

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutisüsteemide instituut

IAG40LT

Reijo Olavi Komu 143086IASB

**MULTIFUNKTSIONAALNE OTSING
TOIDURETSEPTIDE ANDMEBAASI
NÄITEL**

Bakalaureusetöö

Juhendaja: Vladimir Viies

Doktorikraad

Dotsent

Tallinn 2017

Autorideklaratsioon

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

Autor: Reijo Olavi Komu

13.05.2017

Annotatsioon

Lõputöös uurin multifunktsionaalse otsingu realiseerimise võimalust. Prototüübiks on loodud toiduretseptide andmebaas koos multifunktsionaalse otsingu võimalusega. Lõputöö sissejuhatavas osas vaatlen, mida tähendab multifunktsionaalne otsing. Samuti annan ülevaate hulgateooria ajaloo kohta, kuidas antud matemaatiline teooria alguse sai ning kes oli selle rajajaks. Töös toon välja hulgateooria põhimõisted, milleks on : alamhulk, ühend, ühisosa, vahe, hulk ja tühihulk ning kuidas saaks seda kasutada multifunktsionaalse otsingu realiseerimiseks. Töö teises osas sean teatud nõuded ja piirangud multifunktsionaalse otsingu realiseerimiseks ning annan ülevaate tehnoloogiatest, mida saab selle realiseerimiseks kasutada, et süsteem oleks kasutaja sõbralikum. Lõputöö viimases osas analüüsin loodud prototüüpi ning selgitan, kuidas antud multifunktsionaalne otsing realiseeriti.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 32 leheküljel, 3 peatükki, 19 joonist, 4 tabelit.

Abstract

Multifunctional search using food recipe database

In this thesis I give an over view how to construct multifunctional search using set theory and what could be the resources for creating one.

In the instruction part of the thesis I give an overview what does multifunctional search mean. Also, this part explores the history of the set theory and who was the main reason that set theory even exists today and what were the difficulties of that branch of mathematics to come to existence. I also define the main operations like: set,subset,empty set,union,intersection that are need to create a multifunctional search.

In the second part of the thesis I set some requirements and limitations to multifunctional search that are required to implement it. Also I survey the technologies that were used and give some overview of those technologies and what are they mainly used for in computer science.

In the final part of the thesis I analyse the prototype that was created and give on overview how multifunctional search uses set theory. Also I analyse the reason why that type of a database was used and how big is the given database that is used to construct the multifunctional search.

The thesis is in Estonian and contains 32 pages of text, 3 chapters, 19 figures, 4 tables.

Lühendite ja mõistete sõnastik

ASI	TTÜ Arvutisüsteemide instituut
DPI	Dots per inch, punkti tolli kohta
CSS	Cascading Style Sheets,Dokumendivälisilmekirjeldamiseformaalkeel [1]
HTML	HyperText Markup Language,hüperteksti märgistus keel [1]
PHP	PHP: Hypertext Preprocessor,hüpertekstieelprotsessor [1]
SQL	Structured Query Language,struktureeritudpäringute keel [1]
MYSQL	Relatsioonbassihaldur [1]

Sisukord

Sissejuhatus	9
1 Multifunktsioonalne otsing.....	10
1.1 Hulgateooria ajalugu.....	10
1.2 Hulgateooria rakendamine multifunktsionaalses otsingus	12
2 Toiduretseptide andmebaasi rakendus.....	18
2.1 Rakenduse nõuded	18
2.2 Tehnoloogiate valik	19
2.2.1 CSS	19
2.2.2 JQUERY	19
2.2.3 MYSQL	19
2.2.4 PHP.....	20
3 Prototüüp	21
3.1 Registreerimine.....	22
3.2 Logimine.....	23
3.3 Sobivate toiduainete lisamine otsingusse	24
3.4 Mittesoovitavate toiduainete lisamine otsingusse	25
3.5 Otsingu teostamine	26
Kokkuvõte	29
Kasutatud kirjandus	30
LISA 1	31
LISA 2	32

Jooniste loetelu

Joonis 1.Hulk A hulga B alamhulk.....	12
Joonis 2.Hulga A ja B ühisosa.....	13
Joonis 3. Hulga A ja B vahe.	13
Joonis 4.Hulga A ja B ühend.	14
Joonis 5.Toiduainete retseptide ühisosad	15
Joonis 6.Sobivate ja mitte sobivate retseptide vahe	16
Joonis 7.Hulga B ja A vahe	17
Joonis 8.Hulk A hulga B alamhulk.....	17
Joonis 9.Prototüübi struktuurskeem	21
Joonis 10.Registreerimine	22
Joonis 11.Rakenduse Registreerimine	22
Joonis 12.Logimine esivaade.....	23
Joonis 13.Logimine	23
Joonis 14.Toiduainete lisamine	24
Joonis 15.Ebameeldivate toiduainete lisamine ja eemaldamine.....	25
Joonis 16.Andmebaasi mudel.	26
Joonis 17. Otsinguriba	26
Joonis 18. Otsingu algoritimi vooskeem	27
Joonis 19.Otsingu tulemuste kuvamine	28

Tabelite loetelu

Tabel 1.Toiduainete (Hulgad) ja retspide (Hulgad) seosed.....	14
Tabel 2.Retspide ja mitte sobivate retseptide seos.	15
Tabel 3.Hulga A ja B ühend.	16
Tabel 4.Hulga B alamhulgad	17

Sissejuhatus

Elame modernses ühiskonnas, kus elu tempo on kiire ning aega info otsimiseks on vähe. Antud töö autoril tekkiski idee luua multifunktsionaalne otsing, mis kasutab oma andmetena toiduretsepe, et aidata inimestel toituda mitmekesisemalt.

Lõputöö esimeses osas analüüsin, mis on multifunktsionaalne otsing ning kuidas oleks seda võimalik realiseerida tuginedes hulgateooriale. Hulgateooria on matemaatika haru, mis uurib hulkade üldisi omadusi, samuti järjestusi ja muid seoseid. Töös toon välja lühiülevaate hulgateooria rajajast Georg Cantor ning tema publikatsioonidest.

Töö teises osas paneb autor paika protüübi nõuded ja piirangud ning annab ülevaate vahenditest, mida kasutatakse antud töös prototüübi loomiseks. Lisaks annab autor kasutatud tehnoloogiatest ja teekidest ning milleks neid kasutatakse.

Kolmandas osas tuuakse välja prototüübi põhi funktsionaalsus ning antakse ülevaade, kuidas on realiseeritud multifunktsionaalne otsing ning, missuguseid funktsioone kasutati, et emuleerida hulgateooria loogikat, et saada soovitud tulemus. Samuti annan ülevaate prototüüpimisel loodud andmebaasist ning miks on just seda tüüpi andmebaas valitud multifunktsionaalse otsingu realiseerimiseks.

1 Multifunktsioonalne otsing

Multifunktsionaalset otsingut defineerin kui otsingut, mis on võimeline täitma mitut ülesannet. Antud töös on ülesanneteks kihistatud otsing ja liitotsing.

Kihistatud otsing kujutab endas oskust kasutada otsingu tulemuse vastes leiduva informatsiooni uue otsingu teostamiseks ilma kasutaja vahepealse sekkumiseta.

Liitotsing on otsingu tüüp, mille käigus on võimalik piirata või täpsustada otsingu parameetreid enne kui otsing sooritatakse. Liitotsinguga on võimalik täpsemini leida andmebaasist soovitud informatsioon.

1.1 Hulgateooria ajalugu

Hulgateooria kui matemaatika haru alguseks loetakse G.Cantori publikatsiooni "Ueber eine Eigenschaft des Inbegriffes aller reellen algebraischen Zahlen", mis avaldati aastal 1874, kus ta väidab, et algebrailiste reaalarvude arvude hulga elemendid saab panna vastavusse positiivsete täisarvude hulga elementidega. Samuti näitas Cantor, et algebrailiste reaalarvude hulk on võimalik esitada lõpmatu arvudejadana, kus iga hulga element esitub vaid ühe korra. [2]

Oma järgmises publikatsioonis, mille avaldamisega Cantoril oli probleeme, tutvustas ta ideed, et kaks hulka saavad olla võrdsed ehk sama võimsusega, kui nad on võimalik panna üksühesesse vastavausse teineteisega. Ta suutis tõestada, et ratsionaalarvudel on väiksem lõpmatu võimsus ja \mathbf{R}^n on sama võimsusega, kui \mathbf{R} [3], [4].

Hulgateooria arengule oli ohuks Leopold Kronecker, kes oli üks mõjukamaid ja austatumaid matemaatikuid sellel ajastul. Kroneckeri arvates oli Cantori töö aja raiskamine, sest Kronecker aksepteeris ainult matemaatilisi objekte, mis tuginesid naturaalarvudel kuna ta ei näinud mõttet näha vaeva irratsionaalarvudega kuna neid ei eksisteeri. Tänu Cantori sihikindlusele õnnestus tal avaldada veel publikatsioone, kus ta analüüsis täielik järjestus hulkasid ning tutvustas ordinaal arve, kui elemendi indekseid täielikult järjestatud hulkade jadas [2], [4].

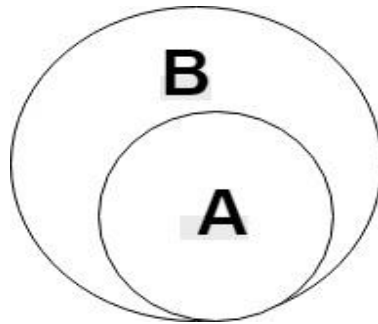
Viimastes publikatsioonides arendas ta edasi teooriat kardinaalarvude ja ordinaalarvude kohta. Tema teooria hulga tüüpide kohta muutus, varem defineeritud ordinaalarvud muutusid nüüd erijuhuks. Oma viimastes publikatsioonides hulgateooria kohta, mis ta avaldas 1895 ja 1897. Andis ta selged definitsioonid kõikidele hulgateooria põhimõistetele ning samuti tõestas ta, et kui A ja B on hulgad ning A on võrdne B alamhulgaga ja B on võrdne A alamhulgaga, siis A ja B on samuti omavahel võrdsed hulgad [2].

1.2 Hulgateooria rakendamine multifunktsionaalses otsingus

Hulgateooriat rakendades on võimalik luua multifunktsionaalne otsing. Selle rakendamiseks tuleb teada, mis on: hulk, osahulk ja peamisi tehteid hulkadega: ühisosa, ühend, vahe.

Hulk on selline omavahel erinevate objektide kogu, millest saab mõelda kui tervikust. hulka kuuluvad objektid on omavahel erinevad. Teine oluline tunnus seisneb selles, et mistahes objekti korral peab olema võimalik üheselt otsustada, kas ta kuulub vaadeldavasse hulka või mitte. Objekte, mis moodustavad hulga (kuuluvad hulka), nimetatakse hulga elementideks [5]. Hulka, milles ei esine mitte ühtegi elementi, nimetatakse tühjaks hulgaks ja tähistatakse sümboliga \emptyset [6].

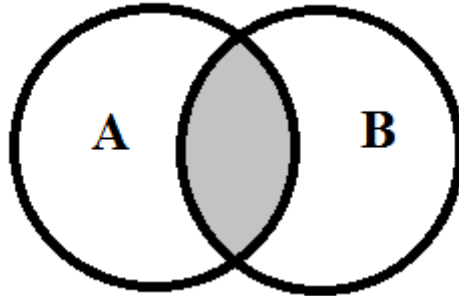
Osahulgaks ehk alamhulgaks kutsutakse hulka A, siis kui kõik hulga A elemendid on hulga B elementideks [7]. Seda tähistakse valemiga $A \subseteq B$. Hulga esitamisel tema elementide loeteluna pole elementide loetlemise järjekord oluline [8]. (vt Joonis 1).



Joonis 1. Hulk A hulga B alamhulk

Kahe hulga A ja B ühisosaks, kutsutakse seda kui hulga A elemendid kuuluvad samal ajal ka hulka B [7] (vt Joonis 2). Ühisosa tähistatakse $A \cap B$ ning see esitub samuti kujul (1).

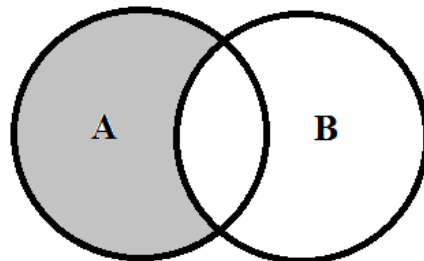
$$A \cap B = \{ a \mid (a \in A) \& (a \in B) \}. \quad [6] \quad (1)$$



Joonis 2. Hulga A ja B ühisosa

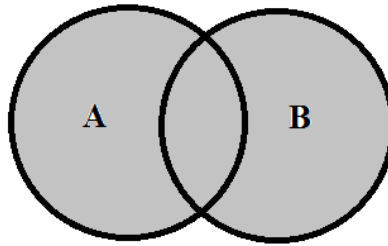
Hulkade vaheks nimetatakse seda, kui elemendid, mis kuuluvad hulka A ja samal ajal ei kuulu hulka B [7] (vt Joonis 3). Seda tähistatakse kujul (2).

$$A \setminus B = \{ a \mid (a \in A) \& (a \notin B) \}. \quad [7] \quad (2)$$



Joonis 3. Hulga A ja B vahe.

Kahe hulga A ja B ühendiks nimetatakse seda, kui elemendid kuuluvad hulka A või hulka B [7](vtJoonis 4). Hulga A ning B ühendit tähistatakse sümboliga $A \cup B$ [6].



Joonis 4.Hulga A ja B ühend.

Selleks, et hulgateooriat rakendada peab eksisteerima hulk, milles on elemendid. Tühja hulgaga pole võimalik otsingut teostada.

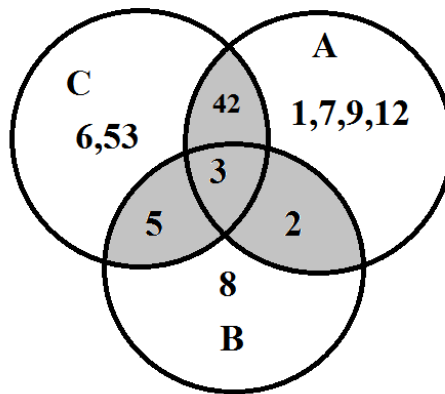
Antud töös on hulkade nimedeks toiduainete nimed ning nende hulkade elementideks on retseptide numbrid kuhu antud toiduained kuuluvad. [vt Tabel 1]

Tabel 1.Toiduainete (Hulgad) ja retspide (Hulgad) seosed.

Komponendi nimetus	Retseptid
Munad	1,2,7,9,12,3,42
Juust	5,3,2,8
Piim	3,6,53,5,42

Selleks, et määratleda milline hulga element on kõige levinum tuleb kasutada hulkade ühisosa. Seda kasutades on võimalik välja selgitada, missuguse retsepti jaoks on olemas kõige rohkem komponente (vt Joonis 5). Joonis valemi kujul (3).

$$(A \cap B) \cup (A \cap C) \cup (C \cap B). \tag{3}$$



Joonis 5. Toiduainete retseptide ühisosad

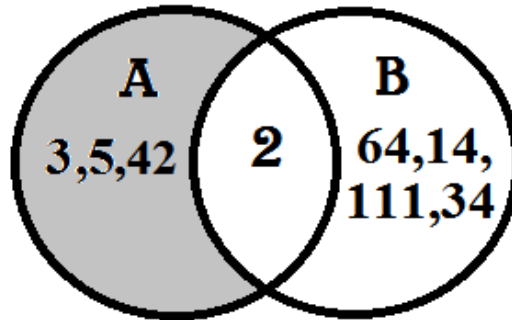
Rakendades hulgaaritmeetilist tehet vahe on võimalik filtreerida varem leitud retseptide hulgast A välja mitte sobivad retseptid. Eksiteerib hulk B, kuhu on ära määratletud kõik retsepti numbrid, mis tuleks eemaldada. Hulga A elemendid on varem leitud ühisosa retseptid (vt Tabel 2).

Tabel 2. Retspide ja mitte sobivate retseptide seos.

Hulgad	Retseptide numbrid
Ühisosade retseptid (A)	$a = \{3, 5, 2, 4\}$
Mitte sobivad retseptid (B)	2, 6, 14, 34, 111

Leides hulga A ja B vahe on näha, et sobivateks retseptideks osutuvad ainult retseptid numbriga 3,5 ja 42. (vt Joonis 6). Valemi kujul (4).

$$A \setminus B = \{ a \mid (a \in A) \& (a \notin B) \}. \quad (4)$$



Joonis 6. Sobivate ja mitte sobivate retseptide vahe

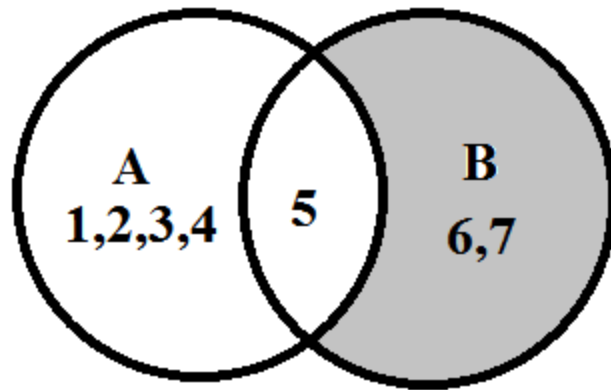
Samuti on võimalik kontrollida kihistatud otsinguga ega ükski retsepti komponent ise ei eksisteeri retseptina. Antud olukorras kui eksiteerib komponent ka resteptina ning retsept pole eelneva retsepti pärisalamhulk tuleb kontrollida nende hulkadkomponentide vahe (5).

$$B \setminus A = \{ a \mid (a \in B) \& (a \notin A) \} \quad (5)$$

Vahe sisu võib sisaldada keelatud aineid peidetud kujul, mida ilma kihistatud otsinguta poleks olnud võimalik tuvastada. Allesjäänud vahe hulgal ei tohi olla ühiosa keelatud komponentidega. Hulk A on esmane retsept koos oma komponentidega. Hulk B on üks komponent, mis osutuks retseptiks ning mille komponendid polnud hulga A pärisalamhulga osa. (vt Tabel 3) (vt Joonis 7).

Tabel 3. Hulga A ja B ühend.

Komponendi hulgad	Komponentide väärtused
A	1,2,3,4,5
B	5,6,7
B \setminus A	6,7

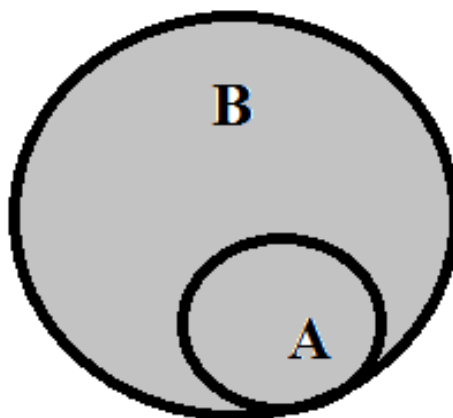


Joonis 7. Hulga B ja A vahe

Alamhulki vaadeldes saame alternatiivselt määratleda, kas kihistatud otsingu tulemus on sobilik. Selleks tuleb vaadelda retseptide numbreid juhul, kui $A \subseteq B$ on tegemist mittesobiliku retseptiga. Hulga B elementideks on mittesobilike retseptide hulk ning A hulga jaoks on retsept, mida vaadeldakse. (vt Tabel 4) (vt Joonis 8)

Tabel 4. Hulga B alamhulgad

Hulgad	Retseptide numbrid
A	1
B	1, 2, 3, 4



Joonis 8. Hulk A hulga B alamhulk

2 Toiduretseptide andmebaasi rakendus

Antud töö esimeses pooles autor analüüsis, kuidas oleks võimalik rakendada hulgateooriat multifunktsionaalse otsingu realiseerimiseks.

Käesolevas peatükis autor seab nõuded loodavale rakendusele ja toob välja kasutatud vahendid rakenduse loomiseks.

2.1 Rakenduse nõuded

Rakenduse loomiseks on autor seadnud teatud nõuded.

- Rakenduse kasutajal peab olema võimalus registreerida süsteemi.
- Rakenduse kõigi funktsionaalsuse kasutamiseks peab olema registreeritud kasutaja.
- Otsing peab rakendama seoseid $A \setminus B$, $A \cup B$, $A \cap B$.
- Kasutaja peab saama lisada ja eemaldada endale meeldivaid komponente.
- Peab eksisteerima algne kasutajaliidese disain.
- Võimalus otsida retsepte ilma registreerimata kasutades komponente.
- Kasutaja peab saama lisada ja eemaldada oma kasutaja konto alla toiduained, mis talle ei meeldi.
- Kasutaja peab saama oma kontolt välja logida.
- Kasutajale tuleb kuvada retseptse, mis eksisteerivad teiste retseptides kuni sügavuseni kaks.
- Erineval sügavusel eksisteerivad retseptid peavad olema eristatavad.
- Kasutaja peab saama sisselogida süsteemi.

2.2 Tehnoloogiate valik

Antud peatükis autor toob välja töös kasutatud tehnoloogiad. Milleks on CSS,JQUERY,MYSQL,PHP. Tehnoloogiate valikut mõjutas ,kui kiiresti on võimalik antud tehnoloogiat omastada, antud tehnoloogia paindlikkus ja prototüübi väljatöötamise hõlbustamine.

2.2.1 CSS

CSS ehk Cascading Style Sheets on kujunduse keel,mis võimaldab HTMLi elemente paremini visualiseerida. Samuti teeb CSSi kasutamine veebileheküljed paremini hoovatavamaks ning võimaldab veebisaitidel olla paindlikum [9]. CSSi üheks tugevaimaks küljeks on kuvada informatsiooni erinevalt, sõltuvalt kasutaja kuvari resolutsioonist.CSSi süntaks tugineb tugevasti inglise keelele, mille tõttu on omastamine üsnagi lihtne [10].

2.2.2 JQUERY

JQuery on platvormist sõltumatu Javascripti teek,mis hõlbustab kliendi poolset HTMLi modifitseerimist. JQuery süntaks on disainitud viisil,mis hõlbustaks dokumentides navigeerimist,animatsioonide loomist ja sündmuste haldamist [11], [12]. JQuery võimaldab arendajal luua lisa tarkvaramooduleid,mis toetuvad Javascripti teegile [12]. Tänu millele on võimalik luua suures hulgas lisa funktsionaalsust ilma, et tuleks implementeerida teisi programmeerimis keeli. JQuery on kasutusel 72.3% kõikidest eksisteerivatest veebilehtedest seisuga 1.mai.2017 [13].

2.2.3 MYSQL

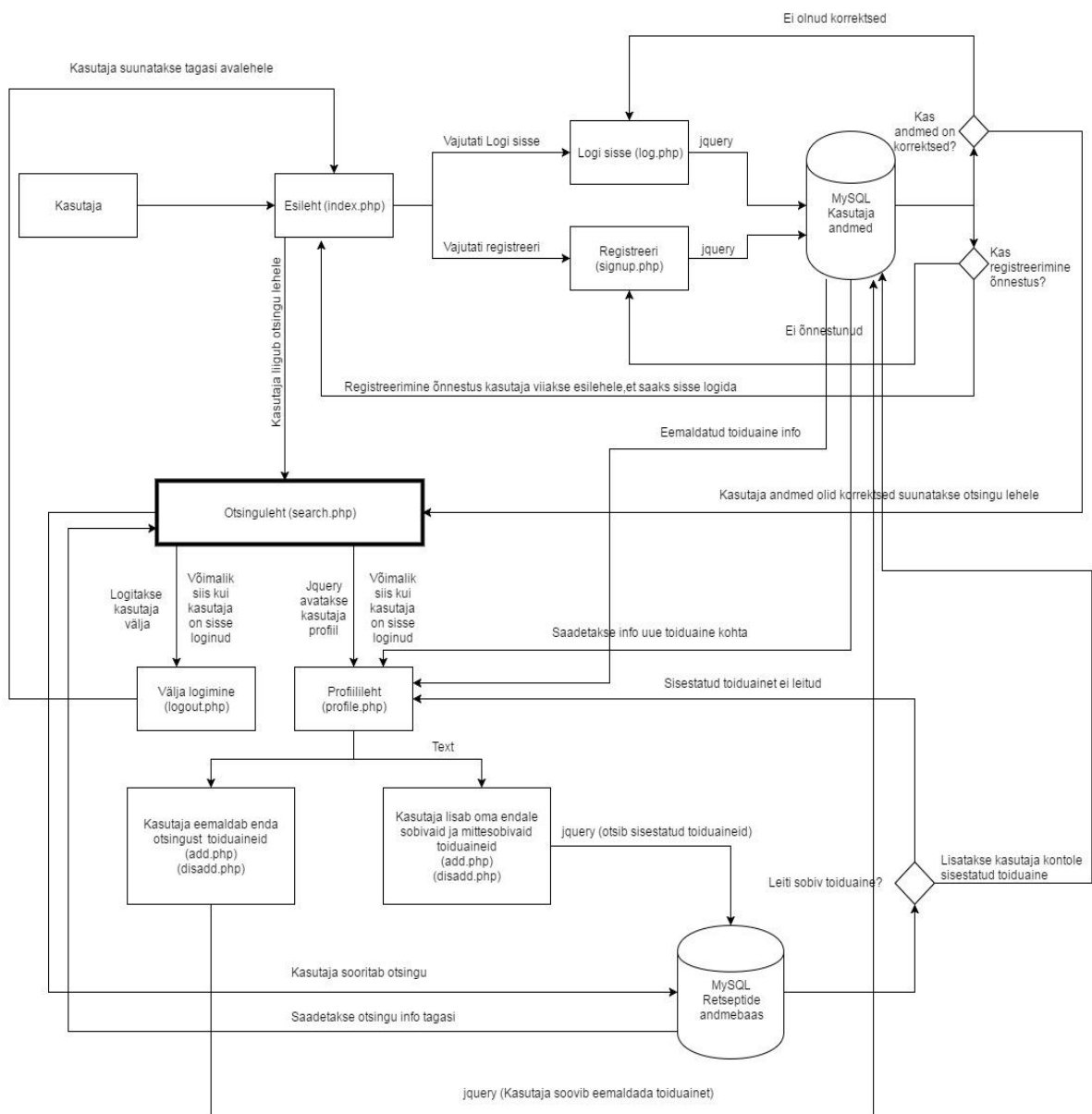
MySQL on avatud lähtekoodiga relatsiooniline andmebaasi haldamise süsteem. Kasutatakse tihti väiksete ja keskmiste lahenduste haldamiseks,mis baaseruvad ühel serveril. MySQL võimaldab arendajatel luua oma kitsendusi ja reegeleid ,mis hõplsutaksid [14]. Samuti pakub MySQL võimalust turvaliselt talletada informatsiooni andmebaasis.MySQL on maailmas levikult teise kohal, 1.mai seisuga 2017 aastal [15].

2.2.4 PHP

PHP ehk Hypertext Preprocessor on skriptimiskeel, mida kasutatakse serveripoolsetes lahendustes dünaamiliste veebilehtede loomisel. PHP on loodud aastal 1994 Rasmus Lerdorfi poolt, et pakkuda lihtsamat võimalust veebiarenduseks. PHP toetab objektorienteeritud programmeerimist, andmebaasidega suhtlemist ning moodulite kaudu veel paljusid protokolle ja rakendusi [16]. Kuna PHP võimaldab lihtsalt siduda HTMLi ja JQueryit ning on mugav kasutada andmebaasidega suhtlemiseks, siis sellest tulenevalt sai keel valitud prototüübi loomiseks.

3 Prototüüp

Antud peaktükis antakse ülevaade prototüübist, mis annab lihtsa ja mugava viisi inimestele, et oma igapäeva toitumisharjumusi mitmekesistada. Antud prototüüp implementeerib oma eesmärgi täitmiseks hulgateooria loogikat, et suurendada ning täpsustada kasutajale orienteeritud vasteid. Prototüübi töö mõistmiseks on loodud struktuurskeem (vt Joonis 9).



Joonis 9. Prototüübi struktuurskeem

3.1 Registreerimine

Konto loomine toimub esilehelt (vt Joonis 9). Esilehe üleval paremas nurgas on registreerimise nupp (vt Joonis 10). Sellel klikkides kuvatakse kasutajale registreerimise vorm (vt Joonis 11). Kasutaja konto loomiseks tuleb täita kõik väljad ning sisestatud kasutajanimi ja email peavad olevama unikaalsed. Kasutaja huvides ei ole rakendusele loodud Facebooki ega Google+ kontoga autentimis võimalust. Kasutaja sisestatud paroolist genereeritakse bcrypt funktsiooniga 60 sümbolist koosnev hash, mis salvestatakse andmebaasi [17]. Pärast registreermist on kasutajal võimalik kohe rakendust kasutama hakata.



Joonis 10. Registreerimine

The image shows a 'Sign Up' registration form. At the top, the text 'Sign Up' is written in a bold, black, serif font. Below this, there are four input fields, each with a label above it: 'Username' (with 'Enter Username' inside), 'Email' (with 'Enter Email' inside), 'Password' (with 'Enter Password' inside), and 'Repeat Password' (with 'Repeat Password' inside). Below the input fields, there is a line of text: 'By creating an account you agree to our [Terms & Privacy](#).' At the bottom of the form, there are two buttons: a green button labeled 'Sign Up' and a red button labeled 'Cancel'.

Joonis 11. Rakenduse Registreerimine

3.2 Logimine

Sisselogimine toimub rakenduse esilehelt (vt Joonis 9). Sisselogimiseks on esilehel eraldi nupp, mis on veebilehe üleval paremas nurgas (vt Joonis 12). Klõkkides nuppule kuvatakse kasutajale vorm (vt Joonis 13). Pärast sisselogimist suunatakse kasutaja profiili lehele. Juhul, kui kasutaja sisestab kasutajanime või parooli valesti kuvatakse talle kiri „Wrong Details“. Kui kasutajanime lahter on tühjaks jäänud, siis kasutajale öeldakse „et „Username required“. Juhul kui parooli lahter on tühi siis kuvatakse kiri „Password required“. Mõlema lahtri tühjaks jätmise korral tuleb ette teade „All fields required“



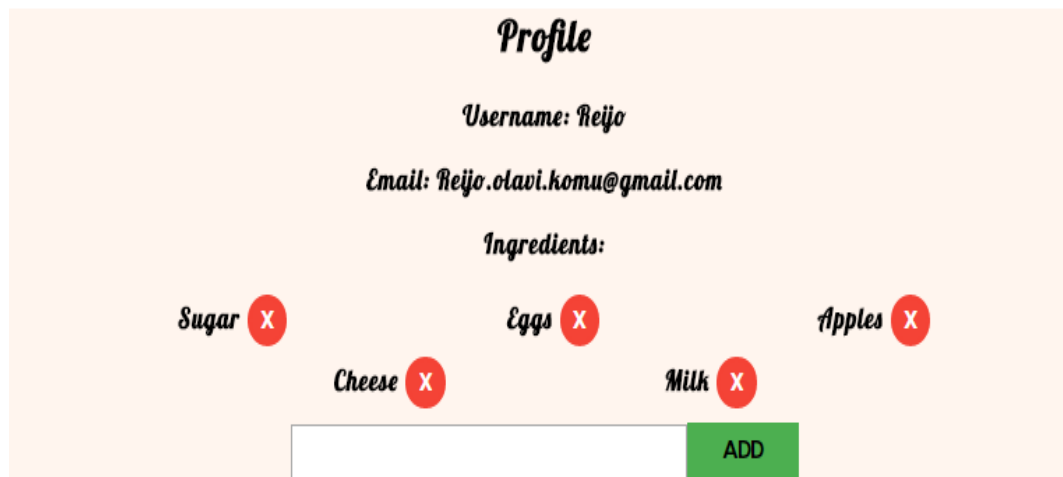
Joonis 12. Logimine esivaade

The image shows a login form titled 'Sign In' in a large, bold, black font. Below the title, there are two input fields. The first is labeled 'Username' and contains the placeholder text 'Enter Username'. The second is labeled 'Password' and contains the placeholder text 'Enter Password'. Below these fields is a large green button with the text 'Login' in white.

Joonis 13. Logimine

3.3 Sobivate toiduainetelisamine otsingusse

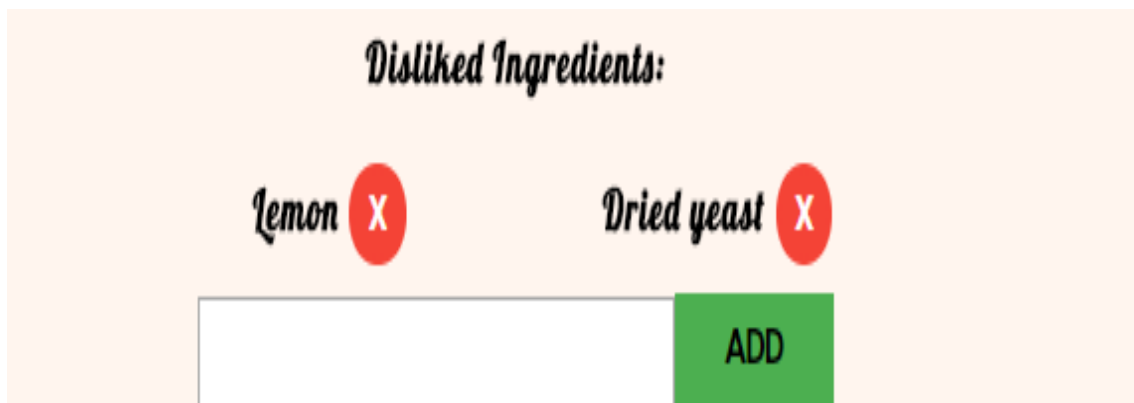
Kasutajal on võimalus lisada ja kustutada retseptides olemasolevaid toiduaineid, et lihtsustada otsimist (vt Joonis 9). Kõik toiduained, mis kasutaja lisab, peavad olema inglise keeles. Juhul kui kasutaja sisestatud toiduaine eksisteerib andmebaasis, siis tekib kasutaja konto alla vastav toiduaine. Juhul kui kasutajal on soov toiduaine oma nimekirjast eemaldada, siis selleks eksisteerib toiduaine nime taga punane nupp, mida klikkides kustutatakse kasutaja konto alt ära nimetatud toiduaine. Maksimaalne komponentide arv, mida kasutaja saab oma kontole lisada ei ole limiteeritud. (vt Joonis 14)



Joonis 14. Toiduainete lisamine

3.4 Mittesoovitavate toiduainete lisamine otsingusse

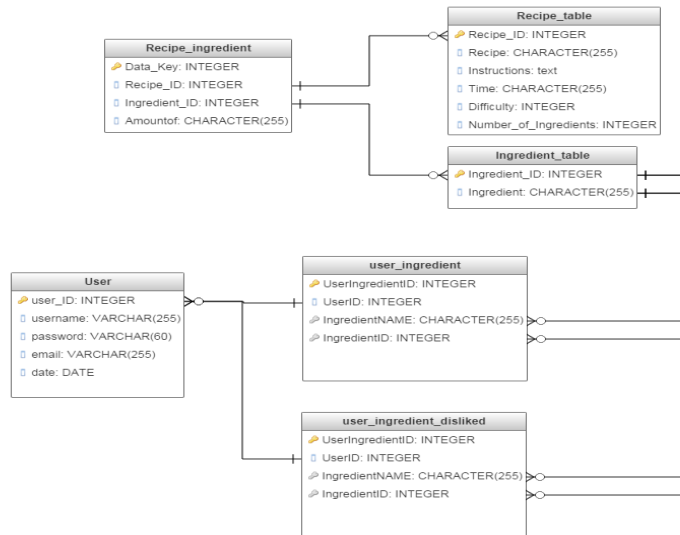
Igal kontol leidub ebameeldivate toiduainete lisamise ja eemaldamise võimalus (vt Joonis 9). Lisamiseks tuleb täita lahter inglisekeelse toiduaine nimetusega ja klikkida „ADD“ nuppu. Kui toiduaine eksisteeris kuvatakse profiili all „Disliked Ingredients“ lisatud toiduainet. Soovi korral on võimalus sisestatud toiduaine nimekirjast eemaldada, kui kasutaja klikib komponendi nime taga olevat „X“ nuppu (vt Joonis 15). Seda funktsionaalsust kasutades on võimalik elimineerida otsingu tulemusest retsepte, mis sisaldavad kasutaja poolt sisestatud toiduaineid.



Joonis 15. Ebameeldivate toiduainete lisamine ja eemaldamine

3.5 Otsingu teostamine

Selleks, et tagada võimalikult paindlik otsing tuli luua relatsiooniline andmebaas. Relatsioonilistes andmebaasides teostatakse seosed erinevate tabelite vahel andmeväljade väärtuste kaudu (vt Joonis 16). Antud andmebaas sisaldab kokku 63 retsepti ja 635 toiduainet.



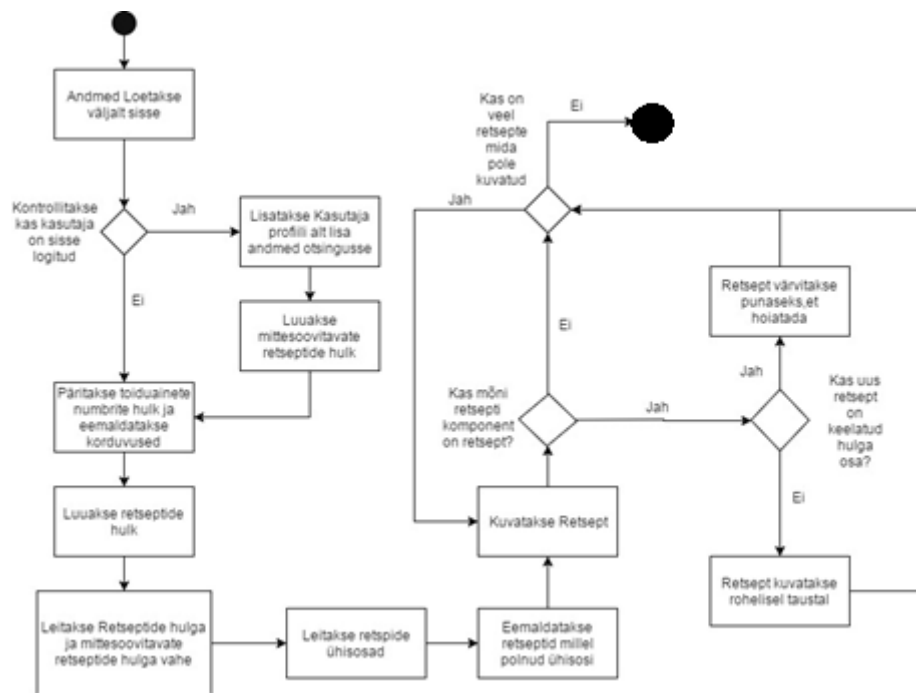
Joonis 16. Andmebaasi mudel.

Retseptide otsimiseks on kasutajal mitu valikut. Vaatleme juhtu, kui kasutaja on sisestanud oma otsingureale kolm komponenti. Milleks on „Dry pancake mix“, „Milk“, „Eggs“ (vt Joonis 17). Samuti soovib kasutaja välistada retseptidest toiduaine „Lemon“. See tuleb lisada kasutajal oma profiili alt. Klõkkides nupule „Search“, mis asub otsingurea kõrval,



Joonis 17. Otsinguriba

käivitub serveris skripti nimega search.php (vt Joonis 18).



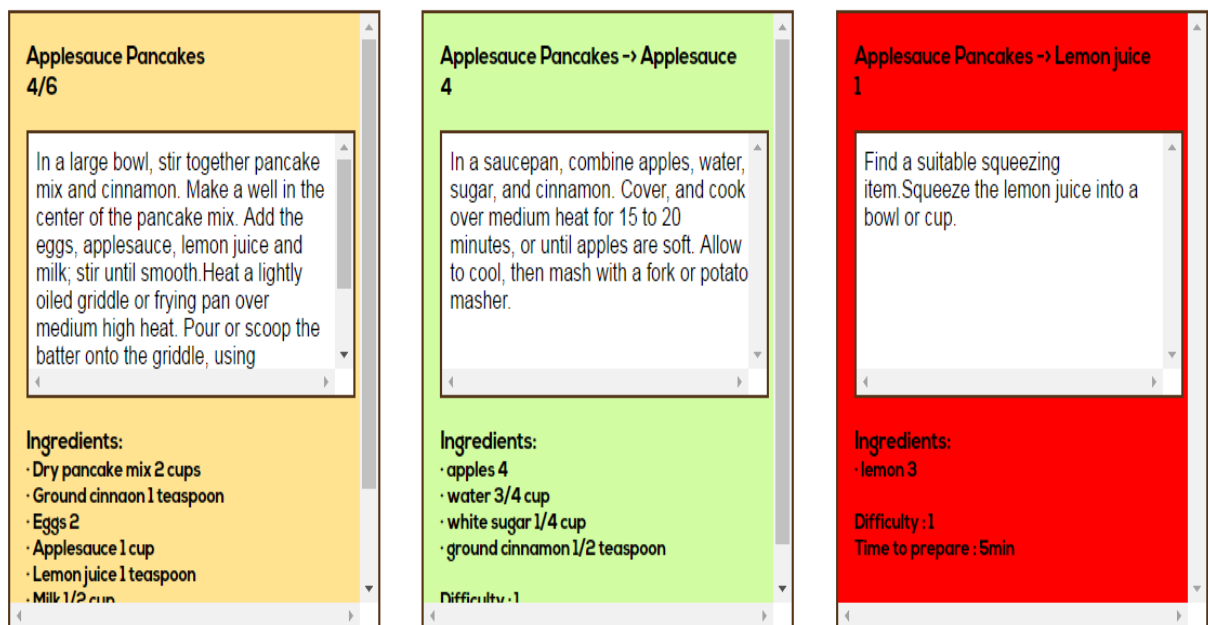
Joonis 18. Otsingu algoritimi vooskeem

Kõigepealt skript loeb andmed sisse otsingu rea kaudu. Seejärel kontrollitakse, kas tegemist oli registreeritud kasutajaga. Kohe kui on kindlaks tehtud, et tegemist on kasutajaga, sooritab skript päringu andmebaasi. Andmebaasist võetakse kasutaja poolt valitud toiduained, mille alusel luuakse hulk nimega „Ingredients_array“. See liidetakse kokku otsingurea hulga. Sellele järgnevalt teostatakse järgmine päring andmebaasi, mille alusel luuakse toiduainete hulk, mida kasutaja ei soovi retseptides kohata nimega „Hated_Ingredients_array“. Seda hulka kasutatakse sisendparameetrina, et luua mittesobivate retseptide hulk nimega „Hated_Recipe_List“ (vt LISA 1).

Hulka „Ingredients_array“ kasutatakse sisendparameetrina päringusse, et saada kõikide toiduainete indeksid, mis talletatakse muutujasse „Numbers_array“. Hulka „Numbers_array“ kasutades luuakse hulk „Recipe_list“ läbi järgmise pöördumise andmebaasi poole. Programmeerimise lihtsustamise eesmärgil on kõik retseptid ühes hulgas. „Recipe_list“ ühisosa leidmiseks kasutatakse array_count_values funktsiooni, millega tehakse kindlaks suurimate ühisosadega retseptid. Seejärel hulk sorteeritakse ning jäetakse alles vaid need retseptid, mis kuulusid ühisosadesse. Kuna kõik retseptid on ühes hulgas ei ole oluline, kas vahe leitakse enne või pärast ühisosa

leidmist. Antud juhul leitakse vahe enne võrreldes „Recipe_List“ hulka „Hated_Recipe_List“, millest alles jääb sobilik retseptide hulk. (vt LISA 2). Seejärel tehakse päringud, et kokku panna sobilik hulk. Seejärel tehakse päring iga retsepti komponendi kohta, et kontrollida ega komponent pole retsept. Juhul, kui komponent on retsept, kontrollitakse ega saadud retsept pole keelatud retseptide alamhulk.

Kasutajale kuvatakse otsingu tulemuse otsevastad beežil taustal, kus kaldkriipsust vasakul pool on ühisosade arv ning paremal maksimaalne võimalik komponentide ühisosade arv antud retsepti kohta. Samuti kuvatakse kasutajale rohelisel ja punasel taustal retsepte, mis olid beežil taustal oleva retsepti sees (vt Joonis 19).



Joonis 19. Otsingu tulemuste kuvamine

Juhul, kui retsepti taust on punane tähendab, et retsept kuulus keelatud retseptide hulka. Rohelisel taustal pakutakse kasutajale võimalust see komponent ise koostada.

Kokkuvõte

Käesoleva töö põhieesmärk oli luua multifunktsionaalne otsing. Toiduretseptide andmebaasi näitel, mis aitaks muuta inimeste toitumist mitmekesisemaks.

Töö käigus analüüsiti hulgateoria põhimõisted ning kuidas oleks neid võimalik implementeerida multifunktsionaalsesse otsingusse, mis kasutab oma andmetena toiduretsepte. Samuti töös analüüsiti erinevaid tehnoloogiaid, mida saab rakenduse loomiseks kasutada.

Töö tulemusena loodi multifunktsionaalse otsingu prototüüp, mis kasutab andmete talletamiseks MySQL andmebaasi ning realiseerib oma otsingu kasutades PHP-ed. Antud prototüüp võimaldab igal inimesele ennast kasutajaks registreerida. Igal kasutajal on võimalik lisada ja eemaldada talle meeldivaid ja ebameeldivaid toiduaineid erinevates retseptides. Sisestatud andmetega on võimalik sooritada multifunktsionaalne otsing, mis soovitab ning hoiatab retseptide komponentide osas. Andmete otsing tugineb hulgateoorial, mida kasutades valideeritakse andmete sobilikust.

Loodud prototüüpi otsing tugineb praegu PHP-l. Edasiarendusena oleks võimalik luua antud otsing mõnes muus keeles, mis võimaldaks kiiremat otsingut. Samuti oleks võimalik luua antud prototüüpile lisa funktsionaalsust. Kasutajad võiksid saada lisada oma retsepte andmebaasi ning kõikidel restepile oleks võimalik anda tagasisidet. Kindlasti oleks võimalik luua edasiarendusena ka iseläienev andmebaas, mis suudaks otsida internetist olemasolevaid retsepte ning neid automaatselt lisada ning seda kasutades muuta andmebaas mitmekeelseks. Parema otsingu saavutamiseks oleks võimalik rakendada iseõppimist, mis annaks täpsemaid ning rohkem kasutajate vajadustest ja soovidest orienteeruvaid otsingutulemusi.

Kasutatud kirjandus

- [1] „<http://akit.cyber.ee>,” Mai 2017. [Võrgumaterjal].
- [2] J. Ferreirós, Labyrinth of Thought: A History of Set Theory and Its Role in Modern Mathematics, 1999.
- [3] „http://storyofmathematics.com/19th_cantor.html,” Mai 2017. [Võrgumaterjal].
- [4] „http://www-groups.dcs.st-and.ac.uk/history/HistTopics/Beginnings_of_set_theory.html,” Mai 2017. [Võrgumaterjal].
- [5] P. Oja, Hulgateooria, Tartu: Tartu Ülikooli Kirjastus, 2001.
- [6] P. Loorents ja V. Viies, „Lühiülevaade IT alal olulisematest hulgateooria mõistetest,” 2003.
- [7] H. Lensen ja M. Kruus, Diskreetne matemaatika, Tallinn: TTÜ kirjastus, 2012.
- [8] M. Kilp ja U. Nummert, Hulgateooria elemendid, Tartu: Tartu Ülikooli Kirjastuse trükikoda, 1994.
- [9] „https://www.w3schools.com/css/css_intro.asp,” Mai 2017. [Võrgumaterjal].
- [10] „<https://www.w3.org/standards/webdesign/htmlcss#whatcss>,” [Võrgumaterjal].
- [11] „https://www.w3schools.com/jquery/jquery_intro.asp,” Mai 2017. [Võrgumaterjal].
- [12] „<http://jquery.com/>,” Mai 2017. [Võrgumaterjal].
- [13] „<https://w3techs.com/technologies/details/js-jquery/all/all>,” Mai 2017. [Võrgumaterjal].
- [14] „<https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>,” [Võrgumaterjal].
- [15] „<https://db-engines.com/en/ranking/relational+dbms>,” Mai 2017. [Võrgumaterjal].
- [16] „<http://php.net/manual/en/history.php.php>,” Mai 2017. [Võrgumaterjal].
- [17] „<http://php.net/manual/en/function.password-hash.php>,” Mai 2017. [Võrgumaterjal].

LISA 1 - Väljalõige search.php-st , kus kontrollitakse kasutaja sisse logitust ja luuakse esmased hulgad otsingu teostamiseks

```
if($user->is_loggedin()!="")
{
    $stmt = $DB_con->prepare("SELECT * FROM user_ingredient WHERE UserID=:user_id");
    $stmt->execute(array(":user_id"=>$user_id));
    while($userRow=$stmt->fetch(PDO::FETCH_ASSOC))
    {
        $Ingredients_array[]=$userRow['ingredientName'];
    }
    if(!empty($Ingredients_array))
    {
        $dataArray=array_merge($Ingredients_array,$dataArray);

        $stmt = $DB_con->prepare("SELECT * FROM usingredientdisliked WHERE UserID=:user_id");
        $stmt->execute(array(":user_id"=>$user_id));
        while($userRow=$stmt->fetch(PDO::FETCH_ASSOC))
        {
            $Hated_Ingredients_array[]=$userRow['IngredientID'];
        }
        if(!empty($Hated_Ingredients_array))
        {
            $connect->Select_Recipe_Ingredient($Hated_Ingredients_array);
            $Hated_Recipe_List=$connect->Return_Recipe_ID_Array();
            $Hated_Recipe_List=array_unique($Hated_Recipe_List);
        }
    }
}
```

LISA 2 - Väljalõige search.php-st ,kus näidatakse ühisosade leidmist ja ühisosa hulgast vahe eemaldamist.

```
if(!empty($Number_Array))
{
    $i=0;
    $connect->Select_Recipe_Ingredient($Number_Array); //For Selecting Recipe_ID From the Recipe ingredient table

    $Recipe_list=array();

    $Recipe_list=$connect->Return_Recipe_ID_Array();

    if(!empty($Hated_Ingredients_array))
    {
        $Recipe_list=array_diff($Recipe_list,$Hated_Recipe_List);//Sets Complement is found
    }

    //The Biggest Intersections are found
    $Recipe_list=array_count_values($Recipe_list);
    //Array is sorted so the Biggest union would be at the top
    arsort($Recipe_list);

    //Used for Intersection

    if((sizeof($Recipe_list)>1))
    {
        foreach($Recipe_list as $key => $value)
        {
            if($value>1)
            {
                $key_array[]=$key;
                $value_array[]=$value;
                if($i==$maxamountofrecipes) break;
                $i++;
            }
        }
        unset($Recipe_list);
        if($i)
        {
            //Only These recipes remain that have atleast 1 Intersection with other set.
            $Recipe_list=array_combine($key_array,$value_array);
        }
    }
}
```