

TARTU ÜLIKOO
MATEMAATIKA-INFORMAATIKA TEADUSKOND
ARVUTITEADUSE INSTITUUT

Matemaatikaolümpiaadide portaali andmebaasi ja kasutajaliidese arendamine

Bakalaureusetöö

Autor: Karl-Martin Ilves

Juhendaja: Härmel Nestra

Autor: „juuni 2011

Juhendaja: „juuni 2011

Professor: „juuni 2011

Tartu 2011

Sisukord

1. Sissejuhatus	3
2. Matemaatikaolümpiaadide portaal	5
2.1 Tulemuste lisamine	6
2.2 Tulemuste otsimine	8
2.2.1 Veebikülastaja vaade.....	9
2.2.2 Administraatori vaade	10
3. Portaali arhitektuur	12
3.1 Füüsiline arhitektuur	12
3.2 Loogiline arhitektuur.....	12
3.2.1 Mudel	12
3.2.2 Vaade.....	13
3.2.3 Kontroller	13
3.3 PHP Zend Framework.....	14
4. Testimine.....	16
5. Kokkuvõte	19
6. Resume.....	20
7. Viited.....	21
Lisa 1. Kasutatud tehnoloogiad.....	22
Lisa 2. Portaali kood	23

1. Sissejuhatus

Matemaatika olümpiaade on peetud väga pikka aega ja nende tulemusi hakati digitaalsel kujul sisestama peaaegu 20 aastat tagasi. Hetkel saab matemaatika olümpiaadide tulemusi näha matemaatikaolümpiaadide portaalis[7], kus on võimalik vaadata tulemusi, ülesandeid ning ülesannete lahendusi. Probleem seisneb selles, et praegu kasutusel olev süsteem on aegunud, andmete kuvamine ja sisestamine võiksid olla tunduvalt kasutajasõbralikumad. Tingitud on see tehnika arengust, sest tänapäeval on veebirakenduste võimalused oluliselt laienenud ja paranenud. Teine suurem probleem on käesoleva süsteemi edasiarendatavus. Nimelt pole kasutatud üldisi arendamise skeeme ega raamistikke, mis muudab projekti eelnevat arendustööd mittetundval arendajal töö väga raskeks. Seepärast peab kasutusele võtma üldtuntud võtted, mida üleüldiselt veebikeskkondade arendamisel järgitakse.

Käesoleva töö peamiseks eesmärgiks on ehitada infosüsteem, kus administraatorid saavad mugavalt tulemusi sisestada ning hiljem vajadusel ka muuta. Loodud süsteemi kaasautoriks on Tartu Ülikooli tudeng Margus Sellin[12]. Mõlemad tudengid arendasid matemaatikaolümpiaadide portaali bakalaureusetöö raames. Karl-Martin Ilvese panuseks portaalis on võimaldada veebikülastajatel näha erinevate olümpiaadide tulemusi, tehes veebipõhiseid päringuid erinevate ajavahemike ja võistluste kohta. Selleks tuli arendada otsingusüsteem, kus kasutaja saab valida talle sobivad parameetrid ning nende põhjal süsteem otsib kõik sobivad kirjed. Kui on leitud sobivad kirjed, saab kasutaja ka näha, mis ülesannete eest keegi kui palju punkte on saanud. See tähendab, punktide peale vajutades viib süsteem kasutaja vastava ülesande juurde. Ülesande näitamisel võimaldatakse näha nii ülesande tekste, lahendusi ja vastuseid. Veel jäi Karl-

Martin Ilvese ülesandeks võimaldada portaali administraatoritel käsitsi tulemusi lisada, muuta ja kustutada. Sellest on täpsemalt räägitud peatükis 2.2.2 Administraatori vaade.

Margus Sellini ülesandeks sai arendada automaatne tulemuste lisamise süsteem, kas siis väliselt veebilehelt või etteantavast failist. Süsteem oskab välistelt lehtedelt lugeda HTML tabelitest tulemusi ning need andmebaasi lisada. Samuti tegeles Margus Sellin portaali kujundusega, lisas portaalile tegevuste logimise süsteemi ja võimaldas kasutajatel näha statistikat tulemuste raames.

Töö jaguneb neljaks suuremaks osaks: alguses kirjeldatakse portaali funktsionaalsust, seejärel on välja toodud portaali arhitektuur, siis näitab käesolev töö missuguste tehnoloogiate abil on eesmärk saavutatud ja lõpetuseks näeb testimise tulemusi ning sellest järeldatavaid portaali tugevusi ja nõrkusi.

Portaali on võimalik näha aadressil <http://ats.cs.ut.ee/olymp/>, kus asub Karl-Martin Ilvese ja Margus Sellini bakalaureuse tööde tulemus.

Portaali kood on lisas.

2. Matemaatikaolümpiaadide portaal

Matemaatikaolümpiaadide portaal on infosüsteem, kus veebikülastajad saavad näha erinevate matemaatikaolümpiaadide tulemusi. Veebikülastajad ehk tavakasutajad moodustavad ühe osa portaali kasutajatest. Teiseks osapooliks on administraatorid, kes kontrollivad andmete õigsust ning tegelevad ka uute andmete sisestamisega. Viimaste jaoks on matemaatikaolümpiaadide portaalil eraldi administreerimise liides. Käesoleva töö eesmärgiks oli eelkõige võimaldada administraatoritel administraatori paneelis (Joonis 3::3) uusi tulemusi sisestada, olemasolevaid tulemusi kas siis omakorda muuta või kustutada. Teiseks ülesandeks sai tulemuste otsimine. Lisaks tulemuste otsimisele pidi süsteem võimaldama näha ülesandeid, mille eest punkte saadi. Ehk tulemusreas, kus on näha punkte iga ülesande jaoks eraldi, peab olema võimalik näha, mis ülesande eest punktid on saadud. See lahendati kasutaja jaoks võimalikult kergelt: kõik punktid peale „Test“ veeru omade on lingid (Joonis 3::6), millele vajutades avatakse uus vaheleht, kus on näha vastav ülesanne, mille eest punktid pandi. Samal lehel on näha ka ülesande lahendused ja vastused (Joonis 5).

Arendamise vaatenurgast oli algse andmebaasi ülesehituse tõttu raske kokku viia tulemuse punkte vastavate ülesannetega. Nimelt ei olnud andmebaasis võimalik tulemust otseselt kokku viia rühmaga, mille all tulemus saadi. Küll aga oli iga tulemusega seotud klass, mis näitas, kus klassis osaleja parajasti käis, kui olümpiaadil osales. Kõige targemaks probleemi lahenduseks tundus uue tabeli loomine, kus viiakse vastavusse rühm ja klass. Seda võimaldas seos rühmade ja klasside vahel: iga klass oli rühmade tabelis esindatud (rühma kirjelduses). Küll aga ei eksisteerinud nende vahelist

seost. Uus tabel viis klassi vastavusse rühmaga, mis kokkuvõttes võimaldas punkte vastavusse viia ülesannetega.

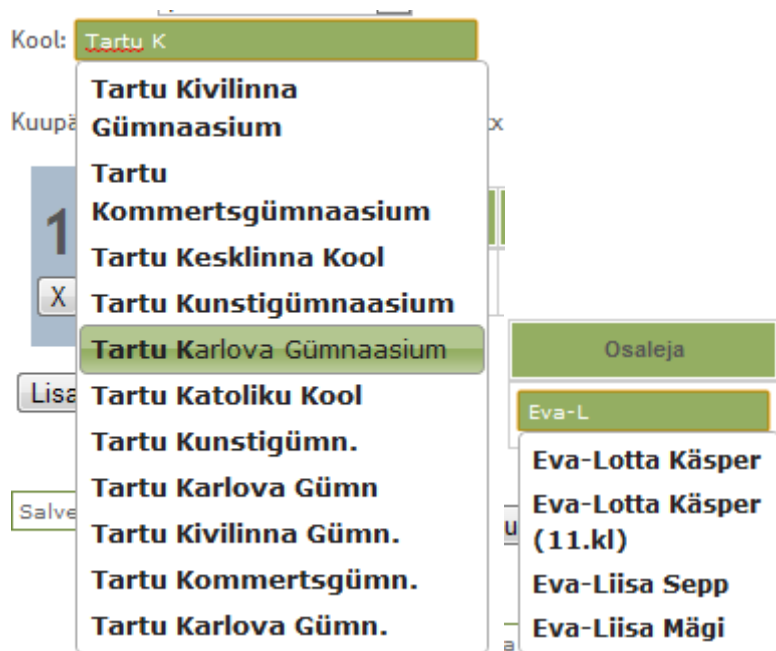
2.1 Tulemuste lisamine

Administreerimise liidese esimeseks põhilisemaks funktsionaalsuseks on uute tulemuste lisamise võimalus. Selle arendamise käigus arvestati administraatori ettepanekuid, mille tulemusena peab kõigepealt valima võistluse, mille alla tulemused kuuluvad. Seejärel tuleb valida kool ja kuupäev. Kuupäeva valiku juures ei pea administraator mitte käsitsi seda sisestama, vaid kuvatakse visuaalne kalender, kus saab navigeerida kuude ja aastate kaupa ning sobiva kuupäeva leidmisel see valida. Jäetud on muidugi ka käsitsi sisestamise võimalus ning on antud ka näide, mis kujul käsitsi kuupäeva sisestama peab (Joonis 2::3),. Peale seda küsitakse administraatorilt osaleja nime, klassi ja punkte erinevate ülesannete eest (Joonis 2::4),. Selliseid ridasid on võimalik ka duplitseerida. Selleks on loodud šabloon (Joonis 2::5), mille saab andmetega täita ning valida, mitu sellist rida soovitakse. Seejärel lisatakse nii mitu uut tulemuste rida eelnevalt täidetud andmetega.

Kokkuvõtlikult, tulemusi saab lisada mitme kaupa, kuid seda võistluse, kooli ja kuupäeva lõikes.

Tulemuse lisamisel tuli administraatoril kool valida mitmesaja olemasoleva kooli seast või siis vajadusel pidi olema võimalus uus kool lisada. Algselt lahendati kooli valik rippmenüüga, kuid selle kasutamine mitmesaja valikuvариandiga on väga ebamugav. Seepärast otsustati kasutusele võtta lahendus, mis lubas kasutajal kooli nime ise trükkida. Süsteem assisteerib kooli valikul vastavalt kasutaja sisestatud sõnele lähimate

koolide nimede pakkumisega: kooli valikul tuleb sobiv kool valida süsteemi poolt pakutavate koolide hulgast: kui administraator parajasti kooli nime trükib, tekib rippvalikmenüü, kust peab valima olemasolevate koolide seast sobiva. Kui sisestatud on 2 tähte, mis on minimaalne tähtede arv valikute pakkumise ilmutamiseks, siis peale iga uue tähe sisestamist uuendatakse rippvalikmenüü valikuvариante. Lühidalt, mida rohkem on sisestatud tähti, seda täpsemaid valikuvариante pakutakse (Joonis 4::1). Samasugune süsteem on ka õpilase nime kirjutamisel, kus õpilase nime sisestamisel pakutakse vastavalt sisestatud tähtedele eelnevalt osavõtnute nimesid. Järgnevad 2 pilti kirjeldavad piltlikult ülaltoodud lahendust.



Joonis 1: Kooli ja osaleja nime pakkuja

Juhul kui süsteemis sisestatud kooli või nime ei ole, lisatakse andmebaasi vastav uus kirje. Kooli sisestamisel teavitatakse sel juhul kasutajat, et tema poolt sisestatud kooli ei ole veel andmebaasis ning salvestamisel tekitatakse andmebaasi uus kool (Joonis 2::1),. Uue kooli lisamisel küsitakse ka lisatava kooli piirkonda (Joonis 2::2), mille saab valida

riipmenüüst. Sel juhul ilmub sisestatud kooli nime alla riipmenüü, kust kasutaja saab valida vastava piirkonna. Piirkonna valikut on vaja piirkondlikel võistlustel määrata, mis piirkonda võistleja kuulub.

Vali võistlus: piirkonnavoore

Kool: Tartu Kunstigümnaasium UUS **1**
Antud kooli ei ole andmebaasis. Uue kooli lisamiseks valige palun piirkond:

Piirkond: Tallinna linn **2**

Kuupäev: kujul: xxxx-xx-xx (aasta-kuu-päev) **3**

1	Osaleja	Klass	Test	p1	p2	p3	p4	p5	p6	4
X		7								

Lisa uusi ridasid **5**

2	Osaleja	Klass	Test	p1	p2	p3	p4	p5	p6
		7							

Mitu sellist rida soovite: 1

Lisa read Puhasta väljad

Salvesta tulemused

Joonis 2: Tulemuste lisamine

2.2 Tulemuste otsimine

Tulemusi otsimine käib nelja parameetri järgi, neist üks on kohustuslik – võistluse määramine. Erijuhtum on „lahtine võistlus“, seal saab aasta juures ka eraldi määrata, kas soovitakse kevadise, sügisese või talvise võistluse tulemusi.

Tulemuste otsimisel saab eristada kahte juhtu. Kui tulemusi otsib administraator, lubatakse tal leitud tulemusi ka muuta/kustutada (Joonis 3::5). Tavaline veebikülastaja saab aga tulemusi otsida ning neid vaadata. Samuti on lisatud võimalus näha, mis ülesande eest punktid on saadud (Joonis 3::6): see funktsionaalsus kujutab endast eelnevalt kirjeldatud olukorda, kus kasutaja saab vajutada tulemuses kajastuvate punktide peale, mille korral avatakse uuel vahelehel vastav ülesanne koos lahenduste ja vastustega (Joonis 5). Peale selle on võimalik vaadata kõiki ülesandeid, mis tulemusega seotud võistlusel olid. Selleks on iga tulemusrea taga link, mis avab uuel vahelehel tulemusega seotud võistluse kõikide ülesannete loetelu (Joonis 3::4). Viimases loetelus on kõikidel loetelus olevatel ülesannetel võimalus näha ka selle lahendusi ja vastuseid, selleks tuleb vajutada vastaval lingil. Lingile vajutades kuvatakse kõik valitud ülesande tekstid, vastused, lahendused ja kommentaarid. Samuti kuvatakse kõik eelnev erinevates keeltes, kui andmebaasis leiduvad vastavad kirjed eri keeltes.

2.2.1 Veebikülastaja vaade

Regulaarne veebikülastaja, st. ilma privileegideta kasutaja, saab otsingu vaates sisestada sobivad parameetrid: võistlus, aasta/aastaaeg, klass ja kool. Peale otsingu toimumist kuvatakse tulemused tabeli kujul, kusjuures näidatakse ainult vajalikke veerge – otsingus märgitud parameetrid ei kajastu enam otsingutulemuste veergude hulgas, välja arvatud kuupäev. Kuvatava otsingu tulemuste tabel on sorteeritav, sorteerimine toimub kliendi masinas. Sorteerimise eest hoolitseb *jQuery*[3] lisamoodul *TableSorter 2.0*[E], mis indekseerib kõik veerud ja nendes asetseva informatsiooni ning vastavalt veergudes asuvatele andmetele oskab neid kasvavalt ja kahanevalt sorteerida. Sorteerimise teostamiseks tuleb vajutada veeru pealdisele (Joonis 3::2). Mitme veeru järgi

sorteerimiseks tuleb all hoida SHIFT nuppu, see võimaldab märkida, missuguste veergude järgi otsimine toimub. Sorteerimisel saab valida, kas veerg tuleks järjestada kasvavalt või kahanevalt, vastavalt nii mitu korda veeru päisele vajutades. Märku annab nool päises, mispidi tulemused parasjagu järjestatud on.

2.2.2 Administraatori vaade

Lisaks kõigele eelnevale, seoses tulemuste otsimisega, on administraatorile antud lisavõimalused. Administraatorile saab otsimisel kuvatavaid tulemusi muuta ja kustutada. Muutmisel viiakse administraator uuele vahelehele (Joonis 4), kus saab muuta väljavalitud tulemuse andmeid, samuti on võimalik naasta eelneva otsingu juurde (Joonis 4::2). Muutmisel on võimalik administraatoril kasutada jällegi nime ja kooli sisestamisel süsteemi automaatpakkujat.

MATEMAATIKAOLÜMPIAADIDE TULEMUSTE INFOSÜSTEEM

Tulemuste otsimine

Võistlus: piirkonnavoore

Aasta: 2001

Klass: 9

Kool: Tartu Kuns

Tartu Kunstigümnaasium

Tartu Kunstigümn.

Otsi

3 ADMINPANEEL

[Staatilised lehed](#) ➤

[Tabelite lisamine](#) ➤

[Käsitsi lisamine](#) ➤

Perekonnanimi	Eesnimi	Test	p1	p2	p3	p4	Summa	Kool	Tegevused
Prostakova	Jekaterina	18	7	7	7	7	46	Pähklimäe Gümnaasium	ülesanded muuda kustuta
Prostakova	Viktorija	16	7	6	7	7	44	Pähklimäe Gümnaasium	ülesanded muuda kustuta
Novozhilov	Fjodor	19	6	7	7	5	44	Narva Humanitaargümnaasium	ülesanded muuda kustuta
Roosmaa	Kadri	17	7	6	7	7	44	Tln. Prantsuse Lütseum	ülesanded muuda kustuta
Märtn	Priit	17	7	7	7	3	41	Elva Gümnaasium	ülesanded muuda kustuta

Joonis 3: Tulemuste otsimine portaalis

ADMINPANEEL

[Staatilised lehed](#) ➤

[Tabeite lisamine](#) ➤

[Käsitsi lisamine](#) ➤

Võistlus	Osaleja	Klass	Kool	Kuupäev
piirkonnavoore	Irina Voronko	9	Kee Koeru Keskkool	2001-01-27

1

Test	p1	p2	p3	p4	p5	p6
15	2	2	0	0		

Salvesta tulemus Tagasi eelmise otsingu juurde

2

Joonis 4: Tulemuse muutmine

Rühma nimetus: 9. klass

Lisatud: 2005-08-20 17:39:48

[Tabelite lisamine](#) ➤

[Käsitsi lisamine](#) ➤

Ülesande tekstid:

Tekst	Keel	Lisatud
Joonisel kujutatud kolm ringjoont on võrdse raadiusega r , nende keskpunktid paiknevad ühel sirgel ja kaks äärmist ringjoont läbivad keskmise ringjoone keskpunkti. Leia neist ringjoontest moodustuva kujundi ümbermõõt.	Eesti keel	2005-08-20 17:39:48

Ülesande vastused:

Vastus	Keel	Lisatud
10πr ² 3.	Eesti keel	2005-08-23 23:37:46

Ülesande lahendused:

Lahendus	Keel	Lisaja	Email	IP	Kontrollitud	Lisatud
Olgu vaadeldavate ringjoonte keskpunktid O1, O2 ja O3 ja lõikepunktid A, B, C ja D, nagu joonisel näidatud. Moodustuva kujundi rajajoon koosneb neljast ringjoone kaarest ~ AB, ~ BC, ~ CD ja ~ DA. Et kolmurgud O1AO2, O1BO2, O2CO3 ja O2DO3 on võrdkülgised (nende kõik küljed on pikkusega r), siis on kaarte ~ BC ja ~ DA pikkus 1 ringjoone ümbermõõdust ning kaarte ~ AB ja ~ BC pikkus 2/3 ringjoone ümbermõõdust. Kujundi ümbermõõt on seega $(2 \cdot 1/6 + 2 \cdot 2/3) \cdot 2\pi r = 10\pi r 3$.	Eesti keel				Jah	2005-08-24 00:43:23

Ülesande kommentaarid: Ülesande kommentaare ei leitud.

Joonis 5 : Ülesande vaatamine

3. Portaali arhitektuur

3.1 Füüsiline arhitektuur

Portaali füüsiline arhitektuur on täpsemalt kirjeldatud Margus Sellini bakalaureuse töös[12], mida kirjutatakse paralleelselt käesoleva tööga. Sealt on võimalik lugeda täpsemalt kasutatava klient-server-andmebaas mudeli kohta.

Peale füüsilise arhitektuuri on järgitud ka loogilise arhitektuuri mustreid.

3.2 Loogiline arhitektuur

Matemaatikaolümpiaadide portaali arendamisel kasutati MVC (*model-view-controller*) ehk eesti keeles mudel-vaade-kontroller arhitektuurimustrit. MVC arhitektuurimuster jagab süsteemi eraldiseisvatesse osadesse: mudel, vaade ja kontroller. See lubab neid osasid eraldiseisvalt arendada ja testida. Teisisõnu on MVC mudelit kasutades eraldi vaadeldavad esitusloogika ja ärilooika. [1]

3.2.1 Mudel

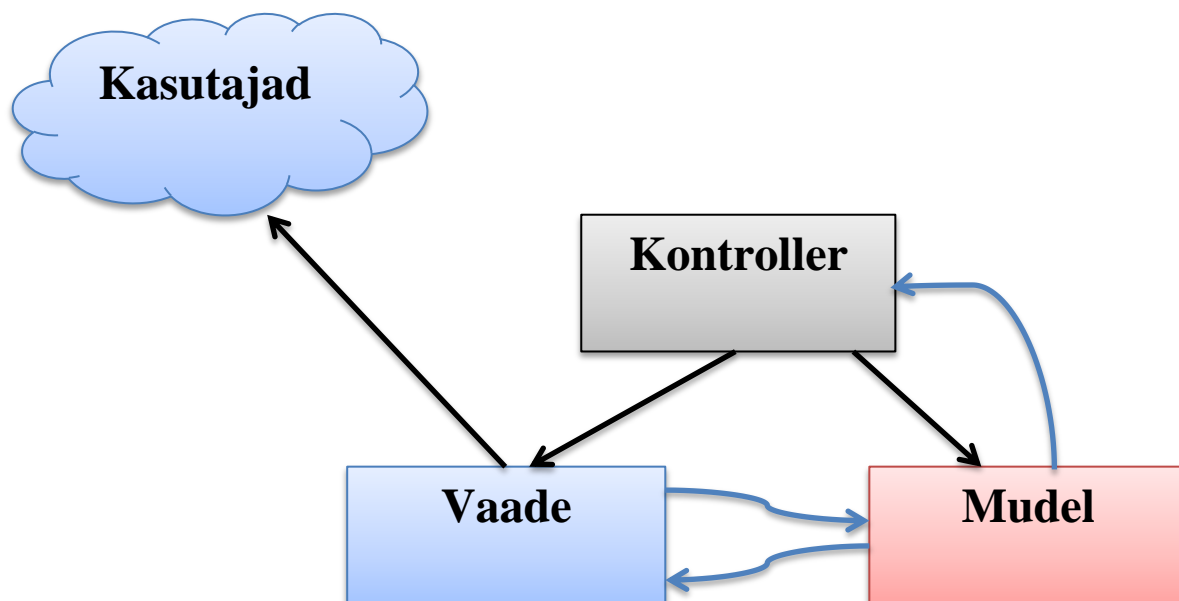
Mudelis toimub andmebaasiga suhtlemine. Mudelis valmistatakse ette päring, mis saadetakse andmebaasi serverile. Andmebaasi serveris täidetakse päring ning päritud andmed saadetakse tagasi mudelisse, kus neid vajadusel veel töödeldakse. Mudeli tööd reguleerib kontroller, kuid ka vaatest saab otse mudeli poole pöörduda andmete pärimiseks. Siiski on eelistatud, et mudeli ja vaate vahelist suhtlust kontrolliks kontroller, sest siis on arendajatel kergem leida andmete liikumise mustreid. [1]

3.2.2 Vaade

Vaate ülesandeks on informatsiooni viimine kasutajale arusaadavale kujule. Vaade vastutab andmete esitluse eest ning võimaldab kasutajal süsteemi funktsionaalsust kasutada. Antud töö raames kasutati palju *JavaScript*-i[10] ühte levinuimat raamistikku, *jQuery*-t. See võimaldas palju operatsioone teostada kliendi arvutis, mis vähendas kontrolleri ja mudeli, seega ka terve süsteemi, koormust. Andmete sorteerimised suudeti tänu *jQuery*-le viia serverilt kasutaja arvuti õlule ning samuti suudeti ära hoida paljud ebavajalikud lehtede laadimised. [1]

3.2.3 Kontroller

Kontroller kontrollib vaate ja mudeli tööd, vastutab info liikumise eest õigest kohast õigesse kohta. Kontroller teeb kindlaks, kust mudelist on parajasti informatsiooni vaja ning vaatab, et see õigesse vaatesse jõuaks. Samuti saab kontroller suunata käesoleva töö edasi järgmisele kontrolleri. Eelistatud on olukord, kus kõik põhioperatsioonid käivad läbi kontrolleri, kuid sellele vaatamata saavad vaade ja mudel ka omavahel suhelda, ilma kontrolleri abi vajamata. Arendajad üritavad arendamise käigus kontrolleri võimalikult lihtsa hoida, lükates kõik võimalikud operatsioonid mudeli või siis vaate õlule. Kontrolleris peab nähtav olema tähtsama informatsiooni liikumine ja tegevuste järjekord, vähem tähtsamad toimingud võib kanda kas siis mudelisse või vaatesse. Järgmisel joonisel näete lihtsat skeemi *model-view-controller* tööpõhimõttest. [1]



Joonis 6: *model-view-controller* (<http://audit.qos.ch/images/architecture-small.jpg>)

Käesoleva töö raames valmis kaks kontrolleriit. Üks neist vastutab otsingu ja kõige sellega seonduva töötamise eest. Teine kontrollib andmete lisamist, muutmist ja kustutamist.

3.3 PHP Zend Framework

Portaali arendamisel kasutati *Zend* raamistikku. See on tasuta, objekt-orienteeritud raamistik, mis on kirjutatud spetsiaalselt PHP 5-e[A] jaoks. Zend raamistik võeti kasutusele, kuna portaali arendajad on sellega eelnevalt kokku puutunud ning Zend omab suurepärast *model-view-controller* mudelit. Viimane on kergesti arusaadav ning võimaldab tulevikus portaali kergema vaevaga edasi arendada. Edasi arendamise muudab hõlpsaks asjaolu, et Zend raamistikul põhinevad projektid tuleb arendada üldtuntuid arendamise võtteid kasutades, seega uus arendaja üldjuhul teab koheselt midagi muutes, kus ta muutused ellu peab viima. [2]

Zend raamistik on mitmed funktsioonid juba valmis kirjutatud, seega ei pea arendajad kõike enam ise uuesti kirjutama. Samuti on Zend raamistikus arendatud projekte kerge hooldada. Kuna Zend raamistik on pideval edasi arendamisel, siis võib tulevikus oodata lisafunktsioone ja uusi lahendusi, mida saab kerge vaevaga matemaatika olümpiaadide portaali integreerida. [2]

Zendi raamistik omab ka väga põhjalikku dokumentatsiooni[8] koos näidetega, kust saab probleemidele hõlpsasti lahendusi leida ning oma teadmisi täiendada. Zend raamistiku populaarsuse tõttu on tekkinud ka palju harrastajaid, kes õpetavad raamistikku kasutama ning tegelevad inimeste probleemide lahendamisega. Kõik see tagab arendajatele hea õppimise ja töötamise võimalused. [2]

Portaali arendamisel kasutatud tehnoloogiad on välja toodud lisas [Lisa 1. Kasutatud tehnoloogiad].

4. Testimine

Süsteemi testimisel võeti vaatluse alla süsteemi kiirus ning korrektsus. Portaali kiirust testiti aga ainult otsingusüsteemi juures, sest mainitud punkt saab olema kasutajate poolt kõige kasutatavam ning kõige suurem andmete voog leiab samuti aset otsingusüsteemis. Samuti ei võimaldanud aja nappus testida kõiki süsteemi punkte.

Testimine toimus PHP-sse sisseehitatud funktsioonidega ning lehe kiiruse hindamisel kasutati ka rakenduse *Google Page Speed*[9] abi. Otsingusüsteemi korrektsust testiti arendamise käigus, täpsemaid aruandeid ette näidata ei ole. Arendamise käigus veenduti, et portaal käituks vastavalt kasutaja sooritatud tegevustele.

Kõigepealt käsitleme lihtsamat süsteemi testimise meetodit, lehe laadimise aja mõõtmist sisseehitatud funktsioonidega. Aega mõõdeti kahes erinevas punktis: esiteks kontrolleris, kus toimub andmete pärimine ning ettevalmistamine vaatesse saatmiseks ja teiseks vaates, kus koostatakse kasutajale kuvatav leht. Põhjus, miks aegu eraldi vaadata, on tingitud asjaolust, et kontrolleris tuleb kõik tulemused ükshaaval läbi käia ning teha päringust saadud andmetega mõningad muudatused.

<i>Tulemusi päringust</i>	<i>Katse 1 (sekundites)</i>	<i>Katse 2 (sekundites)</i>	<i>Katse 3 (sekundites)</i>	<i>Katse 4 (sekundites)</i>	<i>Katse 5 (sekundites)</i>	<i>Keskmine (sekundites)</i>
167	0.05562	0.03945	0.03778	0.04071	0.03913	0.04254
602	0.18925	0.17095	0.17318	0.16586	0.15443	0.17073
9808	2.87617	3.09599	3.27441	3.06869	2.52037	2.96713

Tabel 1 : Lehe ehitamise aeg kuvamiseks

<i>Tulemusi päringust</i>	<i>Katse 1 (sekundites)</i>	<i>Katse 2 (sekundites)</i>	<i>Katse 3 (sekundites)</i>	<i>Katse 4 (sekundites)</i>	<i>Katse 5 (sekundites)</i>	<i>Keskmine (sekundites)</i>
167	0.10372	0.11852	0.08871	0.09839	0.11199	0.10427
602	0.19983	0.16602	0.19074	0.18637	0.16207	0.18101
9808	1.53782	1.82494	1.66837	1.58062	1.18292	1.5589

Tabel 2: Andmete pärimise ja ettevalmistamise aeg kokku (kontroller)

Teiseks kasutati Google-i poolt arendatud rakendust Page Speed, mis hindab lehe laadimise kiirust. Samuti annab see nõuandeid, kuidas leht efektiivsemaks teha. Google Page Speed andis lehe kiiruse hinnanguks 38/100-st punktist. Niivõrd madal hinnang anti seetõttu, et andmed, mis saadetakse serverilt kliendile, ei ole kokku pakendatud. Ehk kõige mõistlikum on andmed enne kliendile saatmist kokku pakkida ning kliendi arvuti pakib need ise lahti. Täpsem olles, kliendi arvutis kasutatav brauser saab aru, et saadud andmed on pakitud kujul ning enne kasutamist on vaja need lahti pakkida. See lubab kliendi ja serveri vahel mahuliselt väiksemat andmeedastust mis omakorda tähendab kiiremat lehe laadimise aega. Aja nappuse tõttu ei ole eelmist andmete pakendamist realiseeritud.

Teine suurem probleem on *Leverage browser caching*, ehk failid, mis on olemuselt staatilised, *CSS* stiililehed[11], *Javascript* failid jms, tuleks hoida kliendi arvutis, mitte neid igakord lehe laadimisel uuesti saata. Seega otstarbekam on nad ainult üks kord laadida ning ladustada kliendi masinas, et neid uuesti serverist laadima ei peaks. See vähendab serveri ja kliendi vahelist suhtlust ning väljendub jälle kiiremas lehe laadimises. Ka *Leverage browser caching* jäi aja nappuse tõttu matemaatikaolümpiaadide portaalis realiseerimata. [5]

Antud bakalaureuse töö raames võeti kasutusele minimiseeritud *Javascript* failid, ehk neid on eemaldatud kõik tühikud, reavahetused jms, mis on mõeldud koodi loetavaks muutmiseks inimsilmale. [6]

Kokkuvõttes on portaali nõrgaks küljeks hetkel lehe laadimise kiirus. Andmeedastus serveri ja kliendi on võimalik kiiremaks muuta. Kahjuks aja puuduse tõttu arendajatel selleks aega ei jäänud.

Positiivne on aga asjaolu, et mitmed ülesanded mida tavaliselt täidab server, suudeti kliendi arvuti kanda jätta, mis vähendab serveri koormust. Samuti jätab see kasutajale lehega suhtlemisel sujuvama mulje (toimub vähem lehe laadimisi) ning mida aeg edasi läheb, seda kiiremaks ja võimsamaks muutuvad ka kasutajate arvutid. Viimane asjaolu tingib veelgi kiirema lehe kuvamise.

5. Kokkuvõte

Eesmärk oli valmistada kaasaegne matemaatikaolümpiaadide portaal, kus administraatorid saavad võrreldes portaali eelkäiaga mugavamalt tulemusi sisestada, muuta ja vajadusel ka kustutada. Eesmärkidest sai täidetud kõik andmete lisamise, muutmise ja kustutamisega seonduv, kusjuures antud protseduurid muutusid administraatoritele lihtsalt ning kiirelt teostatavaks.

Valmis ka tulemuste otsingusüsteem, mis lubab võistluse, aasta, klassi ja kooli järgi tulemusi otsida ning sorteerida. Kuigi esialgu ei olnud plaanis ülesannetega seoses käesoleva bakalaureuse töö raames midagi ette võtta, sooviti töö käigus ka võimalust, et punkte vaadates saab näha, mis ülesande eest need pandud on. Lõppkokkuvõttes suudeti ka see funktsionaalsus realiseerida.

Valminud portaali puudusteks on kliendi ja serveri suhtluse puudulik optimeerimine ning vähene testimine, mistõttu võib ette tulla üksikuid vigu, millest ei olda veel teadlikud.

Kokkuvõttes sai loodud matemaatikaolümpiaadide portaal, mis täidab soovitud funktsionaalsuse ning samas lubab tulevikus mugavalt portaali edasi arendada. Loodud on tugev põhi, kuhu saab tulevikus ehitada mahukama ja põhjalikuma olümpiaadide tulemuste keskuse.

6. Resume

The purpose of this paper is to build a modern mathematical olympiads portal, where portal administrators can easily insert, change or delete olympiads results. The current system was intended to be more easier to use and understand than it's predecessor. That was successfully accomplished.

A search engine for olympiads results was also made. It allows to search and sort olympiads results by competition, year, class and school. Although at first functionality regarding olympiads tasks was not intended to accomplish, it was requested to add to the portal, so that users can see, for what tasks points were given. That was also accomplished.

The portal has some minor issues regarding the data transfer between client and server: it's not very well optimized. Also the system need more testing to find more minor bugs, which should be fixed.

Overall, a mathematical olympiads portal was developed, which fulfills the requirements made by administrators. The portal is also ready to be developed further. The system is made so, that all future developers can easily learn the current system. The basic olympiads system has been made, now it's waiting to be developed further.

7. Viited

1. *Model–view–controller* - Wikipedia,
<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
(viimati vaadatud 19.06.2011)
2. *Miks Vali Zend Framework*, <http://www.ire-s.com/et/642/why-choose-zend-framework/> (viimati vaadatud 19.06.2011)
3. *jQuery tutvustus ja õpetused*, *jQuery ametlik kodulehekül*, <http://jquery.com/>,
(viimati vaadatud 19.06.2011)
4. *Zend Framework & MVC Introduction (inglise keeles)*,
<http://framework.zend.com/manual/en/learning.quickstart.intro.html> (viimati
vaadatud 19.06.2011)
5. *Leverage browser caching, sissejuhatus – Google*,
<http://code.google.com/speed/page-speed/docs/caching.html#LeverageBrowserCaching> (viimati vaadatud 19.06.2011)
6. *JavaScript failide minimaliseerimine, õpetus ja ülevaade - Google*,
<http://code.google.com/intl/et-EE/speed/page-speed/docs/payload.html#MinifyJS>
(viimati vaadatud 19.06.2011)
7. *Matemaatikaolümpiaadide portaal*, <http://www.math.olympiaadid.ut.ee/>,
(viimati vaadatud 19.06.2011)
8. *Zend Framework Documentation*, <http://framework.zend.com/manual/> (viimati
vaadatud 19.06.2011)
9. *Google page speed koduleht, tutvustus, näited*, <http://code.google.com/speed/page-speed/> (viimati vaadatud 19.06.2011)
10. *JavaScript - Wikipedia*, <http://et.wikipedia.org/wiki/Javascript> (viimati vaadatud
19.06.2011)
11. *Cascading Style Sheets - Wikipedia*,
http://en.wikipedia.org/wiki/Cascading_Style_Sheets(viimati vaadatud 19.06.2011)
12. *Margus Sellin*, „*Matemaatikaolümpiaadide tulemuste portaal HTML-tabelite automaatse töötlusega*“, bakalaureusetöö, Tartu Ülikool Arvutiteaduse Instituut, 2011

Lisa 1. Kasutatud tehnoloogiad

Tabelis on toodud portaali arendamisel kasutatud tehnoloogiad.

Tehnoloogia nimetus	Kirjeldus	Viide
A. <i>PHP 5.1</i>	Portaal on kirjutatud keeles PHP 5.1, töötab ka uuemate PHP versioonidega (näiteks 5.3)	http://www.php.net/
B. <i>MySQL 5</i>	Andmebaasi mootoriks on MySQL 5	http://www.mysql.com/
C. <i>jQuery 1.4.2</i>	<i>JavaScript</i> -i raamistikuna kasutatakse <i>jQuery</i> -t, sest hetkel üks parimaid raamistikke mis saadaval. Lihtne õppida, kasutada ning töötab kõikide uuemate brauseritega.	http://www.jquery.com
D. <i>jQuery DatePicker</i>	<i>jQuery</i> lisamoodul, mis võimaldab kasutajal visuaalsest kalendrist sobiv kuupäev valida.	http://jqueryui.com/demos/datepicker/
E. <i>jQuery TableSorter 2.0</i>	<i>jQuery</i> lisamoodul, mis võimaldab tabelleid sorteerida ning seda kõike kliendi masinas, ilma serveri ressursi kasutamata.	http://tablesorter.com/docs/
F. <i>jQuery Tablehover</i>	<i>jQuery</i> lisamoodul, mis võimaldab tulemustabelil mugavat navigeerimist.	http://p.sohei.org/jquery-plugins/tablehover/

Tabel 3: Kasutatud tehnoloogiad

Lisa 2. Portaali kood

Portaali kood on alla laetav aadressilt: http://ats.cs.ut.ee/olymp/olymp_source.zip