

TARTU ÜLIKOOL
MATEMAATIKA-INFORMAATIKATEADUSKOND
Arvutiteaduse instituut
Infotehnoloogia eriala

Jevgeni Koltšin
Väike portaal tenniseklubi jaoks
Bakalaureusetöö (6 EAP)

Juhendaja: dots. H. Hein

Autor: “.....” juuni 2011

Juhendaja: “.....” juuni 2011

Lubada kaitsmisele

Professor “.....“ juuni 2011

TARTU 2011

Sisukord

1. Sissejuhatus	4
2. Portaali üldine iseloomustus.....	5
2.1. Olemasolevatest süsteemidest	5
2.2. Projekti plaan.....	6
2.3. Süsteemi üldine kirjeldus	6
2.4. Punktisüsteemi kirjeldus	7
3. Arhitektuur	8
3.1. Andmemudel	8
3.2. MVC mudel.....	10
3.3. Versiooni haldamine	12
3.4. Projekti paigaldamine ja juurutamine.....	12
3.5. Suhtlemine andmebaasiga	13
3.6. Koodi organiseerimine	16
3.7. Autentimine, kasutajate rollid ja turvalisus.....	17
3.8. Keel	18
3.9. Vaate animatsioon	19
3.10. Valideerimine	20
4. Nõuded	22
4.1. Mittefunktsionaalsed nõuded.....	22
4.1.1. Kasutatavus	22
4.1.2. Käideldavus.....	22
4.1.3. Jõudlus.....	22
4.1.4. Toetus.....	22
4.2. Funktsionaalsed nõuded	22
4.2.1. Sisselogimine	22
4.2.2. Registreerimine	24
4.2.3. Väljalogimine.....	25
4.2.4. Profiili muutmine	25
4.2.5. Parooli muutmine	26
4.2.6. Kasutaja blokeerimine/aktiveerimine.....	26

4.2.7. Kuulutuse lisamine	27
4.2.8. Väljakutse sisestamine	28
4.2.9. Väljakutse kinnitamine.....	29
4.2.10. Väljakutsest loobumine	29
4.2.11. Väljakutse tulemuse sisestamine	30
4.2.12. Väljakutse tulemuse kinnitamine	30
4.2.13. Väljakutse tulemustest loobumine.....	31
4.2.14. Väljakutse muutmine või konfliktse situatsiooni lahendamine.....	31
4.2.15. Väljakutse kustutamine	32
4.2.16. Uudise lisamine/uuendamine	33
4.2.17. Uudise kustutamine	33
4.2.18. Uudise kommenteerimine.....	34
4.2.19. Galerii kataloogi lisamine/muutmine	34
4.2.20. Galeriist kataloogi kustutamine.....	35
4.2.21. Pildigalerii lisamine.....	36
4.2.22. Turniiri lisamine/muutmine.....	36
4.2.23. Turniiri kustutamine	37
4.2.24. Mänguväljaku lisamine/muutmine	38
4.2.25. Mänguväljaku kustutamine	39
4.2.26. Klubi reeglite/kontaktinfo muutmine	39
5. Testimine.....	40
5.1. Jõudlustestid.....	40
6. Kokkuvõte	42
Small portal for tennis club	43
Kasutatud kirjandus.....	45

1. Sissejuhatus

Paljud spordiklubid seisavad sageli ühe ja sama probleemi ees – on vaja infosüsteemi, mis võimaldaks klubi liikmetel vahetada informatsiooni, olla kursis toimuvate üritustega, tõsta klubi populaarsust jne. Käesoleva bakalaureusetöö eesmärgiks on luua veebirakendus Tartu naiste tenniseklubi jaoks, kus liikmed saaksid kergesti sisestada üksteisele väljakutseid, jälgida enda ja teiste liikmete tulemusi, kommenteerida uudiseid, vaadata pilte, jne.

Arenduskeskkonnaks valiti J2EE (Java 2 Enterprise Edition) platvorm, mis on väga võimalusterohke ja kiiresti arenev keskkond infosüsteemide arendamiseks.

Projektis on kasutatud kõige levinumaid raamistikke:

- Spring MVC 3.1 [1];
- Spring Security 3.1 [2];
- Hibernate 3 [3];
- Apache Tiles 2 [4];
- JSR-303 ubade valideerimine [5];
- jQuery [6].

Andmebaasiks on valitud MySQL [7], servlettide konteineriks Tomcat 6 [8]. Lisaks on veel kasutusel Ant [9] ja Apache Ivy [10].

Teises peatükis tehakse ülevaade olemasolevatest analoogilistest süsteemidest Eestis ja analüüsitakse ühte konkreetset süsteemi, tuues välja selle eelised ja puudused. Edasi esitatakse kliendi nõuded loodavale portaalile ja koostakse projekti plaan. Kolmandas peatükis analüüsitakse portaali andmemudelit ning antakse ülevaade projekti arhitektuurist ja kasutatud tehnoloogiatest. Neljandas peatükis esitatakse nõuete analüüs, kus eraldi on vaatluse all mittefunktsionaalsed ja funktsionaalsed nõuded. Funktsionaalsete nõuete osas vaadatakse läbi kõik põhilised juhtumid portaali kasutamisel. Viiendas peatükis kirjeldatakse valminud portaaliga läbiviidud teste ja nende tulemusi.

2. Portaali üldine iseloomustus

2.1. Olemasolevatest süsteemidest

Eestis on loodud mitmeid portaale tenniseklubide jaoks, näiteks Forseliuse Tenniseklubi [11], Tartu Tennisekool [12], Tenniseklubi Sinnet [13] ja Promenaad [14] portaalid. Antud projekt on realiseeritud pidades silmas lähima konkurendi - Reval Ladies tenniseklubi portaali (<http://www.revalladies.ee>).

Reval Ladies portaal on programmeeritud keeles PHP, kusjuures kasutatakse tasuta sisuhaldussüsteemi Edge CMS (*Content Management System*), milles iga mooduli lisamine (nt. kuulutused, bännerid, jne) on tasuta. Koodi muutmine ei ole alati lihtne, sest tavaliselt sisuhaldussüsteem ei realiseeri tellija jaoks vajalikku spetsiifilist funktsionaalsust, milleks on antud projekti puhul punktisüsteemi arvutamine. Portaalil puudub kasutajate registreerimine. Kasutajaid võib registreerida ainult administraator ja kasutajatel pole võimalust muuta oma profiili.

Vale kasutajatunnuse või parooliga sisselogimisel ei näita portaal ühtegi veateadet.

Portaal on kahekeelne (eesti ja inglise), kuid ingliskeelne osa toimib täiesti iseseisvalt. Esiteks, kui valida inglise keel, siis kasutajal ei ole võimalust sisestada väljakutset. Teiseks, inglise keelt valiv küllastaja ei näe edetabeleid, kuigi antud juhul on erinevus ainult tabelite pealkirjades.

Kui vaadelda punktisüsteemi realiseerimist, siis väljakutsete sisestamisel võib mängu toimumiskoha väli olla suvaline tekst, mis ei ole kuidagi seotud väljakute lehel olevate toimumiskohtadega. Kogu protsess ei ole automatiseeritud, see tähendab, et punktide arvutamine ja võitja otsustamine toimub ainult läbi administraatori.

Positiivsest küljest võib öelda, et portaal töötab päris kiiresti, kui saiti kasutada üksinda. Kui proovida saata korraga 50 JMeter [15] abil genereeritud küllastajat esilehele, siis keskmine lehe avamise aeg on 3.5 sekundit, mis ei ole väga hea tulemus.

Teiseks positiivseks küljeks on otsingu võimalus, mis lihtsustab vajaliku info leidmist.

Käesolevas töös on lahendatud kõik selle portaali negatiivsed küljed.

2.2. Projekti plaan

Portaali loomise protsess on jagatud neljaks osaks ehk iteratsiooniks:

1. Nõuete analüüs, projekti arhitektuuri ja andmemudeli koostamine, ehituskriptide loomine, dokumentatsiooni koostamine.
2. Portaali üldise loogika ja moodulite (registreerimine, sisselogimine, administreerimine, uudised, galerii) programmeerimine ning kujunduse ja testimise dokumentatsiooni koostamine.
3. Funktsionaalsuse täiendamine (väljakutsed, minu konto), silumine, testimine, dokumentatsiooni koostamine.
4. Stabiliseerimine, avalik testimine tellija poolt. Avastatud vigade parandamine, dokumentatsiooni täiendamine.

2.3. Süsteemi üldine kirjeldus

Kliendi soovil peab portaal koosnema mitmest erinevast moodulist.

- Kõige olulisem on väljakutsete sisestamise moodul. Igal klubi liikmel peab olema võimalus kutsuda teisi mängima kas paaris- või üksikmängu. Iga mängu tulemust peab olema võimalik salvestada andmebaasi. Iga mäng annab osalejale teatud arvu punkte, mille põhjal genereeritakse võidufond ja koostatakse edetabel üksik- ja paaris mängu jaoks eraldi. Punktide arvutamine peab toimuma automaatselt. Väljakute loetelu (mängude toimumiskohti) haldab administraator. Administraator võib väljakut lisada, muuta või kustutada.
- Teine moodul on administreerimise moodul. Administraatoril peab olema oma alamleht, kus ta saab kasutajad aktiveerida või blokeerida, lisada või muuta uudiseid, kuulutusi, toetajaid; muuta kasutajate profiili, lahendada väljakutsete kinnitamisel tekkinud probleeme, tühistada kasutajate punkte. See leht ja kõik temaga seotud tegevused peavad olema kättesaadavad ainult administraatorile.
- Järgmine on uudiste ja kuulutuste moodul. Esileht peab kuvama administraatori poolt hallatavaid uudiseid ja kuulutusi. Administraatoril peab olema võimalus vormistada uudiseid omal soovil. Kuulutusi saavad sisestada ka sisselogimata külastajad. Uudistel peab olema kommenteerimise võimalus. Kommenteerida saavad ainult sisselogitud

kasutajad. Lisaks peab esileht kuvama klubi toetajate bannereid, mis on samuti hallatavad administraatori poolt.

- Portaali igal külastajal peab olema võimalus registreerida ennast kasutajaks. Kuid kontode aktiveerimisega tegeleb administraator. See tähendab, et peale registreerumist ei saa kasutaja sisse logida, sest enne seda peab administraator kas aktiveerima uue liikme konto või vastupidisel juhul blokeerima selle.
- Igal sisselogitud kasutajal peab olema võimalus vaadata liikmete tabelit koos nende profiilidega, muuta oma profiili, üles laadida oma avatari. Selleks peab olema eraldi „minu konto“ leht, kus kasutajal on võimalus jälgida enda poolt lõpetatud või pooleli olevaid väljakutseid. Lisaks peab igal kasutajal olema võimalus muuta enda staatust - kas kasutaja on aktiivne või mitte ehk kas ta soovib väljakutsetes osaleda. Kui kasutaja pole aktiivne, siis teised liikmed ei saa teda välja kutsuda.
- Galerii. Administraatoril peab olema võimalus luua galerii kataloogi ning lisada või kustutada pilte. Kataloogi kustutamisel kõikide temaga seotud piltide kustutamine peab toimuma automaatselt.
- Turniirid. Administraatoril peab olema võimalus luua ja muuta tulevikus toimuvate turniiride nimekirja (tüüp, koht, aeg).
- Administraatoril peab olema võimalus oma soovil vormistada reeglite ja „Meie kontaktid“ lehekülgi ja vajadused neid muuta.

2.4. Punktisüsteemi kirjeldus

Klubi edetabelis saab klubi liige koha vastavalt kogutud punktide summale. Klubil on kaks edetabelit: üksikmängude ja paarismängude jaoks. Väljakutsemängud toimuvad aastaringelt. Välja kutsuda võib mängijaid terve edetabeli ulatuses, välja arvatud mängijaid, kelle nime järel on tärn (*). Tärn tähistab liikmeid, kes ei soovi väljakutsetes osaleda.

Mängu võitja saab kaotaja koha punktid. Paarismängu võitjad saavad kumbki poole kaotajate kohapunktide summast. Kaotaja saab alati 5 punkti.

3. Arhitektuur

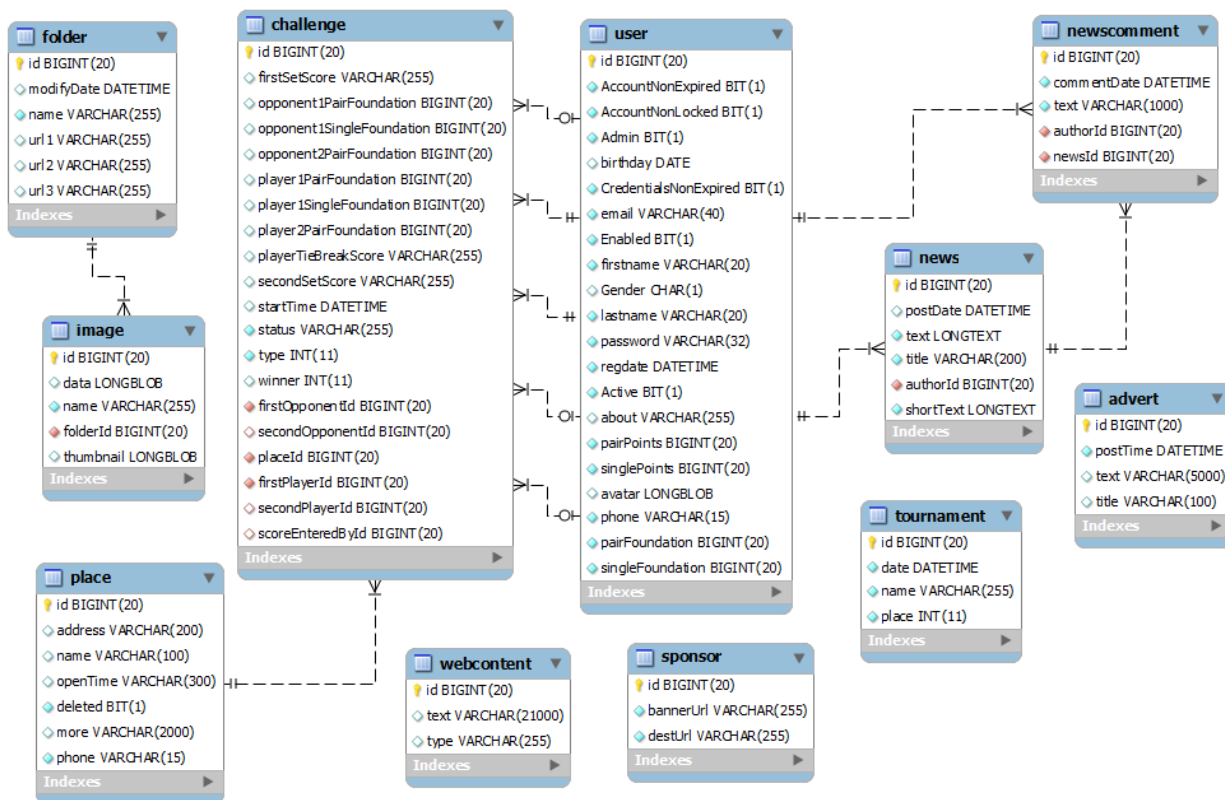
3.1. Andmemudel

Käesoleva projekti andmemudel on esitatud Joonisel 1. Andmebaasis on kokku 11 tabelit:

- **User** – sisaldab infot registreeritud kasutajate kohta. Igal kasutajal on oma id, ees- ja perenimi, email (mis on kasutaja tunnuseks), parool, avatar, info endast, tema üksik- ja paarismängude punktid ja võidufond. Lisaks on olemas veerud, mille abil saab kontrollida, kas kasutaja on administraator, kas kasutaja on aktiivne, st. kas kasutaja on aktiveeritud.
- **Challenge** – funktsionaalsuse seisukohast kõige olulisem tabel, mis sisaldab infot iga väljakutse kohta. Igal väljakutsel on oma id, kaks või neli mängijat (vastavalt üksikmängu või paarismängu korral), mängu toimumiskoht, kuupäev, mängijate võidufondid antud hetkel, väljakutse tüüp, tulemuse sisestaja, mängu skoor, mängu võitja ja väljakutse staatus.
- **Place** – väljakutsete toimumiskohtade tabel sisaldab asukoha id, nimetust, aadressi, telefoninumbrit, infot lahtioleku aegade kohta, lisainfot ja staatust. *Deleted* (kustutatud) staatus on lisatud selleks, et juhul kui administraator tahab asukoha ära kustutada, siis sellega seotud väljakutsete info jääb tervikuna alles.
- **News** – uudiste tabel koosneb järgmistest väljadest: uudise id, tekst, pealkiri, viimane muutmise kuupäev ja lisaja id. Antud tabel on seotud tabelitega *User* (uudise autor) ja *Newscomment* (kommentaari, mis kuuluvad uudise juurde).
- **Newscomment** – tabel uudise kommentaaride jaoks (kommentaari id, kirjutaja id, uudise id, tekst, kirjutamise kuupäev).
- **Tournament** – turniiride tabel, mida kasutatakse turniiride salvestamiseks (turniiri id, toimumiskoht (1,2,3), kuupäev, nimi).
- **Sponsor** – Toetajate tabel sisaldab id ja kahte linki, millest esimene viitab pildile koos bänneriga ja teine lehele, kuhu antud bänner suunab. Seda kasutatakse esilehe bännerite kuvamiseks.
- **Webcontent** – tabel reeglite ja „meie kontaktide“ lehe info hoidmiseks (id, sisu tüüp, html sisu). Tüüp on defineeritud Javas eraldi tüübiga *enum*.

- **Advert** – kuulutuste tabel. Iga kuulutuse kohta salvestatakse järgmine info: kuulutuse id, sisestamise kuupäev, tekst ja pealkiri.
- **Folder** – kausta tabel, mille infot kasutatakse pildigaleriis. Koosneb väljadest: id, muutmise kuupäev, nimi ja kolmest mittekohustuslikust lingist, mis viitavad teistele pildiallikatele.
- **Image** – tabel piltide hoidmiseks. Iga pildi kohta säilitatakse järgmine info: pildi id, nimi, suurus, pilt ise ja vähendatud suurusega pilt, mis on salvestatud baitide massiivina. Antud tabel on seotud tabeliga *Folder*. Ühes kasutas võib olla mitu pilti ja iga pilt on seotud ainult ühe kaustaga (*many-to-one* seos).

Kuna antud portaal ei ole väga suur, siis piltide hoidmine andmebaasis ei muutu pudelikaelaks (kõige nõrgemaks kohaks) ja pole mõtet salvestada pilte otse failisüsteemi. Piltide salvestamine andmebaasi lihtsustab andmebaasi migreerimist ühest masinast teise, sest kõik andmed on koos ühes *dump* failis.



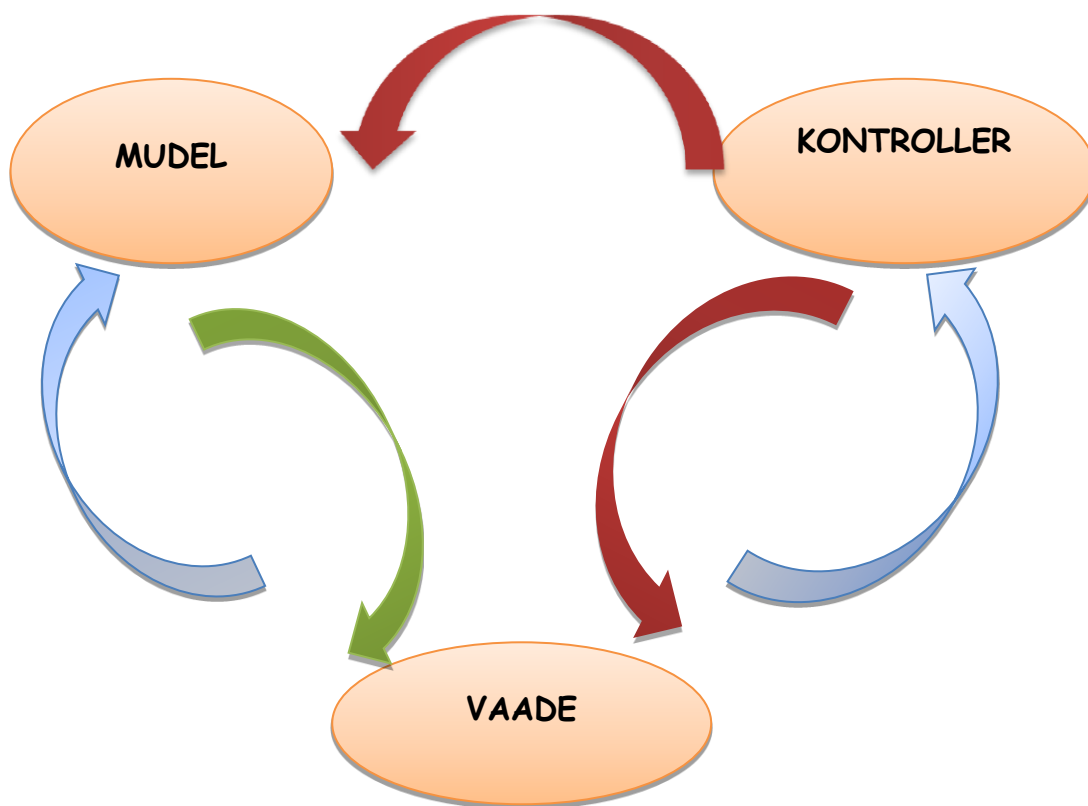
Joonis 1. Rakenduse andmemudel.

3.2. MVC mudel

Rakenduse tuumaks on Spring 3 raamistik. Disaini šablooniks (*pattern*) on võetud MVC (*Model – View - Controller*) mudel, mida realiseerib *Spring MVC* (Joonis 2).

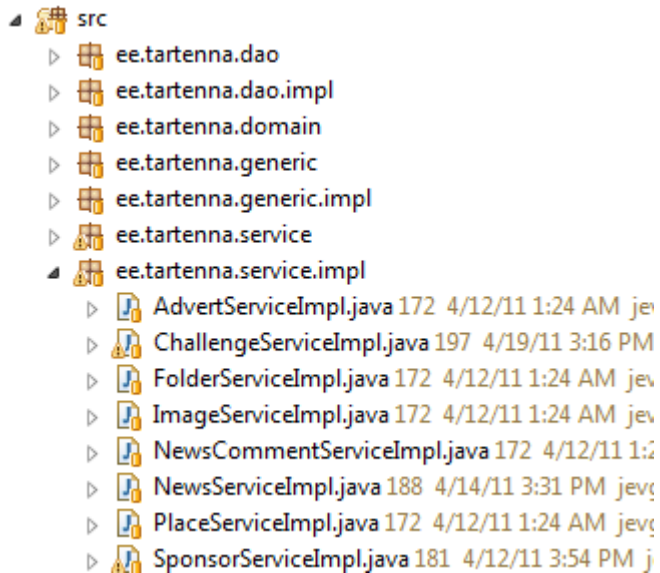
Mudel-Vaade-Kontrolleri eelised [16]:

- eraldab disaini probleemid – erinevad kihid tegelevad andmete püsisalvestamise, presentatsiooni ja rakenduse üldise loogikaga;
- koodi dubleerimise vähendamine – tuleneb tsentraliseeritusest;
- kogu rakenduse juhtimine toimub kontrolleri kihis;
- kuna rakendusi ei tehta tavaliselt kohe algusest lõpuni valmis, on oluline, et nende muutmine oleks lihtsam;
- kuna suuremate rakenduste juures töötavad arendajad on tavaliselt spetsialiseerunud mingile kindlale rakenduse aspektile (näiteks tegevusloogika kirjutaja, kasutajaliidese disainer jne), siis on oluline, et nad saaksid pühenduda just arendatavale küsimusele ja ei peaks tundma kogu rakenduse kõiki aspekte.



Joonis 2. MVC Spring raamistiku korral.

- **Mudel** – sisaldab endas tegevusloogika andmeid. Koosneb domeenist, DAO (*Data Access Object*) ja teenuskihist (vt. Joonis 3).



Joonis 3. Mudeli kihi pakettide loetelu.

- **Vaade** – kannab endas esitlusloogikat ehk kuidas rakendus ekraanil välja näeb. Vaate genereerimiseks on kasutusel Apache Tiles + *jsp*. Apache Tiles on raamistik, mis võimaldab teha lehe šablooni (*layout*) ja tõsta lehe korduvad osad (päis, menüü, jalus) konfiguratsioonifaili. Šabloon ise on *.jsp* fail, milles on defineeritud, kus ja kuidas tema osad paiknevad. Uue lehe loomiseks on vaja ainult lisada uue vaate nimi faili *tiles-layout.xml* ja määrata, kust ta peab võtma vaate.

Näiteks, olgu meil šabloonifailis (*baseTemplate.jsp*) koodifragment

```
<div id="content">
    <tiles:insertAttribute name="content"/>
</div>
<div id="footer">
    <tiles:insertAttribute name="footer"/>
</div>
```

ja konfiguratsioonifailis (*tiles-layout.xml*) järgmine sektsioon

```
<definition name="usersView" extends="baseTemplate">
    <put-attribute name="content" value="/WEB-INF/jsp/general/users.jsp" />
</definition>.
```

Antud juhul vaate *usersView* väljakutsel tagastatakse vaade *baseTemplate*, milles asendatakse *content* osa faili *users.jsp* sisuga.

- **Kontroller** – reageerib päringutele, tagastab õige vaate, st. kontroller on vaate ja mudeli sidumiskohaks.

Kontrollerite defineerimiseks võib raamistikus Spring kasutada kas annotatsioone või defineerida neid *.xml* failis. Antud projektis kasutatakse annotatsioone.

Näiteks, kontrolleris

```
@Controller
public class GeneralController {

    private static final String PORTAL_MEMBERS_VIEW = "usersView";

    @RequestMapping(value = "/members", method = RequestMethod.GET)
    public String getMembers(HttpServletRequest request, Model
model) throws IOException {
        WebUtil.initView(request.getSession(), model);
        model.addAttribute(MODEL_ATTR_USERS,
userService.getActiveUsers());
        return PORTAL_MEMBERS_VIEW;
    }
}
```

meetod *getMembers* tagastab ülaltoodud näites vaate juhul kui kasutaja proovib minna */members* lehele.

3.3. Versiooni haldamine

Kogu lähtekoodi hoitakse Google SVN repositooriumis. Iga iteratsioon on oma harus. SVN kasutamine võimaldab vajadusel kiiresti uuendada portaali. Kui avastati probleem, mis vajab kiiret lahendamist koodis, siis arendaja parandab ja testib seda enda masina koodi abil ja saadab uuendatud versiooni SVN repositooriumi [17]. Live masina administraatorile on vaja saata ainult SVN viimane seis (nt. käsuga *svn up* ja uuendada live: *ant deploy*).

3.4. Projekti paigaldamine ja juurutamine

Projekt töötab Apache Tomcat 6 veebikonteineris Apache 2. Projekti paigaldamiseks käivitatakse Apache Ant käsuga: *ant deploy*; selleks on kirjutatud ehitusskript failis

build.xml. Selleks, et oleks võimalikult lihtne muuta, hoitakse konfiguratsiooni kõiki omadusi eraldi failis *build.properties*.

Vajalike teekide (*.jar*) ja raamistike allalaadimisega tegeleb Ant Ivy, mis on konfigureeritud eraldi failis *ivy.xml*. See võimaldab hoida repositooriumis ainult koodi ilma *.jar* failideta. Sellise konfiguratsiooni korral versiooni uuendamine ei tekita probleeme.

3.5. Suhtlemine andmebaasiga

Rakendus kasutab andmebaasi MySQL. Andmebaasiga suhtlemiseks on kasutusel raamistik Hibernate 3.6, mis võimaldab töötada otse andmetabelite asemel objektidega ja arendaja ei pea kirjutama SQL päringuid, kuigi ka see on võimalik. SQL päringute kirjutamise asemel koostatakse kriteeriumid, mis Hibernate teisendab päringuteks ja vastupidi: Hibernate teisendab päringu vastuse POJO (*Plain Old Java Object*) objektiks [18].

Hibernate kasutab „laiska väärtustamist“ (*lazy loading*), mis tähendab, et Hibernate teeb päringu alles sellel hetkel, mil andmeid reaalselt kasutatakse. Hibernate töötab enamiku andmebaasidega. Iga andmebaasi tüübi jaoks kasutatakse oma dialekti (Tabel 1).

Tabel 1. Hibernate poolt toetatavad andmebaasid ja dialektid.

Andmebaas	Dialekt
DB2	org.hibernate.dialect.DB2Dialect
DB2 AS/400	org.hibernate.dialect.DB2400Dialect
DB2 OS390	org.hibernate.dialect.DB2390Dialect
PostgreSQL	org.hibernate.dialect.PostgreSQLDialect
MySQL	org.hibernate.dialect.MySQLDialect
MySQL with InnoDB	org.hibernate.dialect.MySQLInnoDBDialect
MySQL with MyISAM	org.hibernate.dialect.MySQLMyISAMDialect
Oracle (any version)	org.hibernate.dialect.OracleDialect
Oracle 9i	org.hibernate.dialect.Oracle9iDialect
Oracle 10g	org.hibernate.dialect.Oracle10gDialect
Sybase	org.hibernate.dialect.SybaseDialect
Sybase Anywhere	org.hibernate.dialect.SybaseAnywhereDialect
Microsoft SQL Server	org.hibernate.dialect.SQLServerDialect
SAP DB	org.hibernate.dialect.SAPDBDialect
Informix	org.hibernate.dialect.InformixDialect
HypersonicSQL	org.hibernate.dialect.HSQLDialect
Ingres	org.hibernate.dialect.IngresDialect
Progress	org.hibernate.dialect.ProgressDialect
Mckoi SQL	org.hibernate.dialect.MckoiDialect
Interbase	org.hibernate.dialect.InterbaseDialect
Pointbase	org.hibernate.dialect.PointbaseDialect
FrontBase	org.hibernate.dialect.FrontbaseDialect
Firebird	org.hibernate.dialect.FirebirdDialect

Andmebaasi konfiguratsiooni hoitakse eraldi failis *hibernate.properties*, mis asub kaustas */src*.

Näiteks, meetod

```
@Override
public List <User> getEnabledUsersOrderedByDate() {
    DetachedCriteria criteria = DetachedCriteria.forClass (User.class).
        addOrder(Order.desc("regDate")).add(Restrictions.eq("enabled", true));
    return hibernateTemplate.findByCriteria(criteria);
}
```

tagastab listi, kus on ainult aktiveeritud kasutajad registreerumise kuupäeva kahenevas järjekorras.

Igale tabelile vastab üks klass. Selleks, et Hibernate saaks aru, et tegemist on tabeliga, kasutatakse annotatsiooni @Entity. Tabeli nimeks võetakse vaikimisi klassi nimi kui ei ole defineeritud teisiti.

id - veeru nimeks määratakse vaikimisi id.

- @Id - määrab tabeli primaarse võtme;
- @GeneratedValue – tähendab, et väärtus genereeritakse kasutades automaatset algoritmi. Automaatse algoritmi asemel võiks kasutada ise tehtud järjestusi.

commentDate - veeru nimeks määratakse commentDate.

- @NotNull – tähendab, et veerg on kohustuslik;
- @Temporal (TemporalType.TIMESTAMP) – määrab veeru kuupäeva tüübi.

Näide annotatsioonist:

```

@Entity
public class NewsComment implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @NotNull
    @Temporal(TemporalType.TIMESTAMP)
    private Date commentDate;

```

Rakenduse käivitamisel genereerib Hibernate annotatsioonide põhjal vajalikud tabelid või teeb muudatusi olemasolevatesse tabelitesse.

3.6. Koodi organiseerimine

Parema ülevaate saamiseks on lähtekood organiseeritud pakettidesse. Domeeni klassid asuvad nimeruumis *ee.tartenna.domain*, DAO liidesed ja nende implementatsioonid nimeruumis *ee.tartenna.dao*, teenused ja nende implementatsioonid nimeruumis *ee.tartenna.service* ja kontrollid nimeruumis *ee.tartenna.web*.

DAO ja teenusklasside kirjutamiseks on kasutatud geneerilisi tüüpe (*generics*). See võimaldab luua universaalseid meetodeid ja vältida korduvat koodi, näiteks objektide CRUD (*create, read, update, delete*) operatsioonid.

Näide geneerilist tüüpi rakendavast klassist:

```

public class GenericDaoImpl<T, PK> extends Serializable implements
GenericDao<T, PK> {
    private final Class<T> type = (Class<T>) ((ParameterizedType)
getClass().getGenericSuperclass())
        .getActualTypeArguments()[0];

    @Autowired
    protected HibernateTemplate hibernateTemplate;

```



```

protected Class<T> getType() {
    return type;
}

@Override
public void save(T o) {
    hibernateTemplate.save(o);
}

```

Konkreetses klassis, mis on päritud ülaltoodud klassist, saab automaatselt salvestamise meetodi `save`.

Näiteks, klassist

```

@Repository
public class NewsDaoImpl extends GenericDaoImpl<News, Long> implements
NewsDao {}

```

saab välja kutsuda meetodi `save` uudise salvestamiseks

```

News testNews = new News();
newsDao.save(testNews);

```

3.7. Autentimine, kasutajate rollid ja turvalisus

Projektis on kasutusel raamistik Spring Security 3.1 [19], mis võimaldab lihtsalt jagada rolle, konfigureerida sisse- ja väljalogimist, ligipääsu mingile konkreetsele meetodile või JSP faili osale. Spring Security konfiguratsioon on toodud failides `securityConfig.xml` ja `web.xml`. Näiteks, järgmises `.xml` faili fragmendis

```

<sec:http auto-config="true" use-expressions="true" access-decision-manager-
ref="customAccessDecisionManager"
access-denied-page="/home" disable-url-rewriting="true">
<sec:intercept-url pattern="/admin/**" access="hasRole('ROLE_ADMIN') " />
<sec:intercept-url pattern="/register" access="isAnonymous() " />

```

on konfigureeritud järgmised tegevused:

1. Kõikidele `/admin/` algusega URLidele saadetakse päringud, kui kasutajal on olemas administraatori õigused.
2. Registreerimise lehele lubatakse ainult sisselogimata kasutajat.
Näiteks, kui failis `menu.jsp` on fragment

```

<security:authorize access="isAuthenticated() ">
  <li class="divider"></li>
  <li>
    <a class="nav" href="/logout">
      <spring:message code="portal.menu.logout"/>
    </a>
  </li>
</security:authorize>

```

siis väljalogimise nuppu näidatakse ainult sisselogitud kasutajale.

3.8. Keel

Portaal on esialgu mõeldud kasutamiseks ainult eesti keeles. Keelte konfiguratsioon on kirjas failis *applicationContext.xml*:

```

<mvc:interceptors>
  <bean class="org.springframework.web.servlet.i18n.
LocaleChangeInterceptor" p:paramName="lang" />
</mvc:interceptors>
<bean id="messageSource" class="org.springframework.context.support.
ReloadableResourceBundleMessageSource" p:basenames= "classpath:messages"
p:defaultEncoding ="UTF-8" p:useCodeAsDefaultMessage="true" />
<bean id="localeResolver" class="org.springframework.web.servlet.
i18n.SessionLocaleResolver" p:defaultLocale="et" />

```

Koodist on näha, et keelte vahetus toimub, kui päringuga saadetakse parameeter lang koos keele koodiga. Antud juhul vaikimisi väärtuseks on *et*, mis pannakse sessiooni.

Kõik tõlked on kirjas eraldi failis *messages_et.properties*, mis koosneb paaridest võti=väärtus.

Tõlgete kasutamiseks on raamistikus Spring olemas spetsiaalne märgis `<spring:message>`, mis võtab tõlke võtme ja tagastab väärtuse.

Näiteks,

```

<spring:message code="btn.save"/>

```

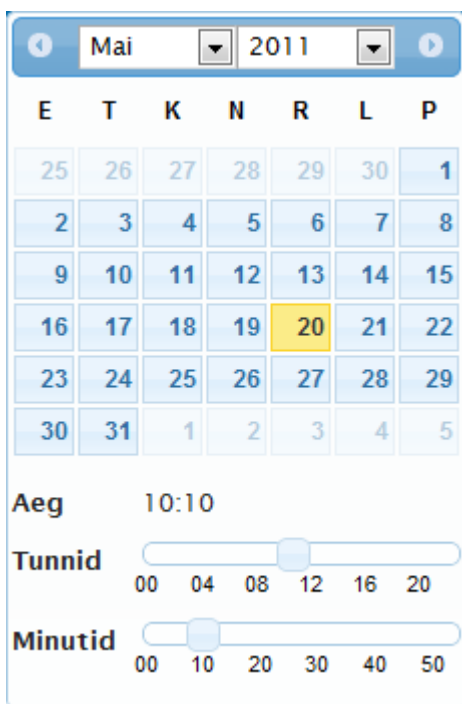
tagastab teksti „Salvesta” sellepärast, et failis *message_et.properties* on olemas paar `btn.save=Salvesta`.

Kui tekib vajadus lisada mingi teise keele tugi, siis piisab uue faili *messages properties* lisamisest koos teise keele koodiga [20].

3.9. Vaate animatsioon

Vaate animeerimiseks kasutati kõige populaarsemat JavaScript raamistikku jQuery.

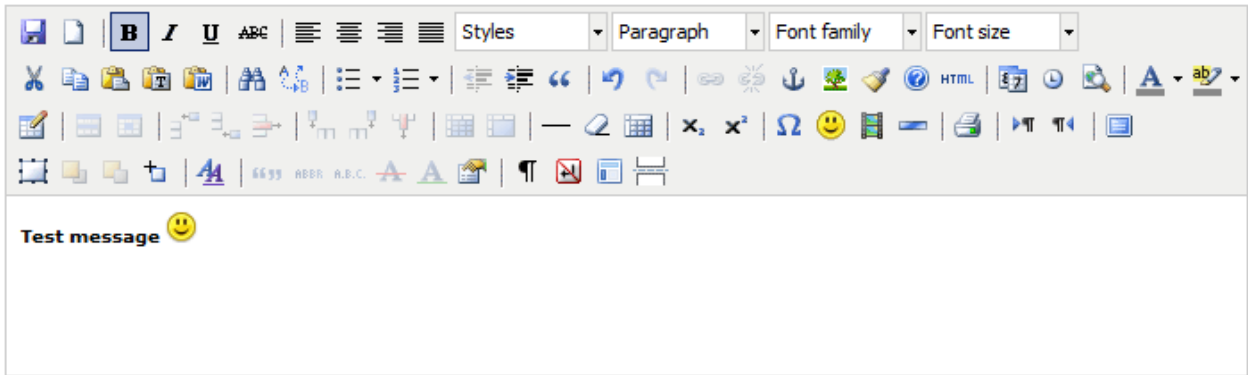
Näiteks, väljakutsete sisestamisel kuupäeva ja aja valimiseks kasutati jQuery aja valimise pluginat *datetimepicker* (Joonis 4).



Joonis 4. Aja valimise komponent.

Keelt JavaScript kasutatakse esilehe banneril piltide liikumiseks ja galeriis.

Uudiste, kontaktinfo ja reeglite lisamiseks/muutmiseks rakendatakse tekstitoimetit TinyMCE (Joonis 5), mis on programmeeritud JavaScriptis. Selle visuaalse redaktori abil on portaali administraatoril väga lihtne vormistada teksti, luua tabeleid jne [21].



Joonis 5. Tekstoimeti TinyMCE.

3.10. Valideerimine

Peale iga vormi kinnitamist toimub andmete serveripoolne valideerimine. Valideerimiseks kasutatakse standardit JSR-303, mis võimaldab kirjeldada valideerimise reegleid kasutades annotatsioone. Lisaks on kasutusel JSR-303 realisatsioon ehk Hibernate validaator ja oma kirjutatud valideerimise annotatsioonid. Järgnevalt on esitatud näide valideerimise annotatsioonist, kus

@FileMaxSize – valideerib faili suurust. Antud juhul faili suurus ei tohi ületada 100000 baiti.

@AllowedContentType – valideerib faili tüüpi. Antud juhul lubatud ainult fail, mille MIME tüüp kas *.bmp*, *gif*, *.jpg*, *jpeg*, *png* või *.tif*.

```
public class AvatarModel {  
    @FileMaxSize(value = 100000)  
    @AllowedContentType(types = { "bmp", "gif", "jpg", "jpeg", "png",  
    "tif" })  
    private MultipartFile file;  
}
```

Valideerimise annotatsioonid asuvad paketis *ee.tartenna.web.validator*. Iga valideerimise reegel koosneb kahest osast:

- Annotatsioon ise, kus on kirjeldatud tema atribuudid, skoop jne;

```
@Target({ FIELD, METHOD, ANNOTATION_TYPE })  
@Retention(RUNTIME)
```

```

@Constraint(validatedBy = FileMaxSizeValidator.class)
@Documented
public @interface FileMaxSize {
    String message() default "wrong.size";
    long value();
    Class<?>[] groups() default {};
    Class<? extends Payload>[] payload() default {};
}

```

- Valideerimise klass, mis sisaldab valideerimise loogikat.

```

public class FileMaxSizeValidator implements
ConstraintValidator<FileMaxSize, MultipartFile> {

    private long value;
    private String message;

    @Override
    public void initialize(FileMaxSize annotation) {
        value = annotation.value();
        message = annotation.message();
    }

    @Override
    public boolean isValid(MultipartFile file,
ConstraintValidatorContext context) {
        return file.getSize() > 0 && value >= file.getSize();
    }
}

```

4. Nõuded

4.1. Mittefunktsionaalsed nõuded

4.1.1. Kasutatavus

- Keskkonna kasutamine peab tavakasutajale olema võimalikult lihtne ja arusaadav.
- Rakenduse veateated peavad olema eestikeelsed.

4.1.2. Käideldavus

- Projekti üleandmisel ei tohi rakendus sisaldada ühtegi teadaolevat funktsionaalset viga.
- Kui välja arvata riistvara rikked, peab rakendus olema kasutatav 99% ajast.
- Projekti üleandmise järel pole arendajal kohustust rakendust hooldada või kasutajatuge pakkuda.

4.1.3. Jõudlus

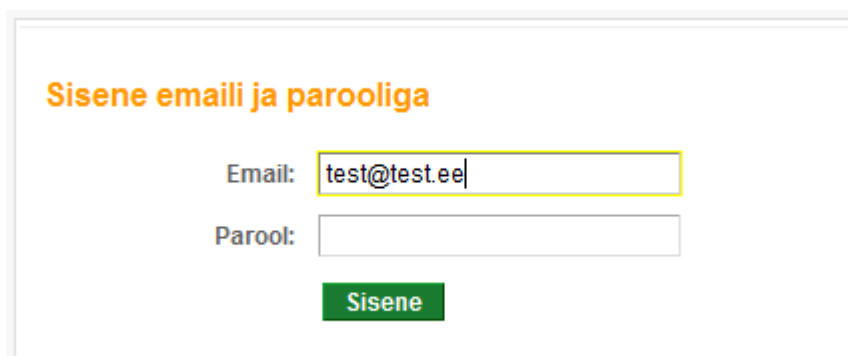
- Rakendus peab suutma teenindada 50 samaaegset kasutajat.
- Kasutaja vastuse koostamine serveris ei tohi võtta kauem kui 2 sekundit.

4.1.4. Toetus

- Veebikeskkond luuakse ühekeelsena (eesti keel).
- Rakendus peab logima töös esinevaid vigu.
- Rakendus peab võimaldama relatsioonilise andmebaasi kasutamist.

4.2. Funktsionaalsed nõuded

4.2.1. Sisselogimine



Sisene emaili ja parooliga

Email:

Parool:

Joonis 6. Sisselogimise vorm.

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on sisselogimise lehel. Kasutaja on sisse logimata.

Põhivoog:

1. Kasutaja kirjutab sisselogimise vormi oma kasutajanime (e-maili aadress) ja parooli (Joonis 6).
2. Kasutaja klikib sisselogimise nupule.
3. Kasutaja suunatakse viimati külastatud lehele.

Tulemus: Kasutaja logitakse sisse.

Alternatiivsed vood:

1. Sisselogimine ebaõnnestus, kasutajanimi või parool on vale.
 - 1.1 Kasutajale kuvatakse veateade.
2. Sisselogimine ebaõnnestus, kasutajanimi ja parool on õiged, aga kasutaja on aktiveerimata.
 - 2.1 Kasutajale kuvatakse veateade.

4.2.2. Registreerimine

Email:*

Parool:* Lubatud pikkus 6 - 10 märki

Parool uuesti:*

Eesnimi:*

Perenimi:* Kohustuslik väli

Telefon:* Kohustuslik väli

Endast:

Joonis 7. Registreerumise vorm.

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on registreerimise lehel, kasutaja on sisse logimata.

Põhivoog:

1. Kasutaja kirjutab registreerimise vormi oma e-maili, nime, telefoni ja kaks korda parooli (Joonis 7).
2. Kasutaja klikib registreerimise nupule.
3. Kasutaja suunatakse lehele registreerimise kinnitamise tekstiga.

Tulemus: Kasutaja on registreeritud, kuid aktiveerimata.

Alternatiivsed vood:

1. Registreerumine ebaõnnestus, mõni kohustuslik lahter on täitmata.
 - 1.1 Kasutajale kuvatakse veateade.
2. Registreerimine ebaõnnestus, sisestatud andmed ei vasta valideerimise reeglitele.

- 2.1 Kasutajale kuvatakse veateade.
- 3. Registreerimine ebaõnnestus, sisestatud kasutajanimi on kasutusel.
- 3.1 Kasutajale kuvatakse veateade.

4.2.3. Väljalogimine



Joonis 8. Väljalogimise nupp.

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on sisse logitud.

Põhivoog:

1. Kasutaja klikib väljalogimise nupule (Joonis 8).
2. Kasutaja suunatakse esilehele.

Tulemus: Kasutaja on välja logitud.

4.2.4. Profiili muutmine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on profiili muutmise lehel. Kasutaja on sisse logitud.

Põhivoog:

1. Kasutaja muudab profiili vormis oma infot.
2. Kasutaja klikib 'salvesta' nupule.
3. Kasutajale näidatakse kinnitusteadet, et tema profiil on muudetud.

Tulemus: Kasutaja profiil on muudetud.

Alternatiivsed vood:

1. Profiili muutmine ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.

1.1 Kasutajale kuvatakse veateade.

4.2.5. Parooli muutmine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on parooli muutmise lehel. Kasutaja on sisse logitud.

Põhivoog:

1. Kasutaja sisestab vana parooli ja kaks korda uut parooli.
2. Kasutaja klikib 'salvesta' nupule.
3. Kasutajale näidatakse kinnitusteadet, et tema parool on muudetud.

Tulemus: Kasutaja parool on muudetud.

Alternatiivsed vood:

1. Parooli muutmine ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.
 - 1.1 Kasutajale kuvatakse veateade.
2. Parooli muutmine ebaõnnestus, sisestatud vana parool on vale.
 - 2.1 Kasutajale kuvatakse veateade.

4.2.6. Kasutaja blokeerimine/aktiveerimine

testtest testtest	testtest@testtest.tt	19.03.2011		
test1 test1	test1@test1.ee	22.03.2011		
test2 test2	test2@test2.ee	22.03.2011		
test3 test3	test3@test3.ee	22.03.2011		

Joonis 9. Kasutajate tabel administraatori vaates.

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on kasutajate haldamise lehel. Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Administraator leiab tabelis kasutaja, keda on vaja blokeerida/aktiveerida (Joonis 9).
2. Administraator klikib kasutaja blokeerimis/aktiveerimisnupule.
3. Administraatorile näidatakse kinnitusteadet, et otsitav kasutaja on blokeeritud/aktiveeritud.

Tulemus: Kasutaja on blokeeritud/aktiveeritud.

Alternatiivsed vood:

1. Kasutaja proovib ennast blokeerida.
 - 1.1 Kasutajale kuvatakse veateade.

4.2.7. Kuulutuse lisamine

Pealkiri:*

Sisu:*

Joonis 10. Kuulutuse lisamise vorm.

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on kuulutuste lehel.

Põhivoog:

1. Kasutaja täidab kuulutuse vormi (Joonis 10).
2. Kasutaja klikib 'salvesta' nupule.
3. Kasutajale näidatakse kinnitusteadet, et kuulutus on lisatud.

Tulemus: Kuulutus on salvestatud. Kuulutus on nähtav kõigile esilehel ja kuulutuste lehel.

Alternatiivsed vood:

1. Kuulutuse lisamine ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.
1.1 Kasutajale kuvatakse veateade.

4.2.8. Väljakutse sisestamine

Mängu tüüp:* Üksikmäng Paarismäng

Väljakutsuja:* Eesnimi Perenimi

Väljakutsuja 2:*

Väljakutsutav:*

Väljakutsutav 2:*

Asukoht:*

Aeg:*

Joonis 11. Väljakutse sisestamise vorm.

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on väljakutse sisestamise lehel. Kasutaja on sisse logitud. Kasutaja staatus on aktiivne.

Põhivoog:

1. Kasutaja täidab väljakutse vormi (Joonis 11). Kasutaja valib mängu tüübi ja toimumiskoha.
2. Kasutaja klikib 'salvesta' nupule.
3. Kasutajale näidatakse kinnitusteadet, et väljakutse on saadetud (salvestatud).

Tulemus: Väljakutse on salvestatud. Väljakutsele pannakse staatus „ootan kinnitust“.

Alternatiivsed vood:

1. Väljakutse sisestamine ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.
 - 1.1 Kasutajale kuvatakse veateade.
2. Väljakutse sisestamine ebaõnnestus.
 - 2.1 Kasutajale kuvatakse veateade.

4.2.9. Väljakutse kinnitamine

Tegutseja: Kasutaja.



Eeltingimused: Kasutaja on enda konto väljakutsete lehel. Kasutaja on sisse logitud. Kasutaja staatus on aktiivne. Kasutaja on väljakutsutav.

Põhivoog:

1. Kasutaja klikib kinnitamise nupule.
2. Kasutajale näidatakse kinnitusteadet, et väljakutse on kinnitatud.

Tulemus: Väljakutse on kinnitatud. Väljakutsele pannakse staatus „ootan tulemust“.

4.2.10. Väljakutsest loobumine

Kuupäev	Asukoht	Tulemus	Võitja	Staatus
20.05.2011 03:20	first place			 

Joonis 12. Väljakutsete tabel enne asukoha kinnitamist/loobumist kasutaja vaates.

Tegutseja: Kasutaja

Eeltingimused: Kasutaja on enda konto väljakutsete lehel. Kasutaja on sisse logitud. Kasutaja staatus on aktiivne. Kasutaja on väljakutsutav.

Põhivoog:

1. Kasutaja klikib loobumise nupule (Joonis 12).

2. Kasutajale näidatakse kinnitusteadet, et väljakutse on tühistatud.

Tulemus: Väljakutse on tühistatud. Väljakutsele pannakse staatus „tühistatud“. Väljakutse viiakse lõpetatud väljakutsete tabelisse.

4.2.11. Väljakutse tulemuse sisestamine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on väljakutse tulemuse sisestamiselehel. Kasutaja on sisse logitud. Kasutaja staatus on aktiivne. Kasutaja on väljakutsutav või väljakutsuja.

Põhivoog:



1. Kasutaja sisestab tulemuse.
2. Kasutaja klikib kinnitamise nupule.
3. Kasutajale näidatakse kinnitusteadet, et tulemus on salvestatud.

Tulemus: Väljakutse tulemus on salvestatud. Väljakutsele pannakse staatus „ootan tulemuse kinnitust“.

Alternatiivsed vood:

1. Väljakutse tulemuse sisestamine ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.
 - 1.1 Kasutajale kuvatakse veateade.

4.2.12. Väljakutse tulemuse kinnitamine

Tulemus	Võitja	Staatus
7/6(1);5/7;(12/14)	Eesnimi Perenimi Admin pass	 

Joonis 13. Väljakutsete tabel enne sisestatud tulemust kinnitamist/loobumist kasutaja vaates.

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on oma väljakutsete lehel. Kasutaja on sisse logitud. Kasutaja staatus on aktiivne. Kasutaja on väljakutsutav või administraator.

Põhivoog:

1. Kasutaja klikib tulemuse kinnitamise nupule (Joonis 13).
2. Toimub punktide ümberarvutamine.
3. Kasutajale näidatakse kinnitusteadet, et väljakutse tulemus on kinnitatud. Väljakutse viiakse lõpetatud väljakutsete tabelisse.

Tulemus: Väljakutse tulemus on kinnitatud. Mängijate punktid on ümber arvutatud. Väljakutsele pannakse staatus „lõpetanud“.

4.2.13. Väljakutse tulemustest loobumine

Tegutseja: Kasutaja

Eeltingimused: Kasutaja on oma väljakutsete lehel. Kasutaja on sisse logitud. Kasutaja staatus on aktiivne. Kasutaja on väljakutsutav.

Põhivoog:

1. Kasutaja klikib tulemuse loobumise nupule.
2. Kasutajale näidatakse kinnitusteadet, et väljakutsega tegeleb edaspidi administraator.

Tulemus: Väljakutsele pannakse staatus „lahendamisel“.

4.2.14. Väljakutse muutmine või konfliktse situatsiooni lahendamine

Tegutseja: Kasutaja

Eeltingimused: Kasutaja on väljakutse muutmise lehel. Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja muudab väljakutse andmeid.
2. Kasutaja klikib kinnitus nupule.

3. Toimub punktide ümberarvutamine.
4. Kasutajale näidatakse kinnitusteadet, et väljakutse tulemus on kinnitatud.

Tulemus: Väljakutse tulemus on kinnitatud. Mängijate punktid on ümber arvutatud. Väljakutsele pannakse staatus „lõpetanud“.

Alternatiivsed vood:

1. Väljakutse ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.
 - 1.1 Kasutajale kuvatakse veateade.

4.2.15. Väljakutse kustutamine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on väljakutsete haldamise lehel. Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja leiab väljakutsete tabelis mingi väljakutse, mida on vaja kustutada.
2. Kasutaja klikib kustutamise nupule.
3. Toimub punktide ümberarvutamine.
4. Kasutajale näidatakse kinnitusteadet, et otsitav väljakutse kustutatud.

Tulemus: Väljakutse on kustutatud.

Alternatiivsed vood:

1. Väljakutse kustutamine ebaõnnestus, väljakutse staatus on juba lõpetanud ja väljakutse on toimunud rohkem 30 päeva tagasi.
 - 1.1 Kasutajale kuvatakse veateade.

4.2.16. Uudise lisamine/uuendamine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on uudise lisamise/muutmise lehel. Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja täidab uudise vormi/muudab uudise info.
2. Kasutaja klikib nupule 'salvesta' või 'uuenda'.
3. Kasutajale näidatakse kinnitusteadet, et uudis on lisatud või uuendatud.







Tulemus: Uudis on salvestatud või uuendatud.

Alternatiivsed vood:

1. Uudise lisamine või kustutamine ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.
 - 1.1 Kasutajale kuvatakse veateade.

4.2.17. Uudise kustutamine

Uudised

Kuupäev	Pealkiri	Author		
06:14 19.04.2011	Federer alustas hooaja esimest saviliivaväljaku turniiri võimsalt	Eesnimi Perenimi		
06:13 19.04.2011	USKUMATU: Kui Nadal viimati kaotas, mängis Jordan veel NBAs	Eesnimi Perenimi		
06:12 19.04.2011	Kanepi hoiab WTA edetabelis 16. kohta, Zopp tõusis kariääri kõrgemale kohale	Eesnimi Perenimi		

Joonis 14. Uudiste tabel administraatori vaates.

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on uudiste haldamise lehel. Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja leiab uudiste tabelis uudise, mida on vaja kustutada (Joonis 14).
2. Kasutaja klikib kustutamise nupule.
3. Kasutajale näidatakse kinnitusteadet, et otsitav uudis kustutatud.

Tulemus: Uudis ja kõik temaga seotud kommentaarid on kustutatud.

4.2.18. Uudise kommenteerimine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on uudiste haldamise lehel. Kasutaja on sisse logitud.

Põhivoog:

1. Kasutaja täidab kommentaari vormi.
2. Kasutaja klikib kinnitamise nupule.
3. Kasutajale näidatakse kinnitusteadet, et kommentaar on lisatud.

Tulemus: Kommentaar on lisatud ja on nähtav uudise lehel. Kommentaari omanik on kommentaari salvestaja.

Alternatiivsed vood:

1. Kommentaari lisamine ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.
 - 1.1 Kasutajale kuvatakse veateade.

4.2.19. Galerii kataloogi lisamine/muutmine

Kausta nimi:*

url1:*

url2:*

url1:*

Salvesta

Joonis 15. Galeriisse kataloogi lisamise vorm.

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on sisse logitud. Kasutaja on kataloogi lisamise/muutmise lehel. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja täidab kataloogi vormi (Joonis 15).
2. Kasutaja klikib salvestamise/muutmise nupule.
3. Kasutajale näidatakse kinnitusteadet, et kataloog on salvestatud/muudetud.

Tulemus: Kataloog on salvestatud ja nähtav galerii lehel.

Alternatiivsed vood:

1. Galerii kataloogi lisamine või kustutamine ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.
 - 1.1 Kasutajale kuvatakse veateade.

4.2.20. Galeriist kataloogi kustutamine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on galerii lehel. Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja leiab kataloogide tabelist kataloogi, mida on vaja kustutada (Joonis 16).
2. Kasutaja klikib kustutamise nupule.
3. Kasutajale näidatakse kinnitusteadet, et otsitav kataloog kustutatud.

Tulemus: Kataloog ja kõik temaga seotud pildid on kustutatud.

[Lisa uus album](#)

Albumi nimi	Viimati muudetud	Pildid(tk)			
Tishka	01:56 10.04.2011	8	Veel Veel		 

Joonis 16. Galerii tabel administraatori vaates.

4.2.21. Pildigalerii lisamine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on galerii lehel. Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja leiab kataloogide tabelist kataloogi, millele soovib lisada pildi.
2. Kasutaja klikib pildi lisamise nupule.
3. Kasutaja suunatakse pildi lisamise vormile.
4. Kasutaja täidab pildi lisamise vormi ja klikib lisamise nupule.
5. Kasutajale näidatakse kinnitusteadet, et pilt on lisatud.

Tulemus: Pilt on salvestatud ja kättesaadav õigest kataloogist.

Alternatiivsed vood:

1. Pildigalerii lisamine ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele, näiteks faili tüüp pole lubatud.
 - 1.1 Kasutajale kuvatakse vastav veateade.

4.2.22. Turniiri lisamine/muutmine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on turniiri lisamise/muutmise lehel. Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja täidab turniiri lisamise või muutmise vormi turniiri infoga.
2. Kasutaja klikib nupule 'salvesta' või 'uuenda'.
3. Kasutajale näidatakse kinnitusteadet, et turniir on lisatud või uuendatud.

Tulemus: Turniir on salvestatud või uuendatud (Joonis 17).





Alternatiivsed vood:

1. Turniiri lisamine või muutmine ebaõnnestus, mõne kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.

1.1 Kasutajale kuvatakse vastav veateade.

Turniiride kalender

[Lisa uus](#)

Date	Tartus	Eestis	Maailmas		
12.Dets		Turniir Eestis			
12.Dets	Uus turniir Tartus				

Joonis 17. Turniiride tabel administraatori vaates.

4.2.23. Turniiri kustutamine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on turniiride lehel. Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja leiab turniiride tabelist turniiri, mida on vaja kustutada.
2. Kasutaja klikib kustutamise nupule.
3. Kasutajale näidatakse kinnitusteadet, et vastav turniir on kustutatud.

Tulemus: Turniir on kustutatud.

4.2.24. Mänguväljaku lisamine/muutmine

Väljaku nimi:*

Address:*

Telefon:*

Lahtioleku ajad:*

Lisainfo:

Salvesta

Joonis 18. Väljaku lisamise vorm.

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on mänguväljaku lisamise/muutmise lehel (Joonis 18). Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja täidab mänguväljaku lisamise/muutmise vormi info.
2. Kasutaja klikib nupule 'salvesta' või 'uuenda'.
3. Kasutajale näidatakse kinnitusteadet, et mänguväljak on lisatud/uuendatud.

Tulemus: Väljak on salvestatud/uuendatud.

Alternatiivsed vood:

1. Väljaku lisamine ebaõnnestus, mõni kohustuslik lahter on täitmata või sisestatud andmed ei vasta valideerimise reeglitele.
 - 1.1 Kasutajale kuvatakse vastav veateade.

4.2.25. Mänguväljaku kustutamine



Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on mänguväljakute lehel (Joonis 19). Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

1. Kasutaja leiab mänguväljakute tabelist väljaku, mida on vaja kustutada.
2. Kasutaja klikib kustutamise nupule.
3. Kasutajale näidatakse kinnitusteadet, et vastav mänguväljak on kustutatud.

Tulemus: Mänguväljak on kustutatud.

Nimetus	Aadress	Telefon	Lahtioleku ajad	Lisainfo		
first place	asdasdcwcb		every day daya	test test lisa		

Joonis 19. Väljakute tabel administraatori vaates.

4.2.26. Klubi reeglite/kontaktinfo muutmine

Tegutseja: Kasutaja.

Eeltingimused: Kasutaja on /admin lehel. Kasutaja on sisse logitud. Kasutajal on administraatori õigus.

Põhivoog:

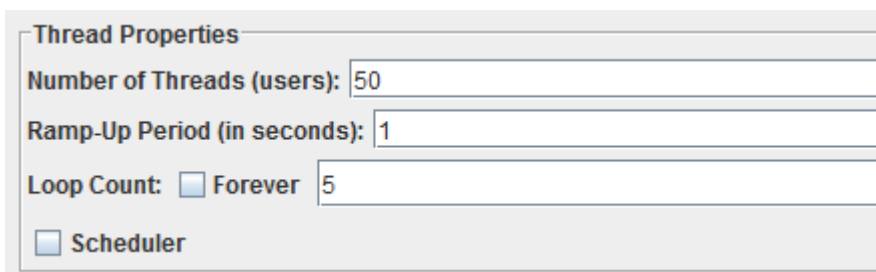
1. Kasutaja klikib reeglite või kontaktinfo muutmise lingile.
2. Kasutaja muudab andmeid tekstitoimeti abil.
3. Kasutaja klikib 'salvesta' nupule.
4. Kasutajale näidatakse kinnitusteadet, et reeglid või kontaktinfo on muudetud.

Tulemus: Reeglid või kontaktinfo on uuendatud. Muudetud andmeid näidatakse reeglite või kontaktinfo lehel.

5. Testimine

5.1. Jõudlustestid

Tellijä nõue oli, et korraga saaks saiti kasutada kuni 50 inimest. Jõudluse testimiseks kasutati programmi JMeter (Joonis 20). Tehti kolm katset, milles igas katses võtsid osa 50 genereeritud kasutajat, ajaintervalliga 1 sekund 5 korda.



Joonis 20. JMeter konfiguratsioon.

Katsete tulemused on esitatud Tabelis 2 ja Tabelis 3. Tabelis 2 on andmed sisselogimata kasutaja ja Tabelis 3 sisselogitud kasutaja kohta. Tabelite ridades on päringud lehtedele ja veergudes vastavalt päringute arv, lehe keskmine avamisaeg (ms), minimaalne avanemisaeg (ms), maksimaalne avanemisaeg (ms) ja vigade arv (%), st. protsent päringute arvust, mil tekkis *timeout* ja leht ei tulnud üldse lahti.

Tabel 2. Sisselogimata kasutaja viibib portaalis.

url	Arv	keskmine(ms)	miinum(ms)	maksimum(ms)	viga(%)
/home	250	1333	287	3507	0
/image/profile/5	250	627	20	1659	0
/ranking	250	1085	151	3237	0
/challenges	250	1479	254	3258	0
/tournaments	250	928	74	2209	0
/rules	250	795	52	2138	0
/places	250	870	63	2528	0
/login	250	489	25	1668	0
/register	250	344	30	1635	0
Keskmine	250	883	106	2427	0

Tabel 3. Kasutaja logib sisse, viibib portaalis ja sisestab kuulutuse.

url	Arv	keskmine(ms)	miinimum(ms)	maksimum(ms)	viga(%)
/login	250	426	30	2025	0
/processlogin	250	1890	214	4603	0
/	250	1359	229	3766	0
/home	250	1553	142	3794	0
/image/profile/5	250	499	16	1966	0
/myaccount	250	412	33	1721	0
/myaccount/avatar	250	216	23	1791	0
/home/read/32	250	870	26	1952	0
/adverts	500	1870	35	6394	0
Keskmine	278	1010	83	3112	0

Mõlemates tabelites vigade arv on 0%. See tähendab, et testimise ajal portaal oli alati kättesaadav ja ei tulnud mitte ühtegi *timeout*-i.

Kõige olulisem veerg on 'keskmine' (ms) – arv, mis näitab keskmist lehe avanemise aega millisekundites. Kõige aeglasem oli kuulutuste leht. See on loogiline, sest seal kuvati korraga 250 kuulutust andmebaasist. Antud portaali sihtgruppi silmas pidades ei saa realses elus niisugust olukorda juhtuda.

Kokkuvõttes mõlemad keskmised tulemused olid 883 ms ja 1010 ms, mis on vastavuses kliendi sooviga.

6. Kokkuvõte

Käesoleva bakalaureusetöö raames programmeeriti veebiportaali tenniseklubile. Kogu portaali arendamisele, alates nõuete analüüsist kuni rakenduse üleandmiseni, kulus umbes 640 tundi. Lõpptulemuseks on valminud testitud ja reaalselt kasutatav portaali, mis töötab J2EE platvormil, millega tellija oli rahul ja mis on kättesaadav lehel <http://www.tartenna.ee>.

Töö jooksul realiseeriti kõik planeeritud moodulid, nimelt

- administreerimise moodul, kus administraator saab hallata kasutajad, väljakutseid, toetajaid ja kuulutusi;
- väljakutsete moodul, kus kasutajad saavad sisestada väljakutseid, vaadata edetabeleid, teiste liikmete tulemusi jne;
- kasutajate registreerimise ja aktiveerimise moodul;
- „minu konto“ moodul, kus kasutaja saab muuta oma profiili, üles laadida avatari, muuta parooli, oma staatust ja vaadata oma väljakutseid;
- administraatori poolt hallatav galerii moodul koos muutmise võimalusega;
- uudiste ja kuulutuste moodul. Uudiste vormistamiseks kasutati tekstitoimetit TinyMCE;
- Turniiride ja väljakute moodul, mis on samuti hallatavad administraatori poolt;
- Reeglite ja „meie kontaktid“ moodulid koos vormistamise võimalusega.

Lisaks võib väita, et seda rakendust võib kasutada ka mõni teine tenniseklubi, kusjuures tingimuseks on sama punktisüsteemi rakendamine. Võimalik on lisada uue portaali jaoks oma kujundus.

Small portal for tennis club

Bachelor Thesis

Jevgeni Koltšin

Summary

The purpose of the present bachelor thesis was to create a small portal for Tartu tennis club on J2EE platform, using the most popular frameworks like Spring MVC, Spring Security, Apache Tiles, Hibernate, JSR-303 bean validation, jQuery. The created portal is running on Apache Tomcat servlet container, behind Apache 2. Portal is accessible by url www.tartenna.ee. Total programming with requirements analyzing took 640 hours. During this time different modules have been created.

- Challenges module is the most important module, where logged in users can create new challenges, accept or decline them, check other player's single and pair game results.
- Administration module, where administrator can modify other user's profile, challenges, banners on home page.
- User registration module. Every guest can register himself on portal. After registration users activation by administrator is mandatory. If the user is not activated, she cannot log in.
- News and adverts module. Administrator can easily create, delete or update news. News visual part is created using TinyMCE html editor. Registered and logged in users can post comments to news.
- Image gallery. Administrator can create folders in gallery, upload pictures, which will be accessible from gallery. Visual effects are created using jQuery framework.
- Tournaments. Separate module, which allows administrator to create tournaments calendar. This module enables to add, modify and delete tournaments in the calendar.
- My account module, which allows users to modify their profiles, upload avatars, change passwords, status, review their statistics, finished and unfinished challenges.

Portal can be easily configured and used by another tennis clubs if this club will decide to have the similar system for saving challenges results, users, news management etc. All they need is to create the new visual part for their portal and launch this application on their server.

Kasutatud kirjandus

- [1] Spring Framework Reference Documentation. [Veebis. 25.05.2011].
<http://static.springsource.org/spring/docs/3.0.x/spring-framework-reference/html/>
- [2] Spring Security 3.1 Reference Documentation. [Veebis. 25.05.2011].
<http://static.springsource.org/spring-security/site/docs/3.1.x/reference/springsecurity.html>
- [3] Jeff Linwood, Dave Minter, Beginning Hibernate, 2nd Edition, Apress, 2010.
- [4] Apache Tiles 2 Documentation. [Veebis. 25.05.2011].
<http://tiles.apache.org/framework/index.html>
- [5] JSR-303 Bean Validation Specification. [Veebis. 25.05.2011].
<http://jcp.org/aboutJava/communityprocess/final/jsr303/index.html>
- [6] J. Chaffer, K. Swedberg, jQuery 1.4 Reference Guide, Packt Publishing, 2010.
- [7] Charles A. Bell, Expert MySQL, Apress, 2007.
- [8] Apache Tomcat Documentation. [Veebis. 25.05.2011].
<http://tomcat.apache.org/tomcat-6.0-doc/index.html>
- [9] Apache Ant manual. [Veebis. 25.05.2011].
<http://ant.apache.org/manual/index.html>
- [10] Apache Ivy Documentation. [Veebis. 25.05.2011].
<http://ant.apache.org/ivy/history/trunk/index.html>
- [11] Forseliuse Tenniseklubi. [Veebis. 25.05.2011].
<http://www.hot.ee/f/forseliusetk/>

[12] Tartu Tennisekool. [Veebis. 25.05.2011].

<http://www.tartutennisekool.ee>

[13] Tenniseklubi Sinnet. [Veebis. 25.05.2011].

<http://www.saaremaatennis.ee>

[14] Promenaad. [Veebis. 25.05.2011].

<http://www.promenaad.ee>

[15] Apache JMeter User Manual. [Veebis. 25.05.2011].

<http://jakarta.apache.org/jmeter/usermanual/index.html>

[16] Craig Walls, Spring in Action, Third Edition, MEAP, 2011.

[17] Apache Subversion Documentation. [Veebis. 25.05.2011].

<http://subversion.apache.org/docs/>

[18] Gary Mak, Hibernate Recipes: A Problem-Solution Approach, Apress, 2010.

[19] Peter Mularien, Spring Security 3, Packt Publishing, 2010.

[20] Willie Wheeler, John Wheeler, and Joshua White, Spring in Practice, MEAP, 2011.

[Veebis. 25.05.2011].

<http://www.manning.com/wheeler/>

[21] TinyMCE Documentation. [Veebis. 25.05.2011].

<http://tinymce.moxiecode.com/wiki.php/TinyMCE>