

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 32 leheküljel, 9 peatükki, 1 joonist, 2 tabelit.

Abstract

The thesis is in estonian and contains 32 pages of text, 9 chapters, 1 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 [36]
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

1	Sissejuhatus	10
1.1	Taust ja probleem	10
1.2	Ülesande püstitus	10
1.3	Metoodika	10
1.4	Ülevaade tööst	11
2	Teoreetiline taust	12
2.1	Andmebaasi avalik liides	12
2.1.1	Vaadete kasutamise eelised	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	Uue süsteemi visioon	15
2.4	Eksisteerivate programmide analüüs	15
2.4.1	Oracle Application Express (APEX)	15
2.4.2	NuBuilder	16
2.4.3	Xataface	17
2.5	Täpsustunud ülesande püstitus	17
2.6	Kasutatavad tehnoloogiad	17
2.6.1	Vagrant 1.8.1	17
2.6.2	Apache HTTP server	17
2.6.3	PHP 5.5.3	17
2.6.4	Postgresql 9.4	18
2.6.5	Javascript	18
2.6.6	AngularJS	18
2.6.7	Bootstrap	18
2.7	Litsents	18
2.7.1	Free Software	18
2.7.2	Open Source	19
2.7.3	<i>Free Software</i> ja <i>Open Source</i> võrdlus	20
2.7.4	Litsentsi valik	20
3	Süsteemi analüüs	22
3.1	Tegutsejad	22
3.2	Terviksüsteemi tükeldus allsüsteemideks	22
3.2.1	Pädevusalad	22

3.2.2	Funktsionaalsed allsüsteemid	22
3.2.3	Registrid	22
3.3	Rakenduste funktsionaalne allsüsteem	23
3.3.1	Eesmärgid	23
3.3.2	Allsüsteemi poolt kasutatavad registrid	23
3.4	Lehtede funktsionaalne allsüsteem	23
3.4.1	Eesmärgid	23
3.4.2	Allsüsteemi poolt kasutatavad registrid	23
3.5	Regioonide funktsionaalne allsüsteem	24
3.5.1	Eesmärgid	24
3.5.2	Allsüsteemi poolt kasutatavad registrid	24
3.6	Navigatsioonide funktsionaalne allsüsteem	24
3.6.1	Eesmärgid	24
3.6.2	Allsüsteemi poolt kasutatavad registrid	25
3.7	Mittefunktsionaalsed nõuded	25
4	Andmebaasi disain	26
5	Kasutajaliidese disain	26
6	Rakenduse disain	26
7	Näide	26
8	Kokkuvõte	26
9	Summary	27
	Kasutatud kirjandus	28
	Lisa 1	31

Jooniste loetelu

1	Andmebaasi avalik liides	12
---	------------------------------------	----

Tabelite loetelu

1	Litsentside võrdlus	21
2	Mittefunktsionaalsed nõuded	25

1 Sissejuhatus

1.1 Taust ja probleem

TTÜ-s õpetatava aine Ändmebaasid 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 [25] ja Oracle [21]. Juhul kui andmebaas on loodud Oracle andmebaasisüsteemi abil, siis on üliõpilastel rakenduse loomiseks võimalus kasutada Oracle APEX-it [20]. PostgreSQL andmebaasisüsteemiga loodud andmebaasi korral tuleb rakendus programmeerida kasutades PHP-d [23]. See tähendab, et üliõpilane ei saa keskenduda täielikult andmebaasi täiustamisele vaid peab tegelema ka lisaprogrammeerimisega, mis olevalt eelnevast kogemusest võib võtta rohekm või vähem aega. Töö tulemusena valmiva prototüübi abil peaks üliõpilastel olema lihtsam luua näidisrakendusi, mis kasutavad andmebaasisüsteemina PostgreSQL-i.

Töö valmis 2016. aasta kevadel Tallinna Tehnikaülikoolis.

1.2 Ülesande püstitus

Antud töö eesmärgiks on uurida, kuidas luua PostgreSQL andmebaasi põhjal meta-andmetega juhutavat veebipõhist rakenduste kiirprogrammeerimiskeskonda ning see realiseerida. Loodav süsteem peab toetama PostgreSQL 9.4 ja PHP 5.5.0. Töö tulem avaldatakse avatud lähtekoodi litsentsi all.

Loodava süsteemi abil peavad kasutajad saama valida, millise andmebaasi põhjal nad soovivad enda rakenduse luua. Kasutades andmebaasi avalikku liidest peab olema võimalik luua lehekülgi, lisada neile raporteid, vorme ning navigeerida erinevate lehtede vahel. Sisselogimine peab olema võimalik, kasutades sama kasutajanime ja parooli, mis on kasutusel andmebaasisüsteemis.

1.3 Metoodika

Süsteemi realiseerimiseks kasutatakse serveri poole pealt andmebaasisüsteemi PostgreSQL 9.4 [25], programmeerimiskeelt PHP 5.5.0 [23], PHP raamistikku Slim 3.3.0 [32]. Klien-

di poole realiseerimiseks kasutatakse lisaks HTML-le ja CSS-le ka AngularJS 1.4 [2] ja Bootstrap 3 [5]. Selleks, et loodud süsteemi oleks ka teistel lihtne üles seada ning kasutada, loon Vagrant-i [34] abil virtuaalmasina, kus on eelnevalt installitud kogu vajalik tarkvara süsteemi arendamiseks ja testimiseks. Javascripti koodi testimiseks kasutan Karma 0.13 [13] ja Jasmine 2.3 [12]. PHP koodi testimiseks kasutan PHPUnit 4.8 [24] ja Mockery 0.9 [15]. Testide automaatseks jooksumiseks ning koodi juurutamiseks (*deploy*) kasutan TravisCI [33] keskkonda. Koodi ennast hoitakse GitHub-i [9] koodihoidlas.

1.4 Ülevaade tööst

TODO

kasutatakse kuritahtlikult valitud funktsioone ja operaatoreid, et näha varjatud infot [29]

- 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.

[6, 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.

[6, 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.[27] 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 kohati edestada jõudluse poolest. Küll aga pole võimalik välja kutsuda välises andmebaasis olevaid funktsioone. [28]

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 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.

2.4 Eksisteerivate programmide analüüs

2.4.1 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 pärngkuid 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.

[20]

2.4.2 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 andmabaasisüsteemis. 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 [18], 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.

[17]

2.4.3 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.[38]

Xataface on avatud lähtekoodiga ning antud välja GPL litsentsi all. Programm on kirjutatud PHP-s [23] ning andmebaasina kasutatakse MySQL-i [16].

Kasutatud on palju väliseid teeke. Programmil on üks põhiline arendaja ning igapäevast arendustööd ei toimu. [39]

2.5 Täpsustunud ülesande püstitus

2.6 Kasutatavad tehnoloogiad

2.6.1 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 [3], *Puppet*-it [30], *Chef*-i [7], *Docker*-it [8] ja *Salt*-i [31]. Tänu Vagrantile saavad kõik luua endale täpselt ühesuguse keskkonna, kus programme jooksutada, vähendades võimalust, et ühes arvutis programm jookseb, teises aga mitte. [35]

2.6.2 Apache HTTP server

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

2.6.3 PHP 5.5.3

PHP (*PHP: Hypertext Preprocessor*) on avatud lähtekoodiga skriptimiskeel, mis on peamiselt mõeldud veebiprogrammeerimiseks. [37] PHP koodi protsessitakse PHP interpretaatori abil. Üldjuhul kasutatakse interpreteerimiseks *Zend Engine*-t, kuid PHP-d võimalik jooksutada kasutada ka *HHVM*-i [11] abil. PHP toetab erinevaid operatsioonisüsteeme,

sealhulgas Windows-i erinevaid versioone ja Linuxi erinevaid distributsioone.

2.6.4 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. [26]

2.6.5 Javascript

2.6.6 AngularJS

2.6.7 Bootstrap

2.7 Litsents

2.7.1 Free Software

Free Software (Vaba tarkvara) tähendab, et kasutajatel on vabadus 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 tasuta 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. [10]

2.7.2 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 - Programmile 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.

[19]

2.7.3 *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. [22]

- 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.

2.7.4 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%). [4]

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 märg	Kaubanduslik kasutamine Jagamine Muutmine Privaatne kasutamine	Võtta vastutusele
Apache License 2.0	Litsents ja copyright märg Teavitus muudatustest	Kaubanduslik kasutamine Jagamine Muutmine Patendi kasutamine Privaatne kasutamine	Võtta vastutusele Kasutada kaubamärki
GNU GPLv3 GNU GPLv2	Lähtekoodi avaldamine Litsents ja copyright märg Sama litsents Teavitus muudatustest	Kaubanduslik kasutamine Jagamine Muutmine Patendi kasutamine Privaatne kasutamine	Võtta vastutusele

Tabel 1. Litsentside võrdlus
[14]

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

3 Süsteemi analüüs

3.1 Tegutsejad

- Administraator.

3.2 Terviksüsteemi tükeldus allsüsteemideks

3.2.1 Pädevusalad

- Administraatori pädevusala.

Antud pädevusala kasutab kõiki allsüsteeme.

3.2.2 Funktsionaalsed allsüsteemid

- 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 Rakenduste funktsionaalne allsüsteem

3.3.1 Eesmärgid

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

3.3.2 Allsüsteemi poolt kasutatavad registrid

Allsüsteem teenindab rakenduste registrit.

Allsüsteem kasutab andmebaasiobjektide registrit.

3.4 Lehtede funktsionaalne allsüsteem

3.4.1 Eesmärgid

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

3.4.2 Allsüsteemi poolt kasutatavad registrid

Allsüsteem teenindab lehtede registrit.

Allsüsteem kasutab mallide registrit.

3.5 Regioonide funktsionaalne allsüsteem

3.5.1 Eesmärgid

- Võimaldada administraatoril saada ülevaade lehel olevatest regioonidest.
- 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.

3.5.2 Allsüsteemi poolt kasutatavad registrid

Allsüsteem teenindab regioonide registrit.

Allsüsteem kasutab mallide registrit, navigatsioonide registrit, andmebaasiobjektide registrit.

3.6 Navigatsioonide funktsionaalne allsüsteem

3.6.1 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 olemas olevaid navigatsioonipunkte.

3.6.2 Allsüsteemi poolt kasutatavad registrid

Allsüsteem teenindab navigatsioonide registrit.

Allsüsteem kasutab lehtede registrit.

3.7 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 veebibrauserid	<ul style="list-style-type: none"> ■ Microsoft Internet Explorer 11 või uuem. ■ Mozilla Firefox 43 või uuem. ■ Google Chrome 49 või uuem.
Andmebaasioperatsioonide töökoorus	Andmebaasioperatsioonid peavad süsteemil aega võtma alla 5 sekundi.

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. (07.03.2016).
- [2] AngularJS. [WWW] <https://angularjs.org/>. (20.02.2016).
- [3] Ansible is Simple IT Automation. [WWW] <https://www.ansible.com>. (07.03.2016).
- [4] 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).
- [5] Bootstrap. [WWW] <http://getbootstrap.com/>. (20.02.2016).
- [6] Larry Burns. *Building the Agile Database - How to Build a Successful Application Using Agile Without Sacrificing Data Management*. Technics Publications, LLC, 1 edition, 2011.
- [7] Chef - Code Can I Chef. [WWW] <https://www.chef.io/>. (07.03.2016).
- [8] Docker - Build, Ship, and Run Any App, Anywhere. [WWW] <https://www.docker.com/>. (07.03.2016).
- [9] GitHub. [WWW] <https://github.com/>. (21.04.2016).
- [10] What is free software? [WWW] <http://www.gnu.org/philosophy/free-sw.html>. (20.02.2016).
- [11] HHVM. [WWW] <http://hhvm.com/>. (07.03.2016).
- [12] Jasmine: Behavior-Driven JavaScript. [WWW] <http://jasmine.github.io/>. (21.04.2016).
- [13] Karma - Spectacular Test Runner for Javascript. [WWW] <https://karma-runner.github.io/0.13/index.html/>. (21.04.2016).
- [14] Licenses. [WWW] <http://choosealicense.com/licenses/>. (20.02.2016).
- [15] Mockery. [WWW] <http://docs.mockery.io/en/latest/>. (21.04.2016).

- [16] MySQL. [WWW] <https://www.mysql.com/>. (08.03.2016).
- [17] nuBuilder. [WWW] <https://www.nubuilder.net>. (29.02.2016).
- [18] GitHub: nuSoftware/nuBuilderPro: Web Application Builder. [WWW] <https://github.com/nuSoftware/nuBuilderPro>. (29.02.2016).
- [19] The Open Source Definition. [WWW] <https://opensource.org/osd-annotated>. (20.02.2016).
- [20] Oracle Application Express. [WWW] <https://apex.oracle.com/en/>. (20.02.2016).
- [21] Oracle Database. [WWW] <https://www.oracle.com/database/index.html>. (20.02.2016).
- [22] Why Open Source misses the point of Free Software. [WWW] <http://www.gnu.org/philosophy/open-source-misses-the-point.html>. (20.02.2016).
- [23] PHP: Hypertext Preprocessor. [WWW] <http://php.net/>. (20.02.2016).
- [24] PHPUnit - The PHP Testing Framework. [WWW] <https://phpunit.de/>. (21.04.2016).
- [25] PostgreSQL. [WWW] <http://www.postgresql.org/>. (20.02.2016).
- [26] PostgreSQL: About. [WWW] <http://www.postgresql.org/about/>. (07.03.2016).
- [27] PostgreSQL: Documentation: 9.4: dblink. [WWW] <http://www.postgresql.org/docs/9.4/static/contrib-dblink-function.html>. (22.04.2016).
- [28] PostgreSQL: Documentation: 9.4: postgres_fdw. [WWW] <http://www.postgresql.org/docs/9.4/static/postgres-fdw.html>. (22.04.2016).
- [29] PostgreSQL: Documentation: 9.4: Rules and Privileges. [WWW] <http://www.postgresql.org/docs/9.4/static/rules-privileges.html>. (21.04.2016).
- [30] Puppet Labs: IT Automation Software for System Administrators. [WWW] <https://puppetlabs.com/>. (07.03.2016).

- [31] SaltStack automation for CloudOps, ITops & DevOps at scale. [WWW] <https://saltstack.com/>. (07.03.2016).
- [32] Slim Framework. [WWW] <http://www.slimframework.com/>. (21.04.2016).
- [33] Travis CI - Test and Deploy Your Code with Confidence. [WWW] <https://travis-ci.org/>. (21.04.2016).
- [34] Vagrant. [WWW] <https://www.vagrantup.com/>. (06.03.2016).
- [35] Vagrant - Why Vagrant. [WWW] <https://www.vagrantup.com/docs/why-vagrant/>. (06.03.2016).
- [36] Vallaste - e-Teatmik: IT ja sidetehnika seletav sõnaraamat. [WWW] <http://vallaste.ee/>. (06.03.2016).
- [37] What is PHP? [WWW] <http://php.net/manual/en/intro-what-is.php>. (07.03.2016).
- [38] Xataface | The fastest way to build a front-end for your MySQL Database. [WWW] <http://xataface.com/>. (08.03.2016).
- [39] 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]