

Tallinna Tehnikaülikool  
Infotehnoloogia teaduskond  
Arvutiteaduse instituut  
Võrgutarkvara õppetool

Veebiliides andmebaasiteegile WhiteDB

Bakalaureusetöö

Üliõpilane:	Mai-Liis Hütt
Üliõpilaskood:	112239 IAPB
Juhendaja:	Tanel Tammet

Tallinn  
2014

## **Autorideklaratsioon**

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

.....  
*(kuupäev)*

.....  
*(allkiri)*

## **Annotatsioon**

Projekti eesmärgiks on luua veebiliides andmebaasiteegi WhiteDB andmebaaside haldamiseks. Veebiliidese abil peab olema võimalik teostada andmebaasis CRUD operatsioone, näha andmebaasi kirjeid ning nende omavahelisi seoseid.

CRUD operatsioonide teostamiseks on vaja luua funktsionaalsus, mille teostamiseks kasutatakse JavaScripti ja jQuery't.

Andmebaasi kirjete mugavaks kujutamiseks võetakse kasutusele Bootstrap teek, mis aitab lahendada kujunduselementide kuvamise erisused veebilehitsejates, ja Handlebars, mis võimaldab valmis teha HTML mallid, et vähendada JavaScripti koodi kirjutatava HTMLi hulka ning muuta koodi selle läbi loetavamaks.

Kirjete vaheliste seoste kujutamiseks võetakse kasutusele JavaScripti teek D3, mis andis kõige paremad võimalused seoste kuvamiseks.

Tulemusena sai loodud veebiliides, millega saavad kasutajad hallata oma WhiteDB andmebaase, teha nendes CRUD operatsioone ning näha ka kirjete vahelisi seoseid.

Lisaks tehakse ülevaade andmebaasi keeltest SQL ja NoSQL ning kõige enam kasutatavatest andmebaasiteekide administreerimisliidestest.

Räägitakse andmebaasiteegist WhiteDB ning selle tööriistast Dserve.

Kirjeldatakse liidese arenduse juures tekkinud probleemide lahendusi ja milliseid tehnoloogiaid kasutati ning tehakse ülevaade kuidas liidest kasutada.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 27 leheküljel, 8 peatükki, 20 joonist.

## **Abstract**

Goal of this project is to make a graphic user interface for database library WhiteDB where users could manage their WhiteDB databases. Users should be able to perform CRUD operations with database entries and see connections between entries as a graph.

To make the functionality necessary for performing CRUD operations JavaScript and jQuery are used.

For displaying database records more conveniently author uses Bootstrap library, WHICH removes differences of displaying desing elements between browsers, and Handlebars library which makes JavaScript code more readable byreducing the amount of HTML written inside it.

For displaying connections between database entries author is using JavaScript library, which gives best opportunities for showing these connections.

The result is a graphic user interface where users could manage their databases and see connections between database entries.

Author makes and overview of SQL and NoSQL and graphic user interface of most commonly used databases.

Author will talk about database library WhiteDB and about its tool Dserve.

Lastly author will describe problems which came up during the development, what kind of technologies were used and how graphic user interface is used.

The thesis is in estonian and contains 27 pages of text, 8 chapters, 20 figures.

## Lühendite ja mõistete sõnastik

AJAX – Asynchronous JavaScript and XML. AJAX-iga saab teha lehe või selle osa muutmiseks päringuid, ilma, et peaks lehte uuesti laadima.

JSON – JavaScript Object Notation. Kergekaaluline andmevahetus formaat, mis on inimestele lihtne kirjutada ja lugeda.

HTML – HyperTextMarkup Language. Peamine keel, mille abil luuakse veebilehti.

SQL – Structured Query Language on keel relatsiooniliste andmebaasidega suhtlemiseks.

String – Eesti keeles sõne, on tervikuna käsitletav elemendi- või märgijada.

Integer – Eesti keeles täisarv, on täisarvuline muutuja.

Boolean – Eesti keeles loogikamuutuja, on kahe võimaliku väärtusega andmetüüp.

Schema – Eesti keeles skeem, on andmebaasi struktuuri kirjeldus.

CGI – Veebiserveri ja CGI programmi vahelise andmevahetuse spetsifikatsioon, mis määrab ära selle, kuidas veebiserver edastab veebikasutaja nõude veebiserveris jooksvale CGI programmile töötlemiseks ning võtab vastu töötlemistulemuse, mis saadetakse tagasi veebikasutajale.

CSV – failivorming, kus iga kirje on üks rida ja iga kirje väljad on üksteisest komadega eraldatud

REST – tarkvara arhitektuuri stiil

Apache server – veebiserver, mida kasutatakse peamiselt staatiliste ja dünaamiliste veebilehtede haldamiseks veebis

CRUD – lühend inglisekeelsest sõnadest *Create, Read, Update, Delete*

## Jooniste nimekiri

Joonis 1: SQL andmebaasi näide.....	10
Joonis 2: JSON struktuuri näide.....	11
Joonis 3: Graafi andmebaasi struktuuri näide.....	11
Joonis 4: Lihtne võti-väärtus paaride näide.....	12
Joonis 5: PhpPgAdmin.....	13
Joonis 6: PhpMyAdmin.....	14
Joonis 7: Neo4j veebiliides.....	14
Joonis 8: MongoDB veebiliides PhpMoAdmin.....	15
Joonis 9: Kõikide andmete kuvamise lehekülg.....	17
Joonis 10: Andmete filtreerimise aken.....	17
Joonis 11: Ühe kirje kuvamise lehekülg.....	18
Joonis 12: Rea muutmise lehekülg.....	18
Joonis 13: Uue kirje lisamise lehekülg.....	18
Joonis 14: Andmebaasi graafina kujutamine.....	19
Joonis 15: Konfiguratsioonifaili vaikimisi väärtused.....	20
Joonis 16: Veergude kuvamine pikkade väljade puhul.....	21
Joonis 17: Pikkade väljade kuvamine kirje lehel.....	21
Joonis 18: Näide kuidas tipud ekraanilt välja lähevad ja millised olid tippude vahelised jooned enne värvi muutmist.....	22
Joonis 19: Näide, kui tippe, millele viidata, on 10.....	23
Joonis 20: jQuery ja JavaScripti DOM manipulatsiooni näide.....	24

## Sisukord

1.Sissejuhatus.....	8
1.1Metoodika.....	8
1.2Ülevaade tööst.....	9
2.SQL ja NoSQL.....	10
2.1SQL.....	10
2.2NoSQL.....	10
2.2.1 Dokumendi-andmebaasid.....	10
2.2.2 Graafi-andmebaasid.....	11
2.2.3 Key-value andmebaasid.....	11
3.Ülevaade veebipõhistest administreerimisliidestest.....	13
3.1SQL andmebaasid.....	13
3.1.1 PhpPgAdmin.....	13
3.1.2 PhpMyAdmin.....	13
3.2Graafi- ja NoSQL-andmebaasid.....	14
3.2.1 Neo4j.....	14
3.2.2 PhpMoAdmin.....	15
4.WhiteDB.....	16
4.1 WhiteDB.....	16
4.2Dserve.....	16
5.Ülevaade rakendusest.....	17
6.Liidese lahendused.....	20
7.Tehnoloogia.....	24
8.Kokkuvõte.....	25
Kasutatud kirjandus.....	26

# 1. Sissejuhatus

WhiteDB on põhimõlul tegutsev NoSQL andmebaasiteek. Selle andmebaasi andmetega tegutsemiseks on kaks võimalust. Esimene võimalus on kasutada käsurea kaudu WhiteDB käsku *wgdb*. Teine variant on kasutada käsurea või veebilehitseja kaudu tööriista *dserve*. Mõlemad võimalused tagastavad andmeid json kujul. Käsurea kaudu andmete küsimine võib olla tülikas, eriti juhtudel kus tegu on pika päringuga. Samuti, kuigi json on enamike jaoks suhteliselt loetav, ei ole pikkade vastuste puhul mugav lugeda ja üksteisest kirjeid eristada. Selleks, et kasutaja ei peaks ise päringuid koostama ja saaks kirjeid ning nendes olevaid välju üksteisest kergemalt eristada oli vaja luua veebiliides.

Esimene eesmärk on luua funktsionaalsus ning vaated, mille abil kasutajad saaksid teostada CRUD operatsioone. See on vajalik, et kasutajad saaksid oma andmebaasi kirjetega tegutseda.

Iga andmebaasi kirje väli võib olla viide teisele andmebaasi kirjele. Sellest kujunes välja kolmas eesmärk, milleks on andmebaasi kirjete vaheliste seoste kujutamine graafina.

Liidese kood on avalikult saadaval GitHubis aadressil <https://github.com/priitj/whitedb/tree/master/Admin> ning on võimalik koos WhiteDB-ga alla laadida.

## 1.1 Metoodika.

CRUD operatsioonide tegemiseks kasutatakse JavaScripti ja AJAX päringuid, mille abil küsitakse andmebaasist infot ning lisatakse sinna uusi andmeid. Andmebaasi kirjete omavaheliste suhete kujutamiseks kasutatakse JavaScripti teeki D3, mis võimaldab joonistada graafe. Selleks, et teha kasutajatele andmebaasi haldamist mugavamaks ning andmete kuvamine oleks erinevate veebilehitsejate kasutamisel välimustelt sama võetakse kasutusele Bootstrap teek.



## 1.2 Ülevaade tööst

Kahes esimeses peatükis tehakse väike ülevaade SQL ja NoSQL andmebaaside struktuurist ning tehakse ülevaade mõningatest SQL ja NoSQL administreerimisliidestest ning andmebaasiteegist WhiteDB.

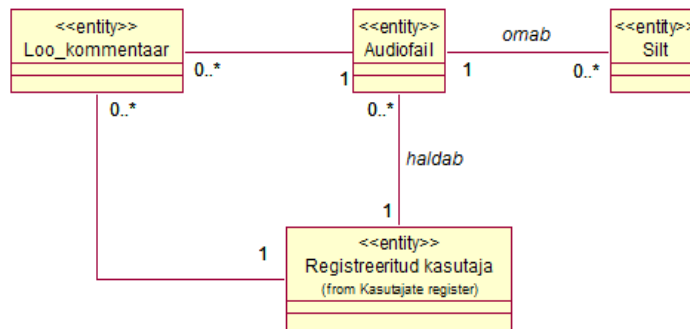
Järgmisena tehakse ülevaade rakendusest, kus kirjeldatakse mida ja kuidas on võimalik andmebaasi haldamiseks vajalikke tegevusi teha.

Viimasena tehakse ülevaade suurematest veebiliidese juures ette tulnud probleemidest, nende lahendusest, samuti tehnoloogiatest, mida veebiliidese tegemisel kasutati.

## 2. SQL ja NoSQL

### 2.1 SQL

SQL ehk Structured Query Language on keel relatsiooniliste andmebaasidega suhtlemiseks. Esimene SQL toode lasti välja 1979 aastal. [7] SQL andmebaasi struktuur koosneb tabelitest, mis omakorda koosnevad ridadest ja veergudest. Igal andmebaasil on oma skeem, kus on kirjeldatud tabelid ning nende veerud. Igal veerul on kindlaks määratud väärtused, millest kõige põhilisemad on sõne, number, kuupäev ja loogikaväärtus. [2]



Joonis 1: SQL andmebaasi näide

### 2.2 NoSQL

NoSQL ehk „not only SQL“ hakati kasutama selleks, et andmebaasid oleksid lihtsamini skaleeritavad ning jõudlus oleks suurem.[1] NoSQL andmebaaside juures räägitakse mõistest CAP, mis on lühend sõnadest *Consistency* – kõik sõlmed näevad samu andmeid, *Availability* – iga päringuga saadakse vastuse kas päring õnnestus või mitte, *Partition tolerance* – süsteem toimib ka siis kui mõni sõnum on kaotsi läinud. Eric Brewer'i „CAP teoreem“, mis väidab, et hajussüsteemides ei saa samaaegselt säilitada neid kolme CAP'i osa, vaid tuleb valida kaks.[8] NoSQL andmebaase on kolme tüüpi: dokumendi-, graafi- ja *key-value* andmebaasid.[1]

#### 2.2.1 Dokumendi-andmebaasid

Dokumendi-andmebaasid salvestavad infot failides, kus kasutatakse JSON formaati. Failid sisaldavad vähemalt ühte välja ja iga välja väärtuseks on sõne, number, loogikaväärtus või massiiv. Kirje ja temaga seotud andmed on kõik ühes dokumendis, mis teeb andmete

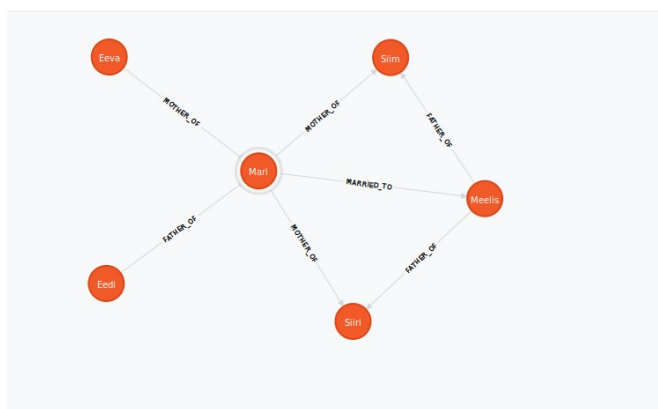
kättesaamise lihtsamaks ja vähendab või kaotab üldse vajaduse keeruliste transaktsioonide ja mitme tabeliga seotud päringute tegemiseks. Skeem on dokumendi-andmebaasis dünaamiline ning iga fail võib sisaldada erinevaid välju.[1] Enimlevinumad neist on MongoDB [3] ja CouchDB [4]

```
{
  "customer": {
    "first_name": "Mary",
    "last_name": "Smith",
    "birth_date": "17.12.1970",
    "children": [
      {"name": "Jane", "birth_date": "18.08.1992"},
      {"name": "John", "birth_date": "15.06.1995"}
    ]
  }
}
```

Joonis 2: JSON struktuuri näide

## 2.2.2 Graafi-andmebaasid

Graafi-andmebaasid kasutavad andmete esitamiseks omadusi ja graafe, mis koosnevad tippudest ning servadest. Selliseid andmebaase kasutakse juhtudel kui andmetevahelised sidemed on andmebaasis tähtsad.[1] Enimlevinumad neist on Neo4j [13] ja HyperGraphDB [5].



Joonis 3: Graafi andmebaasi struktuuri näide

## 2.2.3 Key-value andmebaasid

Key-value tüüpi andmebaasid on NoSQL andmebaaside baastüübid. Nendes andmebaasides hoitakse andmeid võti(*key*)-väärtus(*value*) paarides. Key-value andmebaasides pole erinevaid andmetüüpe nagu näiteks numbrilised või loogikamuutujad, vaid enamasti

salvestatakse andmed sõnedena. Väärtusi on võimalik näha ainult võtme järgi küsides. Kuna key-value andmebaasidel ei ole mingit kindlat andmebaasi skeemi, siis on neid hea kasutada juhtudel kus andmed on struktureerimata.[1] Enimlevinum neist on Redis[6].

Key	Value
Name	Mari Mets
Fruit	Orange
Vehicle	Car

*Joonis 4: Lihtne võti-väärtus paaride näide*

### 3. Ülevaade veebipõhistest administreerimisliidestest

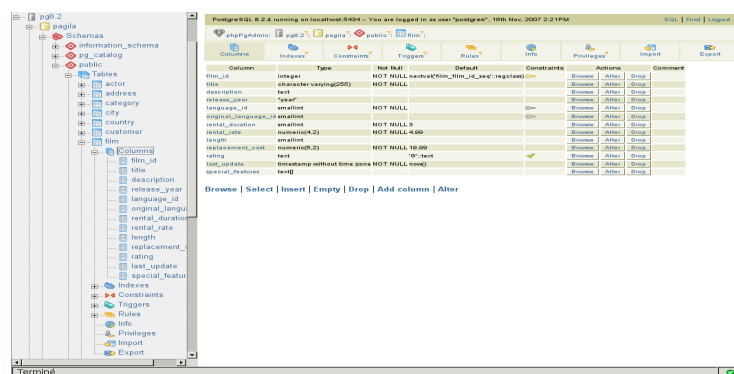
#### 3.1 SQL andmebaasid

Selles peatükis vaadeldakse kahe enimlevinuma andmebaasiteegi, PostgreSQL ja MySQL, liideseid. Need liideseid valiti selle tõttu, et mõlemal on olemas näidis andmebaasid, millega saab liidest kasutada ilma seda alla laadimiseta.

Mõlemas liideseis saab vaadata ja hallata mitut andmebaasi korraga. Saab teha CRUD operatsioone ning kirjutada SQL lauseid andmete kättesaamiseks või muutmiseks. Mõlemad liideseid võimaldavad andmebaasi sisu salvestada faili või andmeid failist andmebaasi importida.

##### 3.1.1 PhpPgAdmin

PhpPgAdmin on vabavaraline veebist haldamise liides andmebaasiteegile PostgreSQL. PhpPgAdmin võimaldab hallata mitut andmebaasi, vaadata ja muuta neis olevate skeemide ja andmebaaside sisu. Lisaks SQL lausetega andmete otsimisele on võimalik otsida mingi kindla nimega andmebaasi objekte milleks on näiteks funktsioonid või tabeli veerud. [9]

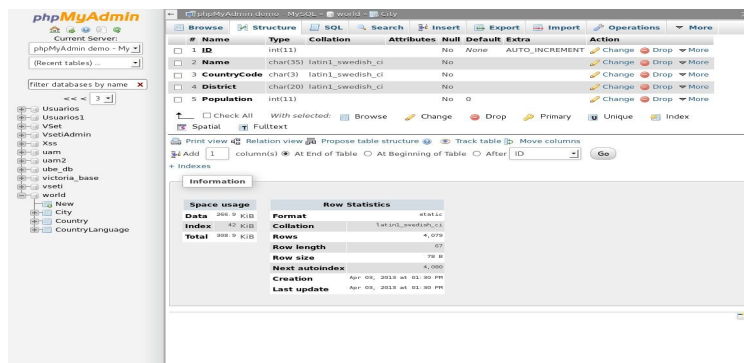


Joonis 5: PhpPgAdmin

##### 3.1.2 PhpMyAdmin

PhpMyAdmin on vabavaraline veebist haldamise liides andmebaasiteegile MySQL. PhpMyAdmin võimaldab hallata mitut andmebaasi, vaadata ja muuta neis olevate skeemide ja andmebaaside sisu. PhpMyAdmin võimaldab andmebaasi sisu ka SQL lausetega pärida ning

lisaks veel ka otsida andmebaasi objekte. PhpMyAdmin abil saab näha, kuidas tabelid on välisvõtmete kaudu üksteisega seotud. [10]



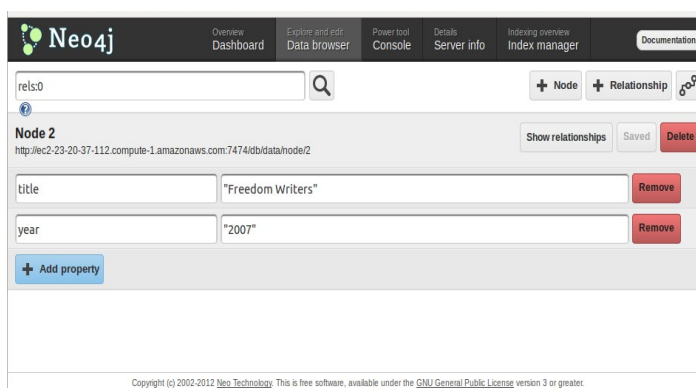
Joonis 6: PhpMyAdmin

## 3.2 Graafi- ja NoSQL-andmebaasid

Selles peatükis vaatan kahe populaarse andmebaasiteegi, Neo4j ja MongoDB, liideseid. Liidestega on võimalik hallata korraga ühte andmebaasi. Mõlemas liideses on võimalik teha CRUD operatsioone ning andmeid sorteerida.

### 3.2.1 Neo4j

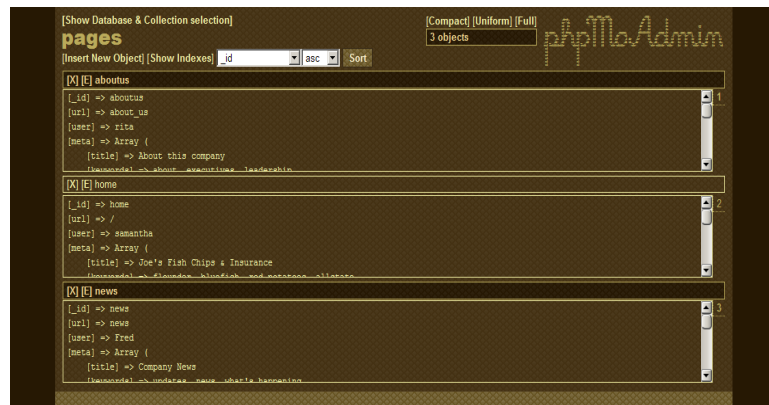
Neo4j veebiliides tuleb andmebaasiteeki alla laadimisel kaasa. Kuna Neo4j on graafi-andmebaas, siis saab andmeid ja nendevahelisi seoseid vaadata ka graafi kujul. Graafi-andmebaasi tüüpilisemaid omadusi on andmebaasi objektide omavahelised seosed ning ka nende seostega on võimalik sooritada CRUD operatsioone. [17]



Joonis 7: Neo4j veebiliides

### 3.2.2 PhpMoAdmin

PhpMoAdmin on vabavaraline veebist haldamise liides MongoDB andmebaasiteegile, mis tuleb eraldi alla laadida ja installida. MongoDB võimaldab andmebaasi JSON formaadis eksportida ja importida. Andmeid saab otsida erinevaid viise pidi, näiteks mingi kindla teksti, regulaaravaldiste või JSON järgi. Samuti saab vaadata andmebaasi statistikat näiteks veateadete ja mälu kohta. [15]



Joonis 8: MongoDB veebiliides PhpMoAdmin

## 4. WhiteDB

Selles peatükis tehakse väike ülevaade andmebaasist WhiteDB, mille veebiliides on selle töö eesmärgiks. Samuti räägitakse Dserve'st, mis on vajalik andmebaasi päringute jaoks.

### 4.1 WhiteDB

WhiteDB on põhimõluses tegutsev noSQL andmebaasiteek kirjutatud programmeerimiskeeles C. Andmed loetakse otse jagatud mälust ning kirjutatakse sinna. Igal kirjel on vähemalt üks väli, mille väärtus võib olla numbriline, sõne või viide teisele elemendile. Andmebaaside nimedeks pannakse numbrilised väärtused, vaikimisi nimi on 1000.[11]

WhiteDB sarnaneb dokumendi-andmebaasile selle poolest, et andmeid saab salvestada JSON faili. Graafi-andmebaasile sarnaneb seetõttu, et andmebaasi kirjel võib olla väli, mis on viide teisele andmebaasi kirjele ning tänu sellele on andmebaasi hea kujutada graafina.[11]

### 4.2 Dserve

Dserve on vajalik andmete pärimiseks WhiteDB andmebaasist. Dserve tõlgib REST päringud andmebaasi päringuteks, mis lihtsustab WhiteDB kasutamist veebiserveris ning teeb päringud kasutajale lihtsamaks ja arusaadavamaks. Dserve tuleb WhiteDB'ga kaasa kuid tuleb eraldi kompileerida ning saadud fail kopeerida Apache serveri *cgi-bin* kausta. Juhul kui kasutajal on *cgi-bin* kaust Apache veebiserveris seadistatud pole Dserve kasutamiseks rohkem seadistusi vaja. [11]

Dserve võimaldab kasutajatel teha kõiki CRUD päringuid. Lisaks neile saab andmebaasi kirjeid ka sorteerida. Päringuid on võimalik teha kas käsurealt, näiteks *dserve 'op=search&from=2&count=3'*, või veebilehitsejast kirjutades aadressiks näiteks *localhost/cgi-bin/dserve?op=search&from=0&count=5*. [11]

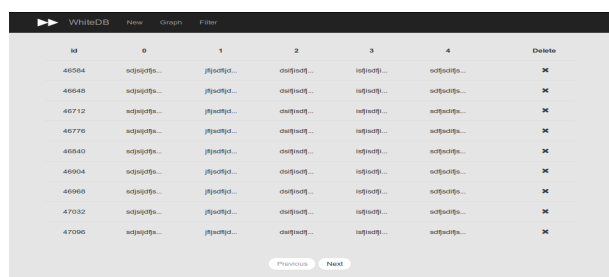


## 5. Ülevaade rakendusest

Selles peatükis tehakse ülevaade rakendusest. Näidatakse kuidas saab teha CRUD operatsioone, ning kus on näha kirjete vahelised seosed.

Enne andmebaasi kasutamist tuleb kasutajatel vajadusel muuta konfiguratsioonifailis *conf.json* olevat andmebaasi nime ja dserve asukohta ning siis saab andmebaasi kasutada.

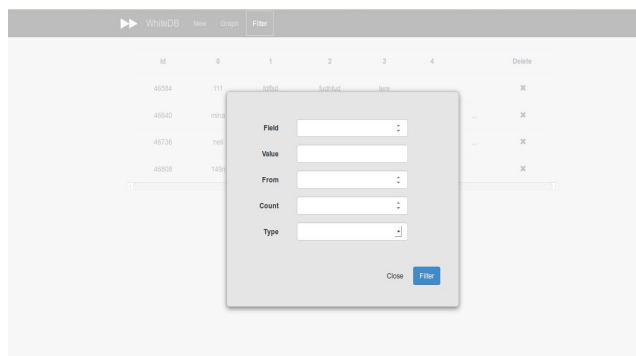
Esilehel on kõige esmane *Read* funktsionaalsus. Lehe avamisel tehakse andmebaasi päring ning kuvatakse andmebaasis olevaid ridu. Kui andmebaasis on rohkem kui 5 veergu siis neid veergusid ei kuvata, vaid näidatakse „...“. Sellel lehel saab ka kirjet kustutada vajutades vastava rea juures risti kujulise nupu peale.



id	0	1	2	3	4	Delete
46564	sdjajdjs...	pljactfd...	dsdjsdjs...	sdjsdjs...	sdjsdjs...	X
46648	sdjajdjs...	pljactfd...	dsdjsdjs...	sdjsdjs...	sdjsdjs...	X
46712	sdjajdjs...	pljactfd...	dsdjsdjs...	sdjsdjs...	sdjsdjs...	X
46776	sdjajdjs...	pljactfd...	dsdjsdjs...	sdjsdjs...	sdjsdjs...	X
46840	sdjajdjs...	pljactfd...	dsdjsdjs...	sdjsdjs...	sdjsdjs...	X
46904	sdjajdjs...	pljactfd...	dsdjsdjs...	sdjsdjs...	sdjsdjs...	X
46968	sdjajdjs...	pljactfd...	dsdjsdjs...	sdjsdjs...	sdjsdjs...	X
47032	sdjajdjs...	pljactfd...	dsdjsdjs...	sdjsdjs...	sdjsdjs...	X
47096	sdjajdjs...	pljactfd...	dsdjsdjs...	sdjsdjs...	sdjsdjs...	X

Joonis 9: Kõikide andmete kuvamise lehekülj

Esilehelt saab andmeid ka filtreerida vajutades nupule „Filter“. Filtreerida saab järgmiste parameetrite järgi: *field* – mitmendast iga kirje väljast alates otsima hakatakse, *value* – millist väärtust otsitakse, *from* – mitmendast andmebaasi kirjest alates otsima hakatakse, vaikumisi 0, *count* – sellega määratakse, mitu tulemust näidatakse, vaikumisi 1000, *type* – väärtuse tüüp.



Field:

Value:

From:

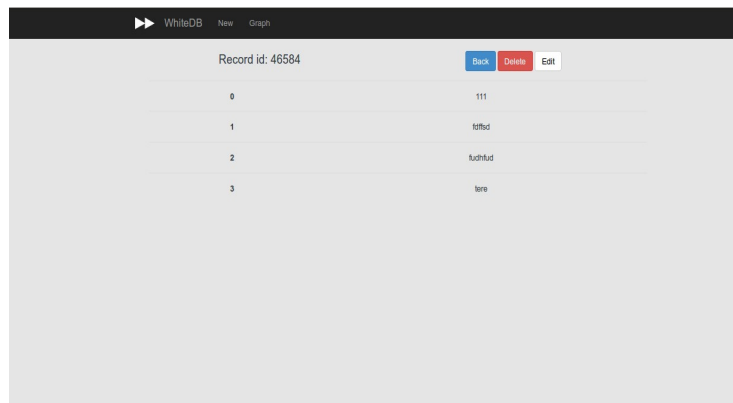
Count:

Type:

Close Filter

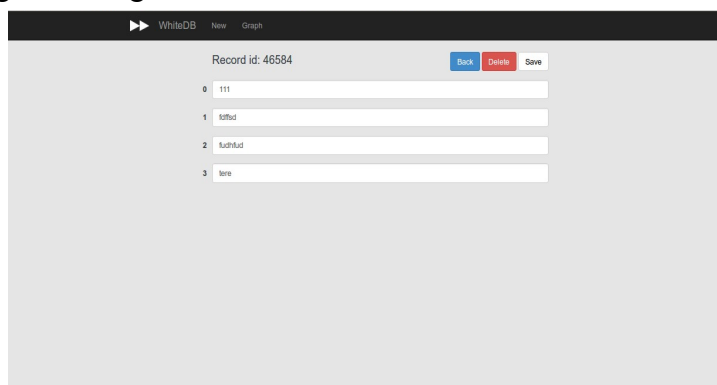
Joonis 10: Andmete filtreerimise aken

Vajutades esilehel mingi kirje peale avaneb leht kus näidatakse vastavat kirjet. Sellelt lehelt saab kirjet kustutada ning vajutades „Edit“ nuppu, avada kirje muutmise leht.



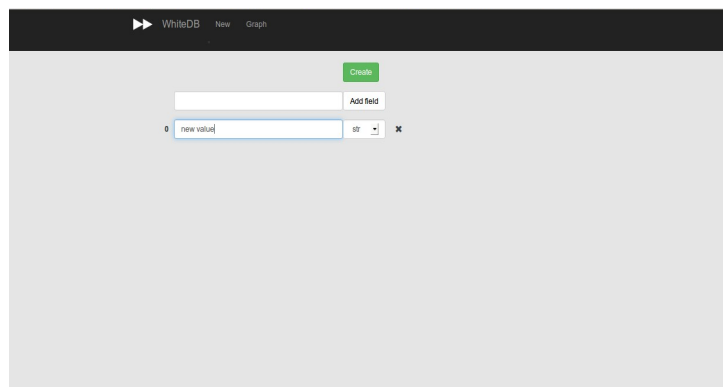
*Joonis 11: Ühe kirje kuvamise lehekül*

Muutmise lehel saab antud kirje veergude väärtusi muuta ehk teostada *Update* operatsiooni. Veerge lisada ega ära võtta ei saa. Samuti ei saa muuta ka veeru tüüpi.



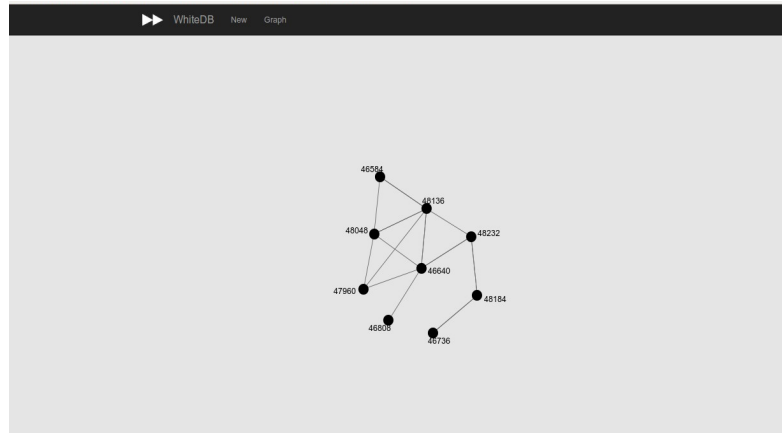
*Joonis 12: Rea muutmise lehekül*

Igalt lehelt saab, vajutades nupule „New“, luua ka uus kirje andmebaasi. Uuele kirjele saab lisada vajalike arvu veergusid. Iga veeru vaikeväärtus on sõne kuid seda saab vajadusel muuta.



*Joonis 13: Uue kirje lisamise lehekül*

Tööriistaribalt saab avada, vajutades nupule „Graph“, lehe kus kuvatakse andmebaasi kirjete vahelisi suhteid graafina. Graafi kujutamine on vajalik selle tõttu, et igal kirjel võib olla väli, mis viitab mingile teisele andmebaasi objektile.



*Joonis 14: Andmebaasi graafina kujutamine*

## 6. Liidese lahendused

Selles peatükis tehakse ülevaade suurematest probleemidest, mis liidese tegemisel ette tulid, ning nende lahendustest.

Liidese tegemisel tuli esimese probleemina lahendada see, millises andmebaasis kasutaja tegutseda tahab. Vaikimisi on andmebaasi nimeks 1000 kuid iga kasutaja võib teha meelepärase numbriga andmebaasi ning samuti võib neid andmebaase olla kasutajal mitu. Teine küsimus, mis lahendust vajas oli, et kus asub Apache *cgi-bin* ja päringute jaoks vajalik dserve. Iga kasutaja võib *cgi-bin'i* ja dserve olla vabalt valitud kohas ja kuidagi peab liides teadma kus need asuvad. Neile kahele probleemile leiti ühine lahendus. Liidesega tuleb kaasa fail *conf.json*. Selles failis hoitakse JSON kujul andmebaasi nime ja dserve asukohta. Liidese kasutamisel salvestatakse need kaks väärtust muutujatesse ja neid kasutatakse päringute tegemisel.

```
{  
  "url" : "http://localhost/cgi-bin/dserve",  
  "database" : "1000"  
}
```

*Joonis 15: Konfiguratsioonifaili vaikimisi väärtused*

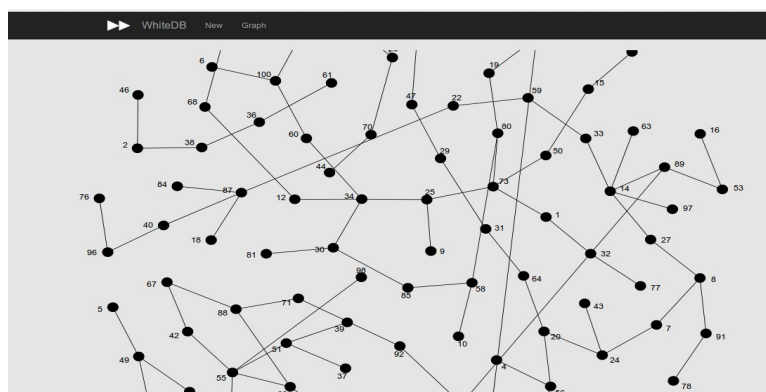
Andmebaasis olevatel kirjetel võib olla erinev arv veergusid. Tuli välja mõelda lahendus kuidas oleks kõige parem kuvada andmeid kasutajale nii, et kogu liides oleks lihtsasti arusaadav ja mugav kasutada. Lahendusena otsustati kuvada iga veeru esimest viite elementi. Juhul kui reas on vähem elemente siis iga järgnev veerg on tühi. Teisel juhul, kui kirjel on üle 5 veeru siis kuvatakse pärast viiendat veergu „ ... “. Selline andmete kuvamine teeb andmebaasi kujutamise kasutajale arusaadavamaks ja lihtsamaks.



Andmebaasis pole peal piirangut, et mitu kirjet andmebaasis olla võib. Selle tõttu oli vajalik testida kuidas andmete suur hulk mõjutab andmete kuvamist esilehel ning samuti graafi kujutamist. Esilehe andmete kuvamise testimiseks hakati andmebaasi järjest lisama 100 kaupa andmeid. Umbes 5000 kirje juures hakkas andmete laadimine ja kuvamine muutuma aeglasemaks. Esilehte laeti veel 9 korda ning kolmel korral veebilehitseja hangus ja muutus mõneks ajaks kasutuskõlbmatuks. Ülejäänud kordadel oli veebileht silmnähtavalt aeglane. Selle lahenduseks otsustati lehele kirjeid laadida 100 kaupa. Järgmiste ja eelmiste kirjete laadimiseks pandi lehele vajalikud nupud. Alternatiivina oli võimalik kasutada veel JavaScripti teeki, mis jagaksid päringute tulemused gruppidesse. See lahendus ei sobinud, kuna päritakse kõik andmebaasi kirjed korraga ning leht on selle tõttu ikkagi aeglane.

Graafi kujutamise testimiseks lisati esialgu andmebaasi 100 kirjet. Igal kirjel oli üks väli, mis viitas teisele kirjele. Graafi kuvamisel jäid mõned graafi tipud ekraanilt välja. Kuna graafisaab tippudest liigutada, siis nende ekraanilt välja jäämine ei sega oluliselt graafi vaatamist. Kui andmebaasi lisada rohkem kui 100 kirjet, pole andmete seoste kuvamine enam mugav, kuna suurem osa andmetest jäävad ekraanilt välja.

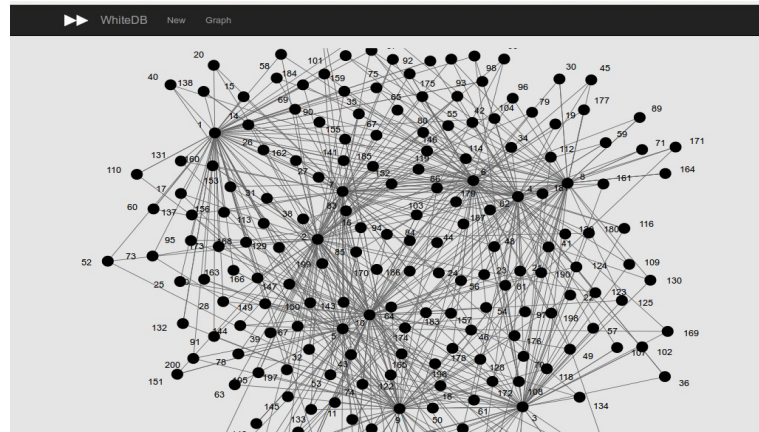
Graafi juures oli vaja järgi proovida ka seda, kui palju andmeid saab andmebaasi lisada enne, kui graafi kuvamine aeglasemaks muutub. Andmebaasi hakati lisama kirjeid 1000 kaupa. Kui andmebaasis oli 4000 kirjet hakkas nende kuvamine muutuma aeglasemaks ning tipud ei liikunud enam sujuvalt. Andmehulga edasisel kasvamisel võib veebilehitseja mõneks sekundiks hanguda.



*Joonis 18: Näide kuidas tipud ekraanilt välja lähevad ja millised olid tippude vahelised jooned enne värvi muutmist*

Lisaks kirjete piirangule pole andmebaasis ka kirjete vahelistel seostel piiri. Seetõttu oli vaja testida kuidas seoste suur hulk mõjutab graafi kujutamist. Andmebaasi lisati 200

kirjet, kus igäühel oli viide kolmele teisele kirjele. Läbi prooviti kolm varianti: iga kirje võib viidata ükskõik millisele kirjele, iga kirje võib viidata kirjetele ühest sajani ja iga kirje võib viidata esimesele kümnele kirjele. Lisatingimusena ei tohi kirje viidata endale. Selgus, et mida väiksem oli kirjete arv millele viidata, seda raskem oli aru saada, mis numbriga tipp on. Selle probleemi lahendamiseks otsustati tippude vahelisi seoseid kujutavad jooned teha numbritest ja tippudest erinevat värvi. See lihtsustas oluliselt tippude eristamist.



*Joonis 19: Näide, kui tippe, millele viidata, on 10*

Viimasena tuli lahendada veahaldus. Dserve vastu päringuid tehes võib vastuseks olla ka veateade, milleks on sõne, mitte json objekt. Kuna veateadet brauser eristada ei suuda ja käsitleb seda kui õnnestunud päringut siis ei leidunud teistsugust lahendust kui see, et kontrollitakse, kas saadud vastus sisaldab sõna „ERROR:“. Kui selline sõna leitakse siis võetakse see teade ja kuvatakse seda kasutajale.

## 7. Tehnoloogia

Selles peatükis räägitakse milliseid tehnoloogiad kasutati liidese tegemisel ning milleks iga tehnoloogia töö juures vajalik on.

Liideses on kasutusele võetud jQuery teek, mis aitab teha koodi lihtsamaks ja ühtlasi ka lühemaks. Liideses on suur osa AJAX päringutel. JQuery aitab teha AJAX-i päringud loetavamaks ning saadavate vastuste käsitlemise lihtsamaks. Samuti on DOM elementide manipulatsioon ja sündmuste haldamine tehtud lihtsamaks. [13]

```
//jQuery
var value = $('#element input:last').val()

//JavaScript
var elements = document.getElementsByTagName('body')[0].getElementsByTagName('input');
var value = elements[elements.length-1].value;
```

*Joonis 20: jQuery ja JavaScripti DOM manipulatsiooni näide*

Liideses on kasutusel Handlebars. See on teek, millega saab teha html malle, ning määrata ära millised andmed kuhu minema peaksid.[12] Javascripti koodis antakse mallile ette andmete hulk ning see hoolitseb, et andmed läheksid õigesse kohta. Mallid aitavad hoida Javascripti failid puhtamad ja paremini loetavad, kuna ei pea kirjutama HTMLi. Handlebars mallides on vaja teha numbrite võrdlust. Kuna sellised operatsioonid ei ole defineeritud Handlebarsis, siis võeti juurde Handlebars-Helpers, milles on ära defineeritud võrdlemiseks vajalikud operatsioonid.[16]

Veebilehitsejad ei toeta üheselt kõiki kujunduselemente.[18] Selle tõttu võeti kujunduses kasutusele Bootstrap teek, mis on probleemid juba ette ära lahendanud nii, et rakendus näeks erinevates veebilehitsejates samasugune välja. [14]

WhiteDB andmebaasi kirjete elemendid võivad viidata teistele kirjetele. See tekitab vajaduse nende seoste kuvamiseks kasutajale. Selle jaoks võeti kasutusele JavaScripti teek D3. See teek valiti, sest selle teegi juures oli kõige suurem ja parem valik graafe.[19] Näidiseks kasutati Moritz Stefaner tehtud graafi, mida muudeti natuke, et see oleks liidesele sobivam. Samuti on selle graafi juures plussiks see, et igat tippu saab ekraanil ringi liigutada, mis aitab tippe vajadusel ümber liigutada.[20]



## 8. Kokkuvõte

Töö põhieesmärk on luua veebiliides andmebaasiteegile WhiteDB. Liidesega peaks olema võimalik teha andmebaasides CRUD operatsioone ning näha ka andmebaasikirjete vahelisi seoseid.

Tulemusena loodi veebiliides, mis tuleb kasutajale andmebaasiteeki WhiteDB alla laadides kaasa.

Igal kasutajal on võimalik oma andmebaaside andmeid muuta, kustutada, vaadata ja muuta. Andmebaasist andmete küsimine tehti jQuery ja JavaScriptiga. Andmete korrektseks kuvamiseks võeti lisaks kasutusele teegid Handlebars ja Bootstrap.

Iga andmebaasi kirje võib viidata ka teisele andmebaasi kirjele. Selle saavutamiseks kasutati JavaScripti teeki D3. Selle abil on kasutajal võimalik näha kirjete vahelisi seoseid graafi kujul.

Kuna Dserve peal pole veel valmis tehtud kirjete muutmise ja uue kirje lisamine, siis antud töö juures tehti ainult vajalikud vaated ning funktsionaalsus tuleb juurde lisada hiljem.

Samuti on Dserve juures puudu muudatus *search* meetodi juures, et oleks võimalik eristada ka viiteid teistele kirjetele. Liidese funktsionaalsuses olemas küll viite olemasolu kontrolliv *if* klausel kuid hetkel see andmete esitlust ja graafide kujutamist ei muuda.

Edasise arenduse käigus tuleks välja mõelda parem lahendus graafi kujutamisele. Praegusel juhul ei ole väga suurte andmekogust juures graafi kasutamine otstarbekas, kuna paljud tipud jäävad ekraanilt välja ning veebilehitseja võib suure hulga andmete korral hanguda.

## Kasutatud kirjandus

1. Top 5 Considerations When Evaluating NoSQL Databases,  
[http://info.mongodb.com/rs/mongodb/images/10gen\\_Top\\_5\\_NoSQL\\_Considerations.pdf](http://info.mongodb.com/rs/mongodb/images/10gen_Top_5_NoSQL_Considerations.pdf) [WWW]
2. M. J. Graham, M. J. Fitzpatrick, and T. A. McGlynn, The National Virtual Observatory Book, ASP Conference Series, Vol. 382, 2008, pp 645-655,  
[http://articles.adsabs.harvard.edu/cgi-bin/nph-iarticle\\_query?db\\_key=AST&bibcode=2007ASPC..382..645M&letter=.&classic=YES&defaultprint=YES&whole\\_paper=YES&page=645&epage=645&send=Send+PDF&filetype=.pdf](http://articles.adsabs.harvard.edu/cgi-bin/nph-iarticle_query?db_key=AST&bibcode=2007ASPC..382..645M&letter=.&classic=YES&defaultprint=YES&whole_paper=YES&page=645&epage=645&send=Send+PDF&filetype=.pdf)  
[WWW] (15.05.2014)
3. Kyle Banker. „MongoDB in action“. Shelter Island (N.Y.) : Manning, 2012.
4. J. Chris Anderson, Jan Lenhardt, Noah Slater. „CouchDB: The Definitive Guide“. Sebastopol: O'Reilly, 2010.
5. „Web-Age Information Management Lecture Notes in Computer Science“ Volume 6185, 2010, pp 25-36
6. Tiago Macedo & Fred Oliveira. „Redis Cookbook“. Sebastopol: O'Reilly, 2011
7. Ron Plew and Ryan Stephens, Sams Teach Yourself SQL in 24 Hours, 3rd Edition,  
<http://www.informit.com/articles/article.aspx?p=29583> [WWW](18.05.2014)
8. Greg Burd, NoSQL  
<https://www.usenix.org/legacy/publications/login/2011-10/openpdfs/Burd.pdf>  
[WWW] (19.05.2014)
9. <http://phpggadmin.sourceforge.net/doku.php> [WWW] (20.05.2014)
10. <http://sourceforge.net/projects/phpmyadmin/> [WWW] (20.05.2014)
11. [www.whitedb.com](http://www.whitedb.com) [WWW] (20.05.2014)
12. <http://handlebarsjs.com/> [WWW] (22.05.2014)

13. [www.jquery.com](http://www.jquery.com) [WWW] (21.05.2014)
14. [www.bootstrap.com](http://www.bootstrap.com) [WWW] (17.05.2014)
15. <http://phpmoadmin.com/> [WWW] (12.05.2014)
16. <https://github.com/danharper/Handlebars-Helpers> [WWW] (22.05.2014)
17. <http://neo4j.com/> [WWW] (21.05.2014)
18. <http://caniuse.com/> [WWW] (17.05.2014)
19. <http://d3js.org/> [WWW] (22.05.2014)
20. <http://bl.ocks.org/MoritzStefaner/1377729> [WWW] (22.05.2014)