

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Infotehnoloogia eriala

Siim Loog
Predikaatarvutuse väljendusülesannete
lahendamise veebikeskkond
Bakalaureusetöö

Juhendaja: dots. R.Prank

Autor“.....” juuni 2011
Juhendaja“.....“ juuni 2011
Lubada kaitsmisele
Professor“.....“ juuni 2011

TARTU 2011

Sisukord

Sissejuhatus.....	4
1. Tehtud valikud ja nendest tulenevad kitsendused.....	6
1.1 Ülesannete valik.....	6
1.2 Valemite süntaks.....	6
2. Rakenduse kirjeldus.....	8
2.1 Sarnased rakendused.....	8
2.1 Kasutajad.....	8
2.2 Süntaksi kontroll.....	8
2.3 Vastuse õigsuse kontroll.....	9
3. Arhitektuur.....	10
3.1 Tehnoloogiad.....	10
3.1.1 HTML5.....	10
3.1.2 CSS3.....	10
3.1.3 Javascript.....	10
3.1.4 PHP.....	10
3.1.5 MySQL.....	11
3.1.6 Ajax.....	11
3.2 Andmemudel.....	12
3.3 Koodi struktureerimine.....	13
3.4 Paigaldamine.....	13
3.5 Kasutajaliidese üldine ülesehitus.....	14
3.5.1 Õpilase menüü.....	14
3.5.1 Õppejõu menüü.....	14
3.5.1 Tööala.....	14
4 Nõuded.....	15
4.1 Mittefunktsionaalsed nõuded.....	15
4.1.1. Kasutatavus.....	15
4.1.2. Käideldavus.....	15
4.1.3. Jõudlus.....	15
4.1.4. Toetus.....	15
4.2 Kasutusjuhud.....	15
4.2.1 Registreerumine.....	15
4.2.2 Sisse logimine.....	16
4.2.3 Töö loomine.....	16
4.2.4 Töösse ülesannete lisamine.....	17
4.2.5 Töö ülesannete redigeerimine.....	18
4.2.6 Töö avamine.....	19
4.2.7 Töö ülesannete lahendamine.....	20
4.2.7 Soorituse hindamine.....	20
5. Täiendamise ja edasiarendamise võimalused.....	22
6. Kokkuvõte.....	23
Web application for expressing predicates as formulas.....	24
Kasutatud materjalid.....	25
Lisad.....	26

Sissejuhatus

Tartu ülikoolis loetaval kursusel Sissejuhatus matemaatilisse loogikasse (SML) on igal semestril tööplaanis lisaks arvutipraktikumidele ka kolm kuni neli tahvlipraktikumi. Nendes praktikumides käsitletakse väljendusülesandeid. Lisaks sellele on ka kontrolltöös väljendusülesanne. Kui predikaatarvutuse valemite analüüsi ja prefikskujule viimise ülesannete harjutamiseks ja lahendamiseks on juba elektroonilised vahendid olemas, siis väljendusülesannete lahendamine on siiani nii harjutamise kui ka kontrolltöö ajal toimunud ainult kirjalikult.

Olukord on selline sellepärast, et puuduvad elektroonilised õppevahendid väljendusülesannete lahendamise harjutamiseks. Harjutuskeskkonnad puuduvad kuna predikaatloogika valemite samaväärsus ei ole algoritmiliselt lahenduv [1].

Küll on aga võimalik hinnata sisestatud valemi süntaktilist korrektsust ja kontrollida, et valemis oleksid õiged vabad ja seotud muutujad. Üliõpilastele võib sellist funktsionaalsust pakkuv rakendus kasulik olla. Tihti viitavad süntaksivead ülesandest valesti aru saamisele. Lisaks aitab see ka juhtida üliõpilase tähelepanu hooletusest tekkinud trükivigadele.

Selle bakalaureusetöö eesmärgiks on luua veebirakendus, milles õppejõul on võimalik sisestada predikaatarvutuse väljendusülesandeid naturaalarvude hulgal, signatuuriga $\langle 0,1;+,*;=\rangle$, neid ülesandeid töödessa organiseerida ning õpilastel nendele küsimustele vastata sisestades valemite mille süntaktilist korrektsust ka kontrollitakse.

Loodud rakendus peaks olema kergesti laiendatav, et tulevikus oleks võimalik kontrollida süntaksi ka siis kui defineeritakse teistsugune signatuur.

Projekt põhineb tehnoloogiatel:

- HTML5
- CSS3
- Javascript
- PHP
- MySQL

Esimeses peatükis on kirjeldatud millised valikud on skoobi kitsendamiseks tehtud ja millised kitsendused nendest valikutest tulenevad.

Teises peatükis antakse rakenduse üldine kirjeldus. Käsitletakse kasutajate rolle, teisi sarnaseid rakendusi ja kirjeldatakse üldsõnaliselt süntaksi valideerimiseks kasutatavaid kontrole.

Kolmandas peatükis käsitletakse rakenduse ülesehitust. Kirjeldatakse arhitektuuri ja andmemudelit ning selgitatakse lähemalt kasutatavaid tehnoloogiaid.

Neljandas peatükis tuuakse välja antud projekti jaoks olulisemad mittefunktsionaalsed nõuded ning kirjeldatakse tähtsamaid kasutusjuhte koos ekraanikuvadega.

Viiendas peatükis kirjeldatakse projekti raames loodud rakenduse edasiarendamise ja laiendamise võimalusi.

1. Tehtud valikud ja nendest tulenevad kitsendused

1.1 Ülesannete valik

Lõputöö keskendub väljendamisülesannetele naturaalarvude hulgal. Signatuur käesoleva projekti raames alati:

- Konstantsümbolid: 0 ja 1
- Funktsionaalsümbolid: + (liitmine) ja * (korrutamine)
- Predikaatsümbol: =

ehk $\langle 0,1; +,*,= \rangle$

Ülesanne koosneb õppejõu poolt teksti kujul sisestatud predikaadist, argumentide loendist ja ühest kuni kolmest õigest vastusest. Lisaks sellele saab õppejõud sisestada ka ülesande eest maksimaalselt saada oleva punktide arvu.

1.2 Valemite süntaks

Fikseerime edaspidise käsitluse tarbeks järgmised sümbolite klassid.

- *Indiviidmuutujad*, mida märgime ladina tähtedega. Võib kasutada nii suuri kui ka väikeseid tähti. Näiteks: A,a,B,b. Iga indiviidmuutuja võib tähistada naturaalarvude hulga ükskõik millist elementi.
- *Konstantsümbolid*, milleks lõputöö skoobis on 0 ja 1. Need tähistavad naturaalarvude hulga vastavaid elemente.
- *Funktsionaalsümbolid*, + (liitmine) ja * (korrutamine), mis tähistavad naturaalarvude hulgal määratud funktsioone.

Avaldise kirjapanekuks pole vaja teada sümbolite täpset tähendust, küll aga reegleid, kuidas sümboleid üksteisega ühendada. Nende reeglite rakendamisel saadud objekti nimetatakse termiks, täpsemalt määratletud järgmise definitsiooniga: [2]

Definitsioon: Termid on parajasti need, mida saab koostada alltoodud reeglite abil,

1. Iga indiviidmuutuja on term.
2. Iga konstantsümbol on term.
3. Kui t_1 ja t_2 on termid, siis $(t_1 + t_2)$ ja $(t_1 * t_2)$ on termid

Kokkuvõttes, term on antud signatuuris kirja pandud aritmeetiline avaldis. Igal konkreetsel juhul võib termi väärtus olla naturaalarv. Termide võrdus on aritmeetika keeles atomaarne valem. Termidest moodustatakse lausearvutuse tehete ja kvantorite abil aritmeetika valemid, nagu näitab järgmine definitsioon.

Definitsioon. Predikaatarvutuse valemid on parajasti need, mida saab koostada alltoodud reeglite abil. [2]

1. Kui t_1 ja t_2 on termid, siis $t_1=t_2$ on aritmeetika valem.
2. Kui F on aritmeetika valem, siis $\neg F$ on aritmeetika valem.

3. Kui F ja G on aritmeetika valemid, siis $(F \& G)$, $(F \vee G)$, $(F \rightarrow G)$ ja $(F \leftrightarrow G)$ on aritmeetika valemid.
4. Kui x on individumuutuja ja F on aritmeetika valem, siis $\forall xF$ ja $\exists xF$ on aritmeetika valemid.

Tabel 1: Näiteid väidete ja neid väljendavate valemite kohta

Predikaat (väide)	Valem
$x < y$	$\exists z(\neg(z=0) \& (x+z=y))$
$z = x/y$	$(x=z*y) \& \neg(y=0)$
x jagub y -ga	$\exists z((x=y*z) \& \neg(y=0))$

2. Rakenduse kirjeldus

2.1 Sarnased rakendused

Ülesannetekomplektide koostamise ja küsimustele vastamise veebilahendusi on mitmeid. Tartu Ülikoolis oli/on kasutusel näiteks Webct[3], mis lisaks muule funktsionaalsusele võimaldab õppejõul koostada küsimustikke, millele õpilased vastata saavad, kuid puuduvad väljendamisülesannete valemite süntaksi kontrolli võimaldavad vahendid.

Rakenduse Tarski's World [4] abil saavad kasutajad leida esimest järku lausete tõeväärtusi lõplikul hulgal.

2.1 Kasutajad

Rakendusel on kahte tüüpi kasutajaid:

- õpilane
- õppejõud

Õpilasel on vahendid

- avatud tööde ülesannete lahendamiseks
- soorituse tagasiside vaatamiseks.

Õppejõul on kõik võimalused, mis õpilasel ja lisaks sellele saab ta

- töid luua
- tööle ülesandeid luua
- töö nime ja kirjeldust muuta
- töid kustutada
- töö ülesandeid redigeerida või eemaldada
- töid lahendamiseks avada ning lahendamisaja lõppedes sulgeda
- vaadata sooritusi
- vaadata valitud kasutaja sooritusi
- sooritusi hinnata määrates igale ülesandele punktid ja tagasiside
- kasutajate nime ja parooli muuta
- kasutajate kontosid blokeerida ja eemaldada

2.2 Süntaksi kontroll

Nii ülesande loomisel kui ülesandele vastamisel kontrollitakse vastuse süntaksi ning kuvatakse teade otsuse kohta. Kus kohane, märgistatakse vea tekitanud valemi osa.

Et koostada kriteeriumid, mida rakendus süntaksi valideerimisel kontrollima peab on arvestatud ka lahenduse programmeerimise lihtsustamist. Kokkuvõttes on need kriteeriumid järgnevad:

- valemis leiduvad ainult lubatud sümbolid
 - $\neg, \&, \vee, \supset$
 - kvantorid
 - ümarsulud
 - signatuuriga määratud sümbolid
 - predikaadiga määratud indiviidimuutujad (vabad muutujad)
 - indiviidimuutujad mis on seotud kvantoriga ja asuvad ainult selle kvantori mõjupiirkonnas, millega nad seotud on
 - üks indiviidimuutuja võib olla seotud mitme erineva kvantoriga, kuid nende kvantorite mõjupiirkondadel ei tohi olla ühisosa
- sümbolite järgnevus. Sümbolite järgnevust illustreeriv tabel toodud lisas 1.
- kvantori mõjupiirkonnas peab leiduma see indiviidimuutuja mida ta seob
- kvantoril peab olema mõjupiirkond
- predikaadiga defineeritud indiviidimuutuja ei tohi olla kvantoriga seotud kuid võib leiduda tema mõjupiirkonnas
- sulud peavad olema suletud s.t igale "("-le peab leiduma ka ")" ja vastupidi.
- Konjunktsiooni, disjunktsiooni, implikatsiooni ja eituse korral peab operaatori iga osavalem olema aritmeetika valem.
- Atomaarsed osavalemid peavad olema sulgudes.
- igas atomaarses valemis peab leiduma täpselt 1 võrdusmärk, s.t ei tohi olla olukorda $x+y=0=z$
- valemis peavad leiduma kõik predikaadiga määratud indiviidimuutujad
- kui valem koosneb ainult ühest atomaarsest valemist ei ole vaja termi sulgudega ümbritseda
- eitusele ei tohi vahetult järgneda eituse. See ei ole süntaktiliselt vale, kuid on ebaotstarbekas.

2.3 Vastuse õigsuse kontroll

Kui õpilane sisestab ülesande lahendamisel süntaktiliselt korrektse valemi, siis seda vastust võrreldakse õppejõu poolt määratud õige(te) vastus(t)ega. Võrdlemisel ei võrrelda mitte valemite tõeväärtusi vaid sõnesid.

Kui õpilase sisestatud valem on sama, mis õpetaja poolt määratud õige vastus või erineb sellest ainult kvantoriga seotud indiviidimuutujate poolest, siis loetakse vastus õigeks.

Näide: Olgu meil väide $x < y$ ja õigeks vastuseks olgu määratud $\exists z (\neg(z=0) \& (x+z=y))$.

Kui õpilane sisestab vastuseks valemi $\exists t (\neg(t=0) \& (x+t=y))$, siis peab rakendus suutma tuvastada, et see vastus on õige.

Kui aga rakendus ei suuda sisestatud valemi õigsust määrata, siis jääb punktide määramine õppejõule.

3. Arhitektuur

3.1 Tehnoloogiad

Lühiülevaade keeltest ja tehnoloogiatest, mida projektis kasutatakse.

3.1.1 HTML5

HTML (inglisekeelne lühend sõnadest HyperText Markup Language 'hüperteksti märgistuskeel') on keel, milles märgendatakse veebilehti. HTML on SGML-i rakendus. HTML-dokumentide (failide) veebist allalaadimiseks ja vaatamiseks kasutatakse veebilehitsejat. Selline elektroonilise dokumendi märgendamise mõte on lahutada sisu, vaade jms objekti omadused üksteisest.

HTML paneb paika üksnes dokumendi struktuuri, kuid võimaldab leheküljele kaasata skripte, eeskätt JavaScript, ning CSS-i millega kirjeldatakse lehe kujundus.[5]

3.1.2 CSS3

CSS (inglisekeelne lühend sõnadest Cascading Style Sheets 'kaskaadlaadistik') on keel, milles märgitakse üles XML ja SGML- keelsete failide kujundust. Eelkõige kasutatakse CSS-i XHTML-i ja viimase eellase HTML-i failide kujunduse loomisel.

CSS-i kombineerimine XHTML-iga võimaldab veebilehe sisu ja kujunduse teineteisest eraldada. See annab parema ülevaate lehekülje ülesehitusest ja on abiks lehekülje hilisemal muutmisel. Lihtsamaks muutub kogu lehekülje haldamine ning eelkõige kujunduse muutmine. [6]

3.1.3 Javascript

JavaScript on Netscape'i poolt loodud objektorienteeritud programmeerimiskeel, mida kasutatakse peamiselt veebilehtede skriptimiseks [7]. Bakalaureusetöö tulemina valminud rakenduses on Javascript kasutusel kasutajaliidese interaktiivseks tegemisel ja sisestatud valemi süntaksi kontrollimisel.

Projektis kasutatakse abivahenditena ka:

- javascripti raamistikku Prototype [8] mille vahendid aitavad kasutada AJAX päringuid ning kergekaalulist andmevahetusformaati JSON [9]
- JavaScript MD5 [10]. Kasutatakse loodud rakenduses et luua parooli *hash* turvalisuse tagamise ees-märgil.
- HTML encoder and decoder tool [11]. Kasutati loodud rakenduses erimärkide korrektseks kuvamiseks ja salvestamiseks.

3.1.4 PHP

PHP: Hypertext Preprocessor on skriptimiskeel, mida kasutatakse peamiselt serveripoolsetes lahendustes dünaamiliste veebilehtede loomisel. PHP toetab objektorienteeritud programmeerimist, andmebaasidega suhtlemist ning moodulite kaudu veel paljusid protokolle ja rakendusi. [12]

PHP ülesanneteks loodud rakenduses on Javascripti poolt edastatud kasutaja käskudele reageerimine, andmebaasiga suhtlemine, kasutaja käskudele vastava HTML sisu genereerimine ning tagastamine ja üliõpilase sisestatud valemi võrdlemine õppejõu poolt sisestatud õige(te) vastus(t)ega.

Projektis kasutatakse abivahendina Smarty templaatimismootorit [13] et lihtsustada kasutajaliidese andmete muutmist.

3.1.5 MySQL

MySQL on relatsioonilise andmebaasi haldamise süsteem, millesse saab päringud esitada SQL standardi järgi.[14]

Loodud rakenduses kasutatakse MySQL-i, et salvestada, uuendada, pärida ja kustutada andmeid.

3.1.6 Ajax

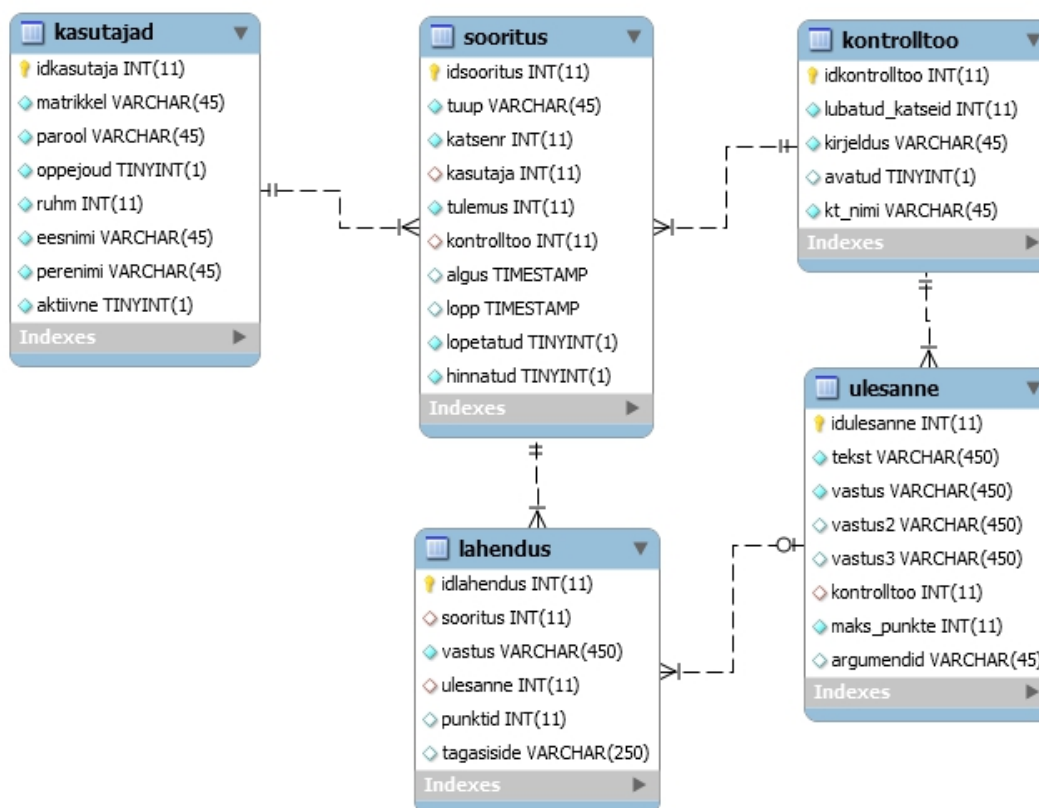
Ajax (inglise keeles Asynchronous JavaScript And XML) on kogum omavahel seotud veebiarenduse tehnikaid, mis on kasutuses rakenduse kliendi poolel, et luua interaktiivne veebirakendus.

Ajax võimaldab veebilehtedel JavaScriptiga serverisse tagaplaanil päringuid teha ja andmeid vastu võtta, segamata avatud lehekülje kuvamist ja olekut.[15]

Projektis kasutatakse Ajaxit peaaegu alati kui on vaja kasutajaliidesel midagi muuta või uuendada. Ainult lehe avamisel või värskendamisel kasutatakse PHP templaatimismootorit Smarty. Põhiline eelis, mille Ajaxi kasutamine kaasa toob on see, et kui muutub ainult mingi osa lehest ei ole vaja uuendada kogu lehte vaid saab uuendada ainult soovitud osa lehest.

3.2 Andmemudel

Projekti andmemudel on kujutatud joonisel 1.



Joonis 1. Andmemudel

- Kasutajad- sisaldab andmeid kasutajate kohta
- kontrolltoo- sisaldab andmeid töö kohta: nimi, kirjeldus, avatud/suletud ja lubatud katsete arv.
- ülesanne – sisaldab andmeid iga üleande kohta: ülesande püstitus, töö millesse ülesanne kuulub, argumendid, õige(d) vastus(ed) ning maksimaalselt saada olev punktide arv.
- lahendus- sisaldab endas õpilase sisestatud valemit, viidet ülesandele millele vastati ning õppejõu poolt määratud tagasisidet ja punkte.
- sooritus- sisaldab andmeid soorituse selle kohta, kes on soorituse autor, kas sooritus on lõpetatud ja hinnatud, millal sooritus lõpetati, millise töö küsimustele vastati ja kui palju punkte kokku saadi.

Lähemalt oleme „sooritus“ kohta:

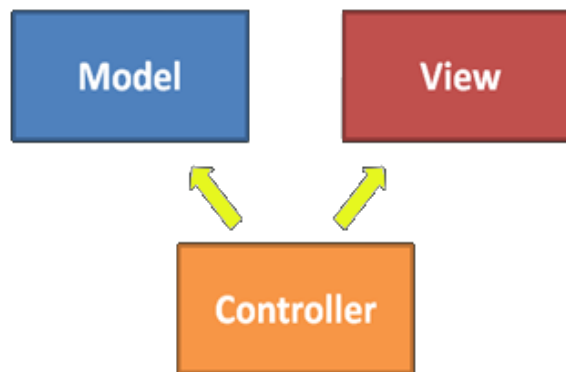
- Sooritus luuakse kui üliõpilane avab esimest korda lahendamiseks töö ülesanded. Igal järgmisel sama töö avamisel kasutatakse juba loodud sooritust, et kuvada juba salvestatud vastused, luua uusi vastuseid või muuta salvestatud vastuseid.
- Töö sooritus loetakse lõpetatuks kui üliõpilane on vajutanud lahendamise aknas nupule „Lõpeta“ ja oma otsust kinnitanud. Peale seda üliõpilane enam selle töö

ülesandeid lahendada ei saa.

- Kui üliõpilane liigub töö ülesannete lahendamise vaatest mõnele muule vaatele, logib välja või sulgeb rakenduse akna, siis see sooritust ei lõpeta.
- Et lõpetatud või töö sulgemisega lõppenuks loetud sooritus muutuks üliõpilasele nähtavaks peab õppejõud selle ära hindama.

3.3 Koodi struktureerimine

Koodi struktureerimisel on lähtutud MVC mudeli variatsioonist *Passive View pattern*.



Joonis 2: MVC mudel [16]

Traditsioonilisemast MVC mudelist eristub see variatsioon sellega, et ei toimu otsesuhtlust moodulite Model ja View vahel.

Struktuur on jagatud kolme ossa

- **Model**– Sisaldab andmeid ja andmetega manipuleerimise vahendeid
- **Controller** – Toimib vahelülina mudeli ja vaate vahel. Kui kasutajaliideses on sooritatud tegevus, mis nõuab mudeliga suhtlemist, siis edastatakse see tegevus kontrolleriile, mis mudeliga suheldes loob vajaliku sisu. See sisu edastatakse seejärel vaatele, mis kuvab sisu.
- **View** – kasutajaliides

3.4 Paigaldamine

Nõuded süsteemile kuhu rakendus installeeritakse:

- PHP tugi (katsetatud versiooniga 5.3)
- MySQL server (katsetatud versiooniga 5.1)

Rakenduse installeerimiseks tuleb alla laadida rakenduse lähtekood [Lisa 2] ning paigutada see sinna, kust see veebile kättesaadavaks tehakse. Peale keskkonnamuutujate seadistamist tuleb jooksutada installatsiooniskripti, mis loob andmebaasi vajalikud tabelid ja ka administratiivkonto. Peale tabelite loomist tuleb

Detailsem installeerimisjuhend on kättesaadav rakenduse praeguses failihoidlas [17].

3.5 Kasutajaliidese üldine ülesehitus

Kasutajaliidese üldine ülesehitus on kujutatud joonisel 3.



The screenshot shows a web application interface with a central table and two side navigation menus. The table has columns for Matrikkel, Nimi, Rühm, Roll, and Aktiivne. A row is visible with the value 'admin' in the Matrikkel column, 'Joud Öppee' in the Nimi column, '1' in the Rühm column, 'Õppejõud' in the Roll column, and 'Aktiivne' in the Aktiivne column. To the right of the table are two buttons labeled 'Redigeeri' and 'Sooritused'. The left navigation menu includes 'Õpilane', 'Kontrolltööd', and 'Minu sooritused'. The right navigation menu includes 'Konto', 'Logi välja', 'Administreeri', 'Kasutajad', 'Kontrolltööd', and 'Sooritused'.

Matrikkel	Nimi	Rühm	Roll	Aktiivne		
admin	Joud Öppee	1	Õppejõud	Aktiivne	Redigeeri	Sooritused

Joonis 3: kasutajaliides

3.5.1 Õpilase menüü

Asub kasutajaliidese vasakul poolel. See menüü sisaldab õpilase jaoks vahendeid avatud tööde ja õpilase soorituste vaatamiseks.

3.5.1 Õppejõu menüü

Asub kasutajaliidese paremal poolel. Sisaldab vahendeid kasutajate, tööde ja soorituste vaatamiseks.

3.5.1 Tööala

Asub kasutajaliidese keskel. Sellel alal kuvatakse kõik nimekirjad, redigeerimisvormid ja lahendusvormid.

4 Nõuded

4.1 Mittefunktsionaalsed nõuded

4.1.1. Kasutatavus

- Keskkonna kasutamine peab tavakasutajale olema võimalikult lihtne ja arusaadav.
- Veateated on süntaksi kontrollimisel on võimalikult informatiivsed

4.1.2. Käideldavus

- Rakendus olema kasutatav 99% ajast. Arvestamata riistvararikkeid.

4.1.3. Jõudlus

- Kasutaja vastuse koostamine serveris ei tohi võtta kauem kui 2 sekundit.

4.1.4. Toetus

- Veebikeskkond luuakse eestikeelsena.

4.2 Kasutusjuhud

Selles peatükis tuuakse välja tähtsamad kasutusjuhud. Kirjeldatud funktsionaalsusele lisanduvad ka vahendid kasutajate administreerimiseks, kuid kuna need pole projekti raames kriitilised siis jäetakse need siinkohas kirjeldamata.

4.2.1 Registreerumine

Kirjeldus: kasutaja registreerib ennast lehel

Kasutaja: õpilane

Eeltingimused: kasutaja on avalehel, kasutaja on sisse logimata

Põhivoog:

1. kasutaja klikib lingil „Registreeri“
2. kuvatakse väljad registreerumiseks vajaliku info sisestamiseks. [joonis 4]



The image shows a registration form titled "Registreeri" with a dark green header. Below the header are several input fields: "Matrikel:" with a white text box, "Parool:" with a white text box, "Parool (re-type):" with a white text box, "Eesnimi:" with a white text box, "Perenimi:" with a white text box, and "Rühm:" with a dropdown menu showing "1". At the bottom of the form is a green button labeled "Registreeri".

Joonis 4: Registreerimisvorm

3. Kasutaja täidab väljad
4. Kasutaja klikib „Registreeri“
5. Kuvatakse teade eduka registreerumise kohta ja kasutaja viiakse tagasi pealehele

Alternatiivvoog:

- kui mõni väljadest on tühi, selle matrikliga kasutaja juba eksisteerib, parool on lühem kui 6 tähemärki või paroolid ei klapi, siis kuvatakse vastav veateade.

Erandid: Kui tekib probleem andmebaasis andmete uuendamisel siis kuvatakse veateade.

Järelingimused: Loodud uus kasutaja. Kasutaja on pealehel. Kasutajat ei logita automaatselt sisse.

4.2.2 Sisse logimine

Kirjeldus: kasutaja logib sisse

Kasutaja: õpilane, õppejõud

Eeltingimused: kasutaja on avalehel, kasutaja on sisse logimata

Põhivoog:

1. kasutaja klikib lingil „Logi sisse“
2. kuvatakse väljad sisse logimiseks vajaliku info sisestamiseks. [Joonis 5]

Joonis 5: Sisselogimisvorm

3. Kasutaja täidab väljad
4. Kasutaja klikib „Sisene“
5. Kasutaja logitakse sisse

Alternatiivvoog:

- kui mõni väljadest on tühi või sellist kasutajat ei eksisteeri siis kuvatakse vastav veateade.

Erandid: Kui tekib probleem andmebaasis andmete uuendamisel, siis kuvatakse veateade.

Järelingimused: Kasutaja on sisse loginud. Kuvatakse kasutaja rollile vastavad menüüd.

4.2.3 Töö loomine

Kirjeldus: õppejõud loob töö

Kasutaja: õppejõud

Eeltingimused: õppejõud on sisse loginud

Põhivoog:

1. kasutaja klikib administreerimismenüüs lingil „Kontrolltöö“

2. Kuvatakse tööde nimekiri ja nupp „Lisa kontrolltöö“
3. kuvatakse väljad kontrolltöö loomiseks vajaliku info sisestamiseks. [Joonis 6]

Joonis 6: Töö loomise vorm

4. Kasutaja täidab väljad
5. Kasutaja klikib „Lisa“
6. Luuakse kontrolltöö
7. Kasutaja viiakse tagasi kontrolltööde nimekirja juurde

Alternatiivvoog:

- kui mõni väljadest on tühi siis kuvatakse vastav veateade.

Erandid: Kui tekib probleem andmebaasis andmete uuendamisel, siis kuvatakse veateade.

Järeldused: Kuvatakse kontrolltööde nimekiri. Andmebaasi on lisatud töö.

4.2.4 Töösse ülesannete lisamine

Kirjeldus: õppejõud loob töö

Kasutaja: õppejõud

Eeltingimused: õppejõud on sisse loginud. Eksisteerib töö, millesse ülesandeid lisada. Kasutaja asub avalehel.

Põhivoog:

1. kasutaja klikib administreerimismenüüs lingil „Kontrolltöö“
2. Kuvatakse tööde nimekiri.
3. Kasutaja klikib „Redigeeri“ selle töö real, millesse ta soovib ülesandeid lisada
4. Kuvatakse väljad töö nime ja kirjeldusega ning nupud töö kirjelduse/nime muutmiseks, töö ülesannete vaatamiseks ning töösse ülesande lisamiseks.
5. Kasutaja klikib nupul „Lisa ülesanne“
6. Kuvatakse ülesande loomiseks vajalikud väljad [Joonis 7]

Kontrolltöö: Eksam
Lisa ülesanne

Tekst:

Ülesande püstitus

Argumendid(eralda komaga):

Maksimumpunktid:

5

Oige vastus 1:

Oige vastus 2:

Oige vastus 3:

Joonis 7: Töösse ülesande lisamise vorm

7. Kasutaja täidab väljad ka klikib nupul „Sisesta“
8. Kuvatakse teade ülesande sisestamise kohta ja väljad tühjendatakse

Alternatiivvoog:

- kui teksti, argumentide või esimese vastuse väli on tühi, kuvatakse vastav veateade. Välju ei tühjendata. Ülesannet ei lisata.
- Kui rakendus tuvastab, et sisestatud vastus ei ole süntaktiliselt korrektne, kuvatakse vastav veateade. Välju ei tühjendata. Ülesannet ei lisata.

Erindid: Kui tekib probleem andmebaasis andmete uuendmisel, siis kuvatakse veateade.

Järeldingimused: Väljad tühjendatakse. Kasutaja saab jätkata ülesannete lisamist.

4.2.5 Töö ülesannete redigeerimine

Kirjeldus: õppejõud muudab töö ülesandeid

Kasutaja: õppejõud

Eeltingimused: õppejõud on sisse loginud. Eksisteerib töö, millesse on lisatud vähemalt 1 ülesanne.

1. kasutaja klikib administreerimismenüüs lingil „Kontrolltöö“
2. Kuvatakse tööde nimekiri.
3. Kasutaja klikib „Redigeeri“ selle töö real, mille ülesandeid ta soovib redigeerida
4. Kuvatakse väljad töö nime ja kirjeldusega ning nupud töö kirjelduse/nime

muutmiseks, töö ülesannete vaatamiseks ning töösse ülesande lisamiseks.

5. Kasutaja klikib nupul „Vaata ülesandeid“
6. Kuvatakse töö ülesanded. Iga ülesande kohta kuvatakse selle ülesande andmed. [Joonis 8]

1.
Ülesanne: Valjenda $x < y$
Argumendid: x, y
Punkte: 5
Vastus 1:
 $\exists z (\neg (z=0) \& (x+z=y))$
Kontrolli süntaksi
Vastus 2:
Kontrolli süntaksi
Vastus 3:
Kontrolli süntaksi
Muuda Kustuta

Joonis 8: Töö ülesande redigeerimise vorm

7. Kasutaja muudab soovitud väljade väärtused
8. Vastuse välja väärtuse muutmisel tuleb süntaksi kontrollimiseks vajutada välja all asuvat nuppu „Kontrolli süntaksi“. Korrektse süntaksi korral kuvatakse nupu kõrval vastav teade.
9. Kasutaja klikib „Muuda“.
10. Kuvatakse teade andmete muutmise kohta.

Alternatiivvoog:

- kui teksti, argumentide või esimese vastuse väli on tühi, kuvatakse vastav veateade. Ülesannet ei muudeta.

Erandid: Kui tekib probleem andmebaasis andmete uuendmisel, siis kuvatakse veateade.

Järeldingimused: Kasutaja näeb endiselt töö ülesandeid ja saab jätkata ülesannete redigeerimist.

4.2.6 Töö avamine

Kirjeldus: õppejõud avab töö lahendamiseks

Kasutaja: õppejõud

Eeltingimused: õppejõud on sisse loginud. Töö mida soovitakse avada on suletud.

Põhivoog:

1. kasutaja klikib administreerimismenüüs lingil „Kontrolltöö“
2. Kuvatakse tööde nimekiri

3. Kasutaja klikib „Muuda olekut“ selle töö real, mida ta soovib avada.
4. Kuvatakse tööde nimekiri
5. Töö olek on nüüd „Avatud“

Erandid: Kui tekib probleem andmebaasis andmete uuendamisel, siis kuvatakse veateade.

Järelingimused: Töö on avatud ja õpilased saavad selle ülesannetele vastata.

4.2.7 Töö ülesannete lahendamine

Kirjeldus: õpilane lahendab töö ülesandeid

Kasutaja: õpilane

Eeltingimused: õpilane on sisse loginud. Eksisteerib avatud töö. Avatud töös on ülesandeid.

Põhivoog:

1. kasutaja klikib õpilase menüüs lingil „Kontrolltööd“
2. Kuvatakse avatud tööde nimekiri
3. Kasutaja klikib „lahendama“ selle töö real, mille ülesandeid ta soovib lahendada.
4. Kuvatakse töö ülesanded [Joonis 9]

Joonis 9: ülesandele vastamine

5. Kasutaja sisestab vastuse
6. kasutaja klikib nupul „Vasta“
7. rakendus kontrollib vastuse süntaksi ja salvestab korrektse süntaksi korral vastuse
8. Kui rakendus suudab tuvastada et vastus on õige, siis kuvatakse ka vastav teade.
9. Nupu „Vasta“ kõrval kuvatakse salvestatud vastus

Alternatiivvoog:

- kui õpilase vastuse süntaks pole korrektne, siis kuvatakse vastav veateade ja vastust ei salvestata.
- Kui töö on vahepeal suletud, siis kuvatakse vastav teade ja kasutaja viiakse tagasi avalehele. Vastust ei salvestata.

Erandid: Kui tekib probleem andmebaasis andmete uuendamisel, siis kuvatakse veateade.

4.2.7 Soorituse hindamine

Kirjeldus: õppejõud hindab sooritust

Kasutaja: õppejõud

Eeltingimused: õppejõud on sisse loginud. Eksisteerib sooritus.

Põhivoog:

1. kasutaja klikib õppejõu menüüs lingil „Sooritud“
2. Kuvatakse soorituste nimekiri
3. Kasutaja klikib nupul „Vaata“ selle soorituse real, mida soovib hinnata
4. Kuvatakse töö ülesanded ja õpilase vastused neile ülesannetele. [Joonis 10]

Ülesanne: $z=x-y$
Õige vastus 1: $x=z+y$
Õige vastus 2: $x=y+z$
Õige vastus 3:
Õpilase vastus: $x=y+z$
Sisesta tagasiside:
tubli
Punkte:
5
Kinnita
Hinda

Joonis 10: Soorituse lahenduse tagasiside määramine

5. Kasutaja sisestab tagasiside ja määrab punktid iga ülesande kohta ja klikib iga ülesande juures Kinnita
6. kasutaja klikib nupul „Hinda“
7. Sooritus loetakse hinnatuks ja kasutajale kuvatakse soorituste nimekiri
8. Nupu „Vasta“ kõrval kuvatakse salvestatud vastus

Alternatiivvoog: kui soorituses pole ühtegi lahendust kuvatakse vastav veateade.

Erindid: Kui tekib probleem andmebaasis andmete uuendamisel, siis kuvatakse veateade.

5. Täiendamise ja edasiarendamise võimalused

Edasiarendamise võimalusi on palju. Ilmne eesmärk oleks täiendada rakendust selliselt, et oleks võimalik lahendada ülesandeid teiste hulkade ja teistsuguse signatuuriga valemite korral.

Lisaks sellele on mõeldav vastuse õigsuse kontrolli täiustamine selliselt, et rakendus arvestaks ka asjaolu, et operaatorite $=$, $\&$ ja \vee argumendid kommutatiivsed. Hetkel ei suuda rakendus tuvastada, et kui õigeks vastuseks on määratud näiteks $x=y+z$, siis tuleks õigeks lugeda ka vastused $x=z+y$, $z+y=x$ ja $y+z=x$.

Ka valminud rakenduse kasutajamugavuse parendamine ei ole välistatud.

6. Kokkuvõte

Bakalaureusetöö tulemusena valmis rakendus, mis täidab püstitatud eesmärgi luua veebirakendus väljendusülesannete lahendamiseks naturaalarvude hulgal, signatuuriga $\langle 0,1;+,*;=\rangle$. Loodud on vahendid kasutajate haldamiseks, tööde administreerimiseks, soorituste hindamiseks ja tagasiside andmiseks. Seda saab rakendust kasutada nii harjutuskeskkonnana kui ka kontrolltöö ülesannete lahendamiseks.

Tuleb mõõnda, et kohati jätab kasutusmugavus soovida ja arvatavasti tuleb tulevikus sellele probleemile tähelepanu pöörata, kuid rakendus on kasutatav ja enamasti kasutajasõbralik.

Web application for expressing predicates as formulas

Bachelor thesis

Siim Loog

Summary

This bachelor thesis set out to create a web application that is able to validate the syntax of a formula defined on the set of natural numbers with a signature of $\langle 0, 1; +, *, = \rangle$. Secondary objectives included creating a framework of tools to create questions, organize questions into assignments, solve assignments, grade the results of assignments, administer assignments and administer users.

To complete these tasks, a variety of technologies were used: Javascript, Ajax, MySQL, HTML, CSS and PHP.

The following tools were developed and included in the application:

- Syntax validator
- User management. Consists of tools used to register, to log in and to administer the account of a user.
- Formula editor, consists of a field where to enter the formula and buttons with which to insert special characters such as \forall , \exists , \neg , \supset , $\&$ and \vee . In some cases the buttons also have to update the status of the syntax check for the formula that is currently being edited.
- Assignment management. Allows the teacher to create and edit assignments. Also has utilities for adding questions to the assignments and editing those questions.
- Grading utility. Enables the teacher to grade assignments and give feedback.

The application is designed to be easy to set up and with minimal requirements. The only requirements set to the system where the environment is set up are:

- PHP support (tested with 5.3)
- MySQL server (tested with 5.1)

Kasutatud materjalid

1. Entscheidungsproblem, <http://en.wikipedia.org/wiki/Entscheidungsproblem>, (20.06.2011)
2. R. Palm, R. Prank „Sissejuhatus matemaatilisse loogikasse“, Tartu Ülikooli kirjastus, 2004
3. Webct: „<http://www.e-uni.ee/index.php?main=177>“ (18.06.2011)
4. J. Barwise, J. Etchemendy, „The language of first-order logic“ Second Edition, Center for the Study of Language and information, 1991
5. HTML, <http://et.wikipedia.org/wiki/HTML>, (18.06.2011)
6. CSS, <http://et.wikipedia.org/wiki/CSS>, (18.06.2011)
7. Javascript, <http://et.wikipedia.org/wiki/Javascript>, (18.06.2011)
8. Prototype, <http://www.prototypejs.org/learn> , (18.06.2011)
9. JSON, <http://et.wikipedia.org/wiki/JSON>, (18.06.2011)
10. JavaScript MD5, <http://pajhome.org.uk/crypt/md5>, (18.06.2011)
11. HTML encoder and decoder tool, <http://www.strictly-software.com/htmlencode>, (18.06.2011)
12. PHP, <http://et.wikipedia.org/wiki/PHP>, (18.06.2011)
13. Smarty, <http://www.smarty.net/>, (18.06.2011)
14. MySQL, <http://et.wikipedia.org/wiki/MySQL>, (18.06.2011)
15. Ajax, <http://et.wikipedia.org/wiki/Ajax>, (18.06.2011)
16. MVC mudel, viide: <http://ahaageeks.blogspot.com/2010/03/mvvm-model-view-viewmodel.html>, (18.06.2011)
17. Rakenduse installeerimisjuhend: <http://vaffel.planet.ee/juhend.pdf>

Lisad

Lisa 1: lubatud järgnevuste tabel

Tabel 2: Lubatud järgnevuste tabel.

Sümbol\alle võib järgneda	argument või 0 või 1	+ ja *	kvantor	seotud \supset	lõpp
indiviidimuutuja või 0 või 1					
+ ja *					
kvantor					
seotud muutuja					
seotud muutuja (kvantori jarel)					
=					
(
)					
&, V, \supset					
\neg (eitus)					
algus					
Sümbol\alle võib järgneda	seotud muutuja (kvantori jarel)	=	()	&,V \neg (eitus)
indiviidimuutuja või 0 või 1					
+ ja *					
kvantor					
seotud muutuja					
seotud muutuja (kvantori jarel)					
=					
(
)					
&, V, \supset					
\neg (eitus)					
algus					

Lisa 2: rakenduse lähtekood: <http://vaffel.planet.ee/valjendamine.zip> või <http://code.google.com/p/predikaat/>