda, kus lühikese aja jooksul mitme filtri lisamisel sooritatakse mitu otsingut, mis juhtuks siis, kui otsing käivitatakse pärast iga filtri muutmist. Pidev otsingutulemuste ootamine võib kasutajaliidesest jätta kohmaka mulje.

Kasutaja näeb otsingutulemuste koguarvu ja saab otsingutulemusi sorteerida nende veergude alusel, mida otsingutulemuses kuvatakse. [61] Sorteerida saab korraga ühe veeru järgi ning nii kahanevalt kui kasvavalt. Vaikimisi sorteeritakse viisiviite järgi kasvavalt ja see on päringutulemuses esimene veerg.

Kasutaja näeb otsingutulemusi samal lehel, kus on otsingu filtrid. Leitud viisi detailide vaatamiseks liigub kasutaja eraldi lehele, sest nii on võimalik kasutajale kuvada korraga rohkem elemente. Kasutaja näeb otsingutulemusi lehel vaikimisi 10 kaupa, kuid saab seda arvu muuta.

Puhta koodi nõudeid järgiti nii palju, kui võimalik. Elementide nimetamisel järgiti sama stiili, mida oli kasutatud teiste sama kategooria elementide puhul. Samuti peeti nimetamisel oluliseks täpsust. See aitab tagada koodi loetavust ja muudab tarkvara edasiarendamise kergemaks. Koodi kirjutamisel välditi koodi kordumist seal, kus see oli mõistlik. Osades kohtades säilitati siiski dubleeriv kood, et parandada selle loetavust.

Ligipääsetavusega seoses arvestasid autorid järgmiste teguritega.

- **Värvikontrast.** Kasutajaliidest disainides valiti liidese põhivärvideks sinine ja hall. Värvide valides veenduti, et kõik vajalikud elemendid paistaksid välja.
- **Skaleeruvus.** Veebirakendust programmeerides veenduti, et lehte suuremaks ja väiksemaks tehes oleksid elemendid paigas ja loetavad.
- **Lehe struktuur.** Lehe struktuuri luues järgiti järjepidevuse standardit [25], et elemendid oleksid kasutajale oodatud kohas.

7. Arendusvaade

Lõputöö tegemise käigus ja EKM-i töötajatega koosolekuid pidades tuli töö autoritel ka ideid edasiarendustest, mis jäid tehtava töö skoobist välja. Antud peatüki eesmärk on need ideed esitada.

7.1 Heli tuvastamine

Arendades avalikku kasutajaliidest ja proovides mõelda tavakasutaja peale, kes pole muusikateooriaga kokku puutunud, tekkis mõte heli tuvastamiseks näiteks ümismise järgi. Käesoleva töö tulemusel loodud lahendus võimaldab leida viisi ABC notatsiooni [2] järgi, mis eeldab kasutajalt muusikaliste teadmiste olemasolu. Statistikaameti andmetel on Eestis igal aastal muusikat õppimas vaid 2% kuni 3% elanikkonnast [62], millest võib järeldada, et Eestis on suur hulk inimesi, kes noodistiku järgi otsingut sooritada ei oska.

7.2 Visualiseeritud otsing

Visualiseeritud otsing on veel üks võimalus, kuidas tarkvara kasutamist lihtsamaks teha. Antud töö raames valminud lahendus võimaldab kasutajal otsida viise ABC notatsioonis, trükkides need otsinguväljale. Kasutajale otsingu lihtsustamiseks tuldi mõttele kuvada kasutajale klaveri klahvistik, mille abil oleks võimalik viisi otsingusse sisestada.

7.3 Kõnesüntesaator

2021. aastal tehtud töö [7], mille edasiarendus on antud töö, mainib kõnesüntesaatori vajadust. Sama vajadus tuli välja ka tellijatega peetud koosolekutel. Praegune lahendus esitab vaid pilliheli, kuid tellijal on soov rahvaviisi puhul ka sõnu kuulda. Süsteemi realisatsioon võimaldab antud lahendust, seega oleks kõnesüntesaatori kasutuselevõtt hea variant edasiarenduseks.

7.4 Tarkvara ühtlustamine

Töö tulemusena valminud süsteemis on otsingusüsteemi tagarakendus PHP-s ja haldussüsteemi tagarakendus Node.js-is. Et lihtsustada süsteemi edasiarendamist ja vähendada halduskulusid, peaks kasutatavaid tehnoloogiaid ühtlustama. Kuna EKM eelistab PHP

kasutamist, siis tuleks haldusliidese tagarakendus tulevikus realiseerida PHP-s.

7.5 Mobiilivaade

Käesoleva töö tulemusena valminud süsteemis pole pööratud tähelepanu sellele, kuidas paistab otsingusüsteem mobiiltelefonis. Tulevaste arenduste käigus võiks lisada ka mobiilivaate.

8. Kokkuvõte

Käesoleva lõputöö tulemusel arendati edasi EKM-i ERA rahvaviiside infosüsteemi tarkvara. Täpsemalt loodi avalikkusele suunatud veebipõhine kasutajaliides rahvaviiside kohta info otsimiseks ning töötajatele mõeldud veebipõhine kasutajaliides viiside andmete haldamiseks. Selle lõputööga arendati edasi süsteemi, mille loomist alustati kahes varasemalt Tallinna Tehnikaülikoolis kaitstud lõputöös.

Töö käigus koguti infot otsingusüsteemide parimate praktikate kohta ja analüüsiti olemasolevaid helide otsingusüsteeme. Töös esitati nõuded süsteemi täiendustele ja kirjeldati nende nõuete alusel rakenduses ja andmebaasis tehtud täiendusi. Veebirakenduse loomiseks kasutati PHP programmeerimiskeelt. Andmebaasisüsteemina on kasutusel PostgreSQL.

Kasutatud kirjandus

- [1] *Web Content Accessibility Guidelines*. URL: <https://wcag.com/> (kasutatud 14.4.2023).
- [2] C. Walshaw. *How Does Shazam Recognize Music Accurately?* URL: <https://abcnotation.com/> (kasutatud 18.3.2023).
- [3] H. Vallaste. *e-Teatmik: IT ja sidetehnika seletav sõnaraamat*. URL: <http://vallaste.ee/> (kasutatud 2.5.2023).
- [4] Tartu Ülikool. *Teadusandmete haldamine*. URL: <https://sisu.ut.ee/teadusandmed/metaandmed> (kasutatud 4.5.2023).
- [5] *Võõrsõnade leksikon*. URL: <http://www.eki.ee/dict/vsl/vsl.html> (kasutatud 1.4.2023).
- [6] K. Avloi and M. Paroll and K. Seemen. “Rahvaviiside infosüsteemi tarkvara arendamine Eesti Kirjandusmuuseumile”. Tallinna Tehnikaülikool, 2020.
- [7] A. Lymar and M. Rajur and J. Susi. “Eesti Kirjandusmuuseumi jaoks rahvaviiside noodistuse standardi juurutamine ja infosüsteemi edasiarendus”. Tallinna Tehnikaülikool, 2021.
- [8] UX World. *How To Improve Advanced Search UX?* 2018. URL: <https://uxdworld.com/2018/07/15/how-to-improve-advanced-search-ux/> (kasutatud 22.3.2023).
- [9] *Unioopedia*. URL: https://et.unionpedia.org/i/Eesti_Rahvaluule_Arhiiv (kasutatud 18.4.2023).
- [10] T. Kannik. “Arhiivide kirjastustegevus Eestis: olukorra kaardistus”. Tallinna Ülikool, 2017.
- [11] *Eesti Rahvaluule Arhiiv*. URL: <https://www.folklore.ee/era/ava.htm> (kasutatud 18.4.2023).
- [12] *Scrum.org*. URL: <https://www.scrum.org/learning-series/what-is-scrum/what-is-scrum> (kasutatud 14.4.2023).
- [13] *ScrumGuides.org*. URL: <https://scrumguides.org/scrum-guide.html> (kasutatud 14.4.2023).
- [14] Agile Alliance. *The 12 Principles behind the Agile Manifesto*. <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/> (kasutatud 14.4.2023).

- [15] E. Eessaar. *Ülikooli infosüsteemi vastuvõtuaegade allsüsteem. Näiteprojekt õppeaines "Andmebaasid II"* (kasutatud. 19.4.2023).
- [16] *dbdiagram*. URL: <https://dbdiagram.io/> (kasutatud 19.4.2023).
- [17] *Wave*. URL: <https://chrome.google.com/webstore/detail/wave-evaluation-tool/jbbplnpkjmeebjpijfedlgcdilocofh> (kasutatud 19.4.2023).
- [18] *Colorblindly*. URL: <https://chrome.google.com/webstore/detail/colorblindly/floniaahmccleoclnneebhhmngdfijgg> (kasutatud 19.4.2023).
- [19] *Page Load Time*. URL: <https://chrome.google.com/webstore/detail/page-load-time/fploionmjgeclbkempmkogoaohcdbig> (kasutatud 19.4.2023).
- [20] *Open Chrome DevTools*. URL: <https://developer.chrome.com/docs/devtools/open/> (kasutatud 8.5.2023).
- [21] A. Roznovsky. *Why Use PHP? Main Advantages and Disadvantages*. URL: <https://light-it.net/blog/why-use-php-main-advantages-and-disadvantages/> (kasutatud 11.5.2023).
- [22] *Best Practices for Search*. URL: <https://www.uxbooth.com/articles/best-practices-for-search/> (kasutatud 17.4.2023).
- [23] *Faceted search*. URL: https://en.wikipedia.org/wiki/Faceted_search (kasutatud 18.4.2023).
- [24] *The Impact of Color Contrast for Accessibility and Inclusivity in UI/UX Design*. URL: <https://artversion.com/blog/the-impact-of-color-contrast-for-accessibility-and-inclusivity-in-ui-ux-design/> (kasutatud 19.4.2023).
- [25] *Maintain Consistency and Adhere to Standards*. URL: <https://www.nngroup.com/articles/consistency-and-standards/> (kasutatud 19.4.2023).
- [26] M. Monaghan. *Website Load Time Statistics: Why Speed Matters in 2023*. URL: <https://www.websitebuilderexpert.com/building-websites/website-load-time-statistics/> (kasutatud 19.4.2023).
- [27] AudioEye. *Website Load Time Statistics: Why Speed Matters in 2023*. URL: <https://www.audioeye.com/post/how-to-make-a-website-accessible-for-the-deaf/> (kasutatud 19.4.2023).
- [28] *Findsounds*. URL: <https://www.findsounds.com/> (kasutatud 21.3.2023).
- [29] D. Sonnenschein. *Sound Design: The Expressive Power of Music, Voice and Sound Effects in Cinema*. Michael Wiese Productions, 2001.

- [30] *Coloring the Audio Waveform Display*. URL: <https://findsounds.com/color.html> (kasutatud 19.4.2023).
- [31] *About Audioburst*. URL: <https://audioburst.com/about> (kasutatud 5.4.2023).
- [32] *Audioburst*. URL: <https://search.audioburst.com/> (kasutatud 21.3.2023).
- [33] *Audioburst Publishers*. URL: <https://audioburst.com/publishers/web> (kasutatud 5.4.2023).
- [34] *Podcast statistics in 2023*. URL: <https://podcastpage.io/podcast-statistics/> (kasutatud 5.4.2023).
- [35] *Shazam*. URL: <https://www.shazam.com/> (kasutatud 21.3.2023).
- [36] J. Awa-abuon. *How Does Shazam Recognize Music Accurately?* URL: <https://www.makeuseof.com/how-does-shazam-work/> (kasutatud 5.4.2023).
- [37] *A Complete Guide To The Shazam Music App*. URL: <https://www.musicgateway.com/blog/music-industry/a-complete-guide-to-shazam-music-app> (kasutatud 19.4.2023).
- [38] *How Many Songs are There in the World?* URL: <https://www.musicianwave.com/how-many-songs-are-there-in-the-world/> (kasutatud 19.4.2023).
- [39] *Audio Archive*. URL: <https://archive.org/details/audio> (kasutatud 21.3.2023).
- [40] *About the Internet Archive*. URL: <https://archive.org/about/> (kasutatud 5.4.2023).
- [41] *Folk tune finder*. URL: <https://www.folktunefinder.com/> (kasutatud 21.3.2023).
- [42] *User Stories*. URL: <https://www.mountangoatsoftware.com/agile/user-stories> (kasutatud 29.4.2023).
- [43] *Infosüsteemide turvameetmete süsteem*. URL: <https://www.riigiteataja.ee/akt/13125331> (kasutatud 29.4.2023).
- [44] R. C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Pearson, 2008.
- [45] *Mastering PostgreSQL Tools: Full-Text Search and Phrase Search*. URL: <https://www.compose.com/articles/mastering-postgresql-tools-full-text-search-and-phrase-search/> (kasutatud 19.4.2023).

- [46] PostgreSQL. *Text Search Types*. URL: <https://www.postgresql.org/docs/current/datatype-textsearch.html> (kasutatud 22.5.2023).
- [47] PostgreSQL dokumentatsioon. *Refresh Materialized View*. URL: <https://www.postgresql.org/docs/current/sql-refreshmaterializedview.html> (kasutatud 22.5.2023).
- [48] Wikipedia. *cron*. URL: <https://en.wikipedia.org/wiki/Cron> (kasutatud 22.5.2023).
- [49] TableEdit. URL: https://tableedit.com/help/english_m/jianpu.shtml (kasutatud 21.5.2023).
- [50] S. Faroult and P. Robson. *The Art of SQL*. O'Reilly Media, 2006.
- [51] B. Karwin. *SQL Antipatterns. Avoiding the Pitfalls of Database Programming*. Pragmatic Bookshelf, 2010.
- [52] M. Winand. *SQL Performance Explained. Everything Developers Need to Know about SQL Performance*. Markus Winand, 2012.
- [53] Saaremaa Ühisgümnaasiumi õppematerjal. *Eesti haldusjaotus ja võõrvõimude vaheldumine läbi aegade*. URL: https://www.syg.edu.ee/oppematerjalid/U_Kiil_ajalugu/voorvoimud.html (kasutatud 28.4.2023).
- [54] Statistikaamet. *Eesti haldus- ja asustusjaotuse klassifikaator*. URL: <https://klassifikaatorid.stat.ee/Item/stat.ee/c4c47742-12d7-4fea-bc8c-5aeca9112e2a> (kasutatud 28.4.2023).
- [55] *SQL Injection*. URL: https://www.w3schools.com/sql/sql_injection.asp (kasutatud 19.4.2023).
- [56] Website Beaver. *PHP PDO Prepared Statements Tutorial to Prevent SQL Injection*. URL: <https://websitebeaver.com/php-pdo-prepared-statements-to-prevent-sql-injection> (kasutatud 19.4.2023).
- [57] *Cross Site Scripting (XSS)*. URL: <https://owasp.org/www-community/attacks/xss/> (kasutatud 19.4.2023).
- [58] *PHPUnit*. URL: <https://phpunit.de/> (kasutatud 28.4.2023).
- [59] *JEST*. URL: <https://jestjs.io/> (kasutatud 28.4.2023).
- [60] *Kivike*. URL: <https://kivike.kirmus.ee/> (kasutatud 8.5.2023).
- [61] A. Prokhorova and L. Vollaire. *Search filters: 5 best practices for a great UX*. URL: <https://www.algolia.com/blog/ux/search-filter-ux-best-practices/> (kasutatud 22.5.2023).

- [62] Statistikaamet. *Huvikoolides õppijad õppevaldkonna, vanuse ja soo järgi*. URL: https://andmed.stat.ee/et/stat/sotsiaalelu__haridus__huviharidus/HTH02/table/tableViewLayout2 (kasutatud 8.5.2023).

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Meie, Joosep Raudväli, Stina Salumets, Triin Sarapuu

1. Anname Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “Eesti Kirjandusmuuseumi rahvaviiside infosüsteemi tarkvara edasiarendamine”, mille juhendaja on Erki Eessaar ,
 - 1.1. 1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. 1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Oleme teadlikud, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autoritele.
3. Kinnitame, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

22.05.2023

¹Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 - Andmebaasi indeksite loomise laused

```
CREATE INDEX IX_county_parishes_counties  
ON folk_tune.county_parishes (county_id);
```

```
CREATE INDEX IX_county_parishes_parishes  
ON folk_tune.county_parishes (parish_id);
```

Joonis 9. Välisvõtmete veergudele lisatavad indeksid.

Tagamaks maakonna nimede unikaalsuse sõltumata sellest, kas neis kasutatakse suur- või väiketähti ning kiirendamiseks maakondade tõstutundetut otsingut, on tabelis *Counties* veerule *title* loodud unikaalsuse funktsioonil **UPPER** põhinev indeks.

```
CREATE UNIQUE INDEX IX_counties_title  
ON folk_tune.counties (UPPER(title));
```

Joonis 10. Funktsioonil põhinev indeks.

Lisa 3 - Andmebaasipäringute näiteid

```
SELECT DISTINCT tune_reference , text_reference ,  
COALESCE(sound_reference , video_reference)  
AS sound_video_reference ,  
parishes.title AS parish , villages.title AS village  
FROM folk_tune.tunes AS tunes  
LEFT JOIN folk_tune.tune_places AS places  
ON tunes.id=places.tune_id  
LEFT JOIN folk_tune.parishes AS parishes  
ON places.parish_id=parishes.id  
LEFT JOIN folk_tune.villages AS villages  
ON places.village_id=villages.id  
WHERE EXISTS (SELECT 1  
FROM folk_tune.tunes_persons_roles AS TPR  
INNER JOIN folk_tune.persons AS persons  
ON TPR.person_id=persons.id  
WHERE TPR.tune_id=tunes.id  
AND tune_person_role_type_id=4  
AND TRIM(COALESCE(given_name , '' ) || ' ' ||  
COALESCE(surname , '')) ILIKE 'juula puu');
```

Joonis 11. Otsing esitaja järgi.

```

SELECT DISTINCT tune_reference , text_reference ,
COALESCE(sound_reference , video_reference)
AS sound_video_reference ,
parishes.title AS parish , villages.title AS village
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS places
ON tunes.id=places.tune_id
LEFT JOIN folk_tune.parishes AS parishes
ON places.parish_id=parishes.id
LEFT JOIN folk_tune.villages AS villages
ON places.village_id=villages.id
WHERE EXISTS (SELECT 1
FROM folk_tune.tunes_persons_roles AS TPR
INNER JOIN folk_tune.persons AS persons
ON TPR.person_id=persons.id
WHERE TPR.tune_id=tunes.id
AND TPR.tune_person_role_type_id IN (1, 2, 3)
AND (persons.given_name ILIKE '%kreek%'
OR persons.surname ILIKE '%kreek%'
OR persons.nickname ILIKE '%kreek%'))
AND EXISTS (SELECT 1
FROM folk_tune.tunes_persons_roles AS TPR
INNER JOIN folk_tune.persons AS persons
ON TPR.person_id=persons.id
WHERE TPR.tune_id=tunes.id
AND TPR.tune_person_role_type_id=4
AND (persons.given_name ILIKE '%eller%'
OR persons.surname ILIKE '%eller%'
OR persons.nickname ILIKE '%eller%'));

```

Joonis 12. Otsing esitaja ja koguja järgi.

```

SELECT DISTINCT tune_reference , text_reference ,
COALESCE(sound_reference , video_reference)
AS sound_video_reference ,
parishes.title AS parish , villages.title AS village
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS places
ON tunes.id=places.tune_id
LEFT JOIN folk_tune.parishes AS parishes
ON places.parish_id=parishes.id
LEFT JOIN folk_tune.villages AS villages
ON places.village_id=villages.id
INNER JOIN folk_tune.tune_songs AS tune_songs
ON tunes.id=tune_songs.tune_id
INNER JOIN folk_tune.tune_songs_song_genres AS genres
ON tune_songs.id=genres.tune_song_id
WHERE EXISTS (SELECT 1
FROM folk_tune.tunes_persons_roles AS TPR
INNER JOIN folk_tune.persons AS persons
ON TPR.person_id=persons.id
WHERE TPR.tune_id=tunes.id
AND TPR.tune_person_role_type_id=4
AND (persons.given_name ILIKE '%mari%'
OR persons.surname ILIKE '%mari%'
OR persons.nickname ILIKE '%mari%'))
AND genres.song_genre_id=5;

```

Joonis 13. Otsing esitaja ja laululiigi järgi.

```

SELECT DISTINCT tune_reference , text_reference ,
COALESCE(sound_reference , video_reference)
AS sound_video_reference ,
parishes.title AS parish , villages.title AS village
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS places
ON tunes.id=places.tune_id
LEFT JOIN folk_tune.parishes AS parishes
ON places.parish_id=parishes.id
LEFT JOIN folk_tune.villages AS villages
ON places.village_id=villages.id
WHERE EXISTS (SELECT 1
FROM folk_tune.tunes_persons_roles AS TPR
INNER JOIN folk_tune.persons AS persons
ON TPR.person_id=persons.id
WHERE TPR.tune_id=tunes.id
AND TPR.tune_person_role_type_id=4
AND (persons.given_name ILIKE '%marie%'
OR persons.surname ILIKE '%marie%'
OR persons.nickname ILIKE '%marie%'))
AND EXISTS (SELECT 1
FROM folk_tune.tunes_persons_roles AS TPR
INNER JOIN folk_tune.persons AS persons
ON TPR.person_id=persons.id WHERE TPR.tune_id=tunes.id
AND TPR.tune_person_role_type_id=4
AND (persons.given_name ILIKE '%anete%'
OR persons.surname ILIKE '%anete%'
OR persons.nickname ILIKE '%anete%'));

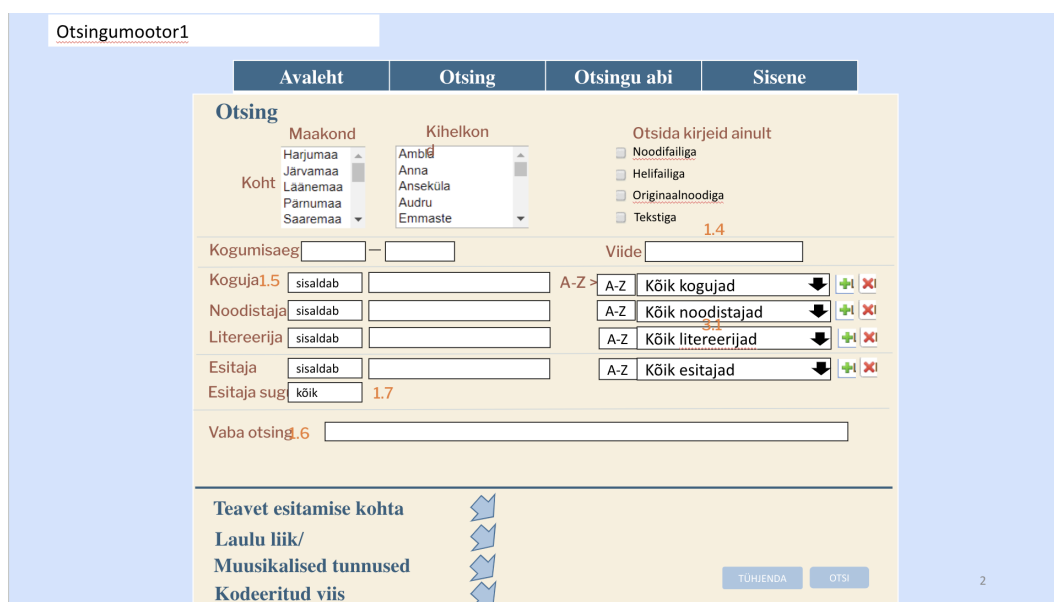
```

Joonis 14. Otsing kahe esitaja järgi.


Lisa 4 – Kavandi kuvatõmmised



Joonis 15. Kavandi avaleht.



Joonis 16. Kavandi otsinguleht.

Otsingutulemused		Avaleht	Uus otsing	Otsingu abi	Sisene			
Otsingutulemused 2.9								
< Tagasi otsingusse		Näitan 1-20 Kokku: 202		Mine lk <input type="text" value=""/> /11	Edasi > Lõppu >		EKSPORRT MULTI	
		Viide	Kihelkonnid	Küla	Päritolu	Esitaja	Laululiik	Aeg
 <input type="checkbox"/> A <input type="checkbox"/> X <input type="checkbox"/> X		EÜS III 56 (1)	Tartu-Maarja	Metsapöole	Puhja	Taive Viiding; Olga Ivaškevits	mardilaul; lastelaul	2016-2018

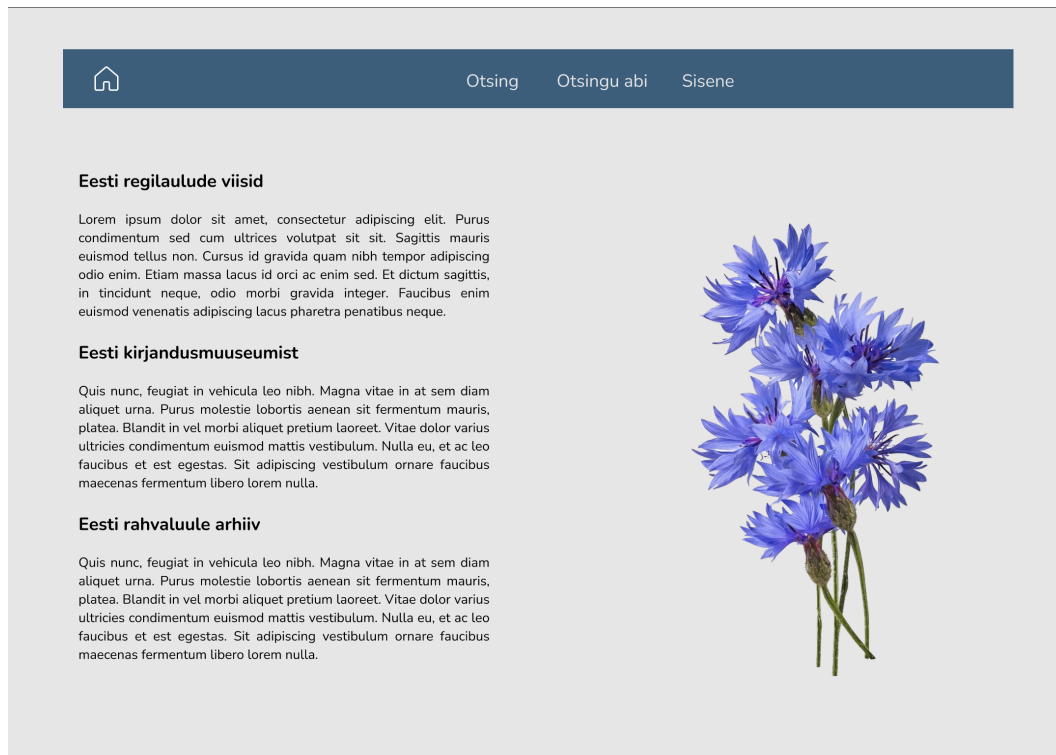
Näitan 1-20	Kokku: 202	Mine lk <input type="text" value=""/> /11	Edasi > Lõppu >	EKSPORRT MULTI
-------------	------------	---	-----------------	-------------------

Joonis 17. Kavandi otsingutulemuste leht.

Avaleht		Otsingutulemused	Otsingu abi	Sisene		
<h3>Kirje</h3> <p>< eelmine 2.7 Teiste järgmine > Kirje kuvamine tavakasutajale /Teade</p> <p>Täisviide: Viisi_viide_<_kihelkond[_,]+vald1[_,]+küla1[_,]+muu_koht_<_kihelkond_pärit[_,]+vald1_pärit[_,]+küla1_pärit[_,]+muu_koht_[-]_ühtlustatud_viisi_koguja[_,]_ühtlustatud_teksti_koguja_<_ühtlustatud_esitaja_nimi1[_,]vanus/sünniaasta[_,]_ühtlustatud_esitaja_nimi2_vanus/sünniaasta2_[(]Kogumisaeg_algus[-]Kogumisaeg_lõpp[)]</p>						
Üldandmed		Viisi viide		Viide märkus	2.6	
		Teksti viide		Viide märkus		
		Heli viide		Viide märkus		
		Video viide		Viide märkus		
		Kogumisaeg		Kogumisaeg märkus		
		Koguja				
		Viisi koguja	Nimi	Isikumärkus		
		Teksti koguja	Nimi	Isikumärkus		
		Esitaja				
		Esitaja	Nimi	Vanus/Sünniaasta	Isikumärkus	
		Koht				
		Päritolu				
		Esitaja	Nimi	Vanus/Sünniaasta	Isikumärkus	
		Koht				
		Päritolu				
		kohta teavet esitamise kohta				
		kuul				
		Muusikalised	Kodeeritud	Märkused		

Joonis 18. Kavandi viisiinfo leht.

Lisa 5 – Figma prototüüp



Joonis 19. Prototüübi avaleht.

Joonis 20. Prototüübi otsinguleht.

Maakond ja kihelkond

Lauale on võimalik otsida nii ühe kui ka mitme maakonna järgi, selleks tuleks teha klõps maakonna väljal ning valida rippmenüüst sobiv maakond; valikut hõlbustab otsitava maakonna nimetuse paari esimese tähe sisestamine. Kui maakond on valitud, siis ilmub selle nimi halli kasti ja edasi saab valida vajadusel teise maakonna. Valiku saab eemaldada, vajutades ristikesele hallis kastis maakonna nime ees. Mitme valitud maakonna korraga eemaldamiseks vajuta ristikesele selle filtri paremas ülanurgas.

Otsingutulemust saab vajadusel kitsendada kihelkondade valikuga. Maakonda valides ilmub paremale kastile klõpsides vastava maakonna kihelkondade loend. Kui maakonda ei vali, ilmub kõikide kihelkondade tähestikuline loend. Kihelkonna filter töötab samamoodi nagu maakonna filter. Maakonna ja kihelkonna filtreid saab rakendada ükshaaval või ka koos – see tähendab, et valida saab korraga maakonna ja kihelkonna tasandi kohti.

Koguja

Koguja järgi otsimiseks on kaks varianti: „on” ja „on muu kui”. Kui soovitakse otsida kindla koguja laule, valitakse koguja nimi (alustades perekonnanimest) ripploendist. Koguja filter töötab samamoodi nagu koha filtrid: koguja saab valida kasutades kerimisriba või esimesi tähti sisestades. Korraga saab valida mitu kogujat. Kui soovitakse mõne koguja laule valimist välistada, valitakse vasakpoolsest kastist „on muu kui” ja parempoolsest koguja nimi/nimed.

Kogumisaeg

Rippenüüs on kõik aastad, mil laule on kirja pandud. Vahel on lauluteksti juures mitu aastaarvu, näiteks kirjanemise aeg ja üleandmise aeg. Sellisel juhul on nn ühtlustatud ajaks, mille järgi andmebaas seda teksti otsib, kirjanemise (kogumise) aeg. Kui lauluteksti kogumisaeg on märgitud mingi ajavahemik, on laulu ühtlustatud ajaks selle ajavahemiku algusaasta.

Kui otsida laule, mille kogumisaasta on teadmata, tuleb valida kogumisaaja alampiiriks „teadmata”.

Kui kogumisaaja ülempiir on valitud ja alampiiri ei ole, siis „teadmata” valiku asemel tuleks määrata „aegade algusest”.

Joonis 21. Prototüübi otsingu abi leht.

Otsingutulemused 1200 tulemust

VIIDE	KIHELKOND	ESITAJA	LAULULIIK	AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG
LAULU VIIDE	LAULU KIHELKOND	LAULU ESITAJA	LAULULIIK	LAULU AEG

Rida lehe kohta 11 1-11 1200st < >

Joonis 22. Prototüübi otsingutulemuste leht.

Sisene

Registreerimiseks kontakteeru info@info.ee

Kasutajanimi

Parool

Jäta mind meelde

Sisene

Joonis 23. Prototüübi sisselogimise leht.

Lisa 6 – Kasutajaliidese kuvatõmmised

[Avaleht](#) [Otsing](#) [Isikud](#) [Sisene](#)

Eesti Rahvaviiside Andmebaas

Andmebaas vahendab muusikapärimust Eesti Kirjandusmuuseumi Eesti Rahvaluule Arhiivi (ERA) kogudest. Siit leiab eesti rahvaluuviiside üleskirjutused ja helisalvestused ning kodeeritud meloodiad. Viise on võimalik vaadata noodistusena, kuulata MIDI-failina ja soovi korral matemaatiliste meetoditega analüüsida. Viiside juurde on lisatud üldandmed, mõningad muusikalised tunnused ja laulutekste arhiiviviited, mille kaudu saab tekste kuvada.

Eesti rahvaviiside andmebaas on praegu katsetusjärgus ja hakkab aastate jooksul täienema. Praegu leiab siit ligikaudu 4000 kahelelise vormiga regiviisi ERA vanematest kogudest. Kaugemaks eesmärgiks on liita sellega kõik kokku kogutud eesti rahvaviisi, mille suurusjärg on arvestuslikult üle pooleaja tuhande.

Ajaloo

Ingrid Rüttil ja Koit Haugas koostasid teadusliku suunitlusega Eesti Rahvamuusika Andmebaasi 1980.-1990. aastatel, mis koondab ligikaudu 9000 kodeeritud regiviisi koos üldandmetega. Nimetatud andmebaas põhines omakorda Herbert Tampere koostatud viisikartoteegil, mis sisaldas ERA kogude koopiaid ning toimus Eesti Kirjandusmuuseumis kuu parima võimaliku rahvaviiside „andmebaasina“. Kartoteek sisaldas eelkõige käsikirjalisi viise ja fonograafisalvestuste noodistusi kuni 1975. aastani. Sinne infosüsteem arendab edasi Rüttili ja Haugase rajatud andmebaasi. Hakkasime seda ette valmistama programmi "Eesti keel ja kultuurimälu" raames 2015, sinne netirakendus valmis 2020-2021.

Teostajad

Eesti Kirjandusmuuseumi Eesti Rahvaluule Arhiivi etnomusikoloog Taive Särg (projektijuht), PhD, digiarhiivaar Olga Ivaškevitš.

Tallinna Tehnikaülikooli IT-osakonna üliõpilased Kristi Seemen, Martin Paroll, Katrin Avloi (2020).



Joonis 24. Rakenduse avaleht.

[Avaleht](#) [Otsing](#) [Isikud](#) [Sisene](#)

Noodifailiga Helifailiga Originaalnoodiga Tekstiga

Asukoht
Maakond Kihelkond

Kogumisaeg
algusaasta - lõpuaasta

Üldine teave

Esitamise teave

Laululiiik

Muusikalised tunnused



Kodeeritud viis

Näita kirjeid 10 kaup

Viisiviide	Tekstiviide	Heli-/videoviide	Kihelkond	Küla	Päritolu	Aeg
8083	EÜS X 2562 (12)		Kadrina	Saksi		1913
A 3476	A 3475 (1)		Tallinn	Leedi	Laiuse	1850
A 3476	A 3475 (1)		Tallinn	Leedi	Laiuse	1923
E 17149 (2)	E 17149 (2)		Tarvastu			1895
E 17150 (4)	E 17150 (4)		Tarvastu			1895
E 36477 (1)			Haljala	Vatku		1898
E 36700			Kadrina			1898
E 38093	E 38093/4		Haljala	Metsiku		1899
E 38200 (a)	E 38200 (a)		Pärnu			1899
E 38201 (c)	E 38201 (c)		Pärnu			1899

Kuvatud: 4,808 kirjet (1-10)

Eelmine 1 2 3 4 5 ... 481 Järgmine


Eesti Kirjandusmuuseum


Joonis 25. Rakenduse otsinguleht, sisselogimata.

Avaleht Otsing Isikud Sisene

märksõna Noodifailiga Helifailiga Originaalnoodiga Tekstiga

Näita kirjeid 10 kaupa

Asukoht Pärnumaa Karksi

Kogumisaeg 1910 - lõpuaasta

Üldine teave +

Esitamise teave +

Laululiiik +

Muusikalised tunnused +

Kodeeritud viis +

Viisiviide	Tekstiviide	Heli-/videoviide	Kihelkond	Küla	Päritolu	Aeg
EÜS VII 2721 (47)	EÜS VII 2639 (15a)		Karksi	Peraküla	määramata	1910
EÜS VIII 1137 (35)	EÜS VIII 1108/9 (87)		Karksi	Pöögile	määramata	1911
EÜS VIII 1138 (37)	EÜS VIII 1112 (89)		Karksi	Pöögile	määramata	1911
RKM II 104, 325 (1)	RKM II 94, 35 (1)		Karksi	Mötsakuru		1960
RKM II 104, 325 (2)	RKM II 94, 36 (2)		Karksi	Mötsakuru		1960
RKM II 104, 325/6 (3)			Karksi	Mötsakuru		1960
RKM II 104, 327 (6)			Karksi	Mötsakuru		1960
RKM II 104, 328 (8)	RKM II 94, 44 (9)		Karksi	Mötsakuru		1960
RKM II 104, 328 (9)	RKM II 94, 44/5 (11)		Karksi	Mötsakuru		1960
RKM II 104, 329 (13)			Karksi	Mötsakuru		1960

Kuvatud: 67 kirjet (1-10) Eelmine 1 2 3 4 5 6 7 Järgmine

Eesti Kirjandusmuuseum

ESTI RAHVALLISLE ARHIIV

Joonis 26. Näide otsingust kihelkonna ja kogumise algusaasta järgi.

Avaleht Otsing Isikud Sisene

märksõna Noodifailiga Helifailiga Originaalnoodiga Tekstiga

Näita kirjeid 10 kaupa

Viisiviide	Tekstiviide	Heli-/videoviide	Kihelkond	Küla	Päritolu	Aeg
8083	EÜS X 2562 (12)		Kadrina	Saksi		1913
A 3476	A 3475 (1)		Tallinn	Leedi	Laiuse	1850
A 3476	A 3475 (1)		Tallinn	Leedi	Laiuse	1923
E 17149 (2)	E 17149 (2)		Tarvastu			1895
E 17150 (4)	E 17150 (4)		Tarvastu			1895
E 36477 (1)			Hajjala	Vatku		1898
E 36700			Kadrina			1898
E 38093	E 38093/4		Hajjala	Metsiku		1899
E 38200 (a)	E 38200 (a)		Pärnu			1899
E 38201 (c)	E 38201 (c)		Pärnu			1899

Kuvatud: 4,808 kirjet (1-10) Eelmine 1 2 3 4 5 ... 481 Järgmine

Eesti Kirjandusmuuseum

ESTI RAHVALLISLE ARHIIV

Joonis 27. Rakenduse otsinguleht peidetud lisafiltritega.

Avaleht Otsing Isikud Sisene

Laulu tekst	Orginaalnoot	Noodifail	Helifailid	Noodistuse andmed		
Viited ▼						
Viisiviide	Tekstiviide	Heli-/Videoviide	Välisviite kirjeldus	Välisviite link	Väljaanded	Kartoteek
E 17150 (4)	E 17150 (4)	-	-	-	-	IX 19
Laul ▼						
Muusikalised tunnused ▼						
Kodeeringud ▼						
Viisiga seotud isikud ▼						

Joonis 28. Rakenduse detailne viisivaade.

Avaleht Otsing Isikud Sisene

Laulu tekst	Orginaalnoot	Noodifail	Helifailid	Noodistuse andmed		
Laulu tekst Mehed tulge meie poole meie lääne teie poole teeme üle neiu linna koame kume kiriku Sissi teeme siidilaeva Väla teeme väistelaeva Otseti odade laeva Nuketi nugade laeva						
	Helifailid puudub					
	Noodistuse andmed					
	Roll	Nimi	Aasta			
	-	-	-			
	Viited ▼					
	Viisiviide	Tekstiviide	Heli-/Videoviide	Välisviite kirjeldus	Välisviite link	Väljaanded
E 17150 (4)	E 17150 (4)	-	-	-	-	IX 19
Laul ▼						
Muusikalised tunnused ▼						
Kodeeringud ▼						
Viisiga seotud isikud ▼						

Joonis 29. Rakenduse viisivaade lauluteksti, helifaili ja noodistuse andmetega.

Laul								
Laulutüüp	Pealkiri	Algusvärss	Laululiik	Viisiliik	Laulu teema	Värsivorm	Refrään	Märkused
Neiu linn	Mehed tulge meie poole	Mehed tulge meie poole	lühirika	-	-	runo	-	-

Muusikalised tunnused							
Hellulatus	Viisi vorm	Teksti vorm	Rütmütüüp	Läbilaulude arv (noot)	Läbilaulude arv (heli)	On varieeruv	Märkused
määramata	Kaherealine	AB	-	2	-	Ei	-

Kodeeringud							
Alter	Tugiheli	Kõrgus	Taktimõõt	Rütmütüüp	Tempo	Noodipikkus	Märkused
-	a	-	2/4	1[2/3]	-	1/8	-

Viisiga seotud isikud
Koht

Joonis 30. Rakenduse viisivaade, täpsema info kuvamine.

Viisiga seotud isikud					
Roll	Nimi	Nime originaalkuju	Vanus	Kogumisaasta(d)	Märkused
viisikoguja	J. Tõllasson	J. Tõllasson	-	1895	-
tekstikoguja	J. Tõllasson	J. Tõllasson	-	1895	-

Koht							
Nimi	Koha liik	Maakond	Kihelkond	Vald	Küla	Muu koht	Märkused
-	elukoht	-	Tarvastu	-	-	-	-

Esitus					
Tegelik esitus	Traditsiooniline esitus	Tegelik tegevus	Traditsiooniline tegevus	Saade	Märkused
teadmata	-	-	-	-	-

Joonis 31. Rakenduse viisivaade, täpsema info kuvamine.

Avaleht Otsing Isikud Sisene


Näita kirjeid 10 kaupa

	Perekonnanimi	Eesnimi	Hüüdnimi	Sünniaasta	Surmaaasta	Sugu
1	Polakess	J.				3
2	Strutzkin	Aino				3
3	Maisa	Liisu				2
4	Neumanoff	Mari				2
5	Pallas	Jüri				1
6	Kääpasuu	Jaani				1
7	Garšnek	Anatoli				3
8	Laugaste	Gerda				3
9	Reinhold	Verner				1
10	Koitmaa	Hermilda				2

Kuvatud: 1,718 kirjet (1-10)

Eelmine 1 2 3 4 5 ... 172 Järgmine

Eesti Kirjandusmuuseum



Joonis 32. Rakenduse isikute vaade.

Avaleht Otsing Isikud Sisene

Isiku Andmed

Eesnimi J.
 Perekonnanimi Polakess
 Hüüdnimi ---
 Sugu teadmata
 Sünniaasta ---
 Surmaaasta ---
 Märkused ---

Seotud viisid


Näita kirjeid 10 kaupa

	Viisiviide	Tekstiviide	Heliviide	Videoviide
3441	EÜS VIII 184 (123)	N		
3969	EÜS VIII 183 (117)	N		
4230	EÜS I 202 (62)			

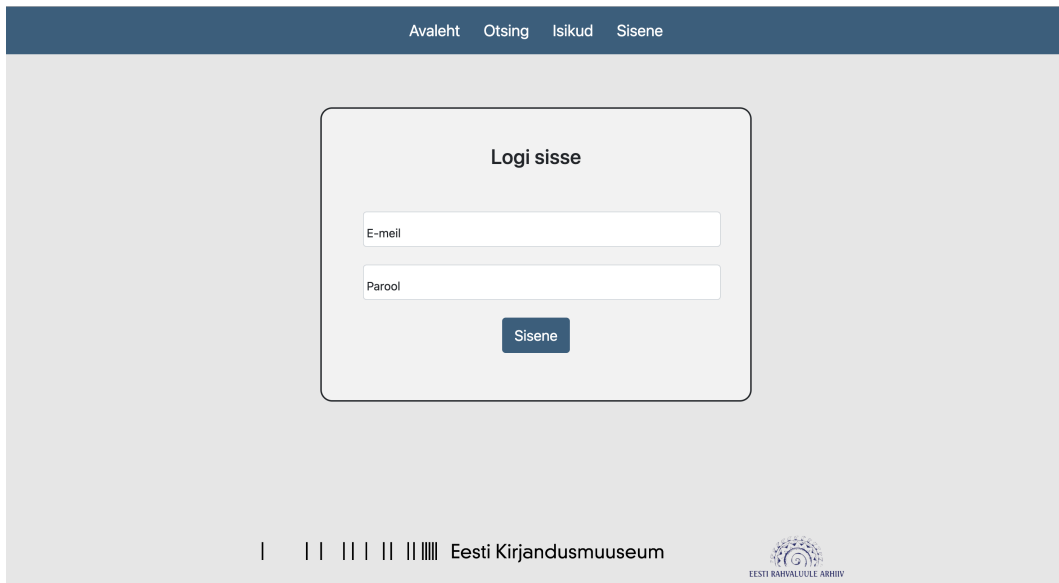
Kuvatud: 3 kirjet (1-3)

Eelmine 1 Järgmine

Eesti Kirjandusmuuseum



Joonis 33. Rakenduse ühe isiku vaade.



Joonis 34. Rakenduse sisselogimise leht.



Joonis 35. Rakenduse avaleht, sisselogitud.

Avaleht Otsing Isikud Klassifikaatorid Logi välja

märksõna Noodifailiga Helifailiga Originaalnoodiga Tekstiga

Asukoht
Maakond Kihelkond

Kogumisaeg
algusaasta - lõpuaasta

Üldine teave +

Esitamise teave +

Laululik +


Muusikalised tunnused +

Kodeeritud viis +

Näita kirjeid 10 kaupa

Viisiviide	Tekstiviide	Heli-/videoviide	Kihelkond	Küla	Päritolu	Aeg
8083	EÜS X 2562 (12)		Kadrina	Saksi		1913
A 3476	A 3475 (1)		Tallinn	Leedi	Laiuse	1850
A 3476	A 3475 (1)		Tallinn	Leedi	Laiuse	1923
E 17149 (2)	E 17149 (2)		Tarvastu			1895
E 17150 (4)	E 17150 (4)		Tarvastu			1895
E 36477 (1)			Haljala	Vatku		1898
E 36700			Kadrina			1898
E 38093	E 38093/4		Haljala	Metsiku		1899
E 38200 (a)	E 38200 (a)		Pärnu			1899
E 38201 (c)	E 38201 (c)		Pärnu			1899

Kuvatud: 4,808 kirjet (1-10) 1 2 3 4 5 ... 481

Eesti Kirjandusmuuseum 

Joonis 36. Rakenduse otsinguleht, sisselogitud.

Avaleht Otsing Isikud Klassifikaatorid Logi välja

Viited

Viisiviide	Tekstiviide	Heli-/Videoviide	Välisviite kirjeldus	Välisviite link	Väljaanded	Kartoteek
8083	EÜS X 2562 (12)					XIII 540

Laul

Muusikalised tunnused

Kodeeringud

Meloodiad

Viisiga seotud isikud

Koht

Esitus

Joonis 37. Rakenduse viisivaade, sisselogitud.

Viisiga seotud isikud

Roll	Nimi	Nime originaalkuju	Vanus	Algusaasta	Lõppaasta
viisikoguja	Karl Viljak			1913	
Märkused					
<input type="text"/>					
Roll	Nimi	Nime originaalkuju	Vanus	Algusaasta	Lõppaasta
esitaja	Kaarel Veiderpass	Kaarel Weiderbass	45	1913	
Märkused					
<input type="text"/>					
Roll	Nimi	Nime originaalkuju	Vanus	Algusaasta	Lõppaasta
tekstikoguja	Karl Voldemar Rosenstraut			1913	
Märkused					
<input type="text"/>					












Joonis 38. Rakenduse viisivaade, täpsem info, sisselogitud.

Laul

Muusikalised tunnused

Kodeeringud


Võtmemärk	Tugiheli	Kõrgus	Taktimõõt	Rütmitüüp	Tempo
	g			1	
Märkused					
<input type="text"/>					
Meloodiad					
Viisiga seotud isikud					
Koht					
Esitus					











 Eesti Kirjandusmuuseum
 

Joonis 39. Rakenduse viisivaade, täpsem info, sisselogitud.

Kodeeringud	▼						
Meloodiad	▼						
Viisiga seotud isikud	▼						
Koht	▲						
Nimi	Koha liik	Maakond	Kihelkond	Vald	Küla	Muu koht	Märkused
Kaarel Veiderpass	elukoht	-	Kadrina		Saksi		
Esitus	▲						
Tegelik esitus	Traditsiooniline esitus	Tegelik tegevus	Traditsiooniline tegevus	+	Saade		
Ü							
Märkused							
Meestelaul							

| | | | | | | | | | Eesti Kirjandusmuuseum

 EESTI RAHALOOLU ARHIIV

Joonis 40. Rakenduse viisivaade, täpsem info, sisselogitud.

Laul	▲		
Laulutüüp	Pealkiri	Algusvärss	Refrään
Talgulaul/Suur härg		Aituma pereosale, aituma pereemale	
Laululiik	+	Viisiliik	+
Laulu teema	+	Värsivorm	+
jutustav	▼		▼
talgu	▼		▼
Märkused			
Muusikalised tunnused	▼		
Kodeeringud	▼		
Meloodiad	▼		
Viisiga seotud isikud	▼		
Koht	▼		

Joonis 41. Rakenduse viisivaade, täpsem info, sisselogitud.

Meloodiad

0:00

Alter	Tempo	Noodipikkus	Autor
		1/8	EdnaTuvi

Meloodia

dB dB dc AB | ed AB AG GG

Sõnad

Märkused

Lisaväljad

Viisiga seotud isikud

Joonis 42. Rakenduse viisivaade, täpsem info, sisselogitud.

Muusikalised tunnused

Heliulatus Viisi vorm Teksti vorm Rütmitüüp Läbilaulude arv (noot) Läbilaulude arv (heli)

määramata 2 AB 4

On varieeruv

Märkused

Kodeeringud

Meloodiad

Viisiga seotud isikud

Koht

Esitus

Eesti Kirjandusmuseum

EESTI RAHALUULE ARHIIV

Joonis 43. Rakenduse viisivaade, täpsem info, sisselogitud.

Avaleht Otsing Isikud Klassifikaatorid Logi välja

Klassifikaatorid

Viis

Rahvus Keel Riik Seisund

Isik

Isiku roll Sugu

Koht

Koha liik Kihelkond Vald Küla

Esitus

Tegelik esitus Traditsiooniline esitus Tegelik tegevus Traditsiooniline tegevus

Laul

Laulu liik Viisi liik Teema Värsivorm

Joonis 44. Rakenduse klassifikaatorite vaade.

Avaleht Otsing Isikud Klassifikaatorid Logi välja

	Nimetus	Kirjeldus	On aktiivne	Tegevused
1	mees		Yes	Muuda Salvesta
2	naine		Yes	Muuda Salvesta
3	teadmata		Yes	Muuda Salvesta
4	ei ole võimalik määratleda		Yes	Muuda Salvesta

Esti Kirjandusmuuseum

EESTI RAHVALLAULE ARHIIV

Joonis 45. Rakenduse klassifikaatorite vaade, kindla klassifikaatori vaade.

Lisa 7 – Vabaotsingu koodinäide

```
CREATE MATERIALIZED VIEW folk_tune.tune_search_data AS
SELECT tunes.id AS tune_id ,
setweight(to_tsvector(
    coalesce(tunes.tune_reference , '') || ' ' ||
    coalesce(tunes.text_reference , '') || ' ' ||
    coalesce(tunes.sound_reference , '') || ' ' ||
    coalesce(tunes.video_reference , '') || ' ' ||
    coalesce(tunes.catalogue , '') || ' ' ||
    coalesce(tunes.publications , '') || ' ' ||
    coalesce(tunes.remarks , '' )
), 'A') ||
setweight(to_tsvector(
    coalesce(tune_places.other_place , '') || ' ' ||
    coalesce(tune_places.remarks , '') || ' ' ||
    coalesce(tune_performances.accompaniment , '') || ' ' ||
    coalesce(tune_performances.remarks , '' )
), 'B') AS document_tsvector
FROM folk_tune.tunes AS tunes
LEFT JOIN folk_tune.tune_places AS tune_places
    ON tunes.id = tune_places.tune_id
LEFT JOIN folk_tune.tune_performances AS tune_performances
    ON tunes.id = tune_performances.tune_id;
```

Joonis 46. Materialiseeritud vaate loomine.

Lisa 8 – Algne projektiplaan

Algsed sprintide kestvused ja eesmärgid

15.02 - 28.02

- Kasutajaliidese disainimise alustamine
- Andmebaasile ligipääsu saavutamine
- Otsingupäringute koostamise alustamine
- Rakenduse töölesaamine lokaalselt

01.03 - 14.03

- Kirjaliku osa alustamine
- Silbitamise funktsiooni lisamine
- Ühe viite põhjal töötava algse otsingu lisamine (ilma kasutajaliideseta)

15.03 - 28.03

- Kasutajaliidese arendamise algus
- Otsingule viidete lisamine

29.03 - 11.04

- Vabaotsingu lisamine
- Kasutajaliidese edasiarendamine

12.04 - 25.04

- Kirjaliku osa valmimine
- Kasutajaliidese valmimine
- Otsingusüsteemi valmimine

26.04 - 09.05

- Eelkaitsmise põhjal paranduste tegemine

10.05 - 22.05

- Eelkaitsmise põhjal paranduste tegemine
- Kaitsmiseks valmistumine

Lisa 9 – Lõplik projektiplaan

Lõplikud sprintide kestvused ja eesmärgid

15.02 - 28.02

- Kasutajaliidese disainimise alustamine
- Andmebaasile ligipääsu saavutamine
- Otsingupäringute koostamise alustamine

01.03 - 14.03

- Kirjaliku osa alustamine
- Silbitamise funktsiooni lisamine
- Rakenduse töölesaamine lokaalselt

15.03 - 28.03

- Kasutajaliidese arendamise algus
- Ühe viite põhjal töötava algse otsingu lisamine (ilma kasutajaliideseta)

29.03 - 11.04

- Otsingule viidete lisamine
- Kasutajaliidese edasiarendamine

12.04 - 25.04

- Esialgne kirjaliku osa valmimine
- Kasutajaliidese edasiarendamine

26.04 - 09.05

- Kasutajaliidese edasiarendamine
- Eelkaitsmiseks valmistumine

10.05 - 22.05

- Eelkaitsmise põhjal paranduste tegemine
- Vabaotsingu lisamine
- Viisiotsingu lisamine
- Kirjalik osa valmimine
- Haldusliidese valmimine

23.05 - 06.06

- Kasutajaliidese viimistlemine
- Süsteemi üleandmine kliendile
- Kaitsmiseks valmistumine