

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Informaatika instituut

Infosüsteemide õppetool

# PostgreSQL-i põhise meta-andmetega juhitavate veebirakenduste kiirprogrammeerimiskeskonna projekteerimine ja realiseerimine

Magistritöö

Üliõpilane:	Rait Raidma
Üliõpilaskood:	143682IAPM
Juhendaja:	dotsent Erki Eessaar

## 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**

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 41 leheküljel, 9 peatükki, 6 joonist, 2 tabelit.

## **Abstract**

The thesis is in estonian and contains 41 pages of text, 9 chapters, 6 figures, 2 tables.

## Lühendite ja mõistete sõnastik

SQL	<i>Structured Query Language</i> , struktureeritud andmebaasikeel andmete käitlemiseks, õiguste jagamiseks ning andmebaasiobjektide haldamiseks
FSF	<i>Free Software Foundation</i> , MTÜ, mis propageerib arvuti kasutajate vabadust ja kaitseb vaba tarkvara kasutajate õigusi
OSI	<i>Open Source Initiative</i> , Organisatsioon, mis propageerib avatud lähtekoodiga tarkvara
Juurutama	<i>Deploy</i> , Tarkvara või riistvara töölepanekuga seotud protsesside - installeerimine, konfigureerimine, käitamine, testimine - läbimine [28]
CRUD	<i>Create Read Update Delete</i> , Lühend, mis tähistab andmetega manipuleerimise nelja põhitegevust: loomine, lugemine, muutmine ja kustutamine
Meta-andmed	Andmed andmete kohta

# Sisukord

<b>1</b>	<b>Sissejuhatus</b>	<b>10</b>
1.1	Taust ja probleem . . . . .	10
1.2	Ülesande püstitus . . . . .	10
1.3	Metoodika . . . . .	10
1.4	Ülevaade tööst . . . . .	11
<b>2</b>	<b>Teoreetiline taust</b>	<b>12</b>
2.1	Andmebaasi avalik liides . . . . .	12
2.1.1	Vaadete kasutamise eelised ja võimalused . . . . .	12
2.1.2	Rutiinide kasutamise eelised . . . . .	13
2.2	Ühendumine teiste andmebaasidega . . . . .	14
2.2.1	dblink . . . . .	14
2.2.2	postgres_fdw . . . . .	15
2.2.3	Mooduli valik . . . . .	15
2.3	Andmebaasi objektide kirjelduste küsimine . . . . .	15
2.3.1	information_schema . . . . .	16
2.3.2	pg_catalog . . . . .	16
2.4	Uue süsteemi visioon . . . . .	17
2.5	Eksisteerivate programmide analüüs . . . . .	17
2.5.1	Oracle Application Express (APEX) . . . . .	17
2.5.2	NuBuilder . . . . .	18
2.5.3	Xataface . . . . .	19
2.6	Täpsustunud ülesande püstitus . . . . .	19
2.7	Kasutatavad tehnoloogiad . . . . .	19
2.7.1	Vagrant 1.8.1 . . . . .	19
2.7.2	Apache HTTP server . . . . .	19
2.7.3	PHP 5.5.3 . . . . .	20
2.7.4	Postgresql 9.4 . . . . .	20
2.7.5	Javascript . . . . .	20
2.7.6	AngularJS . . . . .	20
2.7.7	Bootstrap . . . . .	20
2.8	Litsents . . . . .	20
2.8.1	Free Software . . . . .	20
2.8.2	Open Source . . . . .	21
2.8.3	<i>Free Software</i> ja <i>Open Source</i> võrdlus . . . . .	22
2.8.4	Litsentsi valik . . . . .	22
<b>3</b>	<b>Süsteemi analüüs</b>	<b>24</b>

3.1	Tegutsejad . . . . .	24
3.2	Terviksüsteemi tükeldus allsüsteemideks . . . . .	24
3.2.1	Pädevusalad . . . . .	24
3.2.2	Funktsionaalsed allsüsteemid . . . . .	24
3.2.3	Registrid . . . . .	25
3.3	Rakenduse funktsionaalne allsüsteem . . . . .	25
3.3.1	Eesmärgid . . . . .	25
3.3.2	Allsüsteemi poolt kasutatavad registrid . . . . .	25
3.3.3	Allsüsteemi kasutusjuhtude eskiismudel . . . . .	25
3.4	Rakenduste funktsionaalne allsüsteem . . . . .	27
3.4.1	Eesmärgid . . . . .	27
3.4.2	Allsüsteemi poolt kasutatavad registrid . . . . .	27
3.4.3	Allsüsteemi kasutusjuhtude eskiismudel . . . . .	27
3.5	Lehtede funktsionaalne allsüsteem . . . . .	29
3.5.1	Eesmärgid . . . . .	29
3.5.2	Allsüsteemi poolt kasutatavad registrid . . . . .	29
3.5.3	Allsüsteemi kasutusjuhtude eskiismudel . . . . .	30
3.6	Regioonide funktsionaalne allsüsteem . . . . .	30
3.6.1	Eesmärgid . . . . .	30
3.6.2	Allsüsteemi poolt kasutatavad registrid . . . . .	31
3.6.3	Allsüsteemi kasutusjuhtude eskiismudel . . . . .	31
3.7	Navigatsioonide funktsionaalne allsüsteem . . . . .	33
3.7.1	Eesmärgid . . . . .	33
3.7.2	Allsüsteemi poolt kasutatavad registrid . . . . .	33
3.7.3	Allsüsteemi kasutusjuhtude eskiismudel . . . . .	33
3.8	Mittefunktsionaalsed nõuded . . . . .	35
<b>4</b>	<b>Andmebaasi disain</b>	<b>35</b>
<b>5</b>	<b>Kasutajaliidese disain</b>	<b>35</b>
<b>6</b>	<b>Rakenduse disain</b>	<b>35</b>
<b>7</b>	<b>Näide</b>	<b>35</b>
<b>8</b>	<b>Kokkuvõte</b>	<b>35</b>
<b>9</b>	<b>Summary</b>	<b>36</b>
	<b>Kasutatud kirjandus</b>	<b>37</b>
	<b>Lisa 1</b>	<b>40</b>

## **Jooniste loetelu**

1	Andmebaasi avalik liides . . . . .	12
2	Rakenduse funktsionaalse allsüsteemi kasutusjuhtude eskiismudel . . . .	26
3	Rakenduste funktsionaalse allsüsteemi kasutusjuhtude eskiismudel . . . .	28
4	Lehtede funktsionaalse allsüsteemi kasutusjuhtude eskiismudel . . . . .	30
5	Regioonide funktsionaalse allsüsteemi kasutusjuhtude eskiismudel . . . .	32
6	Navigatsioonide funktsionaalse allsüsteemi kasutusjuhtude eskiismudel .	34



## **Tabelite loetelu**

1	Litsentside võrdlus . . . . .	23
2	Mittefunktsionaalsed nõuded . . . . .	35

# **1 Sissejuhatus**

## **1.1 Taust ja probleem**

TTÜ-s õpetatava aine “Andmebaasid II” raames tuleb üliõpilastel ühe õpiväljundina luua andmebaas koos seda kasutava rakendusega, kus rakendus suhtleb andmebaasiga läbi avaliku andmebaasiliidese. Andmebaasi loomiseks võib kasutada andmebaasisüsteeme PostgreSQL [18] ja Oracle [15]. Juhul, kui andmebaas on loodud Oracle andmebaasisüsteemi abil, siis on üliõpilastel rakenduse loomiseks võimalus kasutada Oracle APEX-it [14]. PostgreSQL andmebaasisüsteemiga loodud andmebaasi korral tuleb rakendus programmeerida kasutades PHP-d [17]. See tähendab, et üliõpilane ei saa keskenduda täielikult andmebaasi täiustamisele vaid peab tegelema ka lisaprogrammeerimisega. Töö tulemusena valmiva süsteemi abil peaks üliõpilastel olema lihtsam luua näidisrakendusi, mis kasutavad andmebaasisüsteemina PostgreSQL-i.

Töö valmis 2016. aasta kevadel Tallinna Tehnikaülikoolis.

## **1.2 Ülesande püstitus**

Töö eesmärgiks on disainida ning realiseerida PostgreSQL põhine meta-andmetega juhitav veebipõhine veebirakenduste kiirprogrammeerimiskeskond. Süsteem peab suhtlema teiste andmebaasidega läbi avaliku liidese ning andmebaasi poole peal genereerima kasutajale kuvatava vastuse.

Loodav süsteem peab toetama PostgreSQL 9.4 ja PHP 5.5.0 ning tuleb välja anda vabavara litsentsi all.

## **1.3 Metoodika**

Esiteks tuleb uurida, kas liideste kasutamine andmebaasi poole peal annab süsteemile mingi eelise. Seejärel tuleb selgeks teha, kas ja kuidas saab ühest andmebaasist suhelda teiste andmebaasidega. Ning viimasena, kuidas küsida teistest andmebaasidest infot liideste kohta. Lisaks uurin, milliseid sarnaseid süsteeme on veel olemas ning kuidas need on üles ehitatud.

Töö tulemusena valmib süsteem, mille abil saab luua veebipõhiseid rakendusi, mis kasutavad andmete edastamiseks liideseid. Töö tulemuse valideerimiseks loodakse näidisrakendus, kus realiseeritakse õppejõu poolt ette antud kasutusjuht.

## **1.4 Ülevaade tööst**

TODO



kasutatakse kuritahtlikult valitud funktsioone ja operaatoreid, et näha varjatud infot [23]

- Võimaldavad pärida andmeid erinevatest tabelitest ja andmebaasidest, peites kasutajate eest päringu tegeliku keerukuse. Vaate koostamiseks vajalik päring on eelnevalt kompilleeritud ja optimeeritud, et tagada parem jõudlus. Vaated kasutavad päringu täitmisel baastabelitele loodud indekseid.
- Võimaldavad varjata rakenduse eest baastabelites olevaid disaini -ja andmevigasid, andes lisaaega nende parandamiseks.
- Võimaldavad kuvada samu andmeid erineval kujul ühendatuna, kasvõi nt XML-na või JSON-na.
- Läbi vaadete, mis vastavad teatud tingimustele, on võimalik teha andmemuudatusi baastabelites, kui realiseerida INSTEAD OF triggerid.

[4, lk 172-173]

### **2.1.2 Rutiinide kasutamise eelised**

- Üle võrgu saadetavate andmete ja SQL koodi hulk hoitakse minimaalsena, mille tulemusel suureneb rakenduse jõudlus.
- Rutiinide kood on andmebaasi serveris eelnevalt kompilleeritud ja optimeeritud, suurendades rutiini täitmise efektiivsust.
- Andmetöötluse jaoks kasutatakse andmebaasiserveri jõudlust, mitte rakendusserveri ega kliendi masina oma.
- Rutiinis olevat SQL koodi on lihtsam testida ja optimeerida, kui rakendusse sisse kirjutatud SQL-i.
- Rutiinide käivitusõiguste abil saab piirata ligipääsu teatud rollidele ning suurendada seeläbi turvalisust.
- Rutiinis käivituvad laused tehakse ühe transaktsiooni jooksul. See aitab vältida osalisi andmemuudatusi, kus üks osa muudatustest läks läbi, teine osa aga mitte.

[4, lk 179, 195]

## 2.2 Ühendumine teiste andmebaasidega

Loodava süsteemi üheks tingimuseks on, et selle abil peab saama luua rakendusi erinevate andmebaaside põhjal. PostgreSQL andmebaasisüsteemis pole realiseeritud andmebaaside vahelisi viitasid ning seetõttu ei saa koostada päringuid kujul:

```
1 select * from other_db_name . schema_name . table_name ;
```

Eelnev päring annab tulemuseks veateate:

```
ERROR:  cross-database references are not implemented: "  
        other_db_name . schema_name . table_name "
```

Selleks, et ühenduda väliste PostgreSQL andmebaasidega, tuleb kasutada kas dblink või postgres\_fdw moodulit.

### 2.2.1 dblink

Mooduli installeerimine:

```
1 CREATE EXTENSION IF NOT EXISTS dblink ;
```

Andmete küsimiseks välisest andmebaasist tuleb ette anda andmebaasi nimi, kasutaja ja parool ning lause, mida käivitada soovitakse. Päring käivitatakse välises andmebaasis. Päringuks võib olla iga SQL lause, mis tagastab read.[20] Allpool on toodud näide päringu koostamisest dblink mooduli abil.

```
1 SELECT schema_name , owner_id  
2 FROM dblink (  
3     'dbname=external_database_name user=  
4     external_database_user password=  
5     external_database_user_password' ,  
6     'SELECT upper(nspname) , nspowner FROM pg_catalog .  
7     pg_namespace ;'  
8 ) AS (  
9     schema_name varchar ,  
10    owner_id int  
11 );
```

### 2.2.2 postgres\_fdw

Selle mooduli poolt pakutav funktsionaalsus kattub suurel määral *dblink* 2.2.1 mooduli funktsionaalsusega, kuid pakub standardsemat süntaksit päringute koostamiseks ning võib *dblink*-i kohati edestada jõudluse poolest.

*postgres\_fdw* loob ühenduse välise serveriga siis, kui tehakse esimene päring välise tabeli vastu. Seda ühendust hoitakse alles ning kasutatakse järgmiste päringute jaoks sama sessiooni piires. Kui väliselt serverilt küsitakse infot erinevate kasutajatena *user mappings*, siis luuakse iga kasutaja jaoks uus ühendus.

*postgres\_fdw* üritab optimeerida väliseid päringuid, et vähendada küsitavate andmete hulka. Selleks saadetakse koos päringuga *WHERE*-tingimus ning ei laeta alla veerge, mida pole päringu täitmiseks vaja. Selleks et vältida valesid päringutulemusi, ei saadeta *WHERE*-tingimusi, kui kasutatakse midagi peale sisse ehitatud andmetüüpide, operaatorite ja funktsioonide või kui operaatorid ja funktsioonid pole muutumatud (*immutable*). [21]

*postgres\_fdw* võimaldab lisaks andmete küsimisele (*SELECT*) ka andmeid lisada (*INSERT*), muuta (*UPDATE*) ja kustutada (*DELETE*) välisest tabelist. Küll aga ei võimalda antud moodul välja kutsuda välises andmebaasis olevaid funktsioone kujul:

```
SELECT function_from_external_database ();
```

### 2.2.3 Mooduli valik

Kuna loodav süsteem peab suutma välja kutsuda välistes andmebaasides olevaid funktsioone, siis pole tuleb kasutada *dblink* 2.2.1 moodulit.

## 2.3 Andmebaasi objektide kirjelduste küsimine

*Information\_schema* sisaldab vaateid andmebaasis olevate objektide kohta. Kuna *information\_schema* on defineeritud SQL standardis, siis võib eeldada, et selle formaat ei muutu. [22] Küll aga ei sisalda *information\_schema* infot PostgreSQL-spetsiifiliste võimaluste kohta. Selleks tuleb pöörduda *pg\_catalog*-i poole. [24]

### 2.3.1 information\_schema

schemata	Sisaldab kõiki skeeme, millele kasutajal on ligipääs.
views	Sisaldab kõiki vaateid, mis asuvad antud andmebaasis. Näidatakse ainult selliseid vaateid, millele kasutajal on ligipääs. Paraku ei saa sealt aga infot materialiseeritud vaadete kohta.
columns	Sisaldab infot andmebaasis olevate tabelite ja vaadete veergude kohta. Näidatakse ainult neid veerge, millele kasutajal on ligipääs. Kui tagastatav tüüp on massiiv, siis saab selle kohta infot information_schema.element_types vaatest. Kui tagastatav tüüp on USER-DEFINED, siis saab selle kohta infot udt_name veerust. Kui veerg on loodud domeeni põhjal, siis saab domeeni nime domain_name veerust.
routines	Sisaldab infot andmebaasis olevate funktsioonide kohta, millele kasutajal on ligipääs. data_type veerg sisaldab infot tagastatava tüübi kohta. Kui tagastatav tüüp on massiiv, siis saab selle kohta infot information_schema.element_types vaatest. Kui tagastatav tüüp on USER-DEFINED, siis saab selle kohta infot type_udt_name veerust.
parameters	Sisaldab infot andmebaasis olevate funktsioonide parameetrite kohta. Parameetreid näidatakse ainult nende funktsioonide kohta, millele kasutajal on ligipääs.
element_types	Sisaldab infot massiivi tüüpide kohta.

[22]

### 2.3.2 pg\_catalog

Kuna SQL standardiga paika pandud information\_schema ei sisalda infot samasse klastrisse kuuluvate andmebaaside ega materialiseeritud vaadete kohta, siis tuleb seda infot küsida pg\_catalog-st. Lisaks ei saa information\_schema abil kontrollida kasutajanime ja parooli klappivust.

pg_database	Säilitab infot olemas olevate andmebaaside kohta. Erinevalt enamikest süsteemi kataloogidest on pg_database jagatud kõikide klastrisse kuuluvate andmebaaside vahel.
-------------	--



<code>pg_namespace</code>	Säilitab infot nimeruumide kohta. Sealt on võimalik kätte saada andmebaasis olevad skeemid.
<code>pg_shadow</code>	Sisaldab infot kasutajate kohta, kellel on sisselogimisõigus. See tabel sisaldab paroole kujul <code>'md5'    md5(parool  kasutajanimi)</code> .
<code>pg_class</code>	Sisaldab infot kõige kohta, millel on veerud, või on mõnes muus mõttes tabeliga sarnane. Sealt saab infot vaadete ja materialiseeritud vaadete kohta. Selle tabeli pealt on tehtud ka vaates <code>pg_views</code> ja <code>pg_matviews</code> , millest on samuti võimalik küsida infot vastavalt vaadete ja materialiseeritud vaadete kohta. Lisaks ei pea kasutajatel olema reaalne ligipääs antud objektidele, et näha infot nende objektide kohta.
<code>pg_attribute</code>	Sisaldab infot veergude kohta.
<code>pg_type</code>	Sisaldab infot andmetüüpide kohta. Siin tabelis on esindatud nii põhiandmetüübid, kasutaja loodud tüübid, domeenid ja komposiitandmetüübid, mis luuakse iga andmebaasis oleva tabeli jaoks.
<code>pg_proc</code>	Sisaldab infot funktsioonide kohta.

[24]

## 2.4 Uue süsteemi visioon

Eesmärk on luua lihtsasti kasutatav veebipõhine süsteem, mille abil saaks kiiresti luua teisi veebipõhiseid rakendusi. Loodavates rakendustes peab andmete pärimine, lisamine, muutmine ning kustutamine käima läbi andmebaasi avaliku liidese. Kogu vajalik info rakenduse loomiseks tuleb hoida andmebaasis.

## 2.5 Eksisteerivate programmide analüüs

### 2.5.1 Oracle Application Express (APEX)

Oracle APEX on veebipõhine rakendus loomaks kiirelt ja lihtsalt teisi veebipõhiseid rakendusi. Kogu süsteem on juhitud andmebaasis hoitavate metaandmetega. APEX kasutab tööks Oracle andmebaasisüsteemi.

APEX (v 5.0.3.00.03) koosneb neljast põhiosast:

- **Application Builder** - Võimaldab luua ja hallata uusi rakendusi. Rakendused koosnevad lehtedest. Lehed omakorda sisaldavad regioone. Regioonides võib kuvada raporteid, graafikuid, vorme jpm. Regioonid sisaldavad komponente, mille abil on võimalik kasutajalt infot küsida ning seda esitada. Lisaks on võimalik näha lehtede statistikat ning hallata seadeid.
- **SQL Workshop** - Võimaldab näha ja hallata andmebaasiobjekte, jooksutada päringuid, importida/exportida andmebaasis olevaid andmeid, koostada päringuid graafilise liidese abil, luua RESTful liideseid jpm.
- **Team Development** - Tööde- ja vigadehaldus süsteem. Võimaldab arendajatel ülesandeid planeerida ja hallata.
- **Packaged Apps** - Galerii näidisrakendustest, mida on võimalik kohe kasutamiseks installeerida.

[14]

### 2.5.2 NuBuilder

NuBuilder on veebipõhine arendusplatvorm loomaks veebipõhiseid rakendusi. Lehtede kirjeldused (sh PHP, JS ja SQL päringud) hoitakse andmebaasis, mis muudab rakenduse varundamise lihtsaks.

NuBuilder on kirjutatud PHP-s ning andmeid hoitakse MySQL andmebaasisüsteemis. Tabelite põhjal on võimalik luua lihtsaid CRUD vorme, kus on võimalik tabelis olevaid andmeid lugeda, lisada, muuta ja kustutada. SQL päringute põhjal on võimalik luua raporteid, mida arendaja saab veebiliidese kaudu disainida. Oma kodulehel väidavad nad, et tegu on *Open Source* tarkvaraga ning lähtekood on avalikult üleväl [12], kuid kusagil pole mainitud, millise *Open Source* litsentsi alt on tarkvara välja antud.

Koodi puhul täheldasin mitut puudujääki:

- Failid on kehvasti struktureeritud - php, js, png ja gif failid on kõik koos ühes kaustas.
- PHP ja HTML on kirjutatud läbisegi, mis teeb disaini muutmise keeruliseks.
- Kasutatakse \$GLOBALS muutujat - see raskendab arusaamist, kus võidakse muutujale programmi töö ajal väärtusi omistada.

- Funktsioonid on liiga pikad - paljud funktsioonid täidavad korraga liiga palju ülesandeid ja seetõttu on raskendatud nendest arusaamine.

[11]

### 2.5.3 Xataface

Xataface on programm, millega saab tabelite põhjal genereerida vorme ja kuvasid. Pärast genereerimist tuleb loodud failid serverisse üles laadida. Lehtede konfigureerimine toimub INI failide abil.[30]

Xataface on avatud lähtekoodiga ning antud välja GPL litsentsi all. Programm on kirjutatud PHP-s [17] ning andmebaasina kasutatakse MySQL-i [10].

Kasutatud on palju väliseid teeke. Programmil on üks põhiline arendaja ning igapäevast arendustööd ei toimu. [31]

## 2.6 Täpsustunud ülesande püstitus

## 2.7 Kasutatavad tehnoloogiad

### 2.7.1 Vagrant 1.8.1

Vagrant on käsureaprogramm, millega saab hallata virtuaalmasina elutsüklit. Vagrant isoleerib programmilised sõltuvused ja nende konfiguratsioonid ühtsesse eraldiseisvasse keskkonda. Keskkonna konfigureerimiseks saab kasutada käsurea käsklusi, *Ansible*-t [2], *Puppet*-it [25], *Chef*-i [5], *Docker*-it [6] ja *Salt*-i [26]. Tänu Vagrantile saavad kõik luua endale täpselt ühesuguse keskkonna, kus programme jooksutada, vähendades võimalust, et ühes arvutis programm jookseb, teises aga mitte. [27]

### 2.7.2 Apache HTTP server

Apache HTTP server on avatud lähtekoodiga HTTP veebiserver.[1]

### 2.7.3 PHP 5.5.3

PHP (*PHP: Hypertext Preprocessor*) on avatud lähtekoodiga skriptimiskeel, mis on peamiselt mõeldud veebiprogrammeerimiseks. [29] PHP koodi protsessitakse PHP interpretaatori abil. Üldjuhul kasutatakse interpreteerimiseks *Zend Engine*-t, kuid PHP-d võimalik jooksutada kasutada ka *HHVM*-i [8] abil. PHP toetab erinevaid operatsioonisüsteeme, sealhulgas Windows-i erinevaid versioone ja Linuxi erinevaid distributsioone.

### 2.7.4 Postgresql 9.4

PostgreSQL on avatud lähtekoodiga objekt-relatsiooniline andmebaasisüsteem, mis vastab täielikult *ACID* nõuetele. See toetab *foreign key*-sid, *join*-e, *view*-sid, *trigger*-eid ja salvestatud protseduure. PostgreSQL toetab erinevaid operatsioonisüsteeme, sealhulgas Windows-i erinevaid versioone ja Linuxi erinevaid distributsioone. [19]

### 2.7.5 Javascript

### 2.7.6 AngularJS

### 2.7.7 Bootstrap

## 2.8 Litsents

### 2.8.1 Free Software

*Free Software* (Vaba tarkvara) tähendab, et kasutajatel on vabads tarkvara jooksutada, kopeerida, levitada, uurida, muuta ja täiustada. Seega *Free Software* on kasutaja vabaduse, mitte hinna küsimus.

Tarkvara on *Free Software*, kui selle kasutajate jaoks on täidetud neli olulist kriteeriumit:

- Vabadus 0: jooksutada programmi oma suva järgi, ükskõik mis eesmärgil
- Vabadus 1: uurida, kuidas programm töötab ja seda muuta (eeldab ligipääsu lähtekoodile)
- Vabadus 2: levitada antud tarkvara

- Vabadus 3: levitada antud tarkvara muudetud kujul (eeldab ligipääsu lähtekoodile)

Vabadus levitada (vabadused 2 ja 3) tähendab vabadust jagada antud tarkvara muudetud või muutmata kujul kas tasu eest või tasuta - selleks ei pea kelleltki luba küsima. Küll aga peab jagatav koopia sisaldama nii lähtekoodi kui ka käivitavat programmi (kui programmeerimiskeel toetab seda võimalust)

*Free Software* ei tähenda, et tegu ei võiks olla kommertstarkvaraga. *Free Software* võib omandada tasuta või raha eest. Vaatamata sellele, kuidas koopia antud tarkvarast omandati, jääb omandajale vabadus antud tarkvara jagada, muuta ja müüa. [7]

## 2.8.2 Open Source

*Open Source* (Avatud lähtekood) ei tähenda ainult ligipääsu lähtekoodile. Tarkvara levitamisel peab lätuma järgmistest reeglitest:

1. Vaba jagamine - Litsents ei tohi piirata ühtegi osapoolt tarkvara müümast või jagamast.
2. Lähtekood - Tarkvara peab sisaldama lähtekoodi ning lähtekoodi ja kompileeritud koodi jagamine peab olema lubatud. Kui tarkvara ei jagata koos lähtekoodiga, peab lähtekood olema mujalt mõistliku vaevaga kättesaadav.
3. Tuletatud tarkvara - Litsents peab lubama muudatusi ja tuletatud tarkvara ning peab lubama nende jagamist samadel litsentsitingimustel.
4. Autori lähtekoodi terviklikkus - Litsents võib keelata muudetud lähtekoodi jagamist üksnes siis, kui on lubatud jagada paikefaile (*patch file*), et muuta programmi lähtekoodi selle loomise mingis järgus (*build time*). Litsents peab selgelt lubama muudetud lähtekoodiga tarkvara jagamist. Litsents võib nõuda, et tuletatud tarkvara kannaksid teist nime või versiooninumbrit, kui originaaltarkvara.
5. Isikute või gruppide diskrimineerimiskeeld - Litsents ei tohi diskrimineerida ühtegi isikut või isikute gruppi.
6. Tegevusvaldkonna diskrimineerimiskeeld - Litsents ei tohi piirata ühtegi konkreetset tegevusvaldkonda.
7. Litsentsi jagamine - Programmile sätestatud õigused kehtivad kõigile, kellele programm on jagatud, ilma, et osapooled vajaksid täiendavat litsentsi.

8. Litsents ei tohi olla tootespetsiifiline - Programile sätestatud õigused ei tohi sõltuda sellest, kas programm kuulub mõne teise programmi koosseisu.
9. Litsents ei tohi piirata teisi tarkvarasid - Litsents ei tohi panna piiranguid teistele tarkvaradele, mida jagatakse koos antud tarkvaraga.
10. Litsents peab olema tehnoloogiliselt neutraalne - Ükski klausel ei tohi viidata konkreetsele tehnoloogiale, stiilile või liidesele.

[13]

### **2.8.3 *Free Software* ja *Open Source* võrdlus**

*Open Source* kriteeriumid on veidi vabamad kui *Free Software* omad. Kõik eksisteerivad *Free Software* programmid kvalifitseeruvad *Open Source* tarkvara alla. Enamik *Open Source* tarkvarast on *Free Software*, kuid leidub ka erandeid. [16]

- Mõlemad nimed ei väljenda täpselt seda, mida nende all tegelikult on mõeldud.
- Mõlemad lubavad tarkvara jagada tasuta või seda müüa.
- *Open Source* kriteeriumid kehtivad ainult lähtekoodile, mitte aga kompilleeritud programmile.

### **2.8.4 Litsentsi valik**

Üheks töö eesmärgiks oli avaldada loodava prototüübi lähtekood avatud tarkvarana. Olemasolevaid litsentse on väga palju. Selleks, et valida välja litsents, mille all avaldada loodav tarkvara, leian esiteks populaarseimad litsentsid ning võdlen neid omavahel. GitHub-i poolt avaldatud statistika kohaselt on populaarseimad litsentsid: MIT (44,69%), GPLv2 (12,96%), Apache (11,19%) ja GPLv3 (8,88%). [3]

Kõik eelpool nimetatud litsentsid täidavad nii *Open Source* kui ka *Free Software* tingimusi. Tabelis 1 on välja toodud litsentside võrdlus.

	Nõutud	Lubatud	Keelatud
MIT	Litsents ja copyright märged	Kaubanduslik kasutamine Jagamine Muutmine Privaatne kasutamine	Võtta vastutusele
Apache License 2.0	Litsents ja copyright märged Teavitust muudatustest	Kaubanduslik kasutamine Jagamine Muutmine Patendi kasutamine Privaatne kasutamine	Võtta vastutusele  Kasutada kaubamärki
GNU GPLv3  GNU GPLv2	Lähtekoodi avaldamine  Litsents ja copyright märged Sama litsents Teavitust muudatustest	Kaubanduslik kasutamine Jagamine  Muutmine Patendi kasutamine Privaatne kasutamine	Võtta vastutusele

Tabel 1. Litsentside võrdlus  
[9]

Valitud sai MIT litsents, kuna see seab kasutajatele kõige vähem piiranguid ning arendajale kõige vähem kohustusi.

## **3 Süsteemi analüüs**

### **3.1 Tegutsejad**

- Administraator.
- Kasutaja.

### **3.2 Terviksüsteemi tükeldus allsüsteemideks**

#### **3.2.1 Pädevusalad**

- Administraatori pädevusala.
- Kasutaja pädevusala.

Administraatori pädevusala kasutab kõiki allsüsteeme.

Kasutaja pädevusala kasutab ainult rakenduse allsüsteemi.

#### **3.2.2 Funktsionaalsed allsüsteemid**

- Rakenduse funktsionaalne allsüsteem.
- Rakenduste funktsionaalne allsüsteem.
- Lehtede funktsionaalne allsüsteem.
- Regioonide funktsionaalne allsüsteem.
- Navigatsioonide funktsionaalne allsüsteem.
- Mallide funktsionaalne allsüsteem.

Antud töös ei realiseerita mallide funktsionaalset allsüsteemi.



### **3.2.3 Registrid**

- Andmebaasiobjektide register.
- Rakenduste register.
- Lehtede register.
- Regioonide register.
- Navigatsioonide register.
- Mallide register.

## **3.3 Rakenduse funktsionaalne allüsteem**

### **3.3.1 Eesmärgid**

- Võimaldada administraatoril ja kasutajal kasutada loodud rakendust.

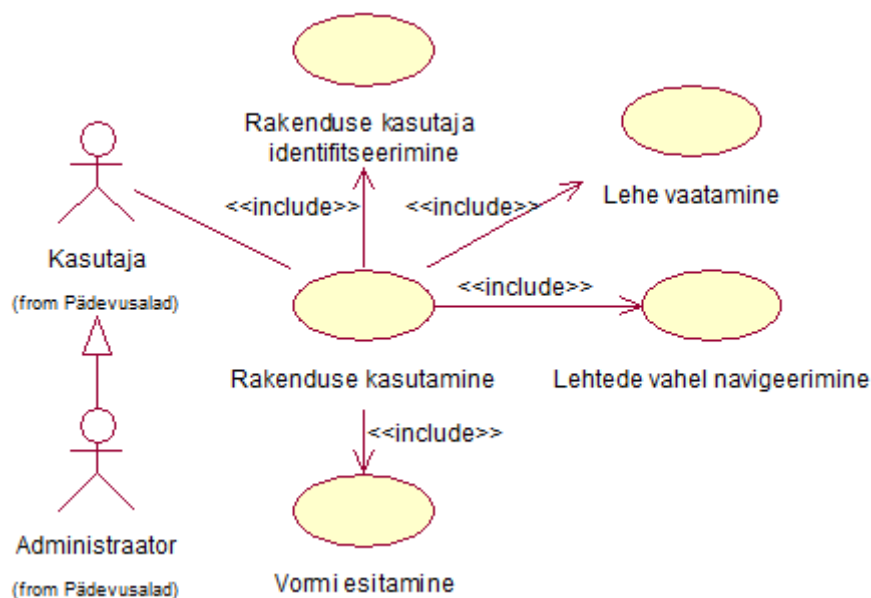
### **3.3.2 Allüsteemi poolt kasutatavad registrid**

Allüsteem ei teeninda ühtegi registrit.

Allüsteem kasutab andmebaasiobjektide registrit, rakenduste registrit, lehtede registrit, regioonide registrit, navigatsioonide registrit, mallide registrit.

### **3.3.3 Allüsteemi kasutusjuhtude eskiismudel**

Järgnevalt on esitatud rakenduse funktsionaalse allüsteemi kasutusjuhtude eskiismudel ja seal esitatud tekstikirjeldused kõrgtaseme formaadis.



Joonis 2. Rakenduse funktsionaalse allsüsteemi kasutusjuhtude eskiismudel

#### **Kasutusjuht: Rakenduse kasutamine**

**Tegutsejad:** Kasutaja

**Kirjeldus:** Kasutaja saab kasutada loodud rakendust.

#### **Kasutusjuht: Rakenduse kasutaja identifitseerimine**

**Tegutsejad:** Kasutaja

**Kirjeldus:** Kui rakenduses kuvatav lehekülg nõuab, et kasutaja on autenditud, siis kuvatakse kasutajale autentimisvorm, kus küsitakse kasutajanime ja parooli. Kui kasutaja poolt sisestatud kasutajanimi ja parool on korrektsed, siis lubatakse kasutajal näha kaitsitud lehekülgi. Kasutaja peab sessiooni jooksul autentima ainult ühe korra.

#### **Kasutusjuht: Lehe vaatamine**

**Tegutsejad:** Kasutaja

**Kirjeldus:** Kasutaja näeb lehel loodud regioonide sisu. Lehel võidakse kuvada navigatsioone, raporteid, vorme ja HTML teksti.

**Kasutusjuht:** Lehtede vahel navigeerimine

**Tegutsejad:** Kasutaja

**Kirjeldus:** Kasutaja saab navigeerida erinevate lehtede vahel vajutades navigatsiooni olevatele linkidele.

**Kasutusjuht:** Vormi esitamine

**Tegutsejad:** Kasutaja

**Kirjeldus:** Kasutajale kuvatakse vorm, mis võib olla juba eelnevalt täidetud. Kasutaja sisestab ormi väljadesse andmed ning saadab need vormi esitamisega töötlemisse.

## **3.4 Rakenduste funktsionaalne allsüsteem**

### **3.4.1 Eesmärgid**

- Võimaldada administraatoril saada ülevaade loodud rakendustest.
- Võimaldada administraatoril luua uus rakendus.
- Võimaldada administraatoril muuta olemasolevate rakenduste seadeid.
- Võimaldada administraatoril kustutada olemasolevaid rakendusi.

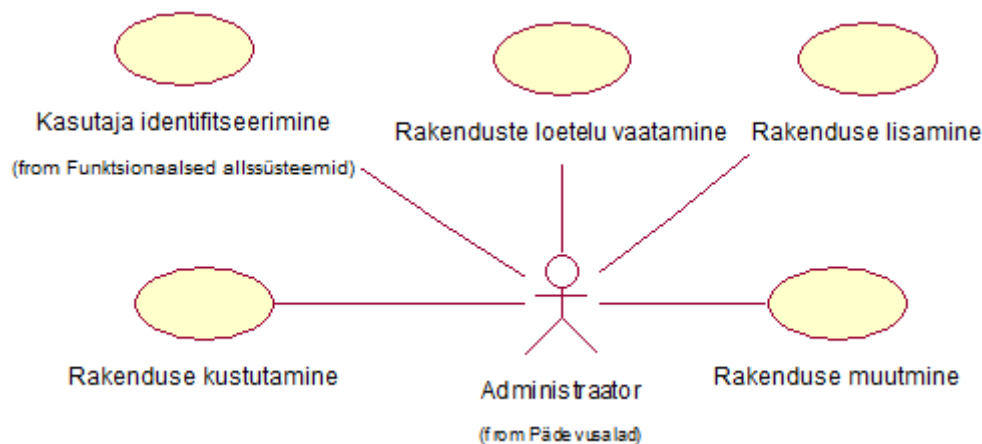
### **3.4.2 Allsüsteemi poolt kasutatavad registrid**

Allsüsteem teenindab rakenduste registrit.

Allsüsteem kasutab andmebaasiobjektide registrit.

### **3.4.3 Allsüsteemi kasutusjuhtude eskiismudel**

Järgnevalt on esitatud rakenduste funktsionaalse allsüsteemi kasutusjuhtude eskiismudel ja seal esitatud tekstikirjeldused kõrgtaseme formaadis.



Joonis 3. Rakenduste funktsionaalse allsüsteemi kasutusjuhtude eskiismudel

#### **Kasutusjuht: Kasutaja identifitseerimine**

**Tegutsejad:** Administraator

**Kirjeldus:** Administraator identifitseerib ennast sisestades kasutajanime ja parooli. Kui sellise kasutajanime ja parooliga kasutaja on andmebaasis olemas ning kasutajal on SUPERUSER õigused, siis lubatakse administraatoril süsteemi siseneda, vastasel juhul mitte.

**Märkus:** Kasutusjuht “Kasutaja identifitseerimine” on kasutusel ka järgnevates allsüsteemides: lehtede funktsionaalne allsüsteem, regioonide funktsionaalne allsüsteem, navigatsioonide funktsionaalne allsüsteem.

#### **Kasutusjuht: Rakenduste loetelu vaatamine**

**Tegutsejad:** Administraator

**Kirjeldus:** Administraator vaatab, mis rakendused on loodud. Süsteem kuvab administraatorile loetelu rakendustest, kus on esitatud rakenduse nimi.

#### **Kasutusjuht: Rakenduse lisamine**

**Tegutsejad:** Administraator

**Kirjeldus:** Administraator valib rakendusele nime, aliase, andmebaasi, mille põhjal rakendus luuakse ning sisestab andmebaasi kasutajanime ja parooli, kellena süsteem andmebaasiga suhtleb. Kui sisestatud andmed on korrektsed ning sellise kasutajanime ja pa-

rooliga kasutaja eksisteerib, siis luuakse uus rakendus.

#### **Kasutusjuht: Rakenduse muutmine**

**Tegutsejad:** Administraator

**Kirjeldus:** Administraator valib rakenduse, mida ta soovib muuta. Administraatorile kuvatakse rakenduse nimi, alias, andmebaas, mille põhjal rakendus on loodud ning andmebaasi kasutajanimi. Administraator saab kuvatud andmeid muuta. Salvestamiseks peab ta sisestama ka andmebaasi kasutajale vastava parooli. Kui sisestatud andmed on korrektsed, siis muudatused salvestatakse.

#### **Kasutusjuht: Rakenduse kustutamine**

**Tegutsejad:** Administraator

**Kirjeldus:** Administraator valib rakenduse, mida ta soovib kustutada. Enne kustutamist küsitakse administraatorilt kinnitust. Kui administraator kinnitab kustutamise, siis rakendus ning sellega seotud info kustutatakse.

### **3.5 Lehtede funktsionaalne allsüsteem**

#### **3.5.1 Eesmärgid**

- Võimaldada administraatoril saada ülevaade rakendusele kuuluvatest lehekülgedest.
- Võimaldada administraatoril luua uusi lehekülgi.
- Võimaldada administraatoril muuta olemasolevate lehekülgede seadeid.
- Võimaldada administraatoril kustutada olemasolevaid lehekülgi.

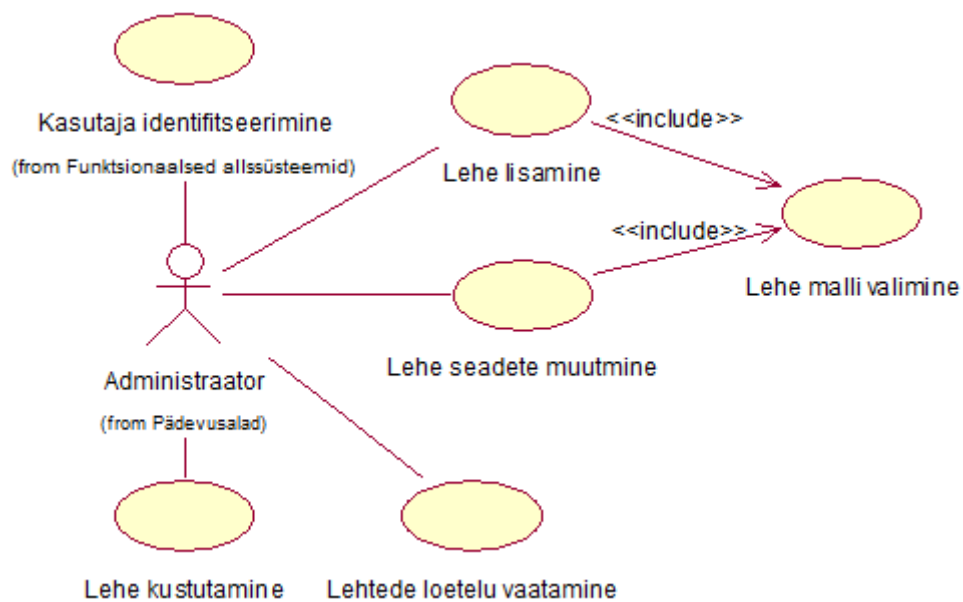
#### **3.5.2 Allsüsteemi poolt kasutatavad registrid**

Allsüsteem teenindab lehtede registrit.

Allsüsteem kasutab mallide registrit.

### 3.5.3 Allsüsteemi kasutusjuhtude eskiismudel

Järgnevalt on esitatud lehtede funktsionaalse allsüsteemi kasutusjuhtude eskiismudel ja seal esitatud tekstikirjeldused kõrgtaseme formaadis.



Joonis 4. Lehtede funktsionaalse allsüsteemi kasutusjuhtude eskiismudel

## 3.6 Regionide funktsionaalne allsüsteem

### 3.6.1 Eesmärgid

- Võimaldada administraatoril saada ülevaade lehel olevatest regionidest.
- Võimaldada administraatoril luua navigatsiooni tüüpi regioone.
- Võimaldada administraatoril luua HTML tüüpi regioone.
- Võimaldada administraatoril luua raporti tüüpi regioone.
- Võimaldada administraatoril luua vormi tüüpi regioone.

- Võimaldada administraatoril muuta olemasolevaid regioone.
- Võimaldada administraatoril kustutada olemasolevaid regioone.

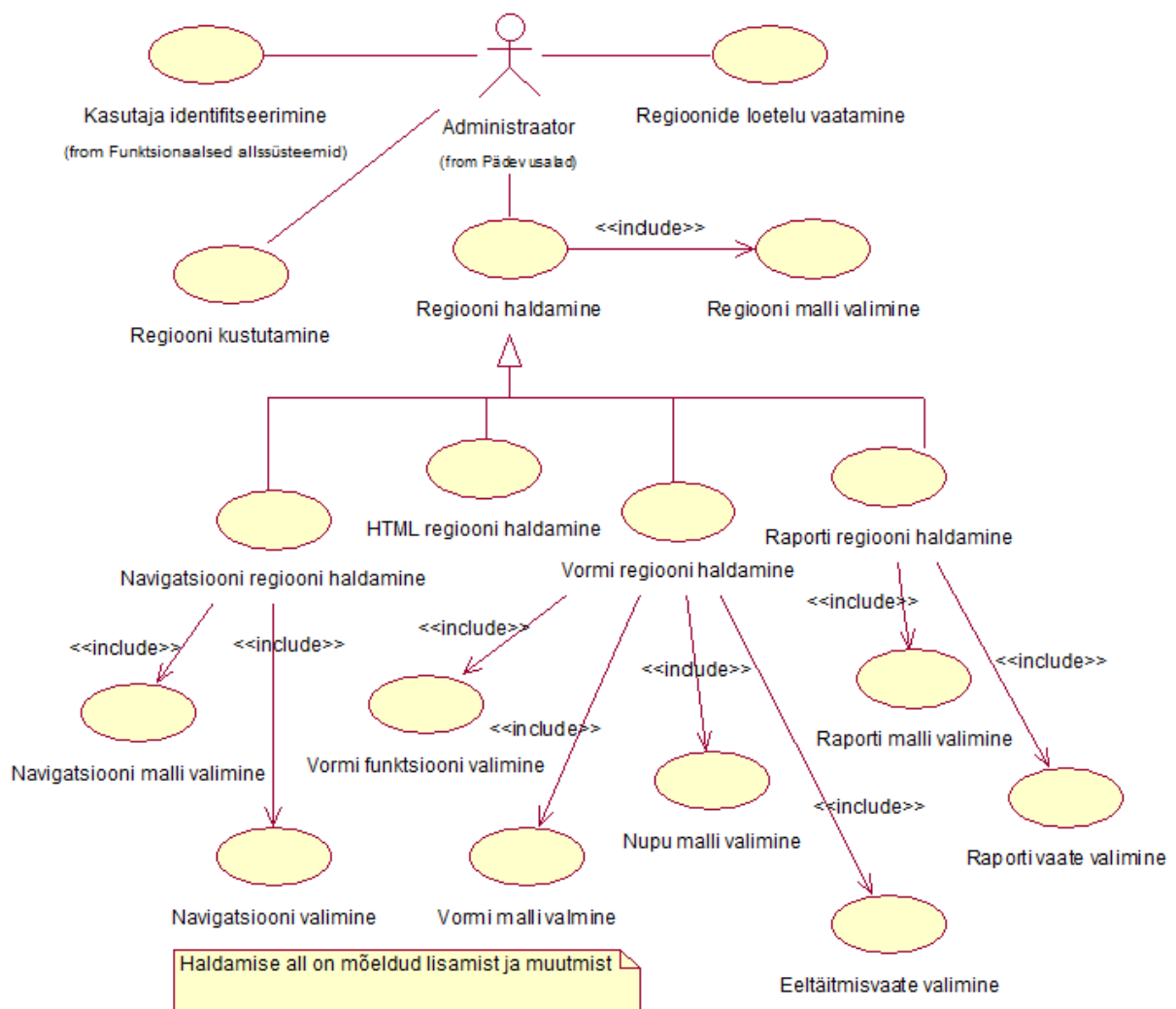
### **3.6.2 Allsüsteemi poolt kasutatavad registrid**

Allsüsteem teenindab regioonide registrit.

Allsüsteem kasutab mallide registrit, navigatsioonide registrit, andmebaasiobjektide registrit.

### **3.6.3 Allsüsteemi kasutusjuhtude eskiismudel**

Järgnevalt on esitatud regioonide funktsionaalse allsüsteemi kasutusjuhtude eskiismudel ja seal esitatud tekstikirjeldused kõrgtaseme formaadis.



Joonis 5. Regioonide funktsionaalse allsüsteemi kasutusjuhtude eskiismudel



## **3.7 Navigatsioonide funktsionaalne allsüsteem**

### **3.7.1 Eesmärgid**

- Võimaldada administraatoril saada ülevaade rakendusele kuuluvatest navigatsioonidest.
- Võimaldada administraatoril luua uusi navigatsioone.
- Võimaldada administraatoril muuta olemasolevate navigatsioonide seadeid.
- Võimaldada administraatoril kustutada olemasolevaid navigatsioone.
- Võimaldada administraatoril lisada olemasoleva navigatsiooni alla navigatsioonipunkte.
- Võimaldada administraatoril muuta olemasolevaid navigatsioonipunkte.
- Võimaldada administraatoril kustutada olemasolevaid navigatsioonipunkte.

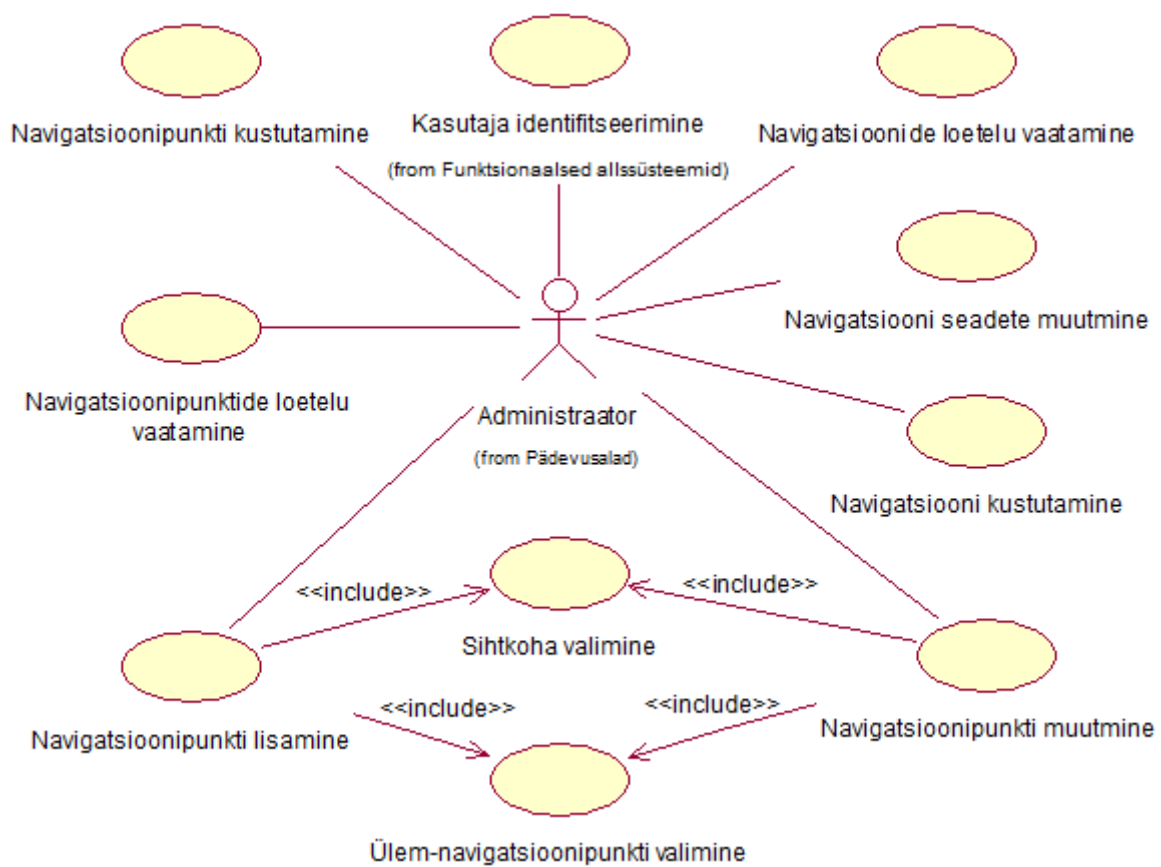
### **3.7.2 Allsüsteemi poolt kasutatavad registrid**

Allsüsteem teenindab navigatsioonide registrit.

Allsüsteem kasutab lehtede registrit.

### **3.7.3 Allsüsteemi kasutusjuhtude eskiismudel**

Järgnevalt on esitatud navigatsioonide funktsionaalse allsüsteemi kasutusjuhtude eskiismudel ja seal esitatud tekstikirjeldused kõrgtaseme formaadis.



Joonis 6. Navigatsioonide funktsionaalse allsüsteemi kasutusjuhtude eskiismudel

### 3.8 Mittefunktsionaalsed nõuded

Tüüp	Nõude kirjeldus
Serveri tarkvara	Andmete hoidmiseks peab kasutama andmebaasisüsteemi PostgreSQL 9.4 või uuemat. Rakendus tuleb luua kasutades PHP 5.5.0 või uuemat.
Keel	Süsteemi kasutajaliides peab olema ingliskeelne.
Kasutajaliides	Kasutajaliides peab olema veebipõhine ning arvestama erinevate resolutsioonidega.
Toetatud veebibrause- rid	<ul style="list-style-type: none"><li>■ Microsoft Internet Explorer 11 või uuem.</li><li>■ Mozilla Firefox 43 või uuem.</li><li>■ Google Chrome 49 või uuem.</li></ul>
Andmebaasioperatsioonide töökoorus	Andmebaasioperatsioonid peavad süsteemil aega võtma alla 5 se- kundi.

Tabel 2. Mittefunktsionaalsed nõuded

## 4 Andmebaasi disain

## 5 Kasutajaliidese disain

## 6 Rakenduse disain

## 7 Näide

## 8 Kokkuvõte

Kokkuvõte

## 9 Summary

Kokkuvõte

## Kasutatud kirjandus

- [1] About the Apache HTTP Server Project - The Apache HTTP Server Project. [WWW] [https://httpd.apache.org/ABOUT\\_APACHE.html](https://httpd.apache.org/ABOUT_APACHE.html). (07.03.2016).
- [2] Ansible is Simple IT Automation. [WWW] <https://www.ansible.com>. (07.03.2016).
- [3] Ben Balter. Open source license usage on GitHub.com. [WWW] <https://github.com/blog/1964-open-source-license-usage-on-github-com>, 2015. (20.02.2016).
- [4] Larry Burns. *Building the Agile Database - How to Build a Successful Application Using Agile Without Sacrificing Data Management*. Technics Publications, LLC, 1 edition, 2011.
- [5] Chef - Code Can I Chef. [WWW] <https://www.chef.io/>. (07.03.2016).
- [6] Docker - Build, Ship, and Run Any App, Anywhere. [WWW] <https://www.docker.com/>. (07.03.2016).
- [7] What is free software? [WWW] <http://www.gnu.org/philosophy/free-sw.html>. (20.02.2016).
- [8] HHVM. [WWW] <http://hhvm.com/>. (07.03.2016).
- [9] Licenses. [WWW] <http://choosealicense.com/licenses/>. (20.02.2016).
- [10] MySQL. [WWW] <https://www.mysql.com/>. (08.03.2016).
- [11] nuBuilder. [WWW] <https://www.nubuilder.net>. (29.02.2016).
- [12] GitHub: nuSoftware/nuBuilderPro: Web Application Builder. [WWW] <https://github.com/nuSoftware/nuBuilderPro>. (29.02.2016).
- [13] The Open Source Definition. [WWW] <https://opensource.org/osd-annotated>. (20.02.2016).
- [14] Oracle Application Express. [WWW] <https://apex.oracle.com/en/>. (20.02.2016).

- [15] Oracle Database. [WWW] <https://www.oracle.com/database/index.html>. (20.02.2016).
- [16] Why Open Source misses the point of Free Software. [WWW] <http://www.gnu.org/philosophy/open-source-misses-the-point.html>. (20.02.2016).
- [17] PHP: Hypertext Preprocessor. [WWW] <http://php.net/>. (20.02.2016).
- [18] PostgreSQL. [WWW] <http://www.postgresql.org/>. (20.02.2016).
- [19] PostgreSQL: About. [WWW] <http://www.postgresql.org/about/>. (07.03.2016).
- [20] PostgreSQL: Documentation: 9.4: dblink. [WWW] <http://www.postgresql.org/docs/9.4/static/contrib-dblink-function.html>. (22.04.2016).
- [21] PostgreSQL: Documentation: 9.4: postgres\_fdw. [WWW] <http://www.postgresql.org/docs/9.4/static/postgres-fdw.html>. (22.04.2016).
- [22] PostgreSQL: Documentation: 9.4: The Information Schema. [WWW] <http://www.postgresql.org/docs/9.4/static/information-schema.html>. (20.02.2016).
- [23] PostgreSQL: Documentation: 9.4: Rules and Privileges. [WWW] <http://www.postgresql.org/docs/9.4/static/rules-privileges.html>. (21.04.2016).
- [24] PostgreSQL: Documentation: 9.4: System Catalogs. [WWW] <http://www.postgresql.org/docs/9.4/static/catalogs.html>. (20.02.2016).
- [25] Puppet Labs: IT Automation Software for System Administrators. [WWW] <https://puppetlabs.com/>. (07.03.2016).
- [26] SaltStack automation for CloudOps, ITops & DevOps at scale. [WWW] <https://saltstack.com/>. (07.03.2016).
- [27] Vagrant - Why Vagrant. [WWW] <https://www.vagrantup.com/docs/why-vagrant/>. (06.03.2016).
- [28] Vallaste - e-Teatmik: IT ja sidetehnika seletav sõnaraamat. [WWW] <http://vallaste.ee/>. (06.03.2016).

- [29] What is PHP? [WWW] <http://php.net/manual/en/intro-what-is.php>. (07.03.2016).
- [30] Xataface | The fastest way to build a front-end for your MySQL Database. [WWW] <http://xataface.com/>. (08.03.2016).
- [31] shannah/xataface: Framework for building data-driven web applications in PHP and MySQL. [WWW] <https://github.com/shannah/xataface>. (08.03.2016).

## **Lisa 1 - [Pealkiri]**