

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Arvutiteaduse instituut

Spordialaliidu infosüsteem

Bakalaureusetöö

Üliõpilane: Rasmus Lelumees
Üliõpilaskood: 112162 IAPB
Juhendaja: Jaagup Irve

Tallinn
2014

Autorideklaratsioon

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

26.05.2014

.....
(kuupäev)

Rasmus Lelumees

.....
(allkiri)

Annotatsioon

Käesoleva töö eesmärgiks on luua infosüsteem ühele spordialaliidule, täpsemalt Eesti Aerutamisföderatsioonile. Infosüsteem peab pakkuma võimalust uudiste, meedia ja sündmuste mugavaks publitseerimiseks ning karikavõistluste koondtulemuste avalikustamiseks. Karikavõistlustele registreerumine, tulemuste sisestamine ning koondtulemuste arvutamine peaks olema digitaliseeritud ning toimuma autentimist nõudvas keskkonnas.

Töö peamiseks probleemideks on kaardistada spordialaliidu vajadused, seejärel leida neile parim tehnoloogiline väljund ning lõpuks kogu süsteem realiseerida. Realiseerimisel on põhirõhk turvalise siseveebi loomisel, mis võimaldaks senisest efektiivsemat võistluste korraldamist ning tulemuste üle arvestuse pidamist. Siseveeb peaks automatiseerima seni manuaalselt toimunud arvutused ning võistlustele registreerumised. Oluliseks osaks on ka atraktiivne avalik veeb, mis peaks olema kasutatav nii arvutil kui ka mobiilsetel seadmetel.

Töö tulemusena valmib infosüsteem, mis võimaldab pidada karikaarvestust sportlaste vahel. Samuti valmib avaliku veebi osa, kus on võimalik publitseerida meediat, sündmusi ja uudiseid ning mis kuvab sünkroniseeritud teavet eelnevalt siseveebis leitud karikaarvestuse koondtulemuste kohta. Infosüsteem läbib kasutajate testperioodi ning selle tulemusena tehakse kokkuvõtteid kasutajakogemuse kohta ning funktsionaalsuste toimimise kohta. Lisaks pakutakse välja edasiarenduse võimalusi ning lisafunktsionaalsusi, mida saaks siseveebis realiseerida.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 41 leheküljel, 6 peatükki, 10 joonist ja 5 näidiskoodi.

Abstract

The main purpose of this thesis is to create an information system for a sports association called Estonian Canoeing Federation. The system must allow easy and maintainable publishing for news, media and sports events. Competition registrations, entering the results and calculation of aggregate results should be digitalized and take place in authenticated environment.

The main problems are: mapping the needs of Estonian Canoeing Federation, finding the best solutions for those needs and eventually developing the whole system. During development, the main focus is to create a secure internal web that enables more efficient arrangement and maintenance of results. The internal web should also automate the calculations and digitalize the registrations. The attractive and responsive public web part is also essential.

As a result, an information system is created that features keeping record over athletes' cup results. In addition, the public web part is developed that enables publishing news, events and media. The public web also displays synchronized information about the aggregate results that are calculated in internal web. The created system passes the test period and as a result, conclusions of user experience and functionalities are made. Finally, some improvement possibilities for internal web are outlined, so that the system would be even more efficient and useful.

The thesis is in Estonian and contains 41 pages of text, 6 chapters, 10 figures and 5 sample codes.

Lühendite ja mõistete sõnastik

EAF	<i>Estonian Canoeing Federation</i> Eesti Aerutamisföderatsioon
JAR	<i>Java Archive</i> Java arhiivifail, tüüpiliselt kasutusel paljude Java class failide konteinerina, faililaiendiks .jar
XML	<i>Extensible Markup Language</i> Märgistuskeel, mis on nii inimloetav kui ka masinloetav ja mida kasutatakse andmete saatmiseks üle võrgu
POM	<i>Project Object Model</i> Maveni konfiguratsioonifail, mis defineerib kogu projekti olemuse
MVC	<i>Model-View-Controller</i> Mudel-Vaade-Kontroller ehk arhitektuurimuster, mida kasutatakse veebirakendustes
CSRF	<i>Cross Site Request Forgery</i> Kuritahtliku küberründe üks tüüpe, kus autoriseerimata transaktsioone teostatakse võõralt saidilt läbi kasutaja sessiooni veebilehele, mis sessiooni tõttu vastava kasutaja tegevusi usaldab
JDBC	<i>Java Database Connectivity</i> Java teek andmebaasiga ühendumiseks
JSP	<i>JavaServer Pages</i> Tehnoloogia, mis võimaldab arendajatel luua dünaamiliselt genereeritud HTML või XML-põhiseid veebilehti
JSTL	<i>JavaServer Pages Standard Tag Library</i> Tehnoloogia, mis defineerib uusi <i>tag-e</i> /märgendeid JSP lehtede jaoks.

AJAX	<i>Asynchronous JavaScript And XML</i> Tehnoloogia, mida kasutatakse sisu kiireks asünkroonseks laadimiseks, kiirendades seeläbi lehel tehtud toiminguid
API	<i>Application Program Interface</i> Kogum rutiine, protokolle ja tööriistu tarkvararakenduste ehitamiseks
CMS	<i>Content Management System</i> Sisuhaldussüsteem
W3C	<i>World Wide Web Consortium</i> Rahvusvaheline kommuun, kus töötatakse välja veebistandardeid
JSON	<i>JavaScript Object Notation</i> Andmevahetusformaad, alternatiiv XML-ile, väga minimalistliku struktuuriga
Front-end	<i>Interface between user and back-end</i> Liides kasutaja ja <i>back-end</i> 'i vahel
Back-end	<i>Server-side logic, serves user indirectly via front-end</i> Serveripoolne loogika, teenindab kasutajat kaudselt <i>front-end</i> 'i vahendusel
CORS	<i>Cross-origin resource sharing</i> Mehhanism, mis võimaldab veebilehe ressursse mõnest teisest domeenist pärida

Jooniste nimekiri

Joonis 1. MVC mustri toimimine	14
Joonis 2. Siseveebi andmebaasi struktuur	22
Joonis 3. Siseveebi avaleht ning koondtulemuste vaade	23
Joonis 4. Siseveebi ja avaliku veebi vaheline suhtlus	27
Joonis 5. EAF-i olemasolev veebileht ja Sõudeliidu leht.....	29
Joonis 6. EAF-i uue veebi põhilised disainielemendid	30
Joonis 7. Tava- ja mobiilivaate võrdlus.....	31
Joonis 8. Kalendri sündmuse vaade.....	32
Joonis 9. Ekraanitõmmis siseveebi prototüübist	33
Joonis 10. Avaliku veebi avakuva muudatused.....	35

Näidiskoodide nimekiri

Näidiskood 1. Autentimise konfiguratsioon.....	24
Näidiskood 2. JSTL core taglib-i kasutamise näide	25
Näidiskood 3. Avalehe kontrolleri.....	26
Näidiskood 4. Andmebaasiühenduse loomine.....	26
Näidiskood 5. CORS filtri koodinäide	27

Sisukord

1. Sissejuhatus	11
2. Süsteemi ülevaade	12
2.1 Süsteemi eesmärgid	12
2.2 Metoodika ja tehnoloogiad	12
2.2.1 Apache Maven.....	12
2.2.2 Java Spring framework ja MVC.....	13
2.2.3 Spring Security	14
2.2.4 PostgreSQL ja JDBC.....	15
2.2.5 JSP ja JSTL.....	15
2.2.6 jQuery	15
2.2.7 Bootstrap.....	16
2.2.8 Wordpress ja PHP.....	16
2.2.9 MySQL	17
2.2.10 HTML5	17
2.2.11 Jackson, JSON ja AJAX.....	17
3. Süsteemi taust.....	19
3.1 Valdkond	19
3.2 EAF-i struktuur.....	19
3.3 Sarnased lahendused ja alternatiivid.....	20
3.4 Sihtgrupp	20
4. Arendusprotsess.....	22
4.1 Siseveeb	22
4.1.1 Andmebaas	22
4.1.2 Disain ja staatiline kasutajaliides.....	23
4.1.3 Veebirakendus	24
4.2 Avalik veeb.....	27
4.2.1 Kasutajaliides ja disain	28
4.2.2 Wordpress.....	31
4.2.3 Kalendri plugin.....	31
5. Süsteemi testimine.....	33
5.1 Esialgne prototüüp.....	33
5.1.1 Lahenduse testimise keskkond	33

5.1.2 Tagasiside	34
5.1.3 Tehtud muudatused	34
5.2 Valminud siseveeb.....	36
5.2.1 Tulemus	36
5.2.2 Lahendused.....	36
5.2.3 Kokkuvõte	36
6. Kokkuvõte	37
Summary.....	39
Kasutatud kirjandus	41

1. Sissejuhatus

Interneti kaudu saadava informatsiooni osakaal on üha suurem, jättes teised infokanalid selgelt varju. Ühe spordialaliidu jaoks on oluline suur harrastajate arv, mistõttu üritatakse inimesi erinevate infokanalite kaudu just nende ala juurde meelitada. Et internet domineerib selgelt infokanalite seas, siis peetakse spordialaliitudes atraktiivse avaliku veebi olemasolu väga tähtsaks. Käesolev bakalaureusetöö keskendub ühe spordialaliidu vajadustele, võttes näiteks Eesti Aerutamisföderatsiooni (edaspidi ka EAF). Töös kaardistatakse EAF-i vajadused ning realiseeritakse vajadustele vastav infosüsteem.

Aerutamislüüdi näitel oli olemasolev veebileht aegunud ja raskesti hallatav. Haldusliides toimus vaid Internet Explorer brauseris, kusjuures soovitatud versiooniks oli Internet Explorer 7. Lisaks toimub EAF-is ja ka teistes spordialaliitudes palju erinevat paberimajandust seoses võistlustele registreerumise, tulemuste sisestamise ning koondtulemuste arvutamisega. Seda kõike oleks võimalik automatiseerida, et kiirendada kogu protsessi ning vältida inimlikke vigu arvutustes ning sellest tulenevat segadust autasustamisel. Näiteks viibis 2011. aasta hooaja lõppüritusel arvutusvigade tõttu autasustamine 2 tundi ning lõpuks esitati ikka kahes karikaklassis protest vigase järjestuse kohta.

Käesolevas töös loodava süsteemi võib jagada kaheks. Esimene pool, avalik veeb, peab olema atraktiivne, kasutatav erinevatel seadmetel (sh mobiilid ja tahvelarvutid) ning võimaldama mugavat sisuhaldust uudiste, meedia ja sündmuste publitseerimiseks. Töö teine pool, siseveeb, seab eesmärgiks karikaarvestuse digitaliseerimise, mis põhineb süsteemi kaudu võistlustele registreerumisel, tulemuste sisestamisel ning automaatsel koondtulemuste arvutamisel. Siseveebis arvutatud koondtulemused peaksid olema kokkuvõtlikult jälgitavad avalikus veebis.

Avaliku veebi puhul kasutatakse mõnd eksisteerivat sisuhaldussüsteemi ning konfigureeritakse seda vastavalt aerutamislüüdi vajadustele. Siseveebi puhul on kriteeriumid spetsiifilisemad ning kogu arendus tuleb algusest lõpuni ise teha.

2. Süsteemi ülevaade

Antud osas antakse ülevaade loodavast süsteemist - käsitletakse lahendusele seatavaid eesmärke ning realiseerimiseks kasutatavaid tehnoloogiaid.

2.1 Süsteemi eesmärgid

Eesti Aerutamisföderatsiooniga koostöös on selles punktis sõnastatud konkreetsed eesmärgid süsteemi komponentidele. Süsteem koosneb kahest komponendist, mis täidavad pisut erinevaid eesmärke, seetõttu on neid kohati käsitletud eraldi.

- Avalik veeb peab võimaldama uudiste, sündmuste ja meedia (pildid ja videod) postitamist sisuhaldusliidese kaudu.
- Avalik veeb peab olema väga hea kasutajaliidesega, atraktiivne ning ligitõmbav, kasutatav erinevatel seadmetel ja ekraanilaiustel.
- Siseveeb peab toetama karikaarvestuse digitaliseerimist. Selleks peab siseveeb:
 - Võimaldama võistlustele registreerumist
 - Võimaldama karikapunktide sisestamist
 - Arvutama automaatselt hooaja lõikes karikapunktid kokku ning moodustama sportlaste paremusjärjetuse
- Avalik veeb peab kuvama kokkuvõtet siseveebis leitud paremusjärjestuste kohta.

2.2 Metoodika ja tehnoloogiad

Käesolevas punktis kirjeldatakse lähemalt süsteemis kasutatavaid tehnoloogiaid ning metoodikaid. Igas alampunktis räägitakse ühest tehnoloogilisest komponendist ning lühidalt selle kasutusviisist loodud süsteemis. Pikemalt kirjeldatakse tehnoloogiate kasutamist punktis Arendusprotsess.

2.2.1 Apache Maven

Maven, täpsemalt Apache Maven projekt, loodi esialgselt selleks, et lihtsustada Jakarta Turbine projekti ehitust. Eesmärgiks oli luua arendusprojektide valmistamise standard, selge

definiitsioon, mis annaks ülevaate projekti koostisosadest ning võimaldaks lihtsalt publitseerida projekti informatsiooni. Lisaks sooviti leida moodus Java arhiivide (JAR) jagamiseks mitme projekti vahel. Tulemusena valmis arendustööriist, mida saab kasutada iga Java-põhise projekti ehitamiseks ning haldamiseks. [1]

Maveni aluseks on üks XML konfiguratsioonifail, kus kirjeldatakse kogu tarkvaraprojekti olemus – projekti informatsioon, sõltuvused, ehituse järjekord ja meetodikad, vajalikud pluginad jpm. Tüüpiliselt on selle faili nimeks pom (Project Object Model). Sõltuvuste realiseerimiseks laeb Maven need automaatselt repositooriumist alla ning salvestab lokaalsesse süsteemi. [2]

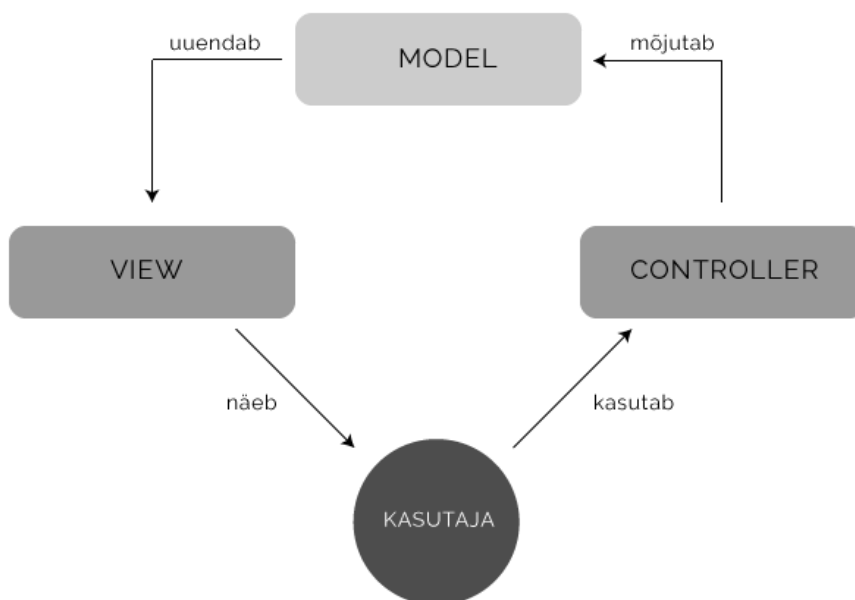
Käesolevas projektis on Maven aluseks siseveebi projektile. Siseveebi projekt on kirjutatud Java keeles ning vajab sõltuvusi, mida on kerge Maveni abil alla laadida ning projekti lisada. Hiljem on erinevate komponentide versioonide uuendamine tänu Mavenile oluliselt lihtsam. Samuti kiirendab Maven siseveebi projekti kokku ehitamist ja annab projekti kohta standardse ülevaate arendajale.

2.2.2 Java Spring framework ja MVC

Java Spring on kõikehõlmav arendusraamistik, mis sobib nii lihtsamatele kui ka väga komplitseeritud projektidele. Spring sisaldab endas palju elementaarset, mida arendaja suure tõenäosusega paljudes projektides tegema peab. Sealjuures on palju tähelepanu pööratud just veebiprojektidele ning nende peamistele vajadustele. Java Spring lihtsustab annotatsioone kasutades eelkõige klassisõltuvuste defineerimist, aga pakub lisaks veel palju muud, näiteks logimise, testimise ja andmete sisestamise lihtsustamine. Kokkuvõtlikult loob Spring projektile algse infrastruktuuri, mis kiirendab edasist arendust. [2]

Java Springi baasil on ehitatud siseveebi osa. Maveni sõltuvustena on defineeritud nii Spring-core, Spring-web kui ka Spring-MVC. Spring-core sisaldab kogu Springi tuumikfunktsionaalsust, Spring-web sisaldab kõike vajalikku veebiprojekti jaoks ning Spring-MVC toob sisse MVC (Model-View-Controller) arhitektuuri realiseerimiseks vajalikud lisafunktsionaalsused. Siseveebi projekt on ehitatud Springi raamistikule ning MVC põhimõtetel.

MVC ehk *Model-View-Controller* (Mudel-Vaade-Kontroller) on tarkvaraarenduses kasutatav arhitektuurimuster kasutajaliideste realiseerimiseks, mis jagab süsteemi kolmeks komponendiks: *model* ehk mudel, *view* ehk vaade ning *controller* ehk kontrolleri.



Joonis 1. MVC mustri toimimine

MVC mustri kohaselt pöördub kasutaja päringutega alati kontrolleri poole, mis otsustab päringu põhjal, mida süsteem edasi teeb. Kogu keerulisem äriloogika asub mudelis. Kontrolleris delegeeritakse päring mudelile, mis oma funktsioonidega uuendab lõpuks informatsiooni vaates. Kasutajale tagastatakse uuenenud vaade. Joonis 1 illustreerib MVC loogikat.

2.2.3 Spring Security

Spring Security on Java Springi kasutavatele rakendustele loodud deklaratiivne turvalisusmehhanism. [3] Ta pakub võimalusi autentimise, autoriseerimise ja teiste turvalisusega seotud funktsionaalsuste realiseerimiseks. [2] Spring Security püüab seda kõike saavutada arendajasõbralikult ja paindlikult. Vaatamata sellele, et Spring Security kasutusala on valdavalt veebirakendused, saab teda kasutada ka iseseisvates rakendustes. Spring Security on avatud lähtekoodiga tarkvara, omades suurt testijate ning edasiarendajate kogukonda. Arendajale tähendab see tugevat tugisüsteemi ning huvitavat võimalust uurida kasutatava tarkvara toimimist lähtekoodi kaudu – seda komponentide mõistmiseks või hoopis nende kohandamiseks ja edasiarendamiseks. [4]

Spring Security on käesolevas töös kasutusel siseveebi poolel. Ta loob elementaarse põhja kasutajate autentimiseks ning erinevate rollide haldamiseks või piiramiseks. Üle on kirjutatud Spring Security `AuthenticationProvider` (Spring Security autentimisprotsessi teostav klass), et oleks võimalik kasutada EAFi siseveebile omast kasutajate struktuuri ning sellele vastavaid tabeleid andmebaasis. Samuti kontrollib ülekirjutatud `AuthenticationProvider` mitte ainult kasutajanime ja parooli, vaid ka seda, kas kasutaja on aktiveeritud moderaatori poolt.

Spring Security abil on defineeritud sisse ja välja logimise elementaarsed parameetrid (kuhu suunatakse sisseloginud kasutaja, kuhu väljaloginud kasutaja jne). Lisaks on defineeritud Spring Security konfiguratsioonis CSRF (Cross Site Request Forgery) kaitse.

2.2.4 PostgreSQL ja JDBC

PostgreSQL (tihti lühendatult „Postgres“) on üks võimekamaid vabavaralisi relatsioonilisi andmebaasisüsteeme maailmas. [4] Postgres'i kasutab EAF-i siseveeb andmete salvestamiseks, kusjuures Postgres'iga ühendumiseks on kasutusel JDBC (Java Database Connectivity) teek. JDBC võimaldab defineerida andmebaasiühendusi ning luua ettevalmistatud päringuid. Siseveebis just nimelt seda tehaksegi.

2.2.5 JSP ja JSTL

JSP ehk JavaServer Pages on tehnoloogia, mis võimaldab arendajatel luua dünaamiliselt genereeritud veebilehti, mis baseeruvad HTML-il, XML-il või mõnel muul dokumenditüübil. JSP lehel on võimalik kirjutada Java koodi staatilise HTMLi vahele ning seeläbi muuta HTMLi sisu ning struktuuri.

JSTL ehk JavaServer Pages Standard Tag Library laiendab JSP spetsifikatsiooni, asendades tavalisemad Java toimingud (itereerimine, tingimused jm) HTML-ile lähedasemate *tag*-idega.

Siseveebis on kasutatud just JSTL-i, täpsemalt JSTL core teeki, et teha lihtsamaid operatsioone vaadetes ning dünaamiliselt andmeid lehel muuta.

2.2.6 jQuery

Veebiarendajate seas väga populaarne jQuery on Javascripti lisa, mis lihtsustab ja kiirendab elementaarsete JavaScripti toimingute tegemist: HTML objektide manipuleerimine, sündmuste käsitlemine, animeerimine ja AJAX (Asynchronous JavaScript And XML). jQuery on

tänapäeval saamas juba standardiks, kuigi hetkel kasutatakse teda ikkagi imporditud API-na. Käesolevas töös on jQuery't kasutatud nii avalikus veebis kui ka siseveebis. Peamiselt on kasutatud jQuery omadust lihtsalt ja kiirelt HTML elemente leida ning nende sisu või stiili (CSS) muuta. Lisaks on kasutatud jQuery pluginaid kasutajakogemuse parandamiseks ning animeerimise hõlbustamiseks.

2.2.7 Bootstrap

Bootstrap on veebiarenduse *front-end* raamistik, mis hõlmab endas valmis kirjutatud stiilipõhja CSS-is ja Javascriptil põhinevat kasutajaliidese funktsionaalsust. Sealjuures on Bootstrap loonud põhja, mida toetavad peaaegu kõik (ka vanemad) veebilehitsejad ning mis on struktuurilt kohanduv iga ekraanilaiusega. [5]

Bootstrap on käesolevas töös aluseks mõlemale komponendile – nii siseveebile kui ka avalikule osale. Bootstrap'i põhjal on saavutatud mõlema poole kohalduvus erinevate ekraanidega. Samuti lihtsustab Bootstrap baaskujunduse ning lehe struktuuri loomist. Bootstrap aitab kokku hoida aega ning vähendas kirjutatud koodi hulka.

2.2.8 Wordpress ja PHP

Wordpress on PHP-l põhinev ning MySQL-i andmebaasi kasutatav sisuhaldussüsteem (CMS ehk Content Management System), mis on peamiselt mõeldud ajaveebide ning blogide loomiseks. Wordpress sisaldab endas mugavat haldusliidest ning pakub tohutult erinevaid pluginaid ja lisasid. Seega on Wordpress kergesti kohandatav ning ideaalne lihtsama funktsionaalsusega veebilehe loomiseks.

Wordpressi peamisteks alternatiivideks on Joomla ning Drupal. Joomla on samuti väga võimekas CMS, mis on isegi parema uudiste haldamise liidese ja võrdlemisi kasutajasõbralik. Negatiivse küljena aga ei ole Joomla'l väga palju lisasid ning eksisteerivaid laiendusi ei ole kõige lihtsam kasutada. Drupal on kolmest CMS-ist kõige rohkem kohandatav. Ta on vahest paremgi lahendus arendajatele, aga mittearendajast süsteemi kasutajale on Drupal natuke liiga keeruline. Eesti Aerutamisföderatsiooni puhul oli mugav ja lihtne haldussüsteem väga oluline, sest süsteemi kasutajateks on inimesed, kelle arvutioskus on tugevalt alla keskmise ning kes kipuvad keerukamate süsteemidega hätta jääma.

Wordpressi kasutati käesolevas süsteemis avaliku veebi baasina. Wordpress ise lahendas avalikus veebis automaatselt uudiste publitseerimise tingimuse, sest blogimine või uudiste

lisamine on Wordpressi sisse ehitatud peamine funktsioon. Sellele lisaks oli olemas tasuta galerii plugin Wordpressile (Nextgen-gallery), mis pakkus palju võimalusi meediafailide publitseerimiseks. Sündmuste publitseerimiseks tuli ise kirjutada EAF-i vajadustele vastav plugin.

Wordpressi väliseks kujundamiseks valmis EAF-i nõudmistele vastav teema, mis kasutab Bootstrapi põhja ning on kasutatav erinevatel ekraanilaiustel.

2.2.9 MySQL

MySQL on hetkel populaarseim vabavaraline andmebaasisüsteem maailmas. [5] MySQL-i populaarsus on tihedalt seotud PHP programmeerimiskeele laialdase kasutusega, mida tihti kombineeritakse MySQL-iga, teenides hüüdnime „dünaamiline duo“. MySQL-i kasutavad paljud populaarsed portaalid ja saidid, näiteks Facebook (oma põhiandmebaasina) või Amazon (oma arvukates siseprojektides). [6]

EAF-i infosüsteemi raames on MySQL andmebaasisüsteem kasutusel avaliku veebi osas. MySQL-i kasutuse tingis avaliku veebi põhjaks olev Wordpress. Wordpress on PHP-l põhinev süsteem ning vajab toimimiseks just MySQL andmebaasi.

2.2.10 HTML5

HTML5 on veebi tuumikkeelee HTML viies põhiredaktsioon. 2007. aastal võttis HTML5 omaks W3C töörühm, varasemalt oli HTML5 nimetus Web Applications 1.0. HTML5 toob kaasa palju uusi HTML elemente ja atribuute, näiteks `<nav>`, `<footer>`, `<video>`, `<audio>` või `<section>` ning input elementide atribuutidena `time`, `email` ja `url`. HTML5 jätab välja mõned esitluseks mõeldud elemendid, mille funktsionaalsust saab saavutada CSS-i abil, näiteks `<center>`, `<front>` ja `<strike>`.

HTML5 kasutamine võimaldas EAF-i avalikus veebis puhtamalt ning kergema vaevaga lahendada mõningad komponendid, näiteks täisekraanil mängiv video kasutab HTML5 `<video>` elementi ning ülamenüüs on kasutatud `<nav>` elementi.

2.2.11 Jackson, JSON ja AJAX

Jackson on Javale mõeldud andmetöötuse teek, mille põhifunktsionaalsuseks on JSON-i parsimine ja genereerimine. Jackson projekt katab lisaks JSONile ka teisi

andmevahetusformaate, näiteks XML, CSV, Avro, CBOR jt. Käesolevas projektis oli oluliseks aga JSONi parsimise ja loomise tugi.

JSON (JavaScript Object Notation) on kergekaaluline andmevahetusformaad, mis põhineb JavaScripti keelestruktuuril. Ometi pole JSON JavaScriptiga otseselt seotud ning tähistab universaalset andmevahetuse formaati, mis on kujunenud populaarseks alternatiiviks XML-ile.

AJAX (Asynchronous JavaScript And XML) on tehnoloogia, mida kasutatakse veebilehtede kiiremaks toimimiseks. AJAX teeb serverile väiksemaid päringuid, nii et kogu lehte ei laeta uuesti, vaid ainult mõningaid komponente vahetatakse välja. Tulemuseks on palju kiirem veebileht, mis tagab parema kasutusmugavuse. AJAX kasutab andmevahetusformaadina XML-i või JSON-it.

Kõik kolm tehnoloogiat (AJAX, JSON ja Jackson) teenivad käesolevas projektis ühte eesmärki, milleks on siseveebis leitud karikaarvestuste koondtulemuste kuvamine avalikku veebi. Siseveebis on kasutusel Jacksoni teek, mis genereerib HTTP GET päringule vastuseks tulemuste JSON massiivi vastava Java klassi baasil. Vastava päringu teeb avalikus veebis AJAX, mis laeb asünkroonselt JSON massiivi ning seejärel saadab JavaScripti töötluse tulemusena parsitud andmed avalehele HTML konteinerisse.

3. Süsteemi taust

Antud osas kirjeldatakse valdkonda, kuhu süsteem kuulub ning sihtgruppi, kes loodavat süsteemi kasutab. Tuuakse välja analoogid ning loetletakse lehe potentsiaalseid kasutajaid ning nende ootusi.

3.1 Valdkond

Käesolev rakendus kuulub spordi ja huvitegevuse valdkonda. Loodav süsteem on kohandatud Eesti Aerutamisföderatsiooni vajadustele, aga sarnane lahendus sobiks ka teistele spordialaliitudele või spordiklubidele.

3.2 EAF-i struktuur

Eesti Aerutamisföderatsioon (lühendatult EAF) on avalikes huvides ja heategevuslikel eesmärkidel tegutsev iseseisev mittetulundusühingute liit, kes ühendab Eestis aerutamise tegelevaid mittetulundusühinguid selle spordiala edendamiseks ning tegevuse kordineerimiseks. EAF-i liikmeteks on aerutamisklubid.

Eesti aerutamise alustalaks on treenerid, kes kuuluvad EAF-i liikmeklubidesse. Treener on sportlase juhendaja ning vastutab käesoleva süsteemi kontekstis selle eest, et sportlase andmed oleksid korrektsed, sportlane oleks karikaarvestusele ning etappidele üles antud (registreeritud) ning et sportlasele kirjutatud tulemus oleks korrektne.

Korraldajad on need inimesed, kes teevad karikaetappide toimumise võimalikuks. Karikaetapi korraldamise eest vastutab üldjuhul üks kindel klubi. Erandiks on vaid Pühajärve regatt, kus lisaks Tartu aerutamisklubile on kaaskorraldajaks EAF ise. Käesoleva süsteemi kontekstis on korraldajaks isik, kes vastutab karikaetapi toimumise ning võistlusinfo kättesaadavuse ja korrektsuse eest. Korraldaja avab etapile registreerumise ning etapi möödudes sisestab etapil osalejatele tulemuse.

Selleks, et kogu korralduslik pool sujuks ning asjaajamine oleks korrektne, on EAF-is tööl peasekretär, kelle rolliks antud süsteemis (siseveebis) on administraator. Hetkel vastutab administraator kasutajaõiguste eest – see tähendab, et iga soovija ei saa omale siseveebi kasutajat registreerida ning kohe näiteks treenerina sportlasi luua ja neid võistleva panna.

Kasutajakonto aktiveerimiseks on vaja administraatori kinnitust. Peasekretär on sisuliselt ka ülevaataja, kes tagab EAF-i poolt korralduse nõuetele vastavuse.

3.3 Sarnased lahendused ja alternatiivid

Eesti mastaabis pole välja töötatud ühtset lahendust, mis sobiks erinevatele spordialaliitudele. Käesoleva töö puhul ei saa seetõttu välja tuua konkreetset konkurenti. Küll aga võib vaadelda, kuidas on erinevad alaliidud tänase seisuga lahendanud oma infosüsteemid.

Eesti Sõudeliit on samuti veespordiliit ning ilmselt võistluste iseloomult ning süsteemi nõuetelt EAF-iga kõige sarnasem. Sõudeliidul eksisteerib avalik veeb (www.soudeliit.ee), mis kasutab Joomla sisuhaldussüsteemi ja pakub samuti võimalusi uudiste ja meedia publitseerimiseks ning sündmuste kuvamiseks. Sõudeliidu avalik veeb oli heaks eeskujuks loodava aerutajate infosüsteemi baasfunktsionaalsuste kirjeldamisel. Sõudeliidul aga puudub siseveeb, kus saaks pidada võistluste üle arvestust ning mis võimaldaks võistlustele registreerumist. Sõudeliidu veebihalduri sõnul oli kunagi plaan siseveebi luua täiesti olemas, aga see jäi teostajate puudumise tõttu soiku. Siseveebi plaaniti kasutada eelkõige dokumentide digitaalseks jagamiseks ning sportlaste üle arvestuse pidamiseks.

Taani aerutamislüit on Taani vaste EAF-ile. Taanlastel on samuti olemas avalik veeb, mis on ühena vähestest spordialaliitude veebidest ka kohalduva disainiga ning mugavalt kasutatav mobiili ekraanil. Kahjuks puudub info selle kohta, kas ja millist sisuhaldussüsteemi taanlased kasutavad. Taanlastel on olemas meedia, sündmuste ning uudiste publitseerimise võimalus ning lisaks ka avalik foorum. EAF-i juhtkond aga arvas, et Eesti Aerutamisföderatsioonile pole foorum oluline, sest pigem kasutatakse tänapäeval suhtlemiseks sotsiaalmeediat ning foorumid kipuvad välja surema.

3.4 Sihtgrupp

Loodava süsteemi puhul on sihtgrupiks kõik aerutamise seotud inimesed – Eesti Aerutamisföderatsiooni juhatuse liikmed, treenerid, ürituste korraldajad, sportlased ning klubide liikmed. Avaliku veebi puhul lisandub kindlasti kasutajate hulka aerutamishuviline, kes ei pruugi olla aktiivne sportlane või harrastaja.

Avalik veeb on oma eesmärkidelt suunatud eelkõige sportlastele ning huvilistele. Sportlased peaksid leidma avalikust veebist informatsiooni võistluste ning teiste aerutajate saavutuste

kohta. Huvilise jaoks on oluline meedia (pildid ja videod) ning kogu staatiline info aerutamise ja EAF-i kohta. Huvilise puhul on oluline just aerutamise vastu huvi tekitamine, mistõttu peaks meedia olema lehel väga hästi presenteeritud. EAF-i jaoks on olulisem just nooremaealine huviline, kellest võiks tulevikus sirguda edukas sportlane.

Siseveebi peamine kasutaja on treener. Treenerile annab siseveeb võimaluse saada ülevaate oma sportlastest, neid võistlustele ilma suurema vaevata registreerida ning hiljem tulemusi vaadata. Teiseks siseveebi rolliks on korraldaja, kes saab siseveebis avada registreerumise oma sündmusele (karikaetapile), karikaetapi kätte jõudes saab korraldaja registreerumise sulgeda ning sisestada tulemusi. Siseveebis on kõikidel kasutajatel võimalik jälgida karikaarvestuse koondtulemusi, mis on oluline EAFi juhtkonnale ning sekretärile.

Avaliku veebi puhul on sihtgrupp pigem nooremapoolne inimene, kes võib olla juba aerutamisega seotud (sportlane) või alles oma teed aerutamiseni leidmas (huviline). Noore sihtrühma puhul on oluline mobiilne kasutatavus ning erinev meediamaterjal (pildid ning videod). Hea kasutajakogemus on samuti väga oluliseks faktoriks, et noor huviline lehel püsiks.

Siseveeb on mõeldud pigem töövahendiks ning seetõttu ei ole selle puhul atraktiivsus ja disain niivõrd oluline. Välimus üritati hoida asjaliku, minimaalse ning lihtsana. Et treenerid ja aerutajad üldiselt ei kipu väga arvuti taga istuma ning teevad oma arvutiga seotud toiminguid tihti hoopis kuskil õues mobiilsetel seadmetel, siis tagati siseveebi loomisel ka mobiili või tahvelarvuti kasutamise võimalus.

4. Arendusprotsess

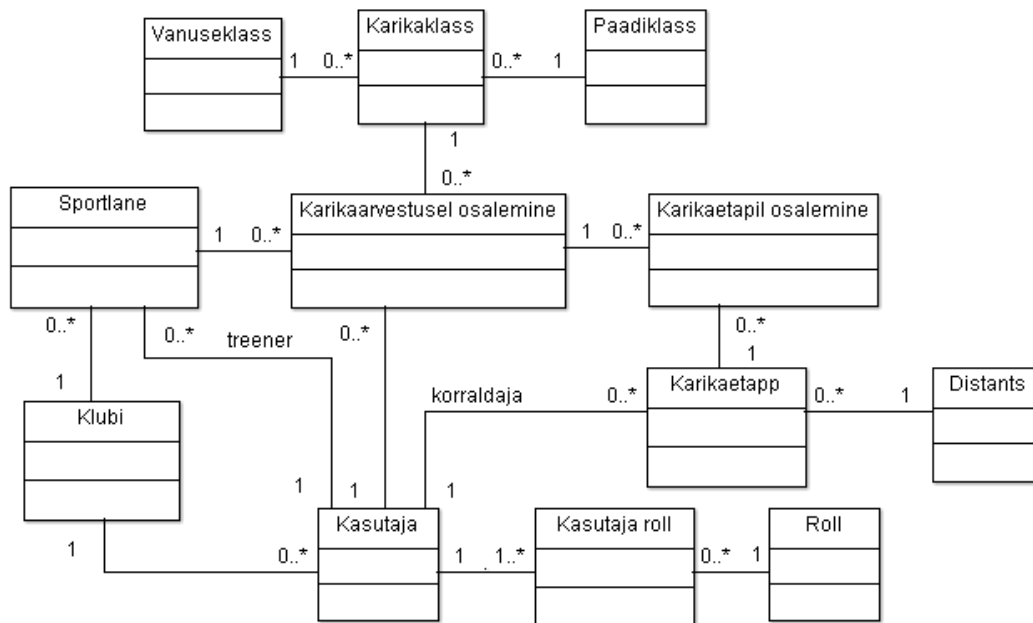
Käesolevas alajaotuses kirjeldatakse süsteemi valmimist, tuues välja arendusprotsessi olulisemad sammud. Et siseveeb ja avalik veeb valmisid iseseisvate komponentidena, siis käsitletakse nende valmimist eraldi.

4.1 Siseveeb

Antud punktis räägitakse sammhaaval siseveebi valmimisest. Arendusprotsess algas olemitüüpide kaardistamisest ja andmebaasi projekteerimisest. Seejärel valmisid ilma funktsionaalsuseta vaated, mida sai näidistena presenteerida EAF-ile. Kui vaadete struktuur ja kujundus sai paika, hakkas pihta veebirakenduse loomine. Neid kolme etappi on alljärgnevalt natuke detailsemalt kirjeldatud.

4.1.1 Andmebaas

Siseveebis kasutatav andmebaas on PostgreSQL 9.3. Sõltumata andmebaasisüsteemist ning versioonist tuli aga esmalt kaardistada EAF-i vajadused siseveebile ning konstrueerida andmebaasidiagramm. Joonisel 2 on toodud loodud andmebaasi struktuur.



Joonis 2. Siseveebi andmebaasi struktuur

Vastaval skeemil on näha põhiobjektid ning nendevahelised seosed. Kasutajale on andmebaasis võimaldatud mitme rolli olemasolu, kuigi Eesti Aerutamisföderatsiooni sõnul hetkel sellist

olukorda ei teki, kus treener ja korraldaja on samas isikus. Tuleviku perspektiivi mõttes sai aga selline olukord andmebaasitasandil võimaldatud. Eraldi tabelitena on lahendatud ka karikaarvestusel osalemine ning karikaetapil osalemine, sest karikasarja sportlase lisamine ei tähenda automaatselt, et vastav sportlane osaleb kõikidel etappidel. Distantis eraldi tabelina on vajalik eelkõige seetõttu, et karikaarvestuse järjestamise loogikas võib tulevikus tulla muutusi ning võrdsete punktide arvu korral saaks sellisel juhul määravaks olümpiadistantside tulemused. Niisiis määraks distantsi mitte ainult pikkus, vaid ka olümpiadistantsi tõeväärtus. Praeguses süsteemis järjestatakse sportlased võrdsete tulemuste korral võitude põhjal.

4.1.2 Disain ja staatiline kasutajaliides

Siseveebi rakenduse arendamine algas esialgu *front-end*'i loomisest, et demonstreerida esmalt EAF-ile staatilisi lehekülgi, vajadusel midagi kasutajaliideses muuta ning seejärel rakendus kõikide funktsionaalsustega realiseerida. Siseveebi disain on võrdlemisi askeetlik, sest tegu on pigem töövahendiga. Disain ja liidese karkass loodi Bootstrapi raamistikule. See andis oluliselt kiirema tulemuse ning hoidis kõvasti kokku koodiridu.

The screenshot displays the EAF SISEVEEB application interface. On the left, there is a login form with the title 'EAF SISEVEEB' and 'LOGI SISSE'. The form includes input fields for 'treener@eaf.ee' and 'Parool', a 'Sisene' button, and a link 'Pole veel kasutaja? Registreeri!'. On the right, the main content area shows 'EAF SISEVEEB' with navigation links: 'Kodu', 'Lisa sportlane', 'Haida sportlasi', 'Karikasari', 'Karikaetapid', 'Etapi tulemused', and 'Koondtulemused'. Below this, there are filters for 'Vali hooaeg: 2014' and 'Vali karikaklass: Mehed süst'. A table titled '2014 MEHED SÜST' displays race results with columns for 'Koht', 'Eesnimi', 'Perenimi', 'Klubi', 'Narva regatt', 'Puhajärve regatt', 'Tallinna MV', 'Tartu Kevadregatt', and 'Kokku'.

Koht	Eesnimi	Perenimi	Klubi	Narva regatt	Puhajärve regatt	Tallinna MV	Tartu Kevadregatt	Kokku
1	Rasmus	Testing	SAK Tartu	7	12	9	7	35 punkti
2	Kaipar	Sula	SAK Tartu	9	9	12	0	30 punkti
3	Martin	Luik	SAK Tartu	12	7	0	5	24 punkti
4	Im	minion	SAK Tartu	0	0	0	12	12 punkti
5	Peeter-Paul	Panov	SAK Tartu	6	0	0	0	6 punkti
6	Kevin	Kriisa	SAK Tartu	2	0	0	0	2 punkti

Joonis 3. Siseveebi avaleht ning koondtulemuste vaade

Bootstrap sisaldab valmis disaini põhilistele HTML elementidele, näiteks vormidele ning vormiväljadele. Joonisel 3 on toodud ekraanitõmmised vaadetest (avaleht ja koondtulemused), mis erinevaid Bootstrap'i elemente kasutavad. Samuti pakub Bootstrap *grid*-süsteemi, mis tagab korrektse kasutamise korral kohalduva veebilehe. *Grid*-süsteem tähendab, et veebileht on mõtteliselt jagatud 12-ks kohalduvaks tulpaks. See tähendab, et mobiili- ning teiste väikeste ekraanide tugi on juba olemas. [5]

4.1.3 Veebirakendus

Veebirakenduse *back-end* oli siseveebi viimaseks lüliks, mis pidi ühendama staatilised vaated andmebaasiga ning kõik vajalikud funktsionaalsused realiseerima.

4.1.3.1 Maven projekt ja XML konfiguratsioon

Back-end'i loomist alustati Maven projekti ülesseadmisega ning Springi raamistiku kasutuselevõttuga. Arendusvahendina oli kasutusel Eclipse Kepler, mis võimaldab veebirakendusi kohe programmisiseselt jooksutada mõnel sisseehitatud serveril, antud juhul oli serveriks Tomcat 7.

Maven projekti puhul on nurgakiviks `pom.xml` fail, kus defineeritakse projekti teave ning vajalikud sõltuvused. Käesoleva projekti puhul olid vajalikud sõltuvused Spring, Spring Security, Jackson, Javax servlet ning Postgres-i JDBC.

XML konfiguratsioon hõlmab veebirakenduse puhul ka `web.xml` ja servlet'i konfiguratsiooni kirjutamist. Et antud projektis oli kasutusel Spring Security, osutus vajalikuks ka turvalisusega seotud konfiguratsiooni kirjutamine. Ülaltoodud konfiguratsioonid on antud projekti puhul väga standardsed, mistõttu neid detailselt välja siin ei tooda. Ainsaks erandiks on turvalisuskonfiguratsioonis defineeritud `AuthenticationProvider`, mis vaikimisi töötab Spring Security reeglite järgi. Siseveebis oli aga oluline, et arvestataks ka kasutaja aktiveeritust sisselogimisel. Seetõttu pidi konfiguratsioonis defineerima näidiskoodis 1 toodud read. Klass `InternalWebAuthProvider` tagab EAF-i jaoks kohandatud autentimise, kontrollides lisaks parooli ja kasutajanime vastavusele ka kasutaja aktiveeritust.

```
<authentication-manager>
  <authentication-provider ref="internalWebAuthProvider"/>
</authentication-manager>

<beans:bean id="internalWebAuthProvider"
class="ee.aerutaja.internalweb.authentication.InternalWebAuthProvider"/>
```

Näidiskood 1. Autentimise konfiguratsioon

4.1.3.2 Vaated ja JSP-d

Kuna siseveebi teises etapis sai loodud staatiline sisu, mis põhines HTML-il, CSS-il ja JavaScriptil, siis Java veebirakenduse jaoks oli vaja need vaated JSP-failideks teha. Sisuliselt piisab laiendi muutmisest ning JSP-le omase päise kirjutamisest. JSP päises on oluline defineerida lehe info (keeleks Java, vajadusel sessiooni tõeväärtus jm parameetrid). Lisaks

defineeritakse päises vajalikud JSTL *taglib*-id, antud näites core ja security elemendid. Hiljem saab defineeritud *prefix*-i järgi *taglib*-e kasutada.

```
<c:if test="{sportsman.gender == 'M'}"><c:out value="selected=selected"/></c:if>
```

Näidiskood 2. JSTL core taglib-i kasutamise näide

Näidiskoodis 2 on toodud lihtne näide, kuidas core *taglib*-i saab kasutada *if*-lause moodustamiseks. Kasutatud väli `sportsman.gender` tuleb mudelile kaasa pandud atribuudist.

4.1.3.3 Kontrolleriid ja annotatsioonid

Kontrollerite defineerimine on Springis tehtud lihtsaks annotatsioonide abil. Suvalise Java klassi saab teha Spring MVC mõistes kontrolleriiks `@Controller` annotatsiooniga.

Lisaks kasutatakse kontrollerites veel `@RequestMethod`, `@RequestMapping` ja `@RequestParam` annotatsioone, mis seovad kontrolleri funktsiooni kindla URL-i, meetodi ja nõutud parameetritega. Sellisel kujul oskab kontroller valida päringu jaoks õige meetodi ning sellele vastavalt edasi toimida.

Näidiskoodis 3 on toodud koodinäide kontrollerist, kus on demonstreeritud annotatsioonide kasutamist.

```

@Controller
public class IndexController {

    @Autowired
    DBConnector dbConnector;

    @RequestMapping(value = { "/", "/login" }, method = RequestMethod.GET)
    public ModelAndView login(
        @RequestParam(value = "error", required = false) String error,
        @RequestParam(value = "logout", required = false) String logout) {

        ModelAndView model = new ModelAndView();
        if (error != null) {
            model.addObject("error", "Kasutajanimi või parool ei sobi");
        }

        if (logout != null) {
            model.addObject("msg", "Oled edukalt välja logitud");
        }
        model.setViewName("login");

        return model;
    }
}

```

Näidiskood 3. Avalehe kontrolleri

4.1.3.4 Andmebaasiga ühendumine

Loodavas rakenduses on kasutusel Postgres andmebaas ning sellega ühendumiseks oli eelnevalt defineeritud Maveni sõltuvusena Postgres'i JDBC teek.

```

try {
    Class.forName("org.postgresql.Driver");
    conn = DriverManager.getConnection( DB_url, properties );
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} catch (SQLException sqlE) {
    sqlE.printStackTrace();
}

```

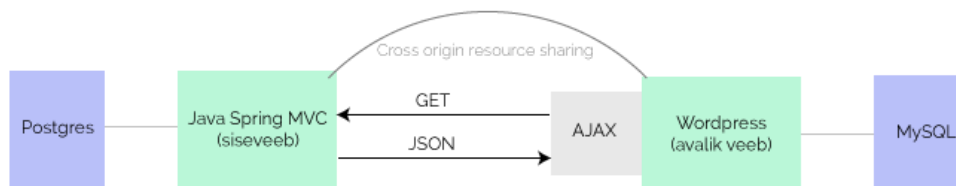
Näidiskood 4. Andmebaasiühenduse loomine

Näidiskoodis 4 on toodud andmebaasiga ühendumise näide Postgres'i JDBC draiveri abil. Ühendus luuakse serveri käimapanekul ning seda kasutatakse korduvalt andmebaasipäringutes.

4.1.3.5 Teenus avaliku veebi jaoks, CORS

Selleks, et avalik veeb saaks küsida informatsiooni siseveebist, tuli luua meetod, mis väljastaks JSON kujul karikaarvestuse koondtulemusi kindlale päringule. JSONi võimaldamine vajab vaid `@ResponseBody` annotatsiooni kontrolleriile. Et aga teises domeenis asuv server saaks

tulemusi küsida, pidi konfigureerima siseveebi poolt selliselt, et CORS (Cross-origin resource sharing) oleks võimaldatud. Joonisel 4 on toodud arhitektuurijoonis siseveebi ning avaliku veebi vahelisest suhtlusest.



Joonis 4. Siseveebi ja avaliku veebi vaheline suhtlus

CORS on mehhanism, mis võimaldab veebilehe ressursse (JSON, JavaScript, fondid jne) mõnest teisest domeenist pärida. Kui CORS poleks võimaldatud, ei lubaks brauser vastavat päringut teha *Same-origin policy* määrusele kohaselt.

Siseveebi projekti puhul tuli CORS-i võimaldamiseks defineerida uus servleti filter, mis lubaks HTTP päises `Access-Control-Allow-Origin` väärtusena kõikvõimalikud variandid. Näidiskoodis 5 on toodud kohandatud filtri peameetod, mis illustreerib HTTP *header*'i konstrueerimist.

```
public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain)
throws IOException, ServletException {
    HttpServletResponse response = (HttpServletResponse) res;
    response.setHeader("Access-Control-Allow-Origin", "*");
    response.setHeader("Access-Control-Allow-Methods", "GET");
    response.setHeader("Access-Control-Max-Age", "3600");
    response.setHeader("Access-Control-Allow-Headers", "x-requested-with");
    chain.doFilter(req, res);
}
```

Näidiskood 5. CORS filtri koodinäide

Lisaks tuli määrata loodud filter `web.xml` failis URLile, kust avalik veeb JSON kujul andmeid pärib.

4.2 Avalik veeb

Antud alajaotuses räägitakse avaliku veebi poole valmimisest. Avaliku veebi nurgakiviks oli hea disain ja kasutajaliides, mistõttu algaski arendusprotsess just sellest osast. Edasi valmistati ette Wordpressi põhi ning juurutati loodud disain Wordpressi teemana süsteemi. Viimase

etapina loodi ja integreeriti funktsionaalsuste realiseerimiseks vajalikud Wordpressi pluginad ning seoti avalik osa siseveebiga.

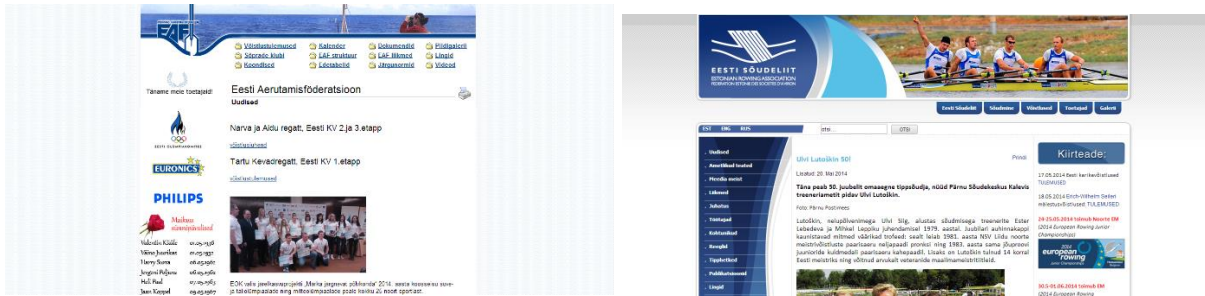
4.2.1 Kasutajaliides ja disain

Et üheks põhiliseks avaliku veebi eesmärgiks oli olla atraktiivne, siis oli disaini ning kasutajaliidese loomine väga oluline. Antud punktis analüüsitakse olemasoleva veebi ja süsteemi analoogide välimust, kirjeldatakse disaini tausta ning mobiiliveebi valmimist.

4.2.1.1 Olemasolev veeb ning analoogid

EAF-i eelmine avalik veeb on loodud ligi 10 aastat tagasi. Loomulikel põhjustel ei olnud selleaegsed veebisaidid arvestanud veel mobiiliekraanidega või vastupidi, kõrge resolutsiooniga. Vana veebi puhul on muutuvat sisu vähe ning peamiselt on tegu staatiliste lehtedega. Avalehele kuvatakse uudiseid ning olemas on ka galerii, aga ebamugava ja keeruka haldusliidese tõttu ei olnud keegi võimeline sealset galeriid pidevalt värskendama. Menüü oli vanas veebis kogu ulatuses alati nähtav, mis on kasutajaliidese mõttes positiivne. Negatiivse küljena aga oli menüü struktureerimata linkide kogumik, kus oli keeruline vajalikku linki üles leida.

Analoogina toodud Eesti Sõudeliidu veebisait on aerutajate eksisteerivast lehest sisukam. Disaini mõttes on jällegi tegu vanema stiiliga, kus lehel on fikseeritud laius ning sisu ei kohaldu ekraani laiusega. Menüüsid on Sõudeliidu veebis kaks ning need on mõlemad ühetasandilised. Positiivse küljena on ühetasandilises menüüs kõik elemendid kogu aeg nähtaval, aga see, et informatsioon on jagatud kahte eraldi menüüsse, ajab kasutaja pigem segadusse ning esmakordsel kasutamisel on tegu ebamugava lahendusega. Sõudeliidu kodulehel loob väga positiivse kuvandi ülaserivas asuv banner, mille vahetuvad sõudmisega seotud pildid. See viib kasutaja kohe teemasse sisse ning pakub ka silmailu. Ideaalis aga võiksid pildid olla suuremad ja kvaliteetsemad. Sellegipoolest pakkus selline temaatikaga haakuv banner inspiratsiooni aerutajate veebilehe loomiseks. Joonisel 5 on toodud ekraanitõmmised EAF-i olemasolevast veebilehest ning Sõudeliidu avalikust veebist.



Joonis 5. EAF-i olemasolev veebileht ja Sõudeliidu leht

Taani aerutamisföderatsiooni koduleht (<http://kano-kajak.org/>) on eelnevatest näidetest kaasaegsem ning on kasutatav ka mobiili ekraanil. Peamenüü on loogiliselt struktureeritud ning sarnaselt Sõudeliidu avalikule veebile on lehe ülaservas kasutusel banner, kus aerutamise pilte kuvatakse. Negatiivse küljena on Taani aerutamislidu lehel väga palju lehe liigendamist ning silmale ei paista kogu sisu ühtse tervikuna.

4.2.1.2 Disaini loomine

EAF-i uue avaliku veebi disaini loomisel lähtuti parimatest lahendustest analoogide juures ning püüti luua pisut teistest eristuv ning meelde jääv leht. Antud punktis tuuakse välja põhilised disainielemendid ning põhjendatakse nende taga peituvat loogikat.

Aerutamise juures on elementaarseteks komponentideks, ilma milleta aerutamist ei toimu, paat ja aer. Tippspordis on väga oluline, et paat ning aer oleksid jäigast materjalist ja kerged, et lisakaal oleks minimaalne maksimaalse voolujoonelisuse ja tugevuse juures. Seetõttu valmistatakse paate ja aere süsinikkiust, mida sulatatakse kärjestruktuuri, et tagada veelgi suurem tugevus ja vastupidavus. Sellest tulenevalt on aerutamise veebis üheks läbivaks disainielemendiks kuusnurk, mis on süsinikkärje baasstruktuur. Süsinikkiudu meenutavaid mustreid on kasutatud igal alamlehel ülaservas asuva menüü ning avalehe video katteks.

Joonisel 6 on toodud EAFi uue avaliku veebi disainielemendid ning põhivärvid. Sinine sümboliseerib vett ja puhtust, tumehall kindlust ja tasakaalu ning punane jõudu, energiat ja kirge. Värvivaliku inspiratsioon tuli aerutamisest enesest - vesi kui aerutamiseks vajalik element, tasakaal kui aerutaja professionaalsuse näitaja ning spordile omased faktorid jõud ning energia.



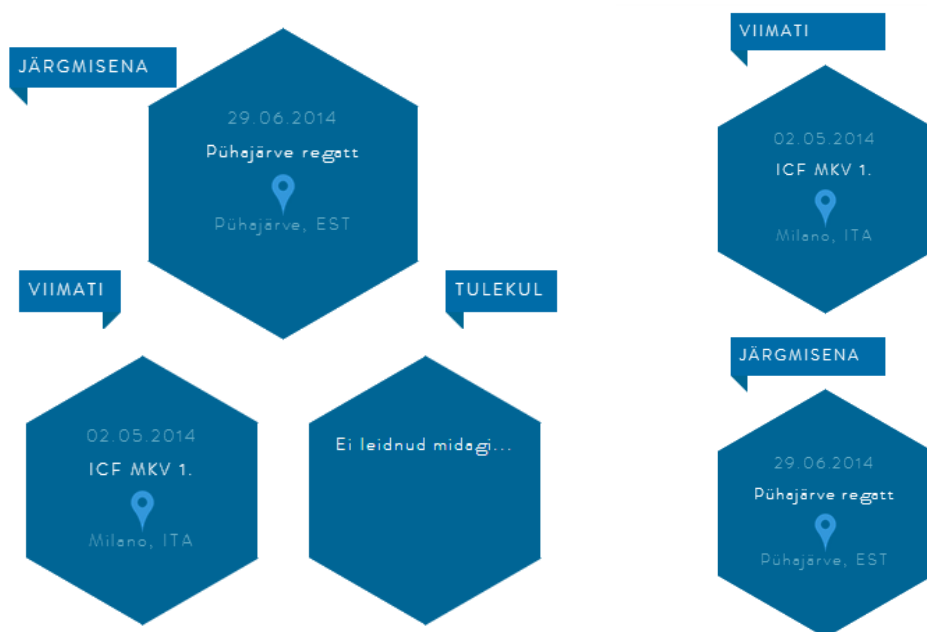
Joonis 6. EAF-i uue veebi põhilised disainielemendid

4.2.1.3 Mobiililahendus

Et avaliku veebi puhul oli oluline mobiilne kasutatavus, tuli kohe arvestada ka mobiililahenduse väljatöötamisega. Bootstrap'i põhi oli heaks baasiks, aga kohandatud Wordpressi teema juures oli elemente, mis vaikimisi kitsa ekraaniga ei kohaldunud (näiteks eelnevalt demonstreeritud kuusnurgad) ning tegid seetõttu kogu lehe katki. Samuti oli proleeme HTML5 video mängimisega täisekraanil.

Mobiilse lahenduse juures on oluline kogu lehe liidest pisut lihtsustada. Suurem osa veebist oli mobiililahenduse jaoks korras, aga uudised ning kalender vajasis kohendamist. Mobiilivaadete kujundamiseks on mõistlik kasutada @media (*media query*) päringuid CSS failides. Nii saab lehe laiuse muutumisel CSSi värskendada ning vajalikke elemente ümber kujundada. Mobiilse kasutamise oluliseks osaks on mobiilimenüü, mis peab alati ekraanile kogu ulatuses ära mahtuma ning suure tõenäosusega tavamenüüst pisut erinevalt käituma, sest mobiili puhul on valdavalt tegu puuetundliku seadmega.

Joonisel 7 on illustreeriv näide mobiililahenduse väljatöötamisest. Kõrvuti on võrdluses kalendri avakuva laiiale (vasakul) ning kitsale (paremal) ekraanile.



Joonis 7. Tava- ja mobiilivaate võrdlus

4.2.2 Wordpress

Wordpress on kuulus oma 5 minuti installatsiooni poolest. Tõepoolest, vajalikud komponendid on vaid töötav MySQL andmebaas ning Wordpress ise. Wordpress tuleb alla laadides juba valmis arhiivifailina, mis tuleb lahti pakkida ja serverisse kopeerida.

Kui wordpress on konfigureeritud (andmebaasiühenduse osas), on sisuliselt töötav sait juba olemas. Vaikimisi kasutab see aga Wordpressi enda kujundust. Avaliku veebi puhul olemasolevatest teemadest ei piisanud ning seetõttu tuli välja töötada EAF-i vajadustele vastav kujundus ja välimus. Sellest oli lähemalt juttu punktis 4.2.1. Loodud stiilifailide kasutamiseks tuleb Wordpressis luua uus teema. Teema jaoks on vajalik korrektne failistruktuur ning teema informatsiooni sisaldavad kommentaarid. Seejärel on teemat võimalik aktiveerida Wordpressi sisuhaldusliidese kaudu.

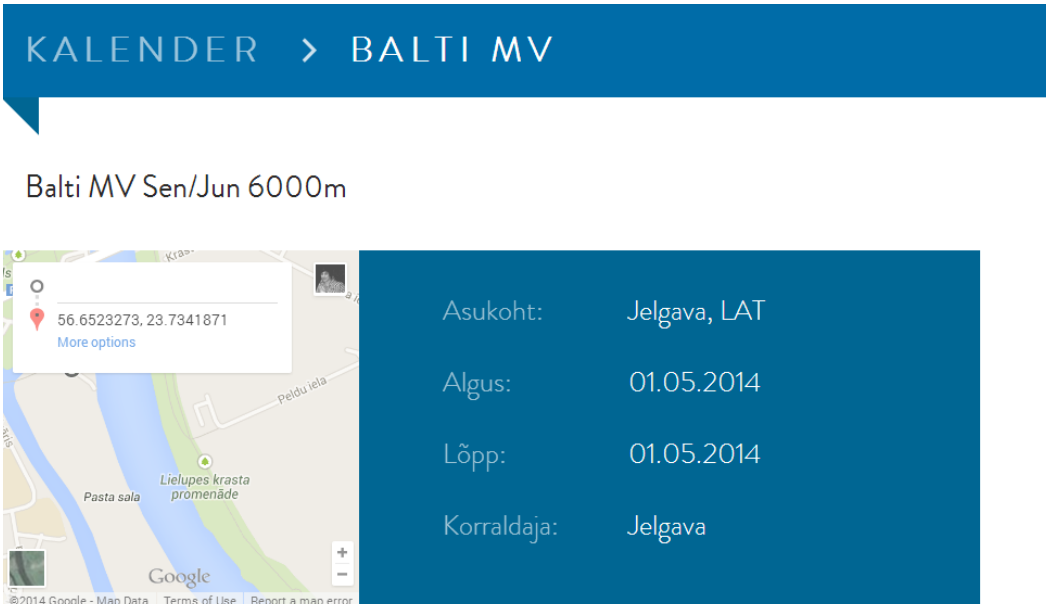
4.2.3 Kalendri plugin

Avaliku veebi üheks eesmärgiks oli sündmuste informatsiooni publitseerimine. Wordpressile on arendatud palju erinevaid kalendri-stiilis pluginaid, aga ükski neist ei vastanud päris hästi

EAF-i vajadustele. Mahult kippusid olemasolevad lahendused väga suured ja kohmakad olema. Seetõttu oli neid keeruline kohandada aerutamismõistluste ja muude sündmuste kuvamiseks.

EAF-i vajaduseks oli luua lihtne haldusliides, kus saaks sündmusi lisada, muuta ja kustutada. Sündmuse olulisteks parameetriteks olid nimi, algus- ja lõppkuupäev, asukoht tekstina ning asukohakaart. Avalikul poolel oli soov informatsiooni kuu ja sündmuse vaates kuvada. Wordpressi plugina loomiseks piisab korrektsest failistruktuurist ning mõnest kommentaarist plugina peamises PHP failis. Selleks, et plugin ilmuks administratiivpaneeli, on vaja kasutada Wordpressi *hook*'e või sidemeid.

Joonisel 8 on toodud sündmuse detailvaate ekraanitõmmis, mis illustreerib EAF-i nõudmisi kalendri pluginale.



The screenshot shows a calendar interface with a blue header bar containing the text 'KALENDER > BALTI MV'. Below the header, the event title 'Balti MV Sen/Jun 6000m' is displayed. The event details are presented in a table format on a dark blue background. To the left of the table is a Google Maps snippet showing the location 'Jelgava, LAT' with coordinates 56.6523273, 23.7341871. The table lists the following information:

Asukoht:	Jelgava, LAT
Algus:	01.05.2014
Lõpp:	01.05.2014
Korraldaja:	Jelgava

Joonis 8. Kalendri sündmuse vaade

5. Süsteemi testimine

Antud osas kirjeldatakse, kuidas testiti esmast prototüüpi ning millist tagasisidet selle abil koguti. Lisaks tuuakse välja, mida tagasiside tulemusena muudeti. Siseveeb läbis valminud kujul ka teise testimise, kus reaalsed lõppkasutajad üritasid neile olulisi toiminguid sooritada süsteemis, kus funktsionaalsus oli realiseeritud.

5.1 Esialgne prototüüp

Esialgse lahenduse käigus valmis avalik veeb, kus sai publitseerida uudiseid ning meediafaile. Siseveebi kohta valmis Bootstrapil põhinev prototüüp, kus sai põhitoominguid läbi mängida, aga päris funktsionaalsust ei olnud veel vaadetega ühendatud.

5.1.1 Lahenduse testimise keskkond

Avaliku veebi puhul oli testijate hulk suurem ning testimist pidi olema võimalik teha erinevates seadmetes erinevate kasutajate poolt. Seetõttu sai esialgne lahendus testserverisse üles laetud ning seal tööle pandud. Testimine toimus reaalselt lõppkasutajate endi seadmete kaudu (sportlaste ning huviliste arvutid ja mobiiltelefonid) ning erinevates võrkudes (kohalik traadiga ning traadita võrk ja mobiilne andmeside).

Siseveeb valmis esialgu UI prototüübina, kasutades ainult staatilist HTML-i, CSS-i ja JavaScripti. Joonisel 9 on toodud ekraanitõmmis esimesest prototüübist, vaateks sportlase lisamine. Siseveebi puhul on kasutajate hulk piiratud ning selle testimist sai teostada lokaalselt EAFi üldkoosolekul ning treenerite nõukogul. Testimiseks kasutati ühte testseadet ning testijateks olid treenerid ning EAF-iga seotud siseringi inimesed (korraldajad, peasekretär, juhtkond jt ametnikud).

The screenshot shows the 'EAF SISEVEEB' web application interface. At the top right, it says 'TREENER TREENERIPNIMI KLUBI: SAK TARTU LOGI VÄLJA'. Below the header is a navigation menu with 'Kodu', 'Lisa sportlane', 'Haldajad sportlasi', 'Kariikasan', 'Kariikaetapid', and 'Tulemused'. The 'Lisa sportlane' button is highlighted. Below the menu, the page title is 'LISA UUS SPORTLANE'. The form contains the following fields: 'Sportlase email' (text input), 'Eesnimi' (text input), 'Perenimi' (text input), 'Sünniaasta' (text input with '0' entered), and 'Sugu' (dropdown menu with 'Mees' selected). At the bottom of the form is a 'Lisa uus sportlane' button.

Joonis 9. Ekraanitõmmis siseveebi prototüübist

5.1.2 Tagasiside

Testimisel kogunes väga palju tagasisidet kasutusmugavuse ja disaini kohta. Mõned tähelepanekud tulid ka funktsionaalsuse kohta. Antud punktis tuuakse välja tagasiside põhjal koostatud kokkuvõtte süsteemi komponentide kaupa.

5.1.2.1 Avalik veeb

Avaliku veebi disaini puhul laekus erinevaid arvamusi avalehel oleva täisekraanvideo osas. Nooremale generatsioonile selline lahendus meeldis ning tundus teistest selgelt eristuv ja atraktiivne. Vanem generatsioon, sealhulgas enamus treenereid, pidas täisekraanil jooksvat videot liiga suureks ning soovis näha avalehel rohkem sisu. Neile tundus harjumuspärasemana nä. tüüpiline lehe struktuur, kus kogu menüü on kohe nähtav, asudes üldjuhul lehe ülaseras, ning külgedel on samuti tulpadena esitatud mingit sisu. Vaidluskohaks oli ka see, kas avalehe video peaks kohe lehe laadimisel mängima hakkama või alles kasutaja klikkamisel.

Uudiste sektsioonis tekitas kuusnurkade ja regulaarsete uudiste kooslus segadust – testkasutajate meelest tulnuks jääda ühe lahenduse juurde, olgu selleks siis kuusnurgad või ristkülikud. Samuti tundus suurelt uudiste väljakuvamise osas kolm uudist liiga vähe, sest pealehe põhiline mõte peakski uudiste kuvamine olema.

Kaks testkasutajat pidasid oluliseks vanas veebis olnud sünnipäevade sektsiooni. Uues veebis sellele esialgu kohta ei nähtud, sest otsest lisaväärtust analüüsi käigus see teave ei omanud.

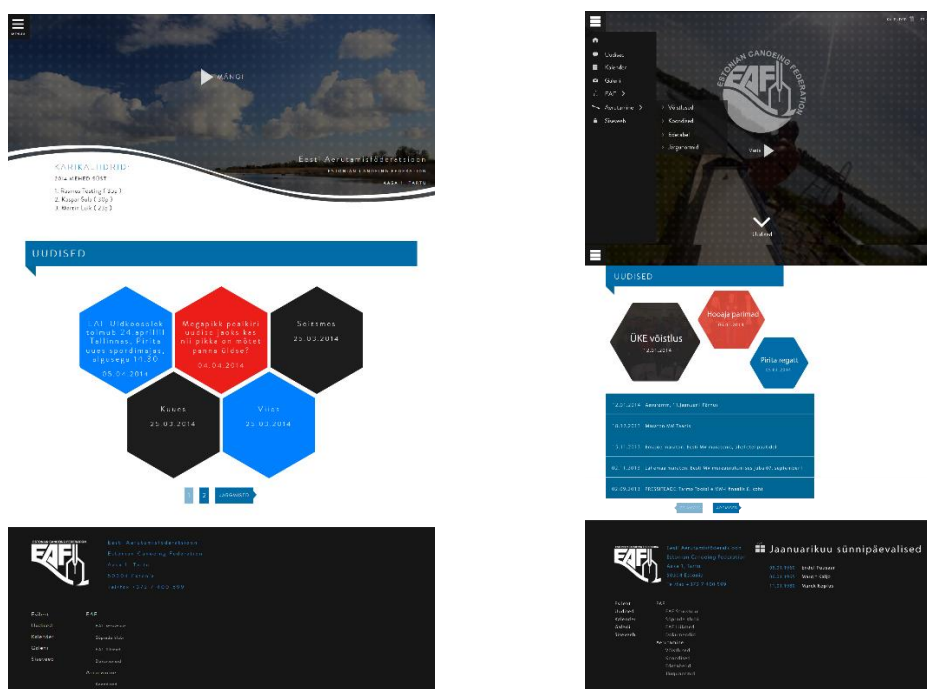
5.1.2.2 Siseveeb

Siseveebi prototüübid ei tekitanud nii suurt diskussiooni nagu avaliku veebi esialgne lahendus. Sellegipoolest tuli välja mõningaid probleeme, mida võiks parandada. Näiteks korraldaja vaates sooviti etappide haldusliidese nimekirjas näha, kas vastav etapp on registreerumisele avatud või mitte. Samuti tekitas segadust tulemuste vaade, kuhu oli kombineeritud kokku nii etapipõhine tulemuste vaatamine kui ka koondtulemuste vaatamine. Värvilahenduste ja disaini kohta negatiivset tagasisidet ei tulnud.

5.1.3 Tehtud muudatused

Kogutud tagasiside oli väga konstruktiivne ning võimaldas teha arvestatavaid muudatusi. Testkasutajad tõestasid, et arendaja või disainer ei näe kõiki valupunkte, mis teinekord on üsna lihtsad ning välditavad.

Avaliku veebi tagasiside puhul oli olulisimaks teemaks just avalehe disain. Täisekraanil jooksev video jäi küll alles, sest see oli eelistatud noorema generatsiooni (ehk peamise sihtgrupi) poolt. See-eest sai avalehele planeeritud karikaarvestuse koondtulemuste ala, mis muudab avakuva informatiivsemaks, ning tausta muudeti selliseks, et video ei kataks enam kogu ekraani ulatust. Video otsustati arutelu tulemusena siiski esialgu seisvana laadida ning alles kasutaja kliki peale mängima panna. Uudiste sektsioon läbis samuti uuenduskuuri ning laial ekraanil jäid alles vaid kuusnurgad. Kitsama ekraani puhul asenduvad kuusnurgad ristkülikutega. Joonisel 10 on toodud võrdlusena kõrvuti muudatuste tulemusena valminud uus avakuva (vasakul) ning esialgne kujundus (paremal).



Joonis 10. Avaliku veebi avakuva muudatused.

Sünnipäevade osas arvas EAF, et vastavaid nimekirju peaks esmalt uuendama, sest olemasolevas veebis on sellekohane informatsioon aegunud. Otsustamata jäi veel ka see, kas sünnipäevi kuvada avalikus veebis või peaks see informatsioon jääma siseveebi.

Siseveebi probleemidest esimene ehk registreerumise avatuse info korraldaja vaates oli lihtne lahendada ning vajas vaid lisaveergu etappide tabelisse. Rohkem tööd oli tulemuste sektsioonis, kus tuli lahku lüüa ühe etapi tulemused ning koondtulemused. Uuenenud liideses asuvad etapipõhised tulemused koondtulemustest eraldi lehel.

5.2 Valminud siseveeb

Realiseeritud funktsionaalsusega siseveeb läbis teistkordse testimise, kus osales 3 süsteemi reaalsel lõppkasutajat. Antud osas on toodud testimise käigus selgunud probleemsed kohad ning nende lahendused.

5.2.1 Tulemus

Testimise tulemusena tulid välja veel mõningad olulised puudujäägid või probleemsed kohad, mida saaks paremini teostada. Esiteks leiti, et prototüübi testimise käigus tehtud muudatus etappide haldusvaates oli mõistlik, aga lisaks infole, kas etapi registreerumine on avatud, peaks vastav tulp olema ka funktsionaalne – registreerumist võiks olla avatabelis otse võimalik muuta. Teiseks oluliseks tähelepanekuks oli, et etapi lisamisel võiks eduka lisamise tulemusena suunata brauseri otse etappide haldusvaatesse, mitte uue etapi lisamise vormi juurde. Seevastu sportlaste lisamisel oli sarnane lahenduskäik väga tervitatav, sest sportlasi lisatakse tihti mitu tükki korraga. Testimise käigus tõdeti ka seda, et tühjade tabelite puhul peaks päise asemel näitama hoopis informatsiooni selle kohta, et ühtegi tabeli kannet andmebaasist ei leitud. Väga lihtsaks, aga samas oluliseks veaks ilmnas see, et vormide täitmisel puudusid viited, millised väljad on kohustuslikud ja millised mitte. Viimaseks testimise käigus ilmnenu probleemiks oli see, et kinnitusaken võiks lisaks küsimusele „Kas oled kindel, et soovid teha tegevust X?“ kuvada ka objekti infot, kelle või mille kohta kinnitust küsitakse. Näiteks sportlase kustutamise korral võiks kinnitusaknas vastava sportlase andmed nähtaval olla.

5.2.2 Lahendused

Testimise tagasisides toodud vead said seekord kompromissitult lahendatud. Etappide haldusvaatesse ilmus võimalus registreerumist avada või sulgeda ning etapi lisamisel suunatakse kasutaja nüüdsest etappide haldusvaatesse. Tühjade tabelite või vormiväljade puhul asendus tabel informatiivse tekstiga, mis annab teada, et vasteid ei leitud. Nõutud vormiväljad said vastavalt tähistatud ning kinnitusaknad muutusid muudatuste tulemusena informatiivsemaks.

5.2.3 Kokkuvõte

Antud testimise etapp tõestas, et esialgse UI prototüübiga pole võimalik kogu süsteemi efektiivselt testida. Kasutajatel tekib küll üldine nägemus disainist ja kasutajaliidesest, aga funktsionaalsuse lisandumisel ilmneb veel palju aspekte, mida varasemalt ei osatud analüüsida.

6. Kokkuvõte

Käesoleva töö põhieesmärgiks oli infosüsteemi loomine Eesti Aerutamisföderatsioonile. Vastav infosüsteem pidi koosnema kahest osast - siseveeb ning avalik veeb. Siseveebi peamiseks eesmärgiks oli karikaarvestuse digitaliseerimine. Avaliku veebi puhul oli peamisteks vajalikeks funktsionaalsusteks uudiste, meedia ja sündmuste publitseerimise võimalus. Samuti oli avaliku veebi puhul oluline kohalduv ja atraktiivne disain ning suurepärase kasutatavus.

Töö tulemusena valmis infosüsteem, mis vastab kõikidele põhieesmärkidele. Valmis siseveeb, mis võimaldab treeneritel sportlasi digitaalselt võistleva panna, vältides seni paberil toimunud registreerumisi. Korraldaja rolliga kasutaja saab luua karikaetappe ning etapi möödudes digitaalselt sportlastele tulemusi määrata. Hooaja lõikes arvutab süsteem automaatselt koondtulemused, mis aitab vältida inimlikke eksimusi ning lubab kogu hooaja vältel jälgida, kellel kuidas läheb. Antud töö tulemusena valmis ka avalik veeb, mis on Wordpressil baseeruv atraktiivne veebiportaal. Avalik veeb on kohalduva disainiga, see tähendab, et veebi on võimalik kasutada nii mobiilsetel seadmetel kui ka suurel ekraanil. Avaliku veebi sisuhaldusliides võimaldab tänu Wordpressi baasfunktsionaalsustele ning lisatud pluginatele publitseerida meediat, uudiseid ning sündmusi – just nagu eesmärgiks seati.

Töö käigus või lõppedes võis tõdeda, et sellise infosüsteemi olemasolu on ühele spordialaliidule väga vajalik. Kinnituseks võib pidada fakti, et Sõudeliiduga suhtlemisel tunti huvi sarnase lahenduse vastu ning pakuti reaalselt võimalust süsteemi loomisega alustada. Aerutajate seas tekitas vastava süsteemi valmimine ning testimine palju kõneainet ning seetõttu kogunes testimise käigus palju konstruktiivset tagasisidet. Testimise järelendusena võib kinnitada, et süsteemi katsetamine lõppkasutaja peal on äärmiselt oluline. Kuitahes ettenägelik või professionaalne arendaja või disainer üritab olla, ei suuda isegi valdkonnaga seotud inimene alati kõiki vigu või võimalikke paranduskohti märgata. Eriti kehtib see väide valdkonnaspetsiifiliste tegevuste juures. Eesti Aerutamisföderatsioon oli antud süsteemiga kokkuvõttes väga rahul ning tõi peamise edusammuna välja just paberimajanduse vähenemise seoses karikaarvestuse digitaliseerimisega.

Kuigi võib väita, et kõik peamised eesmärgid said antud töös täidetud, saab sellegipoolest välja tuua mõningaid edasiarenduse võimalusi, mis muudaksid süsteemi veelgi kasulikumaks Eesti Aerutamisföderatsioonile. Esimeseks võimalikuks laienduseks võiks olla sportlase roll siseveebis. See tähendab, et sportlased saaksid samuti sisse logida ning enda ja konkurentide

tulemusi jälgida. Teiseks võimalikuks lisaarenduseks siseveebis on võistluste definitsiooni laiendamine. Hetkel on siseveeb suunatud just karikaetappidele, aga süsteem võiks võimaldada ka teistele võistlustele registreerumist ning tulemuste jälgimist. Korraldajal võiks olla võimalus väljastada siseveebi kaudu kõikide registreerunute põhjal stardiprotokoll, mis oleks digitaalsel kujul nähtav sportlastele ja treeneritele juba enne võistluspäeva.

Summary

The main purpose of this thesis was to create an information system for Estonian Canoeing Federation. The system had to contain two parts: internal web and public web. The Internal web's main goal was to digitalize the Estonian cup. The public web's main objective was to enable publishing media, news and events. In addition, it was important that the public web would be attractive, responsive and with great usability.

As a result, an information system was created that met all the main targets. One part of the system, the Java Spring based internal web, created the ability to enter sportsmen digitally to competitions, avoiding the paper registrations in the future. The organizer role can create new cup events and insert results afterwards. The internal web system calculates automatically all the aggregate results per one season, avoiding human errors and confusion in the award ceremony. Automatic result calculation also gives the ability to follow the results during the competing season. The other part of the information system, the public web, was developed on Wordpress platform. It features responsive design, which means it is usable on different kinds of devices, including mobile devices. The administrative interface of the public web allows to publish media, news and events thanks to Wordpress base functionalities and additional plugins – just like it was planned.

One could recognize during and after the development process that this kind of information system is extremely necessary for sports associations. As a confirmation, the Estonian Rowing Association showed deep interest in the system that was developed while the differences between their current system and canoers' existing website were discussed. The development of this new system sparked a lot of discussion amongst the canoers which produced great amounts of constructive feedback. As a result of the testing process, one could arrive at a conclusion that testing on the end-user is really important. No matter how provident or professional the developer or designer tries to be, even the person who is deeply related to the field can not always notice all the mistakes. Estonian Canoeing Federation was very pleased with the result and pointed out the reduction of paperwork as the major success in relation to the digitalization of the Estonian cup.

Although one can conclude that the main targets have been reached, it is still possible to bring out some potentialities for further development. The first possible expansion could be the role of sportsman in the internal web. It means that the sportsman can log into the system and get

an overview of his/her results and participations. Secondly, the definition of an event or competition can be expanded. Currently the internal web is targeted at cup events, but this system could also be used for all kinds of competitions. Lastly, the organizer should have the opportunity to print out the start list of the event. The same list could be digitally viewed by coaches and sportsmen before the competition.

Kasutatud kirjandus

- [1] „Maven,“ [Võrgumaterjal]. Available: <http://maven.apache.org/>. [Kasutatud 25 05 2014].
- [2] Srirangan, Apache Maven 3 Cookbook, Packt Publishing, 2011.
- [3] „Spring documentation,“ [Võrgumaterjal]. Available: <http://spring.io/docs>. [Kasutatud 25 05 2014].
- [4] C. Walls, Spring in Action, 3rd Edition, Manning, 2011.
- [5] C. Scarioni, Pro Spring Security, Apress, 2013.
- [6] „PostgreSQL,“ [Võrgumaterjal]. Available: <http://www.postgresql.org/>. [Kasutatud 25 05 2014].
- [7] P. McNeil, Mobile Web Designer's Idea Book, HOW Books, 2013.
- [8] „DB Engines,“ [Võrgumaterjal]. Available: <http://db-engines.com/en/ranking>. [Kasutatud 25 05 2014].
- [9] „MySQL,“ [Võrgumaterjal]. Available: <http://www.mysql.com/why-mysql/>. [Kasutatud 25 05 2014].