

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Tarkvarateaduse instituut

Hendrik Õunapuu 155376 IAPB

# **VIRTUAALSE RAHVAHANKELISE KLAVERI MULTIMEEDIARAKENDUS**

Bakalaurusetöö

Juhendaja: Jaak Henno  
Doktorikraad

Tallinn 2019

## **Autorideklaratsioon**

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

Autor: Hendrik Õunapuu

07.11.2018

## **Annotatsioon**

Eesmärgiks on programmeerida klaveri multimeedia veebirakendus, mis lubab rahval valmistada rahvahanke meetodil ühise jõuga klaverile lugusid ning mis näeks samal ajal kasutajaliidese poolest ilus, kerge ja mugav välja.

Projekti sisuks on kõigepealt muude rahvahankeliste projektide uurimine ning seejärel programmikoodi valmistamine. Uurimised toimusid Vikipeedia, Google'i tõlke, IARPA, reCAPTCHA, Folding@Home ning Akinator kohta. Lisaks uuriti ühte juba varem tehtud Brendon Ferris-i klaveri rahvahankeprojekti.

Programmeerimist alustati parima klaveri liidese valimisest, millele lisati seejärel helid. Pärast klaveri tegemist valmistati andmebaas, kuhu saab vajalikud lood ja muud andmed salvestada. Kolmandaks valmistati veebilehele funktsionaalne implementatsioon ning ühendati see serveri ja andmebaasiga. Viimaseks kujundati veebilehele ilus, mugav ja kerge kasutajaliides.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 25 leheküljel, 5 peatükki, 13 joonist, 0 tabelit.

## **Abstract**

### **Virtual Crowdsourcing Piano Multimedia Application**

The goal of this project was to program a multimedia piano web application, which allows people to use crowdsourcing method for creating a piano song together, and that looks beautiful, simple and comfortable at the same time.

The content of this project involves the research of a couple of other crowdsourcing projects and then the creation of the program code. The research was about Wikipedia, Google Translate, IARPA, reCAPTCHA, Foling@Home, and Akinator. In addition, another previously executed piano crowdsourcing project made by Brendon Ferris was researched as well.

The programming started with choosing the best interface for the piano and the addition of sounds to said piano. After that, the database was created for saving all the songs and other important data. Thirdly, the website functions were implemented and connected to the server and the database. And finally, the website was modified to look beautiful, simple and comfortable.

The thesis is in Estonian and contains 25 pages of text, 5 chapters, 13 figures, 0 tables.

## Lühendite ja mõistete sõnastik

Algoritm	Protsess või reeglite kogum, mida järgitakse arvutustes või muudes probleemi lahenduse operatsioonides.
Teek	Kellegi poolt valmistatud baasfunktsionaalsusi täitev programmikood, mida saab mujal kasutada.
Gradientne	Muutuvate atribuudi omadustega. Siin kontekstis kasutatud kui värvi sujuva üleminekuna ühest värvist teise.

## Sisukord

1 Sissejuhatus .....	9
2 Rahvahankelised projektid .....	10
2.1 Muud tuntud rahvahankelised projektid .....	10
2.1.1 Vikipeedia.....	10
2.1.2 Google'i tõlge .....	11
2.1.3 IARPA .....	11
2.1.4 ReCAPTCHA .....	12
2.1.5 Folding@Home .....	12
2.1.6 Akinator.....	12
2.2 Muusika loomise rahvahankeline projekt.....	13
3 Veebiklaveri funktsionaalsuse valmistamine .....	15
3.1 Baasklaver .....	15
3.2 Klaveri häälestus.....	16
3.3 MySQL andmebaas .....	17
3.4 Baasfunktsionaalsuste implementeerimine.....	19
3.4.1 Lugude automaatselt mängimine.....	19
3.4.2 Üheautorilise loo valmistamine .....	20
3.4.3 Valmisolevate lugude leidmine ja mängimine .....	21
3.4.4 Rahvahankelise loo valmistamine .....	22
3.4.5 Rahvahankeliste lugude töötlemine.....	23
3.4.6 Annetajate nimekiri .....	23
4 Klaveri veebilehe kasutajaliides .....	24
4.1 Klaveri kujundus.....	24
4.2 Pealehe kujundus .....	25
4.3 Üheautorilise lehe kujundus .....	26
4.4 Nimekirjaliste lehtede kujundused .....	28
4.5 Rahvahanke lehe kujundus .....	30
5 Kokkuvõte .....	32
Kasutatud kirjandus .....	34

Lisa 1 – Kõik klaveri klahvide noodid nimed .....	35
Lisa 2 – Kättesaadav ja töötav demo .....	36

## Jooniste loetelu

Joonis 1: Klaveri lõplik visuaal .....	16
Joonis 2: Üheautoriliste lugude tabel .....	18
Joonis 3: Rahvahankeliste lugude tabel.....	18
Joonis 4: Annetajate tabel.....	19
Joonis 5: Pealeht .....	26
Joonis 6: Üheautoriline leht .....	27
Joonis 7: Salvestusaken .....	28
Joonis 8: Lugude otsimine.....	29
Joonis 9: Rahvahanke nimekiri.....	29
Joonis 10: Annetajate nimekiri .....	30
Joonis 11: Rahvahanke lehe avalduse nupp .....	30
Joonis 12: Rahvahanke salvestusaken .....	31
Joonis 13: Rahvahanke avaldusaken .....	31



## 1 Sissejuhatus

Valga muusikakoolil tekkis hiljuti tahtmine osta kontsertklaverit, kuid kuna neil endil raha napib, otsustasid nad teha annetusfondi, kus inimesed saavad raha annetada, kuni lõpuks piisava summa kokku saab, et klaver ära osta. Et annetusfond huvitavamaks teha, mõtlesid nad välja idee, et võiks teha internetisisese interaktiivse annetuslehe. Kuna teemaks on muusika ning ostuplaaniks on klaver, siis võiks teha veebisisese klaveri, ning et interaktiivsust veel huvitavamaks teha, võiks lasta inimestel endil lugusid valmistada läbi rahvahanke meetodi – iga annetusega saab inimene valida ühe klahvi, mis lisatakse loole otsa, kuni lõpuks pärast mitmeid annetusi saab lugu valmis.

Kuna idee iseenesest ei ole eriti raske, siis otsustati ise programmeerimise asemel anda see võimalus ülikooli lõpetavatele tudengitele lõputöö teemaks valida. Selle idee võtsid vastu kaks tudengit Tallinna Tehnika Ülikoolist, seehulgas ka mina. Töö eesmärgiks on eelnevalt kirjeldatud probleem lahendada, ehk programmeerida rahvahankelise klaveri veebirakendus. Töö alla ei kuulu pangaga ühendus – pangandusega seotud teemad lahendab Valga muusikakool ise. Teha on vaja vaid töötav kasutajaliides koos serveriga, mida saaks teoreetiliselt kasutada.

## **2 Rahvahankelised projektid**

Rahvahankelisis võib alguses tunduda kui väga võõras sõna, aga see on vaid seetõttu, et seda sõna spetsiifiliselt eriti ei kasutata. Tegelikuses on maailmas juba üsnagi palju erinevaid rahvahankelisi projekte olnud, millest on osad ka praegu erinevatel platvormidel kättesaadavad. Rahvahankelise alla ei kuulu vaid muusika, vaid sinna alla kuuluvad kõik projektid, millesse on kaasatud rahva enda sisendandmed, olgu need konkreetsed (valikvastused, numbrilised parameetrid, jne) või abstraktsed (arvamused, ennustused, jne). Me võime neid isegi igapäevaselt kasutada, teadmata, et need on tegelikult rahvahankel tehtud.

### **2.1 Muud tuntud rahvahankelised projektid**

#### **2.1.1 Vikipeedia**

Üks kõige tihemini kasutatavaid rahvahankelisi platvorme, mida väga paljud on kasutanud ning siiani kasutavad, on Vikipeedia. Inimesed ei pruugi mõelda sellest kui rahvahankelisest projektist, aga nii see just on. Selle veebilehe avalehel on öeldud, et Vikipeedia on „vaba entsüklopeedia“ ning „avatud veebientsüklopeedia“ ja et seda saavad kõik vabalt kasutada, täiendada ja parandada, mis otseselt viitab sellele, et Vikipeedia on ametlikult rahvahankeline platvorm.

Vikipeedia on viies kõige enim kasutatud veebiplatvorm, mille rajasid Jimmy Wales ja Larry Sanger. Vikipeedia alustas 2001 aasta jaanuaris inglisekeelse veebisõnaraamatuna, mis hiljem sai ka eestikeelse varianti 2002 aasta augustis. See leht koosneb erinevatest artiklitest, mis kõik räägivad mingist kindlast teemast. Nendel artiklitel saab teisi artikleid viidata läbi linkide, millele vajutades kasutaja sinna suunatakse. Läbi kõigi linkide tekkinud linkide rägastik on nii põhjalik ja sügav, et teoreetiliselt peaks võimalik olema igast artiklist igasse teise artiklisse jõudma ainult linkidele vajutades.

Vikipeedia kontsept seisneb selles, et terve domeen on vabalt töödeldav ja täiendatav – lisada saab uusi artikleid, kõikidesse artiklitesse saab midagi kirjutada ning kõiki artikleid saab hiljem parandada ja täiendada. See tähendab, et terve Vikipeedia on põhimõtteliselt

arendatud rahva poolt. Informatsiooni tõesus ei ole seetõttu eriti usaldusväärne, kuna informatsioon võib olla moonutatud erinevate tõekspidamiste, kultuuride jms tagajärjel, või võib mõni pahatahtlik inimene tahtlikult informatsiooni valeks muuta (seepärast pannakse ka pahaks Vikipeedia kasutamist ametlikel dokumentidel, seehulgas ka lõputööd), aga üleüldiselt võib suurema tõenäosusega oletada, et loetud informatsioon on siiski tõene.

### **2.1.2 Google'i tõlge**

Google asutas 2006 aasta aprillis Google'i tõlke platvormi, mille ülesandeks on erinevate keelte tõlkimine ühest keelest teise. Selle asemel, et tõlkida keeli otse, tõlgitakse iga keel vahepeal inglise keelde, kuna see andis paremaid tõlkimistulemusi ning selle jaoks on vähem andmeid vaja. Google'i tõlge kasutas alguses närvivõrgulist õppimist läbi ÜRO ja Euroopa Parlamendi transkriptide, mida kasutati algoritmi õpetamiseks. See ei andnud tihti kõige paremaid tulemusi, mistõttu lisati hiljem veel üks lisafunktsionaalsus.

2014 aastal käivitati Google'i kogukonna platvorm, mis üritab tõlketulemusi parendada läbi rahvahanke meetodi – vabatahtlikud inimesed saavad ülesandeid mingite lausete tõlkimiseks ning sellest kogutud andmeid kasutatakse parema tõlke leidmiseks. Sellest ajast saati on tõetud, et tõlkimise tulemused on tõesti paranenud.

### **2.1.3 IARPA**

IARPA (Intelligence Advanced Research Projects Activity) on organisatsioon, mis investeerib kõrge riskiga, kõrge tasuliste uuringu programmidesse et tegeleda Intelligentsusühenduses mõnede agentuuride ja erialade raskeimate väljakutsetega. Hiljuti hakkasid nad vaatama võimalusi kasutada rahvahankelist meetodit geopoliitiliste sündmuste ennustamiseks [1].

Intervjuus räägitakse, kuidas neil on plaanis ennustada geopoliitilisi sündmusi läbi rahvahanke reaalajas. Et inimesi motiveerida, on mängus 200'000 dollarit. Sellise ettevõtte põhjuseks on parem tõenäosus nende sündmuste ennustamiseks, kuna sadade või tuhandete inimeste arvamused annavad lõppkokkuvõttes parema ülevaate mingi sündmuse toimumise tõenäosusest võrreldes ühe kindla targa inimese ennustusega. Mõnikord võivad need eksperdid õigesti ennustada, mõnikord mitte. Keegi ikka ennustab õigesti, aga selles, kes ennustab, ei saa keegi täielikult kindel olla. Kuid see ei tähenda, et ekspertide arvamused oleksid tähtsusetud – vastupidi, ekspertide arvamused on rahvale

tähtsad, kuna eksperdid pakuvad kriitilist konteksti nähtuste arusaamiseks ning nende arvamuste põhjal tekivad uued arvamused teistel inimestel. Läbi nende muutuste on ka võimalik näha reaalarvamusi inimeste arvamuses, mis esineksid ka antud rahvahanke projektis.

#### **2.1.4 ReCAPTCHA**

CAPTCHA on taoline arvutisisene test, mis üritab aru saada, kas tegemist on inimese või arvutiga. ReCAPTCHA kasutab CAPTCHA't, et kaitsta programmirobotitel sisenemast veebis kohtadesse, mis on rangelt mõeldud vaid inimestele käsitsi sisenemiseks.

Programm kasutab inimeste poolt tehtud pilte erinevatest tekstidest kas raamatutest, tänavatelt, maja peal olevast tekstist või mingi oma looming ning sama inimese poolt antud vastusest, mida see tekst täpselt ütleb. Selle abil saab arvuteid segadusse ajades, kuna arvutid ei ole võimelised neid tekste lugema, kuid inimeste jaoks on see kerge.

#### **2.1.5 Folding@Home**

Folding@Home on rahvahankeline projekt, mis seadis end uurima haigusi. Projekt simuleerib valkude kokkuklapitavust, arvutuslikku ravimi disaini ning muid sorte molekulaardünaamikat. Need uuringud lähevad sisendiks paljudele teistele meditsiinilistele uuringutele, mis uurivad Alzheimeri tõbe, Huntingtoni tõbe ning muid vähi tüüpe.

Projekti sisu seisneb selles, et neid uuringuid saab aidata läbi viia rahvas. Uuringud võtavad palju arvuti protsessori ressursse, mistõttu on vabatahtlike inimeste arvutid siin väga olulised, sest see annab kiiremad tulemused, kui kõik ülesanded korraga laiali jagada ja siis kokku koguda. Inimestele saadetakse mingid ülesanded, millele leitakse aja jooksul vastus, mis seejärel saadetakse tagasi.

#### **2.1.6 Akinator**

Rahvahankelised projektid ei pea alati olema seotud mingi sünge teemaga või olema tingimatult rahvale kasulik. Selle tõestamiseks on olemas selline projekt nagu Akinator, mis valmistati 2007 aastal.

See projekt põhineb ühel mängul nimega „Kakskümmend küsimust“, kus üks inimene valib omale mingi objekti või eriala ning teised üritavad kahekümne küsimusega ära

arvata, millest ta mõtleb, kusjuures kõik küsimused peavad algama sõnaga „Kas“ ning kõik vastused peavad olema kas „Jah“, „Ei“ või „Võib-olla“. See mäng sai hiljem ka Eestis telesarja nimega „Kuulus ja kummaline“, kus kolm tarka teadlast üritasid ära arvata ühe suvaliselt leitud inimese töökoha.

Akinator võttis selle idee ning võttis ära arvamiseks kõik maailma tegelased, millel on nimi, seehulgas ka videomängudest, filmidest ning ka lastesarjadest. Akinator üritas ära arvata 25 (kuni maksimaalselt 75) küsimusega, millisest tegelasest inimene mõtleb. Alguses ei olnud Akinator eriti tark, aga iga kord, kui kasutaja jõuab lõppu ning näeb, et tema mõeldud tegelane ei ole Akinator andmebaasis, saab ta selle tegelase sinna lisada ning kõik eelnevalt vastatud küsimused seostatakse selle tegelasega. Selle tulemusena lisandub see uus tegelane Akinator andmebaasi ning seda on võimalik kindlate küsimustega ära arvata. Mida rohkem kedagi arvatakse, seda suurema tõenäosusega leiab Akinator selle tegelase üles.

Akinator ei oleks kunagi nii populaarseks muutunud, kui inimesed poleks aidanud sellel leida parim algoritm küsimuste esitamiseks ning kui tegelasi ei oleks juurde lisatud. Akinator ise kasutab närvivõrgulist algoritmi, mis üritab iga küsimusega kõik teadaolevad tegelased pooleks jagada. Sellise algoritmiga leiaks Akinator 25 küsimusega üles maksimaalselt  $33'554'432$  tegelast, kui kõik oleksid piisavalt erinevad ning võrdselt jagatud, ning 75 küsimusega tõuseks see arv kosmiliseks.

## **2.2 Muusika loomise rahvahankeline projekt**

Rahvahankelise muusika loomise teemal on üks kindel näide, mida saab välja tuua. Selleks näiteks on üks 2015 aasta septembris loodud projekt, kus 67'000 inimest koostasid ühe meloodia [2]. Programmeerija Brendon Ferris mõtles välja idee, kuidas kasutada interneti ühist jõudu, et üks muusikapala luua. Brendon programmeeris ühe veebilehe, lisas sellele kaks nooti, andis sellele mõned baasreeglid, ning andis ülejäänud loomingulise töö üle internetile.

Ferris arvas, et pika aja jooksul saab ta äkki sada nooti täis, aga tema üllatuseks tõusis tema veebileht esimese päevaga Reddit'i esileheküljele, mille tagajärjel hakkasid tuhanded inimesed laulule noote juurde kirjutama. 11 kuud hiljem oli laul valmis ning nootide koguarvuks jäi 67'167 nooti. Üllataval kombel on aga ka laulu meloodia ise väga

hea. Selle põhjuseks peetakse inimeste teadmiseid selles, mis teeb muusika heaks, isegi kui nad ise üksinda midagi valmistada ei oskaks.

Rahvahanget on kasutatud tuhandetes projektides, aga Ferris'i projekt paistab sealt välja paremini kui paljud teised, sest selle tulemusena tekkis kunstiteos. Tavaliselt ei peeta rahvahanget kunstiks, aga see projekt pani inimesi mõtlema, et äkki rahvahankes on ikkagi mingit kunsti sees.

Ferris kasutas sama meetodit, et anda loodud laulule ka laulutekst. Lisaks kasutab ta ka sama tehnikat, et luua veebileht, kus saab luua rohkem kui ühe laulu. Tema lootuseks on luua Vikipeedia sarnane veebileht, kus saab mitmekesi koos lugusid luua.

### 3 Veebiklaveri funktsionaalsuse valmistamine

Et valmistada veebiklaver, on selle jaoks vaja palju erinevaid etappe. Kõigepealt on vaja teha või leida klaver, seejärel on vaja see modifitseerida selliseks, mida on võimalik kergelt kasutada ja edasi arendada vajalike funktsionaalsuste implementeerimiseks. Seejärel on vaja andmebaasi, kus hoiustada vajaminevaid andmeid, ning lõpuks on vaja klaveri funktsionaalsuste implementeerimine ja selle ühendamise andmebaasiga. Viimaseks oleks vaja ka veebilehe kasutajaliidese viimistlust, et kasutajatel seda võimalikult mugav kasutada oleks.

#### 3.1 Baasklaver

Alustuseks tuleks leida või valmistada sobiv klaver, mida projekti jaoks kasutada. Leidmise all mõeldakse teegi kasutamist – leida ja alla laadida mingisugune teek, mis on kellegi teise poolt valmistatud ning mida saaks kasutada enda programmikoodis. Programmeerimises tihti öeldakse, et „ära leiuta jalgratast“, mis tähendab, et ei ole mõtet leiutada uut programmikoodi, mille jaoks juba kood eksisteerib. Tihti on aga kergem kasutada enda tehtud koodi, sest see on tuttavam ja seal on vaid täpselt need funktsionaalsused, mida sa ise lisasid, ja mitte midagi muud.

Üks parimaid teekke, mida ma leidsin, on Danigb poolt leiutatud helifondi mängija [3]. See on HTML, CSS ja JS-ga koostatud teek, mis tekitab veebilehele koheselt mängitava klaveri, millel on 84 klahvi. Aga nagu karta oli, tekkis teegi modifitseerimisega probleeme, sest klaveri visuaalset poolt oli raske muuta ning klaveri funktsionaalsust oli raske kasutada. Lisaks tekkis sellega kaasa üle 1300 erinevat faili, millest suurem osa ei pruugi isegi kasulik olla. Klaveril mängitavad noodid olid aga väga head ja kvaliteetsed, nii et need võtsin ma enda klaverile kasutamiseks.

Lõppkokkuvõttes otsustasin ma aga klaveri ise valmistada, aga mitte päris nullist, vaid ühe teise veebist saadava klaveri järgi. Klaver, mille järgi ma enda oma tegin, on Apronus lehelt leitav Virtual Piano Online [4]. Selle klaveri visuaal on väga ahvatlev – kõiksugused varjud, kumad ja tuhmumised aitavad klaveri väljanägemist väga parandada, nii et see ei näeks väga monotoonne välja. Küll aga klahvi alla vajutades klahvi nihkumine madalamale polnud kõige parem, nii et selle visuaalse komponendi muutsin ma ära – selle

asemel ma otsustasin klahvi muuta tumedamaks, ning klahvide peale oli kirjutatud vastava klahvi noot, mille ma samuti ära eemaldasin.

Valga muusikakooli nõue oli aga teha klaver, millel on 88 klahvi, kuid antud klaveri on vaid 25 klahvi, aga kuna klaveri visuaali oli väga kerge muuta, siis sai kergelt lisada klahve juurde. Klahvide paigutamine toimus blokkide kaupa – viiesed blokid noodist C kuni E ja seitsmesed blokid noodist F kuni B. Kõik klaveri osade pikkused ja laiused olid ka protsentides kirjeldatud, mille tulemusena oli klahvide lisamine veelgi kergem. Kahjuks oli mul aga ainult 84 nooti, mida mängida, nii et lõppkokkuvõttes jäi tulemuseks 84 klahviga klaver (Joonis 1), mis ei tohiks eriti midagi muuta.



Joonis 1: Klaveri lõplik visuaal

Klaver peaks ka mobiilidel töötama, aga klaveri kujunduse tõttu ei pruugi see visuaalselt nii hea välja näha ega väga mugav kasutada olla, sest klaveril on 84 klahvi, mis kõik ühel real asuvad, ehk mobiilil võivad klahvid olla liiga väikesed. Seda annaks ilmselt parandada, kui mobiiliversioonis klaver ainult vaadeldavaks teha ning teha mingi uus kasutajaliides selle kasutamiseks, aga selleks mul aega ei jäänud, nii et projekti poolest fokuseerisin ma rohkem arvutiversioonile ja et see töötaks kõigis brauserites.

### 3.2 Klaveri häälestus

Nagu varem mainitud, võtsin ma eelmainitud teegist kõik 84 noodi helifaili oma klaveri jaoks kasutusele. Klaverit üles seadistades seadistasin ma ka iga klahvi peale vajutades sellele spetsiifilise helifaili, mida mängima peaks. Klaverit häälestades tekkisid aga kummalised probleemid, mille kohta õppisin ma lisainfot selle kohta, kuidas erinevad brauserid töötavad.

Alguses kasutasin ma MP3 faile, mis töötas väga hästi Chrome brauseris, aga proovides seda mängida näiteks Firefox brauseris ei töötanud. Pärast veidike uurimist sain ma teada, et Firefox ei luba JavaScript-ga tehtud uut audio muutujat teha, kui selle laienduseks on



MP3, ning selle põhjuseks on potentsiaalsete autoriõiguste seaduste rikkumine. MP3 helifail ise on lubatud, kui see on HTML-ga seatud, aga muudel juhtudel see lihtsalt ei lähe läbi. Õnneks laseb Firefox aga kõik muud helifaili laiendustega failid läbi, aga nende puhul tekkis brauserite toetusega probleeme, sest osad brauserid ei toeta ühte laiendust ning osad ei toeta teist ning mitte ükski MP3-st erinev laiendus ei ole toetatud kõigil brauseritel. Lõppkokkuvõttes seadistasin ma helifailid üles nii, et OGG failid on vaikimisi laienduseks kõigil brauseritel väljaarvatud Internet Explorer, Microsoft Edge ja Safari, mille puhul peaks mängima hoopis WAV fail. Helifailide konverteerimiseks kasutasin ma tasuta programmi nimega Audacity. Kuna helifailid on kõik topelt, tuli kõigi helifailide kogusuuruseks 33MB, mis on üpriski kopsakas.

### 3.3 MySQL andmebaas

Andmebaasi tegemiseks oli mitu erinevat võimalust mitme erineva programmeerimiskeelega, nagu Python ja tekstifailid, või hoopis Github, aga kõige rohkem kogemusi on mul PHP ja MySQL-ga, millega ma võisin kõige kindlam olla, et suudan teha töötava andmebaasi. Andmebaasiks kasutasin kooli poolt antud oma Dijkstra alamkataloogi, mis seda toetas.

Mul oli plaan teha mõlemad – nii üksinda tehtava loo funktsionaalsus kui ka rahvahankel tehtava loo funktsionaalsus, nii et mul oli vaja teha mitu tabelit, kuhu need salvestada. Esimeseks on tabel, kuhu saab salvestada üksinda tehtud lood – sellel peab olema loo nimi, loo autor ning loo mängimiseks vaja olev rodu klahvide andmeid, mida salvestama peab (Joonis 2). Teiseks on tabel, kuhu saab salvestada rahva poolt tehtud lood – sellel peab olema loo nimi, klahvide andmed ning loo seisund. Kuna autoriks on palju inimesi, siis autori veergu sinna vaja ei ole (Joonis 3). Viimaseks on tabel, kuhu saab salvestada annetajate andmed – minu algelises versioonis pidasin ma vajalikuks ainult annetaja nime ja annetuse suurust (Joonis 4).

Igal tabelil on esimeseks veeruks id, mis on igal real erinev ning suureneb ühe võrra iga uue rea sisestamisel. See on pigem kohustuslik väli, mis garanteerib, et tabelis pole mitte ühtegi rida, mis oleksid täpselt samad. Nimede veergudeks seadsin ma andmetüübiks VARCHAR(60), mis tähendab, et seal võib olla ükskõik mis täht, ning kokku võib tähti olla kuni 60. Ma pidasin 60 piisavalt suureks numbriks, millega nime määrata. See ei tohiks olla liiga lühike ega liiga pikk.

Klahvide andmete jaoks kasutasin ma VARCHAR(20000), mis lubab kuni 20'000 tähemärki. Selle arvu leidsin ma halvima juhtu arvutades – iga klahvi jaoks seadsin ma maksimaalselt neli tähemärki, iga noodi mängimise vahele seadsin ma 230 millisekundit pausi, sellest järeldades on iga sekundi jaoks halvimal juhul keskmiselt 17,4 tähte, selle korrutasin 60-ga, et saada maksimaalne tähtede arv minutis, ning maksimaalseks halvima juhu pikkuseks seadsin ma 20 minutit. Korrutades viimast arvu 20-ga sain ma arvu 20'870, mille ma ümardasin alla 20'000-ni. Sellest see arv tuleneb.

Rahvahanke laulude tabelis on ka loo seisundi liik andmetüübiga BOOLEAN, mis tähendab, et selle väärtus võib olla kas tõene või väär – selle veeru funktsionaalsus seisneb määramaks, milliseid rahvahanke laule on võimalik veel muuta. Kui laul on valmis, siis seda muuta enam ei saa – siis saab vaid kuulata. Viimaseks on annetajate tabelis annetuse andmetüübiks DOUBLE(6, 4), mis tähendab, et see võimaldab hoida väärtuseid 0,00 kuni 9999,99. Võib üpriski julget oletada, et keegi üle 10'000 euro ei anneta, nii et sellisest andmetüübist peaks piisama.

```
CREATE TABLE t155376_singlesongs(  
    id INTEGER AUTO_INCREMENT,  
    songname VARCHAR(60),  
    author VARCHAR(60),  
    data VARCHAR(20000),  
    PRIMARY KEY (id)  
);
```

Joonis 2: Üheautoriliste lugude tabel

```
CREATE TABLE t155376_crowdsongs(  
    id integer AUTO_INCREMENT,  
    songname VARCHAR(60),  
    data VARCHAR(20000),  
    finished BOOLEAN  
    PRIMARY KEY (id)  
);
```

Joonis 3: Rahvahankeliste lugude tabel

```
CREATE TABLE t155376_donators(  
    id integer AUTO_INCREMENT,  
    name VARCHAR(60),  
    amount DECIMAL(6, 4),  
    PRIMARY KEY (id)  
);
```

Joonis 4: Annetajate tabel

## 3.4 Baasfunktsionaalsuste implementeerimine

### 3.4.1 Lugude automaatselt mängimine

Esimeseks baasfunktsionaalsuseks on mingisuguse loo automaatselt mängimine mingi noodistiku järgi. Selle jaoks on kõigepealt vaja defineerida, kuidas noodistikku lugeda. Mina otsustasin noodistiku defineerida üherealise noodirodu viisil, kus iga noot on eraldatud semikooloniga ning iga noot on oma nimega, kusjuures mustad klahvid on defineeritud kui sellele vasakule poole jääv valge noodi nimi pluss täht „u“. Lisaks defineerisin ma ka pausinoodi, milleks on mul sidekriips. Selle baasiga on näiteks üks sobilik noodirodu „G3;C4;B3;C4;D4;C4;A3u;C4;A3u;-;G3u;-;G3u;-;-;“, mida mängides võib aru saada, et need on esimesed kuusteist nooti laulust „Sinine vagun“. Kokku on 84 võimalikku nooti ning pausinooti (vt Lisa 1).

Nüüd kui meil on olemas viis kuidas noodirodu defineerida, saab lugusid automaatselt mängida. Kui anname programmikoodile selle rodu sisse, saab kood selle jagada koolonite kaupa nimekirjaks. Eelnevalt nimetatud helifailide nimed on täpselt samasuguste nimedega salvestatud, mille tulemusena peab vaid noodi nimele lisama taha brauserile vastava failitüüpi toetava laienduse ning siis saab helifaili välja kutsuda ning seda mängida, väljaarvatud pausinooti, mille puhul ei mängita midagi.

Korruga saab mängida vaid ühte nooti, vähemalt selles versioonis. Ma otsustasin, et iga noodi mängimise järel peaks olema 230 millisekundit pausi. Täpne arv tuli pigem kuulamise ka katse-eksitus meetodi järgi kui mingi kindla arvutuse järgi – see lihtsalt kostus kõige sujuvam ja normaalsem. Lugu saab mängida nupule vajutades ning loo mängimise ajal nupu vajutamine ei tee midagi. Loo lõppedes saab jälle lugu uuesti mängida nupule vajutades.

### 3.4.2 Üheautorilise loo valmistamine

Otsus lisada ka loo valmistamine algusest lõpuni ainult ühe autori poolt tekkis pigem ideena lihtsalt sellepärast, et saada ülevaade, kuidas kõik funktsionaalsus peaks välja nägema ja et saada läbi proovida versioon, mis peaks teoreetiliselt lihtsam olema programmeerida.

Esiteks on mul vaja viisi, kuidas mängitud klahvid kätte saada. Selleks lisasin klahvi vajutuse programmikoodi alla täiendkoodi, mis salvestab mängitud klahvi noodid nime ühte nimekirja. Lisaks arvasin, et oleks hea idee seda nimekirja ka kasutajale näidata, et ta näeks, milliseid klahve ta juba vajutanud on. Selleks lisasin samasse koodi ka HTML visuaali generaatori, mis lisab selleks määratud kohta kõik noodid sellises formaadis, et iga noot oleks eraldi real. See võtab küll rohkesti ruumi, aga see on kõige lihtsam ja loogilisem kui kaks veergu või mingi muu troopiline viis. Lisaks võiks kõige hiljem lisatud noodid olla kõige üleval, et kasutaja ei peaks kõige alla kerima, et viimaseid noote näha, mille jaoks ma lihtsalt käisin nimekirja tagurpidi läbi.

Kasutajal võib ka tekkida tahtmine lisada pausinoot või hoopis mingi noot eemaldada. Noodi eemaldamiseks lisasin ma spetsiaalse nupu, mis eemaldab kõige viimasena lisatud noodid nimekirjast. Nupule mitu korda vajutades saab järjest viimaseid noote eemaldada. Spetsiifilise noodid eemaldamist otsustasin ma mitte lisada, sest siis tekiks nimekirja auke ning nende aukude täitmine osutuks problemaatiliseks ning kasutajale ebamugavaks, nii et ma otsustasin vaid viimase noodid eemaldamise nupu kasuks. Ning kuna klaveri klaviatuuril pausinuppu pole, siis selle jaoks lisasin samuti spetsiaalse nupu, mis nimekirja lisab pausi.

Viimaseks lisasin eelmainitud kahele nupule lisaks veel kaks nuppu. Üheks neist on loo loomise katkestamise nupp, mis lihtsalt viib tagasi pealehele, kus on vaid klaver. See nupp on igaks juhuks sinna lisatud, et kasutaja tunneks ennast turvalisemalt ning ei peaks muretsema, mis saab juhul, kui ta lugu enam jätkata ei tahaks. Teiseks on salvestuse nupp, mis on muidugi kohustuslik, sest muidu ei saaks lugu salvestada. Salvestuse nuppu vajutades tekib ekraanile väiksem aken, kus on kaks tekstivälja ja kaks nuppu, millest üks nupp on jällegi katkestusnupp, mis lihtsalt selle väikse akna ära kaotab ja lugu edasi töödelda laseb.

Salvestusakna kaheks tekstiväljaks on loo nime ja autori nime kastid. Loo salvestamiseks on vaja loole lisada mingi nimi, et selle hiljem ära tunneks. Teiseks oleks vaja loole lisada loo autor – see võib olla nii loo originaalne autor, mille järgi loodud lugu on tehtud, või mingi muu suvaline nimi. Mõlemad nimed ei tohiks olla tühjad sõned – vastasel juhul salvestamise päring ei lähe läbi. Loo andmed ise tulevad nimekirjast automaatselt – seda kasutaja lisama ei pea. Kui lugu on salvestuseks valmis, võib kasutaja vajutada salvestamise nupul. Kõik vajalikud andmed saadetakse edasi serverisse, kus kontrollitakse, et kõik saadetud andmed vastaksid kindlatele reeglitele, mis kindlustab, et mingi pahatahtlikum kasutaja valeandmeid saata ei saaks. Seejärel saadetakse andmed andmebaasi, kuhu need salvestatakse.

### **3.4.3 Valmisolevate lugude leidmine ja mängimine**

Kui juba üks või paar lugu on valmis, siis peaks neid lugusid kuskilt leidma, et neid kuulata saaks. Selleks tekitasin ma ühe lisalehekülje, kuhu lisan kõik valmisolevad lood nimekirjana ülevalt alla. Lugude lisamine toimub läbi serveri, mis need sinna automaatselt lisab.

Server teeb andmebaasist kaks päringut – esimesena leiab ta kõik rahvahankelised lood vastavast tabelist, mille seisund on tõene, ehk kõik lood, mis on lõpetatud ning mida enam muuta ei saa. Praegu pole ühtegi sellist lugu, aga hiljem peaks see päring ka tagastama mõned lood. Teises päringus leiab ta kõik üheautorilised lood. Kuna kõik üheautorilised lood on automaatselt valmis ja lõpetatud, siis siin erandeid ei ole.

Järgmisena lisab server mõlemast päringust saadud vastused järjest nimekirja, mis prinditakse kasutajale välja. Igal nimekirja real on välja toodud nupp, mida vajutades minnakse loo kuulamise leheküljele, kus saab lugu kuulata. Sellele nupule vajutades saadetakse kaasa ka loo andmed selle kuulamiseks. Lisaks on ka sellele reale kirjutatud loo nimi ning selle autor, et lugusid eristada omavahel ning vajadusel mingit spetsiifilist lugu üles leida.

### 3.4.4 Rahvahankelise loo valmistamine

Nüüd tuleb valmistada põhiosa, mida me alguses seadsime eesmärgiks teha – rahvahankelise klaveriloo tegemise veebirakendus. Selleks peab tegema alternatiivse lehe, mis töötab sarnaselt üheautorilise lehega, aga seekord on mängus ainult üks noot.

Kõigepealt laetakse sisse mingi lugu, et seda üldse töödelda saaks (tühja loo puhul laetakse sisse tühi noodistik). Seejärel saab seda lugu kuulata selleks vastavale nupule vajutades (tühja loo puhul ei mängita midagi). Klahvile vajutades lisatakse seekord ainult üks noot loole lisaks. Mitmele klahvile vajutades vahetatakse lihtsalt viimane noot ära. Nupud on ka kõik samad, aga sama põhimõttega – pausi lisamise nupule vajutades lisatakse paus ning mitu korda sellele vajutades asendatakse viimane noot ikka pausiga. Noodi eemaldamise nupp eemaldab ikka viimase noodi nimekirjast, ainult seekord saab vaid uue lisatud noodi ära kustutada ning muu jääb samaks.

Salvestamise nupule vajutades tuleb uuesti väiksem salvestuse aken lahti, aga seekord on väljad veidike erinevad – kuna tegemist on rahvahankelise looga, siis loole nime pole vaja veel sisestada. Noodi lisaja nime on aga ikka tarvis, kuna teiseks tekstiväljaks on seekord annetuse suurus. Annetuse andmetüübiks on arv, mis on suurem kui 0. Salvestust saab jällegi katkestada vastavale nupule vajutades, mis salvestusakna ära peidab ja edasi töödelda laseb.

Salvestamise nupule vajutades saadetakse kõik andmed edasi serverile, mis jällegi kontrollib andmete õigsust, ning mis siis salvestab muudatused, uuendades loo andmeid. Juhul, kui lugu vahepeal uuenes, saadetakse kasutajale tagasiside loo muutuste kohta ning kasutaja võib uue töötluse lisada. Salvestuse päring ise läbi seejuures ei lähe.

Lisaks loo enda salvestamisele salvestatakse ka annetaja andmed selleks vastavasse tabelisse. Nagu eelnevalt mainitud, on annetuse saatmine spetsiifiliselt väga ebaametlik, sest selle funktsionaalsuse peaksid Valga muusikakooli töötajad spetsiifiliselt ise ära täitma, kuna pangakonto andmed ja sellega seonduv protsess jne ei ole ülikooli lõputöö jaoks kõige sobilikumad osad. Seetõttu ei ole erilisi kontrole seal tehtud ning põhjus selleks miks sisendiks on suvaline annetuse saatja nimi ning annetuse suurus.

### **3.4.5 Rahvahankeliste lugude töötlemine**

Et lugusid töödelda, peaks selleks olema ka leht, kus töödeldavaid lugusid laadida saab. Selleks on jällegi uus samalaadne leht valmis olevate lugude otsimise lehega, ainult et seekord otsitakse lugusid ainult rahvahanke lugude tabelist lugusid, kus seejuures peab loo seisundiks olema väär, ehk lugu ei ole veel valmis ja seda saab veel töödelda. Andmed kätte saades lisatakse need samasuguse ridade nimekirjana kasutaja ette, kus kasutaja saab igale reale ette määratud nupule vajutades sellele spetsiifilisele loole mingi noodi juurde lisada. Loo autoriks on igal real määratud „Crowdsource“.

Lisaks sellele on lehel ka nupp, kus saab uue tühja rahvahanke loo lisada. See on juhuks, kui mitte ühtegi pooleli olevat lugu ei eksisteeri või kui keegi tahab uuelt lehelt alustada. Uue tühja loo esitamise jaoks peab lisama sellele algelise nime, millena seda ära tunda, ning seejärel võib vajutada nupule „Lisa uus“, mis selle uue tühja loo nimekirja lisab. Loo nimi ei tohi seejuures olla tühi sõne.

### **3.4.6 Annetajate nimekiri**

Kuna annetajaid oleks ka hea kuskil nimekirja vormis näha, oleks selle jaoks ka ühte lehte vaja. Selle jaoks tegin veel ühe spetsiaalse lehe, mis töötab samas vormis kui teised lehed, millel on vaid nimekiri, ehk siis kõik lehed, kust saab lugusid otsida. Lehele minnes saadab server jälle nimekirjana kõigi annetajate nimekirja, kus on kirjas annetaja nimi ja tema poolt annetatud rahasumma.

Annetajate lehele minemiseks on nupp, mis asub pooleli olevate rahvahankeliste lugude otsimise lehe allosas. Kuna annetamise funktsioon on vaid rahvahankeliste lugude töötlemise jaoks, siis see tundus selle nupu jaoks loogiline koht.

## 4 Klaveri veebilehe kasutajaliides

Funktsionaalsus on olemas, aga et kasutajale seda võimalikult mugavaks teha, on vaja töödelda kõigi lehtede kasutajaliidest, et kõik oleks hästi nähtav, hästi arusaadav ja hästi mugavalt kasutatav, ehk siis et kasutajal tekiks võimalikult vähe frustratsioone kasutajaliidese kasutamisel.

### 4.1 Klaveri kujundus

Üks kõige tähtsamaid komponente terve projekti juures on klaveri enda kujundus, et kasutajale tunduks, et klaver on ilus, mõnus ja et seda oleks turvaline kasutada. Nagu varem mainitud, otsustasin ma klaveri moodustada ise ühe spetsiifilise klaveri järgi, mille ma leidsin ühelt veebilehelt, kuna selle kujundus meeldis mulle kõige rohkem võrreldes mõnede teiste klaveritega ning võrreldes mõne teegiga, mida oleks võinud kasutada klaveri disainimise hõlbustamiseks.

Klaveri kõik valged noodid on ülevalt nurgelise nurkadega ning alt ümarate nurkadega, mis meenutab reaalseid klavereid. Kõigil valgetel klahvidel on ka sisemine tume kuma, mis lisab klahvile kolmedimensioonilise tunde ja vaate – nagu iga klahv olekski eraldi klahv. Lisaks on kõik valged klahvid altpoolt valgemad kui ülevalt, mis lisab veel parema vaate ning julgustab kasutajal sihtima klahve altpoolt, nagu tavalisel klaveril peaks.

Klaveri kõik mustad klahvid on paigutatud täpselt nii, et igal blokil olevate valgete klahvide ülemised mustade klahvidega katmata osad oleksid võrdsete laiustega, mis annab palju reaalsema vaata pärisklaveritele, millel on sama ehitus, kuna professionaalsetel pianistidel on tihti näpud kõveramad loo kohtades, kus on vaja palju musti klahve vajutada, mistõttu on kergem valgetele klahvidele vajutada ülemisest poolest, mistõttu kui mustad klahvid oleksid vaid täpselt valgete klahvide keskel, siis ei jääks iga bloki keskmiste valgete klahvide vahele nii vähe ruumi, et ei mahu enam sõrme vahele panema. Seetõttu on mustade klahvide paigutus ametlikult korrektne ning see annab klaverile professionaalse väärtuse.

Lisaks on kõik klaveri mustad klahvi vastastikuselt valgetele klahvidele altpoolt heledamad, mis jällegi julgustab neid klahve mängima altpoolt. Kõige parem on aga klaveri mustade klahvide juures see, et selle dimensionaalsus on veel parem kui valgetel



klahvidel, mis tuleneb sellest, et mustade klahvide sees veel üks kast, mis on veidike väiksem laiuselt ning altpoolt lühem pikkuselt. See väike detail muudab kõik mustad klahvid järsku palju jälgitavamaks ning huvitavamaks.

Klahvide peal kursoriga hõljudes muutuvad valgete klahvide piirjooned palju kontrastsemaks ja tumedamaks ning mustad klahvid veidike heledamaks. Need on head indikaatorid kasutajale, et klaver peaks olema töökorras.

Klahvide peale vajutades muutuvad valged klahvid tumedamaks ning mustad klahvid heledamaks. Värvimuutus kaob 215 millisekundi pärast, mis on veidike lühem kui 230 millisekundit, mis on klaveri laulude automaatse mängimise nootide mängimise vahe. See annab täpselt piisavalt aega automaatse mängimise ajal klahvi värvi muuta tagasi normaalseks enne järgmise klahvi mängimist, kui mõlemad klahvid peaksid üks ja sama klahv olema.

Kõiki neid kujunduselemente arvesse võttes on see üks parimaid veebisiseseid klavereid, mida ma näinud olen, millepärast ma seda just kasutasingi. Klaveri visuaali võib taas näha enne mainitud joonisel (Joonis 1).

## **4.2 Pealehe kujundus**

Pealehel on oluline, et kasutaja saaks kõik kõige vajalikuma informatsiooni võimalikult kiiresti, efektiivselt ja vaevatult kätte, kuid seejuures lubab kasutajal ka ringi liikuda ning uurida ja avastada. Selleks kõige parem on menüüriba, mille tavaline asukoht on kas vasakul servas või üleval ääres. Kuna klaver on laiuselt väga lai, siis küljele eriti seda mahutada ei anna, mis tulemusena on menüüriba paigutatud üles (Joonis 5).

Menüüriba on värvitud siniseks, kuna sinine on teadatuntud kui värv, mis rahustab inimesi kõige rohkem. Sinine värv rahustab kasutaja maha ning annab teada, et lehekülg on turvaline ja rahulik. Menüüsse on paigutatud kõige üles vasakusse nurka nupp, millega saab kohe tagasi pealehele. Kasutajatele on alati tähtis, et saaks alati ja igal momendil tagasi turvalisele pealehele, kust avastamist alustati.

Menüüreal on veel kolm nuppu, mis viivad vastavatele lehekülgedele, kus saab avastusi teha. Kuna lehekülg on tehtud mõttega, et lugusid valmistada, siis kõigepealt peaksid tulema nupud, mis viivad lugude valmistamise lehekülgedele. Esimeseks on nupp, mis

viib sind lehele, kus saab ise üksinda mingit lugu valmistada. See tuleb esimesena, kuna see on turvalisem ning kindlam, kuna kõik on kasutaja enda poolne ning tõenäosus et ta võib teisi segada ei eksisteeri. Teiseks nupuks jääb rahvahankeliste lugude valmistamise leht, mis viib sind lehele, kus leiab nimekirja lugudest, mida saab töödelda. Viimaseks nupuks on valmis olevate laulude otsing.

Nuppudel puudub kast, et menüüd kergemini hõlmatavamaks teha. Nuppude peal olevad tekstid on tehtud halliks, et nende peal kursoriga hõljudes need valgeks muutuksid, kuna see annab parema tunde kui peale hõljudes valget halliks teha. Nupu peale vajutades hakkab tekst kumama, et viidata nupu töötamisele.



Joonis 5: Pealeht

### 4.3 Üheautorilise lehe kujundus

Vajutades teisele peamenüü nupule viiakse sind loo valmistamise lehele, kus saad ühe terve loo ise valmis teha. Siin on oluline, et lehele tekivad lisaks nupud, mis aitavad loo tegemist, ning suurepärase käitlemine laulu automaatses mängimises (Joonis 6).

Esimeseks nupuks on loo kuulamise nupp, mis on suur ning erilise teistest erineva värviga, et kasutaja seda esimesena näeks. Uue loo tegemisel on tähtsaks komponendiks katsetamine, mistõttu on loo mängimise nupp kõige tihemini kasutatav ja peaks olema kõige suurem ja otse lehe keskel.

Järgmisel real on loo katkestamise nupp, viimase noodi eemaldamise nupp, pausinoodi lisamise nupp ja salvestamise nupp. Nuppude värvid on siinosas väga olulised, sest kasutaja tihti pigem defineerib nuppe värvi järgi, mitte sellel olevate sõnade järgi,

mistõttu on katkestamise nupp punane, kuna see defineerib millegi katkestamist või ebaõnnestumist, ja salvestamise nupp roheline, kuna see defineerib õnnestumist ja millegi positiivse juhtumist. Samuti on mõlemad nupud äärtes, kuna neid kasutatakse kõige vähem. Lisaks on katkestamise nupp just vasakul ja salvestamine just paremal kuna suurem hulk inimesi on loogiliselt üles ehitatud ideega, et edusammud on paremal pool ja loobumised vasakul pool. Pausinoodi lisamise ja viimase noodi eemaldamise nupud on vastavalt menüüribale sinised ning vastavalt kasutussageduse reeglile lehe keskel.

Nuppude all on nimekiri kõikidest eelnevalt mängitud nootidest järjekorras, kus kõige ülemine on kõige hilisemalt lisatud noot, et kasutajale nimekirja järgimine võimalikult mugavaks teha. Kõik read on eraldatud gradientse joonega, et nootide nimesid võimalikult kerge näha oleks.

Kasutajale mugavuse lisamiseks ning liidese ahvatlevamaks tegemiseks on loo automaatse mängimise ajal klaveri klaviatuuril toimumas sama klahvide värvuse muutumine kui käsitsi mängimise ajal, et anda kasutajale parem tagasiside sellest, mis klahvi just äsja mängiti. Lisaks sellele mängib ka kaasa nimekiri, kus automaatse mängimise ajal muutub mängitava noodi tagataust nimekirjas halliks, et näidata, kus kohas laulu mängimise punkt sellel ajahetkel asub.



Joonis 6: Üheautoriline leht

Viimaseks on sellel lehel salvestamise nupp, millele vajutades avaneb väiksem salvestamise aken (Joonis 7). Akna ilmumisel selle taga olev aken kaetakse valge poolläbipaistva värviga, et tuua äsja ilmunud aknale rohkem tähelepanu. Aken on huvitavamaks tegemise nimel tehtud helekollaseks ning seda ümbritseb paks hall piirjoon, mis seda taga olevast aknast eraldaks. Akna sees on kaks tekstivälja, mille kohal on kirjas, mida sinna peaks kirjutama, ning akna all servas on kaks nuppu – katkestamise nupp ja salvestamise nupp. Nupud on jällegi vastavate värvidega kodeeritud ning vastavas järjekorras. Katkestamisel salvestuse aken peidetakse ning kasutaja saab töötlemise juurde tagasi pöörduda. Salvestamisel leht värskendatakse.

Laulu nimi:

Sinu või laulu autor:

Katkesta Salvesta

C3  
G2  
E3

Joonis 7: Salvestusaken

#### 4.4 Nimekirjaliste lehtede kujundused

Peamenüü kahe viimase nupu peale vajutades viiakse kasutaja lehekülgedele, kus talle esitatakse nimekirjad, mis koosnevad ridadest ning kus iga rea alguses on nupp, millele vajutada saab (Joonis 8). See on hea võimalus lasta kasutajal avastada lugusid, nii et liialt juhiseid ei tohiks anda.

Nimekirja alguses on päis, mille peale on kirjutatud, mida mingi veerg tähendab. Päis on värvitud jällegi siniseks, et kasutaja rahulik oleks ega liiga ähmi täis ei läheks kõiki leid ridu vaadates. Nimekiri ise on valgel tausta, mis on eraldatud samade gradientsete joontega. Rea alguses on nupp, millele kasutaja saab peale vajutades loole ligi. Nupp on ikka roheline aga nupu asukoht on seekord vasakul, kuna seekord ei ole asukoht kõige

tähtsam seetõttu, et katkestamise nupp puudub ja seetõttu, et nimekiri tegutseb nagu menüüriba, kus kõik vajalikud nupud on vasakule joondud. Nii on lihtsalt kasutajale mugavam, kui kõik nupud on vasakul ääres sellisel juhul. Ning lehe keskel on kõigi lugude nimed ja paremal lugude autorid. Järjekorra põhjenduseks on inimeste harjumus sellest, et loo nimi kirjutatakse tavaliselt enne loo autori nime.

Klaver Valmista ise üks laul Aita valmistada rahvahankeline laul Kuula valmis olevaid laule		
	Loo nimi	Loo autor
Vaata	Mario3	Crowdsource
Vaata	Mario	Vaza
Vaata	Sinine Vagun	Vaza
Vaata	Luigi's Mansion	Vaza

Joonis 8: Lugude otsimine

Rahvahankelisi laule otsides on lehekülje all veel paar elementi, millega saab kasutaja midagi teha (Joonis 9). Nendeks on uue tühja rahvahankelise loo loomine koos ajutise nimega, mida antakse kasutajale teada läbi kohatäite teksti tekstiväljas. Jätkuvalt esineb selles osas roheline nupp, kuid selle all on navigeerimisnupp, mis on värvitud siniseks, et eristuda teisest nupust. See nupp viib annetajate nimekirja lehele, millel on sama nimekirja disain, aga seal lihtsalt puudub vaatamise nupp, kuna see ei ole seal kontekstis vajalik (Joonis 10).

Klaver Valmista ise üks laul Aita valmistada rahvahankeline laul Kuula valmis olevaid laule		
	Loo nimi	Loo autor
Vaata	testsong	Crowdsource
Vaata	hh	Crowdsource

Loo ajutine nimi

Lisa uus

Annetajad

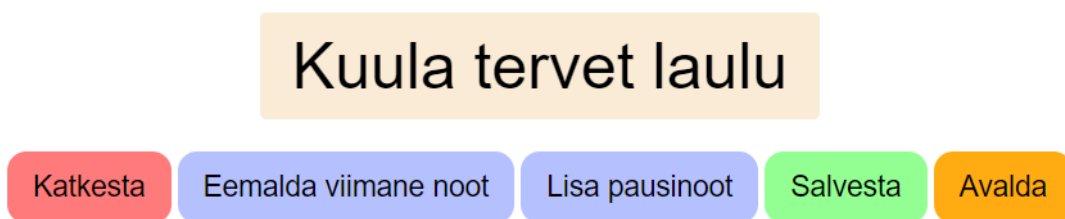
Joonis 9: Rahvahanke nimekiri

Klaver		Valmista ise üks laul	Aita valmistada rahvahankeline laul	Kuula valmis olevaid laule
Annetaja nimi		Annetuse kogus		
Michael Einstein		1.00		
Manlyman		50.00		
Fluffy Menugger		5.00		
hh		1.00		

Joonis 10: Annetajate nimekiri

## 4.5 Rahvahanke lehe kujundus

Rahvahankelise lehe kujundus on väga sarnane üheautorilise lehe kujundusega, aga siin on juures veel avaldamise nupp, mis on värvitud oranžiks (Joonis 11). Selline spetsiifiline värv on antud just seetõttu, et seda eraldada tavalisest salvestamisest ning et anda kasutajale teada, et see nupp on erilise tähtsusega.



Joonis 11: Rahvahanke lehe avalduse nupp

Salvestamise nupule vajutades ilmub seekord sarnane salvestuse aken, aga seekord on tekstiväljad veidike erinevad (Joonis 12). Teiseks tekstiväljaks on seekord numbriline väärtus, mille kohal on antud tekstis kirjas valuuta ning millele on antud algväärtuseks 1.00, et kasutajale näidata, millises formaadis peaks summa esitama ning et on võimalik esitada ka sentide arv. Muu on siin aknas sama.

Avaldamise aknas on aga ainult üks tekstiväli, mille kohal on rohkem teksti kui tavaliselt, mis peaks kasutajalt veidike rohkem tähelepanu püüdma võrreldes ülejäänud kasutajaliidesega ning seetõttu ka veidike ettevaatlikumaks muutma, et ta midagi valesti ei teeks (Joonis 13). Samamoodi on ka avaldamise nupp siin aknas muudetud oranžiks, et järjepidevust pidada.

Teie nimi:

Annetuse suurus (€)

Katkesta Salvesta

Joonis 12: Rahvahanke salvestusaken

Olete kindel, et tahate avaldada?

Pärast avaldamist laulu enam muuta ei saa.

Laulu nimi:

Katkesta Avalda

Joonis 13: Rahvahanke avaldusaken

## 5 Kokkuvõte

Rahvahankelisi projekte on viimase kümne aasta jooksul tehtud tuhandeid, kus seejuures on mõned neist väga populaarsetel ja tihti kasutatavatel veebilehtedel. Näiteks kasutab Vikipeedia rahvahankelist meetodit kõige kohta informatsiooni saamiseks, Google'i tõlge tõlkimise parandamiseks läbi Google'i kogukonna abi, IARPA geopoliitiliste sündmuste ennustamiseks läbi rahva arvamuste, reCAPTCHA programmirobotite takistuseks veebilehtedele sisenemiseks läbi inimese pool sisestatud piltide, Folding@Home haiguste uurimiseks läbi rahva arvuti protsessori ressursside ning Akinator paremini tegelaste arvamiseks läbi inimeste abi lisades veel tegelasi ning küsides paremaid küsimusi. Lisaks suutis Brendon Ferris luua klaveri veebilehe, kus 67'000 inimest valmistasid rahvahankel üllatavalt hästi kõlava muusikalise pala.

Projekti põhieesmärgiks oli programmeerida multimeedia klaveri veebirakendus, mis võimaldab rahval ühise jõuga valmistada klaverile mingi lugu – rahvahankeline lugu, ning see rakendus teha võimalikult mugavaks kasutamiseks.

Esimeseks programmeerisin ma rakenduse kõige tähtsama komponendi – klaveri. Selle jaoks kasutasin ma näidist, mille leidsin ühelt leheküljelt. Klaveri tähtsamateks komponentideks on mugav kasutatavus, kergesti ära tuntavus hästi ligi tõmmatavus ning heade indikaatorite olemasolu klaveri töövõimekusele. Teiseks lisasin ma klaverile helid sisse klahvide vajutamisel, mille juures on oluline, et hääled töötaksid kõigil brauseritel ning kostuksid tuttavalt.

Kolmandaks programmeerisin ma andmebaasi, mida saaks kasutada lugude salvestamiseks. Selles osas on oluline, et kõik vajalikud väljad oleksid tabelites olemas. Neljandaks programmeerisin ma funktsionaalsed implementatsioonid, mis teevad veebilehe töötavaks ning seob veebilehe andmebaasiga. Nende hulgas on põhilisteks osadeks lugude nimekirjade näitamine, lugude töötlemine ja salvestamine, üheautorilise loo valmistamine ja salvestamine, rahvahankelise loo avaldamine ning annetatjate nimekirja näitamine.

Viimasena oli kavas kasutajaliidese korrigeerimine ja töötlemine, et see kasutajale võimalikult mugav ja ahvatlev välja näeks. Selle jaoks kasutasin teadatuntud printsiipe – menüüriba üleval servas, sinised värvused kasutajas rahu säilitamiseks, punased värvid



katkestamise nuppudele, rohelised värvid õnnestumist osutatavatele nuppudele, nuppude asukohad, nuppude interaktiivsus kursoriga, nimekirjade lihtsus ja arusaadavus ning muud väiksemad nüansid.

Algselt oli plaanis ka lisada mobiilse liidese tugi kasutajatele, kas tahavad seda mobiilis kasutada, aga klaveri 84 klahvi kogulaiuse tõttu muutus liides väga ebamugavaks ja naeruväärselt kahvatuks, mille tõttu otsustasin ma pigem fokuseerida arvutiversioonile ning et see töötaks kõigil brauseritel. Mobiiliversiooni annaks sellele programmeerida, kuid siis peaks klaveri tegema ainult vaadeldavaks ning vormistama hoopis mingi teistsuguse liidese selle jaoks, aga selle jaoks ei jäänud mul piisavalt aega.

Mobiiliversioon vahele jättes arvan ma, et projekt oli suhteliselt edukas. Baasfunktsionaalsus on kõik töökorras, kasutajaliides on suhteliselt mugav, müratu ning lihtne, serveri ja andmebaasi vaheline kooskõla on olemas ning lugude tegemine on võimalik. Kasutajaliidest annaks arvatavasti veelgi edasi arendada, et seda veel professionaalsemaks muuta, aga praeguses olukorras on see piisav. Töötav demoversioon on kättesaadav ning kasutatav (vt Lisa 2). Nüüd peaks järgi olema selle saatmine Valga muusikakoolile, kus tehakse viimased viimistlused, pangakontoga ühendused ning pärisveebilehele üles panemine, millega saaksid nad annetustega aja jooksul oma tahetavat klaverit osta.

## Kasutatud kirjandus

- [1] M. L. Kelly, Interviewee, *Intelligence Community Looking At Crowdsourcing For Predicting Geopolitical Events*. [Intervjuu]. 26 January 2018.
- [2] Z. Crockett, „67,000 people composed this melody on the internet — and it sounds pretty good,“ [Võrgumaterjal]. Available: <https://www.vox.com/2016/9/20/12942866/song-crowdsourced-on-internet>. [Kasutatud May 2018].
- [3] Danigb, „Quick soundfont loader and player for browser,“ [Võrgumaterjal]. Available: <https://github.com/danigb/soundfont-player>. [Kasutatud May 2018].
- [4] Apronus, „Virtual Piano Onlilne,“ [Võrgumaterjal]. Available: <https://www.apronus.com/music/flashpiano.htm>. [Kasutatud May 2018].

## Lisa 1 – Kõik klaveri klahvide noodi nimed

Järgnevalt on kõik nimed, mis on noodiroadus defineeritud:

"A1", "B1", "C1", "D1", "E1", "F1", "G1", "A1u", "C1u", "D1u", "F1u", "G1u", "A2",  
"B2", "C2", "D2", "E2", "F2", "G2", "A2u", "C2u", "D2u", "F2u", "G2u", "A3", "B3",  
"C3", "D3", "E3", "F3", "G3", "A3u", "C3u", "D3u", "F3u", "G3u", "A4", "B4", "C4",  
"D4", "E4", "F4", "G4", "A4u", "C4u", "D4u", "F4u", "G4u", "A5", "B5", "C5", "D5",  
"E5", "F5", "G5", "A5u", "C5u", "D5u", "F5u", "G5u", "A6", "B6", "C6", "D6", "E6",  
"F6", "G6", "A6u", "C6u", "D6u", "F6u", "G6u", "A7", "B7", "C7", "D7", "E7", "F7",  
"G7", "A7u", "C7u", "D7u", "F7u", "G7u", "-"

## **Lisa 2 – Kättesaadav ja töötav demo**

<http://dijkstra.cs.ttu.ee/~heouna/piano/>