

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Mario Tosso 112647IAPB

EESTI AIRSOFTI KOMMUUNI INFOSÜSTEEM

Bakalaureusetöö

Juhendaja: Jaagup Irve

Tehnikateaduste
magister

Tallinn 2017

Autorideklaratsioon

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

Autor: Mario Tosso

15.05.2017

Annotatsioon

Käesoleva töö eesmärgiks on luua infosüsteem Eestis tegutsevatele Airsofti harrastajatele. Infosüsteem ei ole ametlikult seotud ühegi organisatsiooniga ning autentimine on avatud kõigile. Süsteemi peamine eesmärk on pakkuda mugavat võimalust Eestis Airsoftiga tegelevatele inimestele sündmuste korraldamiseks ja valdkonnas osalevate inimeste haldamiseks. Ürituste loomine, vaatamine, kommenteerimine ning mängijate haldus peab toimuma digitaalselt ning sõltub süsteemi kasutajate reitingust.

Suurima osa töö keerukusest moodustab täpse funktsionaalsuse vajaduse väljaselgitamine, mille tarvis suheldakse valdkonna aktivistidega ning lähtutakse personaalsest kogemusest. Ürituste loomine peab muutuma senisest efektiivsemaks ning üritustele registreerimine ning registreerinute haldamine kiiremaks. Lisaks on oluline registreeritud kasutajate haldamine ning kasutajate reitingusüsteem, mida haldavad kasutajad ise.

Töö tulemusena valmib Airsofti ürituste ning mängijate halduseks kasutatav infosüsteem, mis läbib reaalse kasutajate käes testiperioodi. Kasutajatel on võimalik anda süsteemi kasutatavuse ja funktsionaalsuse kohta tagasisidet, mille põhjal viiakse sisse muudatused vastavalt individuaalse tagasiside keerukusele ning otstarbekusele.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 31 leheküljel, 6 peatükki, 15 joonist.

Abstract

Information System for Estonian Airsoft Community

The purpose of this thesis is to create an information system for Estonian Airsoft enthusiasts. The information system is not officially connected to any organisation and authentication is open for all. The main purpose of the system is to offer a comfortable way for Estonian Airsoft players to arrange Airsoft events and manage the community's players. Creating, viewing, commentating the events and managing the players has to be enabled digitally and depends of the user's rating.

The main difficulty of the system is finding the exact needs functionality-wise. For this, the author will communicate with activists from the field's playerbase and guides from personal experience. The creation of events has to become more efficient and registering to events and managing the registered players has to be made quicker. Additionally, it is important to manage users who have registered to the information system and the user rating system, which will be handled by the users themselves.

The result of all the work will be an information system, which can be used to create Airsoft events and managing the playerbase. The information system will go through a testing phase in the hands of real-life users who will have the possibility to leave feedback regarding the implemented usability and functionality of the system. Based on that feedback, additional changes will be made, depending on the difficulty and practicality of the suggestions.

The thesis is in Estonian and contains 31 pages of text, 6 chapters, 15 figures.

Lühendite ja mõistete sõnastik

| | |
|------------|---|
| Rails, RoR | <i>Ruby on Rails.</i> Dünaamilisele Ruby keelele ehitatud veebirakenduste loomise raamistik. |
| MVC | <i>Model-View-Controller</i> Mudel-Vaade-Kontroller arhitektuurimuster. |
| JSON | <i>JavaScript Object Notation</i> Inimloetav andmevahetusformaat |
| Airsoft | Militaar-sportlik taktikamäng, milles kasutatakse õhkrelvadest tulistatavaid kuule vastase alistamiseks. Tavaliselt korraldatakse välitingimustes militaarobjektidel. |
| Payload | <i>Front-endist back-endi</i> saadetav parameetrite kogum, kus võib sisalduda süsteemi kasutaja sisestatud info. |
| URL | <i>Uniform Resource Locator.</i> Veebiaadress ehk internetiressursi unikaalne identifitseerija. |
| HTTP | <i>HyperText Transfer Protocol.</i> Interneti alustala ja enim kasutatud andmevahetusprotokoll. |
| CSS | <i>Cascading Style Sheets.</i> HTML elementide stiili ja väljanägemise kirjeldamise keel. |
| PaaS | <i>Platform-as-a-service</i> Tavaliselt pilveteenusena pakutav tarkvaraplatvorm, mida saab kasutada tarkvaraprojektide arenduses ja avalikustamises. |

| | |
|-----------|--|
| Back-end | Tarkvaraprojekti serveri ning taustarvutuste pool. See süsteemi osa, mis pakub <i>front-endile</i> teenust. |
| Front-end | Tarkvaraprojekti kuvapool. Sisaldab kasutajale kuvatavaid vaateid ning suhtleb <i>back-endiga</i> API kaudu. Seob süsteemi kasutajat ning <i>back-endi</i> . |
| API | <i>Application Programming Interface</i> Kogum reegleid ja funktsioone mingi tarkvarasüsteemiga suhtlemiseks. |

Sisukord

| | |
|--|----|
| 1 Sissejuhatus | 10 |
| 2 Infosüsteemi ülevaade | 11 |
| 2.1 Infosüsteemi eesmärgid | 11 |
| 2.2 Arenduses kasutatav meetoodika | 12 |
| 2.3 Arenduses kasutatavad tehnoloogiad | 12 |
| 2.3.1 Ruby on Rails 5.0 ja MVC | 12 |
| 2.3.2 Devise | 13 |
| 2.3.3 Paperclip | 13 |
| 2.3.4 PostgreSQL 9.3 | 14 |
| 2.3.5 jQuery | 14 |
| 2.3.6 Bootstrap | 14 |
| 2.3.7 HTML5 | 14 |
| 2.3.8 eRuby | 15 |
| 2.3.9 Puma | 15 |
| 2.3.10 will_paginate | 16 |
| 2.3.11 has_scope | 16 |
| 3 Infosüsteemi taust | 17 |
| 3.1 Valdkond | 17 |
| 3.2 Sihtgrupp | 17 |
| 3.3 Olemasolevad lahendused | 18 |
| 4 Infosüsteemi arendusprotsess | 19 |
| 4.1 Andmebaas | 19 |
| 4.2 Tagarakendus | 20 |
| 4.2.1 Marsruutimine | 21 |
| 4.2.2 Kontrollerid | 21 |
| 4.2.3 Mudelid | 22 |
| 4.3 Kasutajaliides | 23 |
| 4.3.1 Disan ja kasutusmugavus | 23 |
| 5 Infosüsteemi testimine | 25 |

| | |
|-----------------------------------|----|
| 5.1 Tagaside..... | 25 |
| 5.1.1 Testimise keskkond | 25 |
| 5.2 Muudatused tagasisidest | 26 |
| 5.3 Tulevad arendused | 27 |
| 5.4 Kokkuvõte | 27 |
| 6 Kokkuvõte | 29 |
| Kasutatud kirjandus | 31 |

Jooniste loetelu

| | |
|---|----|
| Joonis 1. MVC Muster | 13 |
| Joonis 2. Puma veebiserveri võrdlus alternatiivsete veebiserveritega..... | 16 |
| Joonis 3. Andmebaasi struktuur | 19 |
| Joonis 4. Kasutaja e-posti aadressi unikaalsuspiirang | 20 |
| Joonis 5. Kasutaja ja ürituse vahelise seose loomine | 20 |
| Joonis 6. Marsruudid | 21 |
| Joonis 7. Ürituste kontrolleri juurdepääsupiirang | 22 |
| Joonis 8. Ürituse loomisel kasutajaliidestest saadetakse lubatud parameetrite loetelu.. | 22 |
| Joonis 9. Ürituse otsimine nime järgi | 22 |
| Joonis 10. Hetkel autenditud kasutaja kõik üritused | 23 |
| Joonis 11. Autenditud administraatorkasutaja avavaade | 24 |
| Joonis 12. Autenditud administraatorkasutaja avavaade kitsal ekraanil | 24 |
| Joonis 13. Uus reitinguvaade | 26 |
| Joonis 14. Reitingu valimine | 26 |
| Joonis 15. Pildifaili lisamise validaatorid | 27 |

1 Sissejuhatus

Pea igas valdkonnas, kus on ühe tegevuse taha kogunenud rohkem kui üks inimene, on oluline, et liikmed saaksid asjadest aru sama moodi. Ilma digitaalse süsteemita on see tänapäeval väga keeruline. Samuti pole selline lahendus skaleeruv ega otstarbekas, kui kogukonnas aktiivsete liikmete arvu soovitakse suurendada. Antud töö keskendub Eestis tegutsevate Airsofti harrastajate kommuunile, võttes arvesse selle valdkonna eripära. Töös kirjeldatakse valdkonna funktsionaalsed vajadused ning luuakse vastavalt nendele infosüsteem.

Eestis tegutsevatel Airsofti harrastajatel ei ole kasutusel ühtegi automatiseeritud süsteemi mängijate ning sündmuste haldamiseks. Siiani on kasutatud tavalisi internetifoorumeid, kus on kogu informatsioon alamfoorumite ning teemade kaupa killustatud ja kohati duplikeeritud. Väga palju on ka sellist infot, mida kuskile kirja ei panda ja vahendatakse ainult näost näkku suusõnaliselt. Sündmuste loomine ning nende registreerimine on võimalik vähemalt osaliselt automatiseerida, et muuta kogu protsess alates sündmuse ideest kuni selle täideviimiseni efektiivsemaks.

Kõik infosüsteemis võimalikud tegevused peaksid olema võimaldatud ainult autenditud ja piisava reitinguga kasutajatele. Mängijate maine on valdkonnas väga tähtis, kuid seni pole olnud seda võimalik kuskil kontrollida. Igal süsteemi kasutajal on oma reiting, mille põhjal on võimalik ürituste loojatel nende ürituste osavõtu kohta omad korrektuurid teha. Reitingut jagavad süsteemi kasutajad üksteisele ise ja on ka hallatav administraatori poolt.

2 Infosüsteemi ülevaade

Selles peatükis luuakse ülevaade loodava infosüsteemi funktsionaalsusest ja tuvustatakse arenduses kasutatavaid tehnoloogiaid.

2.1 Infosüsteemi eesmärgid

Käesolev töö ei ole loodud koostöös ühegi Eesti tegutseva Airsofti klubiga, kuid valdkonnas tegutsevate aktiivsete mängijatega suhtluse tulemusena on kirjeldatud infosüsteemi peamised eesmärgid.

- Infosüsteem peab võimaldama kasutajaks registreerimist.
- Infosüsteem peab võimaldama Airsofti spetsiifikat arvestavate sündmuste loomist ja kommenteerimist.
- Infosüsteemi kasutajatel peab olema võimalus lisada endale personaalne mängijanumber.
- Infosüsteem peab võimaldama loodud sündmustele registreerimist.
- Infosüsteem peab võimaldama sündmusele registreerinud kasutajate haldamist.
- Infosüsteem peab võimaldama registreeritud kasutajate haldamist.
- Infosüsteem peab võimaldama registreeritud kasutajatele reitingu andmist teiste registreeritud kasutajate poolt.
- Kogu infosüsteemi funktsionaalsus peab olema võimaldatud ainult autenditud kasutajatele.
- Piisava reitingu puudumisel peab olema süsteemi kasutajale võimaldatud tegevused piiratud.

2.2 Arenduses kasutatav metoodika

Kuna tegemist on bakalaureusetöö raames loodava süsteemiga, mis tähendab, et arendajaks on üks inimene, selle töö autor, siis kasutatav metoodika erines mõnevõrra traditsioonilistest tarkvaraarenduse metoodikatest. Enne arenduse programmikoodi kirjutamise algust tehti selgeks tervet süsteemi puudutav funktsionaalsus, mis on iseloomulik kose mudelile, kuid arenduse käigus täpsustati detaile jooksvalt, mis on iseloomulik mõnele agiilsele mudelile.

Arendus algas staatiliste kasutajaliidese lehtede valmistamisega, sest sihtgrupil polnud spetsiaalset ettekujutust loodavast süsteemist ning arendaja ei suuda kõiki detaile analüüsidokumendis kirja panna. Kasutajaliidese prototüübi valmistamisest oli kasu töö mahu vähendamisel. Ilma prototüübita oleks olnud tagasisides muudatusettepanekuid rohkem.

2.3 Arenduses kasutatavad tehnoloogiad

Käesolevas punktis kirjeldatakse süsteemi arenduses kasutatavaid tehnoloogiaid. Iga tehnoloogia kohta tuuakse välja konkreetne süsteemi osa, kus seda kasutati.

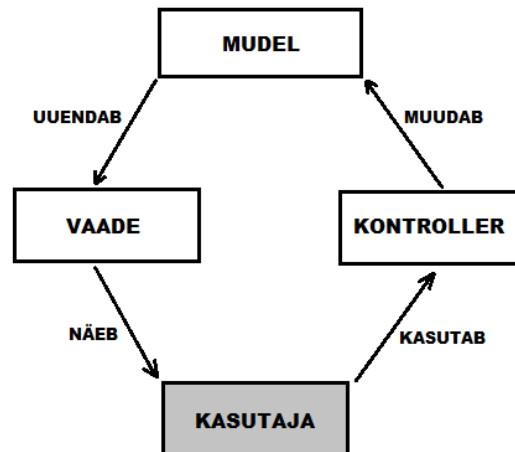
Arendustööriistade ja tehnoloogiate valik tehti üksnes töö autori poolt. Valituks osutus Ruby on Rails oma suhteliselt lihtsa *out-of-the-box* töökindluse tõttu võrreldes konkureerivate tehnoloogiatega. Lisaks oli süsteemi arenduse ajal autori karjääri töö seotud sama raamistikuga, mis lihtsustas mõlema projekti arendust tänu lisandunud kogemustele ja oskustele.

2.3.1 Ruby on Rails 5.0 ja MVC

Ruby on Rails, mõnikord ka lihtsalt Rails, on Ruby keeles kirjutatud laialt levinud veebirakenduste loomise raamistik. See sisaldab endas kõike, mida on vaja ühe andmebaasiga veebirakenduse loomiseks MVC (Model-View-Controller, Mudel-Vaade-Kontroller) mustri põhjal. [1] MVC muster Railsi rakenduse selgroog, mis jagab programmeerimise loogika kolmeks peamiseks kihiks. Mudeli osas on rakenduse äri loogika. Seal hoitakse ning arvutatakse infosüsteemi objektide omadusi. Kontrolleri vastu suunatakse algsed kasutaja päringud ja sealt edasi suuntakse need mudelile ning kõige lõpus kuvatakse HTML või JSON kujul vaate osas arvutuste tulemused. Ruby on Railsis on mudel hallatud komponendi Active Record poolt, mis on andmebaasisuhtluse

raamistik. Kontroller on hallatud Action Controlleri ning vaade Action View komponendi poolt. [2]

Ruby on Rails koos ning MVC muster on kogu rakenduse põhiline tehnoloogiline komponent. Joonis 1 kujutab MVC mustrit.



Joonis 1. MVC Muster

2.3.2 Devise

Devise on üks enim levinud Ruby on Railsi rakendustes kasutatavaid autentimise mooduseid. [3] Nagu paljudes muudes veebirakenduste raamistikes, on ka Ruby on Railsile loodud palju kolmanda osapoole lisasid. Railsi puhul on need lisad tavaliselt gem-i kujul. Devise on kergelt kasutatav ja konfigureeritav autentimise gem, mis koosneb kümnest erinevast moodulist, mida tüüpiliselt veebirakenduse autentimise protsessis kasutatakse. Modulaarsuse põhimõttel loodud Devise lubab kasutajal valida komponentide seast just endale tähtsad. Devise lubab ka integratsiooni levinud sotsiaalvõrgustike kontodega. [4] Antud komponenti on kasutatud süsteemi registreerimise ja autentimise protsessi loomiseks.

2.3.3 Paperclip

Paperclip on Ruby on Railsile loodud pildifailide üleslaadimise teek. Selle põhiline idee on, et arendaja saaks pildifaili mudelile lisamisel kasutada seda nagu teisi kontrollerile saadetavaid parameetreid. Paperclip võimaldab defineerida lisatava pildi jaoks faili

mahu ning nullväärtuste piiranguid. Lisaks on võimalik sellega muuta üleslaetud pildi suurust, et süsteemi vaadetes saaks kasutada originaalpildi pisipilte. [5]

2.3.4 PostgreSQL 9.3

PostgreSQL, mõnikord ka „Postgres“, on võimekas avatud lähtekoodiga objekt-relatsiooniline andmebaasisüsteem. [6] PostgreSQL on loodud kõikidele peamistele operatsioonisüsteemidele ja on oma võimekuse ning usaldusväärsuse tõttu üks kõige suurema kasutajaskonnaga andmebaasisüsteeme maailmas. [7]

Rakenduses kasutatakse PostgreSQL-i süsteemi sisestatud andmete ja süsteemi vaadetes kuvatava informatsiooni salvestamiseks.

2.3.5 jQuery

jQuery on üks populaarsemaid Javascripti teeke, mida hetkel veebiarenduses kasutatakse. jQuery pakub olulist lihtsustust tüüpilisemate Javascripti toimingute jaoks, näiteks HTMLi elementide otsimine ja uuendamine ning Ajax päringute loomine. Rakenduses on jQuery kasutusel just tüüpiliste Javascripti toimingute lihtsustamiseks ja koodi puhtuse hoidmiseks.

2.3.6 Bootstrap

Bootstrap on veebirakenduste valmistamiseks kasutatav HTML, CSS ja Javascript raamistik. Bootstrap sisaldab mitut faili valmis CSS ja Javascripti koodi, mida raamistiku kasutaja kerge vaevaga kasutada saab. [8] 2016. aastal oli Bootstrap maailmas kõige levinum kasutajaliideste valmistamise raamistik ja on siiani suure võimekuse ja sisuka dokumentatsiooni tõttu väga laialt kasutatav. [9] Bootstrap sisaldab endas erinevaid CSSi komponente, mida kasutatakse, et rakenduse kasutajaliides kohandada erinevate ekraanisuurustega. Lisaks on rakenduses kasutatud Bootstrapi komponente erinevates kohtades, näiteks autentimise vormi modaalakna loomisel ning peamenüü lõõtsakomponendi loomisel.

2.3.7 HTML5

HTML5 on ülemaailmse veebi tuumikkeelega HTML viies põhiredaktsioon. HTML5 pakub arvukalt uusi elemente ("märgendeid") ja atribuute, mis iseloomustavad tänapäeva veebikülgede tüüpilist kasutust. Mõni uus on olemuselt sarnane <div> ja märgenditele, kuid mõni nagu näiteks <nav> (veebikülje navigeerimisala) ja

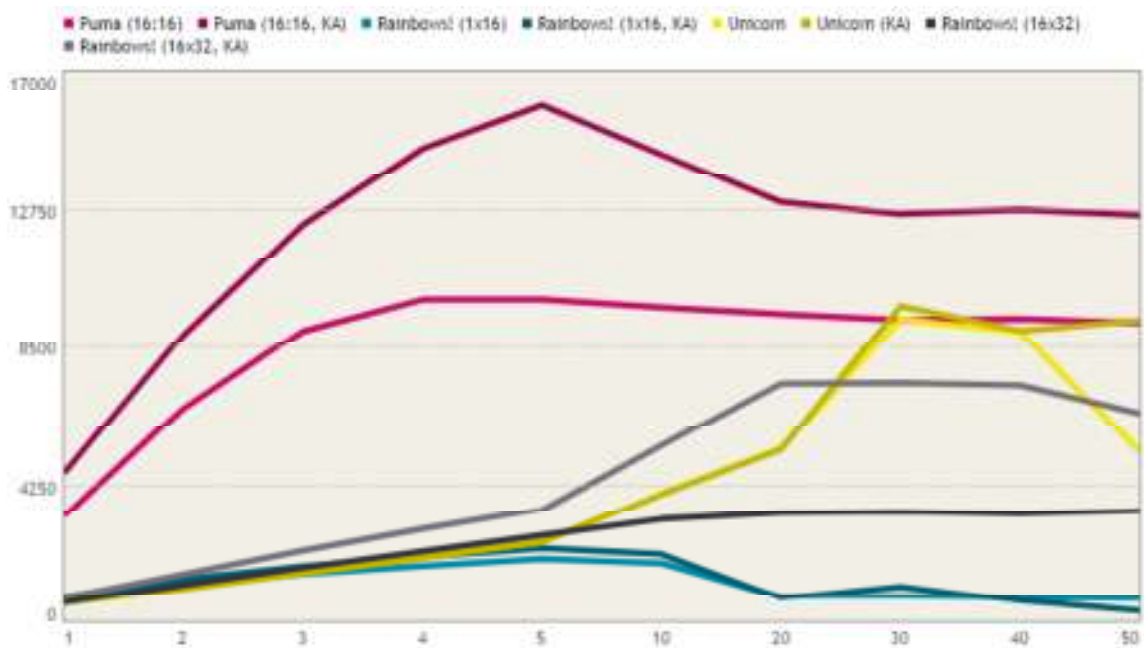
`<footer>` (jalus) omavad paigutamise tähendust veebilehel. Selliste märgendite eesmärk on hõlbustada otsimootori indekseerimist ja veebilehtede näitamist pisikeste ekraanidega seadmete poolt. Muud elemendid lisavad standardiseeritud liidese abil uusi võimalusi, nagu näiteks `<audio>` ja `<video>` elemendid. Välja on jäetud näiteks puhtalt esitluseks kasutatavad elemendid `<center>` ja ``, mille otstarvet saab kaskaadlaadistikuga saavutada. [10] Rakenduses kasutatakse HTML5 komponente kasutajaliidese loomisel.

2.3.8 eRuby

eRuby ehk Embedded Ruby on tehnoloogia, mis võimaldab lisada veebilehtede HTML märgendite vahele Ruby keeles kirjutatud koodi. Selle eesmärk on luua dünaamilisi vaateid ja kontrolleri mudeli abil loodud instantsmuutujate väärtusi rakenduse kasutajale kuvada. Lisaks on võimalik `<% ... %>` märgendite vahel teha Ruby keeles operatsioone, kui see peaks vajalikuks osutama. Hea tava kohaselt seda ilma mõjuva põhjuseta ei tehta. Vaate kihis peaksid olema kõik operatsioonid juba mudeli ja kontrolleri tasemel tehtud. Rakenduses kasutatakse eRuby tehnoloogiat dünaamiliste vaadete loomisel igal rakenduses kuvataval lehel.

2.3.9 Puma

Puma on Ruby lihtne, kiire ja lõimedega HTTP 1.1 veebiserver Ruby rakendustele. Puma on mõeldud kasutamiseks nii arendus- kui tootekeskondades. [11] Puma veebiserverit kasutati infosüsteemi arenduse faasis lokaalse serverina ning testimise faasis Heroku platvormil tootekeskonna veebiserverina.



Joonis 2. Puma veebiserveri võrdlus alternatiivsete veebiserveritega

Joonisel 2 [11] on ära toodud Puma serveri kiiruse võrdlus alternatiivsete Ruby serveritega. Horisontaalteljel on samaaegsete päringute arv ning vertikaalteljel päringute arv sekundis. Lisaks on Puma hetkel turul olevatest Ruby serveritest kõige väiksema mälu kasutusega [11], mis on platvormteenuseid kasutades oluline, sest seal esitatakse kasutajale arve kasutatud arvustuaaja põhjal.

2.3.10 will_paginate

will_paginate on Ruby on Railsi kolmanda osapoole teek, mida kasutatakse leheküljestatud vaadete loomiseks. Loodud süsteemis kasutatakse teeki ürituste vaadetes lehekülgede ning lehekülje navigatsioonivormi loomiseks.

2.3.11 has_scope

has_scope on Ruby on Railsi kolmanda osapoole teek, mida kasutatakse vaadetes loodud otsinguvormide teenindamiseks. Ruby klassi tasandil on võimalik defineerida klassiväljade põhised otsingureeglid, mida kontrolleri instantsimuutujate loomisel kasutatakse. Loodud süsteemis kasutatakse teeki ürituste ning mängijate otsinguvormide teenindamiseks.

3 Infosüsteemi taust

Selles peatükis kirjeldatakse loodava süsteemi valdkonda ning potentsiaalseid kasutajaid. Lisaks luuakse ülevaade hetkel olemasolevatest sama eesmärgiga lahendustest.

3.1 Valdkond

Airsoft on militaar-sportlik taktikamäng, mida korraldatakse tavaliselt välisoludes - mahajäetud linnakutes või maastikul. Infosüsteemi loomisel võeti arvesse valdkonna spetsiifikat ja sihtgruppi. Tüüpilise süsteemi kasutaja jaoks on eelkõige oluline kiirus ja kasutusmugavus. Visuaalne disain on teisejärguline ning pigem kasutusmugavust toetava funktsiooniga.

Eestis tegutsevad Airsofti klubid, aga suurem osa mängijatest ei kuulu ühtegi organisatsiooni. Eksisteerivad erinevad internetifoorumid, mille kaudu mängijad ürituste informatsiooni jagavad, aga tsentraalset süsteemi, millega liitudes mängija reaalselt kasu saaks, ei ole. Keske infosüsteemi puudumine mõjub kahjustavalt niigi killustatud valdkonna elujõulisusele. Lisaks on tegemist kallist ning spetsiaalset varustust nõudva alaga, mistõttu uute mängijate värbamine on väga keeruline.

Valdkonna aktivistide soov on foorumipõhiste infosüsteemide sulgemine, sest mängijate vaheline suhtlus kolib üha enam sotsiaalvõrgustikesse, kus on juba praegu olemas mängijaskonna liikmete vestlusgruppid. Foorumite kasutamine ainult ürituste organiseerimiseks ei ole aga piisavalt efektiivne ega tulus. Samas ei ole võimalik üritusi organiseerida ka ainult suhtlusvõrgustikes.

3.2 Sihtgrupp

Loodava süsteemi sihtgrupp on kõik Airsofti üritustest huvitatud inimesed. Valdkonnas tegutseb kümnekond aktivisti, kelle kohustuseks muutuks kasutajaskonna administreerimine. Kuigi ürituste loomine on piisava reitingu olemasolul võimaldatud kõikidele kasutajatele, näitab praktika, et sellega tegelevad üljuhul ainult need samad aktivistid. Selle üheks põhjuseks on ka väike mängijaskond.

Üritustele registreerimine ja nendega seotud informatsiooni jagamine on nii-öelda tavakasutaja jaoks süsteemi üks põhifunktsionaalsuseid. Lisaks on valdkonnas tähtis

mängijaskonna haldamine. Mängijate ja aktivistide jaoks on nende maine väga tähtis. Mainet võivad alla viia mängus petmine, näiteks kuuliga pihta saamise mitte tunnistamine, liiga võimsate relvade kasutamine või liigne vägivaldsus.

3.3 Olemasolevad lahendused

Sarnast lahendust Eestis ei eksisteeri. Siiani on kasutatud kõikideks toiminguteks tavalisi internetifoorumeid. Ürituste organiseerimine toimub tavaliselt ühes foorumi teemas. Foorumeid on mitmeid ja nende kasutajaskonnad on osaliselt kattuvad. Säärase killustatuse tõttu on ürituste ja mängijate haldamine liiga aeganõudev ja frustreriv.

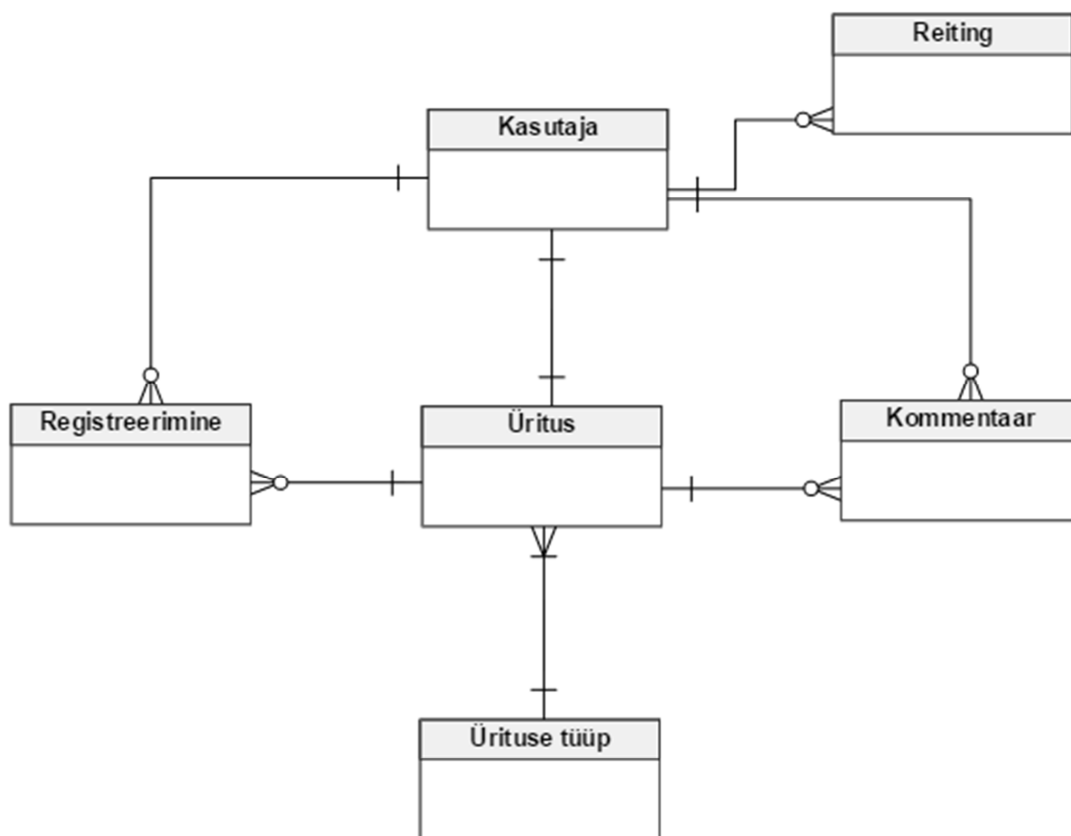
Välismaalt on välja tuua näiteks Belgias kasutusel olev Airsoft United organisatsiooni infosüsteem (www.airsoftunited.be). Põhilised funktsioonid on Belgia infosüsteemil sarnased antud töö raames loodavaga. Lisaks on sellel olemas piiratud kasutatavusega avalik veeb, kus on informatsioon Belgia Airsofti kommuunile olulistest uudistest ning kuulutuste rubriik, kus on võimalik seadusega mitte-reguleeritud Airsofti varustust müüa. Sarnaselt loodavale süsteemile, puudub ka Belgia infosüsteemis foorum, sest nende kasutatavus on sotsiaalvõrgustike tõttu vähenenud.

4 Infosüsteemi arendusprotsess

Selles peatükis kirjeldatakse süsteemi valmimisel kasutatud arendusprotsessi erinevate kõrgtaseme komponentide tasemel.

4.1 Andmebaas

Andmebaasisüsteemina on kasutusel PostgreSQL 9.3. Infovajaduste ja esmase iteratsiooniga loodava süsteemi funktsionaalsuse analüüsi tulemusena valmis andmebaasi struktuur, mis on kujutatud joonisel 3.



Joonis 3. Andmebaasi struktuur

Lisaks joonisel kujutatule on olemas ka kasutusel oleva Ruby on Rails raamistiku spetsiifiline tabel, kus hoitakse andmebaasimigratsioonide informatsiooni. Ruby on Rails pakub andmebaasi loomiseks väga mugavad moodust, milleks on migratsioonid. Migratsioon on Ruby süntaksis kirjutatud andmebaasistruktuuri muudatus. Migratsiooniks võib olla näiteks uue tabeli loomine, indeksi loomine, unikaalsuspiirangu loomine, uue tabeliveeru lisamine jne. Kuna andmebaasi loomine

toimub Ruby süntaksit kasutades, siis on põhimõtteliselt võimalik Ruby on Railsi arendajal luua andmebaasiga veebirakendus SQL süntaksit tundmata.

Ruby on Railsi arendajate seas on andmebaasiobjektide kitsenduste loomisel lahkarvamusi. Ühe võimalusena on kasutusel variant, kus andmebaasi tasemel ei looda ühtegi kitsendust ega seost. Ruby on Rails lubab seda teha edukalt ka mudeli tasemel. Loodud süsteemis on kasutatud seda varianti. Näiteks on võimalik `validates_presence_of` märksõna kasutades kirjeldada mudeli tasemel nullväärtuste mittelubamine. Lisaks on võimalik kirjeldada mudeli tasemel unikaalsuspiiranguid, nagu on tehtud kasutaja mudelis. Joonisel 4 on kasutaja e-posti välja unikaalsuspiirang.

```
validates :email, presence: true, uniqueness: true
```

Joonis 4. Kasutaja e-posti aadressi unikaalsuspiirang

Valideerimine toimub raamistiku poolt automaatselt. Raamistik koostab SQL päringu kasutajate tabeli suunas, otsides sama e-posti aadressiga kirjet. Juhul kui valideerimine tagastab vea, on võimalik selle info põhjal kasutajale kuvada dünaamiline veebileht või modaalaken.

Samuti on võimalik mudeli tasemel kirjeldada seoseid teiste mudelite ehk andmebaasiobjektidega. Joonisel 5 on kasutaja mudelis üritustega seose loomine.

```
has_many :events
```

Joonis 5. Kasutaja ja ürituse vahelise seose loomine

4.2 Tagarakendus

Tagarakendus ehk *back-end* loodi Ruby on Rails raamistikku kasutades ning MVC mustri põhimõttel. Iga kontrolleri puhul on kasutusel raamistiku pakutavad abistavad funktsioonid. Ruby on Rails seob ise kontrolleri funktsioonid neile vastavate vaadete failidega. Näiteks ürituste kontrolleri `index` funktsioon otsib vastavalt marsruudifailile ürituste kaustast faili nimega `index.erb`.

Tagarakenduse üles seadmine on raamistiku poolt oluliselt lihtsustatud võrreldes konkureerivate tehnoloogiatega. Olenevalt arenduses kasutatava arvuti kiirusest ja

installeeritud tarkvara olemasolust, on võimalik primitiivne Ruby on Rails rakendus luua ja lokaalsele serverile käima panna sekunditega.

4.2.1 Marsruutimine

Raamistik pakub marsruutimiseks võimalust kõik rakenduses kasutatavad URLid ühes kohas defineerida. Joonisel 6 on kirjeldatud kõik süsteemis olemasolevad URLid Ruby süntaksis. Kasutatud on nii eraldiseisvaid marsruutide definitsioone, kus on kindlale URLile tehtavale HTTP päringutüübile vastav kontrolleri ja selle funktsioon, kui ka massdefinitsioone märksõnaga `resources`, mis loob kõikide võimalike HTTP päringutüüpide kontrolleri funktsioonid kirjeldatud objekti jaoks.

```
Rails.application.routes.draw do
  devise_for :users, :controllers => { :registrations =>
'registration' }
  root to: 'pages#index'
  resources :events
  resources :users
  get 'register/:id', to: 'events#register'
  delete 'register/:id', to: 'events#del_register'
  delete 'registered/:id', to: 'events#destroy_register'
  post 'comment/:id', to: 'events#add_comment'
  post 'rate/:id', to: 'users#add_rating'
  get 'oldevents', to: 'events#old_events'
  get 'myevents', to: 'events#my_events'
  get 'myregistrations', to: 'events#my_registrations'
  put 'users/trust/:id', to: 'users#set_trusted'
end
```

Joonis 6. Marsruudid

4.2.2 Kontrollerid

Kontrollerid kasutavad puhta koodi põhimõttel koodi duplikeerimise vältimiseks raamistiku poolt pakutavaid abifunktsioone. Enne või pärast kindlate kontrolleri funktsioonide jooksmist on võimalik defineerida funktsioonid, mis kindlasti jooksmas peavad. Näiteks enne ükskõik millise ürituste kontrolleri funktsiooni väljakutset on vajalik kasutaja autentimine. Ühe rea koodiga on võimalik tervele kontrollerile defineerida juurdepääsupiirang. See osutub vajalikuks, kui mingi osa funktsionaalsusest tuleb näiteks autentimata kasutajate eest peita või kindla kasutajatüübi eest kaitsta. Joonisel 7 on kujutatud ürituste kontrolleri juurdepääsupiirang. Juurdepääs on avatud ainult autenditud kasutajatele.

```
before_action :authenticate_user!
```

Joonis 7. Ürituste kontrolleri juurdepääsupiirang

Igale URLile saadetud parameetride ehk *payloadi* kasutamine on kontrolleri vastavas funktsioonis defineeritud. Näiteks uue ürituse loomisel lubatud saadetavate parameetrite list on välja toodud joonisel 8.

```
def allowed_parameters
  permitted = [:name, :description, :date, :eventType,
              :participationCost, :place, :image]
  params.require(:event).permit(permitted)
end
```

Joonis 8. Ürituse loomisel kasutajaliidesest saadetavate lubatud parameetrite loetelu

See on väga lihtne moodus rakenduse turvamiseks ja vähendamaks API vastaste rünnakute toimumiste arvu ja nende edukust.

4.2.3 Mudelid

Lisaks punktis 4.1 Andmebaas kirjeldatud mudelipõhiste andmebaasiobjektide kitsendustele on kasutatud süsteemi otsingufunktsionaalsuse loomiseks teist Ruby on Rails raamistikule loodud teegi poolt pakutavat abifunktsiooni. *Scope* teek lubab mudelipõhiseid otsinguid defineerida vähese vaevaga. Joonisel 9 on kirjeldatud ürituse nime järgi otsimise abifunktsioon ürituse mudelis.

```
scope :event_name, -> name { where("lower(name) LIKE ?",
"%#{name.downcase}%" ) }
```

Joonis 9. Ürituse otsimine nime järgi

Otsingu abifunktsiooni kasutamine jääb kontrolleri kanda. Otsinguvaatele vastava marsruudi kontrolleri funktsiooni instantsmuutujal kasutatakse mudelis loodud abifunktsiooni. Joonisel 10 on näidatud ürituste kontrolleri funktsioon, milles leitakse hetkel autenditud kasutaja üritused. Kui otsinguvormi pole nime sisestatud, tagastatakse kõik leitud üritused.

```
def my_events
  @events = apply_scopes(Event).where(['creator = ?',
    current_user.id]).paginate(page: params[:page], per_page:
    6).order('date ASC')
end
```

Joonis 10. Hetkel autenditud kasutaja kõik üritused

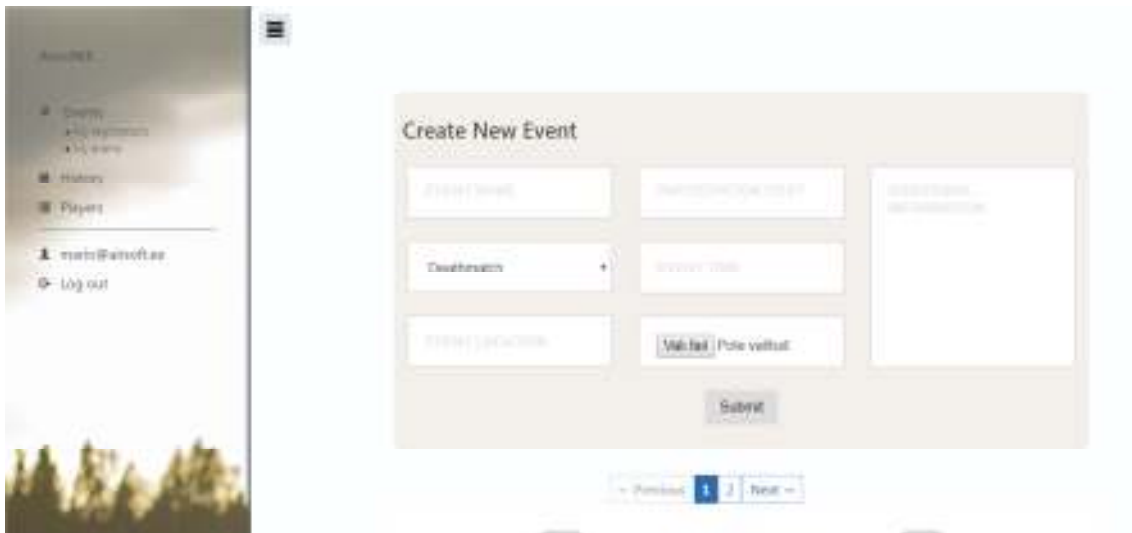
4.3 Kasutajaliides

Kõik vaated on vormistatud eRuby failidena. eRuby on üks põhilisi Ruby on Rails arendajate poolt kasutatavaid templaattehnoloogiaid. eRuby failidesse saab kirjutada tavalist HTML markeeringut ning lisaks `<%= ... %>` ja `<% ... %>` markeeringute vahele ka Ruby süntaksis programmikoodi.

Liidese loomisel on laialdaselt kasutatud Bootstrap raamistikku. Bootstrap võimaldab lihtsustatud korras luua esteetilisi vorme ja vaateid. Kuigi valdkonna liikmete jaoks ei ole disain kuigi tähtis, on Bootstrap hea tehnoloogia vaadetes kasutatavate elementide struktuuri defineerimiseks.

4.3.1 Disain ja kasutusmugavus

Kasutajaliidese prototüüpimine algas staatiliste lehtede loomisega, et saada kasutatavusest paremat ülevaadet. Disainis on kasutatud neutraalseid ja pehmeid toone. Kasutajaliideses ei ole kujutatud militaar- ega sportsümboolikat, et luua valdkonnast kutsuvam maine. Kõik peamenüü ikoonid on võetud Bootstrapiga kaasas olevast ikooniloetelust. Peamenüü taustal on kasutatud töö autori tehtud fotot.



Joonis 11. Autentitud administraatorkasutaja avavaade

Tänu Bootstrapile on võimalik rakendust lihtsamini kasutada erinevate suurustega ekraanidega. Joonisel 12 on sama vaade, mis joonisel 11, kuid palju väiksema ekraanilaiusega seadmel. See on saavutatud kasutades Bootstrapi *grid* süsteemi. Veebilehe saab jagada kuni kaheteistkümneks võrdseks osaks ja olenevalt veebilehte kuvava ekraani ning akna resolutsioonist on võimalik nendes osades olevaid elemente ning infot ümber paigutada vastavalt soovitud disainile.



Joonis 12. Autentitud administraatorkasutaja avavaade kitsal ekraanil

5 Infosüsteemi testimine

Selles peatükis kirjeldatakse valminud prototüübi testimist ning infosüsteemi tulevikku. Samuti tuuakse ära testimise tagasiside tulemusena sisse viidud muudatused.

5.1 Tagaside

Infosüsteemi prototüübi valmimise ajal suheldi pidevalt potentsiaalsete lõppkasutajatega. Sellest tulenevalt oli esialgse valmistoote tagasiside üldjoontes positiivne. Eraldi märgiti ära kergesti mõistetavat kasutajaliidest ning disaini.

Reitingute süsteem oli esialgu binaarne. Kasutajatel oli võimalus teistele kasutajatele jätta kas positiivset või negatiivset tagasisidet. Mõlema variandi jaoks oli eraldi nupp, mille valimisel reiting registreeriti. Testimise käigus leiti, et selline kasutajate hindamise süsteem ei tasu ennast ära, sest kasutajaskond on väike ning hinnangu andmine ei ole alati must-valge. Lisaks toodi ära, et reitingutest üksi ei piisa ning oleks vaja koos hinnanguga ka kommentaari jätmise võimalust.

Ürituste loomise protsessis toodi välja üks puudus. Kasutajate soov oli realiseerida võimalus, aga mitte kohustus, lisada ürituse loomisel sellele pilt. Erinevad ürituste korraldajad kasutavad nii-öelda avataridena enda loodud erksavärvilisi pilte, et need oleks visuaalselt kiirelt teistest eristatavad.

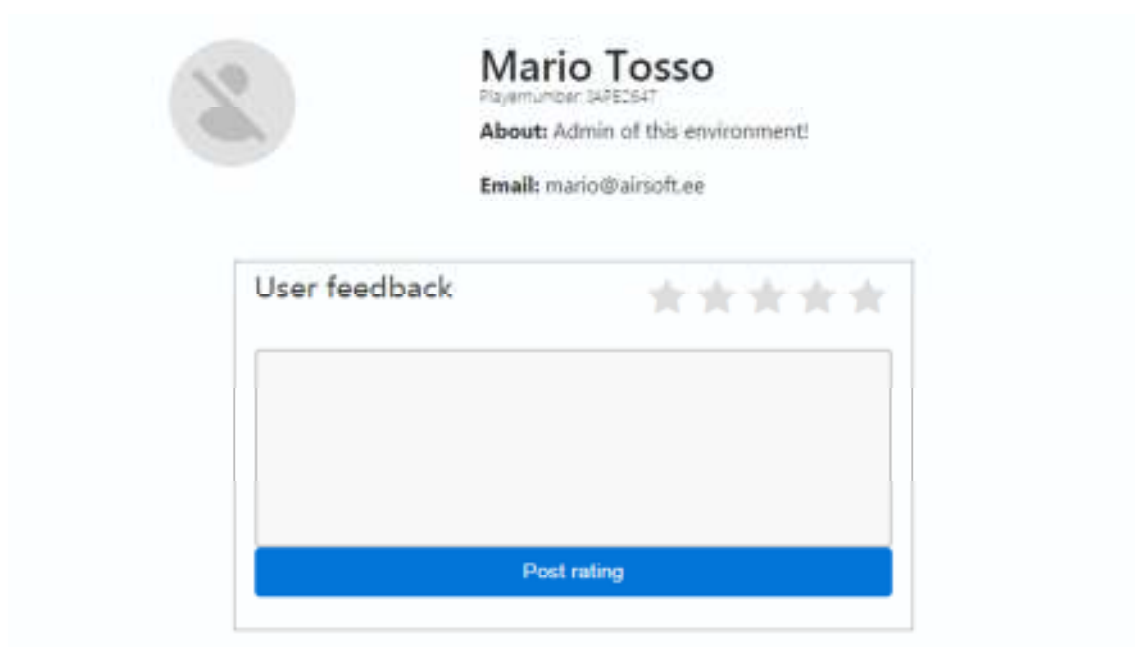
5.1.1 Testimise keskkond

Rakenduse testimiseks kasutati Heroku keskkonda. Heroku on pilveteenusena loodud platvormteenus (PaaS, *platform-as-a-service*), mis laseb arendajatel kiirelt ja mugavalt rakendusi käivitada. Heroku suureks plussiks on automaatne skaleeruvus ja laialdane konfigureeritavus. Herokut kasutades ei pea arendaja ise tegelema füüsiliste serverite ja skaleerumisest tulenevate probleemidega. [12]

Testimise keskkond oli avalik ning ligipääsetav kõigile. Avaliku testkeskkonna olemasolu lihtsustas suhtlust süsteemi tulevaste kasutajatega. Samuti lihtsustas see mõlemapoolset otsustamisprotsessi funktsionaalsuse suhtes, sest Railsi rakenduses tehtud muudatused on võimalik paari minuti jooksul seal kättesaadavaks teha.

5.2 Muudatused tagasisidest

Analüüsi tulemusena jõuti järeldusele, et mõlemad suuremad tagasisidest välja toodud muudatusettepanekud on piisavalt kiiresti realiseeritavad ning kasutajakogemus muutuks nende tulemusena paremaks.



Joonis 13. Uus reitinguvaade

Joonisel 13 on kujutatud modifitseeritud reitinguvaade. Lisatud on kommenteerimise võimalus ning reiting ei ole enam binaarne. Kasutusel on viie palli skaala, kus on võimalik reitingut anda 0.5 punkti täpsusega. Joonisel 14 on kujutatud poole punkti täpsusega valitavat reitingut.



Joonis 14. Reitingu valimine

Teise muudatusena viidi sisse ürituse loomisel sellele pildi lisamise võimalus. Selleks kasutati Ruby on Railsile loodud kolmanda osapoole teeki Paperclip. Teeki sisseehitatud valideerimist kasutatakse failitüübi ning suuruse piiramiseks. Joonisel 15 on kujutatud programmikoodi, millega piiratakse üleslaetava faili suurust maksimaalse

ühe megabaidini ning lubatud failitüüpide nimistusse lisatakse neli tüüpilisemat pildifaili tüüpi: jpg, jpeg, gif, png.

```
validates_with AttachmentSizeValidator, attributes: :image, less_than:
1.megabytes
validates_attachment :image, :content_type => { :content_type =>
["image/jpg", "image/jpeg", "image/gif", "image/png"] }
```

Joonis 15. Pildifaili lisamise validaatorid

5.3 Tulevad arendused

Kuna sihtgrupp luges esialgse ning muudatustejärgse süsteemi õnnestunuks, siis ei ole siiani rohkem laekunud suuremaid muudatusettepanekuid. Sellest hoolimata on infosüsteemile võimalik lisada mõningaid komponente. Kuigi see ei olnud sihtgrupi soov, siis avaliku veebi loomine muudaks kogu valdkonna veel ühtsemaks.

Avalikus veebis võiks olla sarnaselt punktis 3.3 Olemasolevad lahendused kirjeldatud Belgia Airsofti infosüsteemile uudiste ning turu funktsionaalsus. Kuna Eestis tegutsev Airsofti kogukond on liikmete arvult väike, siis vajab see otsus täiendavad analüüsi. Kasutajate ning sisuliste uudiste puudulikkuse tõttu on võimalik, et need avaliku veebi osad jääksid tühjaks samamoodi nagu on tühjaks jäänud seni kasutusel olevate foorumite suhtlusalamfoorumid.

Arenduse käigus selgus, et Airsofti valdkonnas tegutseb inimesi, kes on huvitatud loodud projekti üle võtmisest ning selle API-t kasutades mobiilirakenduse loomisest. Mobiilirakendus ei kuulunud selle töö skoopi, kuid mõningate lihtsate muudatustega on loodud API kasutatav ka mobiilirakenduse loomiseks.

5.4 Kokkuvõte

Testimise faasis esinenud puudujäägid on tarkvaraarenduse protsessis tüüpilised. Arendaja ning sihtgrupp ei suuda kogu vajalikku funktsionaalsust korraka kirjeldada isegi staatilise prototüübi olemasolul. Paratamatult muutub pea iga kose meetodikas alustatud tarkvaraprojekt lõpuks erinevate agiilsete tarkvaraarendusmeetodikate hübriidiks. Sellest hoolimata võib lugeda testimise faasi edukaks. Testimise järgse

arenduse käigus sisse viidud muudatuste mõju kasutajakogemusele on positiivsed. Sihtgrupi ootused said täidetud, kuid arvestades, et eelnevalt polnud sarnast süsteemi keegi kunagi kasutanud, siis võib arvata, et süsteemi igapäevase kasutuse käigus tekib kasutajate tagasisidet veel. Teiste sarnaste süsteemidega võrdlusmomendi puudumine mõjub süsteemiga rahulolule hästi.

6 Kokkuvõte

Käesoleva bakalaureusetöö peamiseks eesmärgiks oli infosüsteemi loomine Eestis Airsofti militaarmänguga tegelevatele inimestele. Infosüsteemi peamiseks eesmärgiks oli ürituste loomise ja nendega liitumise osaline automatiseerimine ning digitaliseerimine. Lisaks oli eesmärgiks mängijaskonna haldamise lihtsustamine ning sellest parema ülevaate saavutamine. Sarnase süsteemi senise puudumise tõttu oli suur rõhk täpse funktsionaalsuse vajaduse väljaselgitamisel.

Töö tulemusena valmis infosüsteem, mis on läbinud sihtgrupi poolse testimise ning saanud nende heakskiidu. Valminud süsteemis on võimalik piisavate õigustega kasutajal luua üritusi, neid kommenteerida ning üritustele registreerida. Lisaks on võimalik anda tagasisidet teiste sihtgruppi kuuluvate süsteemiga liitunud kasutajate kohta. Ürituse korraldaja saab enda loodud ürituse raames registreerimisi hallata. Administraatorkasutajad saavad registreeritud kasutajaid hallata – nende registreerimisi kinnitada ning nende reitingut muuta. Mängijatel on võimalik lisada enda registreeritud kasutajale Airsofti kommuuni poolt antud mängijanumber. Samuti on võimalik mängijaid selle järgi identifitseerida ning otsida. Eraldi on võimalik ülevaade saada ajaloolistest sündmustest ning enda poolt loodud sündmustest. Testimise tagasiside tulemusena lisati ka viie palli skaalal reitingusüsteem, mis osutus algselt planeeritud binaarsest hääletussüsteemist efektiivsemaks.

Töö põhilised eesmärgid said täidetud ning sihtgrupi poolne tagasiside oli positiivne. Sihtgruppi kuulub aktiviste ning tarkvaraarendajaid, kes on avaldanud soovi infosüsteem ja koodibaas üle võtta ning administraatorkohustused enda õlule võtta. Testimise faasis antud tagasiside näitas, et süsteemi testiti piisavalt, et esialgne toode väljastada, kuid rohkemate lõppkasutajate kaasamine süsteemi toob kindlasti kaasa veel muudatusettepanekuid. Kuigi autor oli kunagi valdkonnaga seotud, on igapäevaselt Airsoftiga tegutsevate inimeste pilgu läbi süsteemi testimine ja kasutamine hindamatu tähtsusega.

Arenduse käigus võeti autoriga ühendust sihtgruppi kuuluvate tarkvaraarendajate poolt ning sooviti loodud API-t kasutades luua sama funktsionaalsusega mobiilirakendus. See näitab, et tegu on sihtgrupile olulise temaga ning kasutajate soov loodud infosüsteemi käivitamiseks on olemas. Muudeks lisaarendusteks võib kaaluda avaliku veebi loomist, kuid see vajab sihtgrupi väiksuse tõttu täiendavat analüüsi. Lisaks on Airsoft kallist ning spetsiaalset varustust nõudev ala, mistõttu on vähetõenäoline, et avalik veeb võhikuid kommuuniga liituma kaasaks.

Kasutatud kirjandus

- [1] „Rails,“ [Võrgumaterjal]. Available: <https://github.com/rails/rails>. [Kasutatud 15 05 2017].
- [2] „Ruby on Rails,“ [Võrgumaterjal]. Available: <http://rubyonrails.org/everything-you-need/>. [Kasutatud 15 05 2017].
- [3] „Vexxhost,“ [Võrgumaterjal]. Available: <https://vexxhost.com/resources/tutorials/10-essential-useful-ruby-on-rails-4-gems/>. [Kasutatud 15 05 2017].
- [4] „Devise,“ [Võrgumaterjal]. Available: <http://devise.plataformatec.com.br/>. [Kasutatud 15 05 2017].
- [5] „Paperclip,“ [Võrgumaterjal]. Available: <https://github.com/thoughtbot/paperclip>. [Kasutatud 15 05 2017].
- [6] „PostgreSQL,“ [Võrgumaterjal]. Available: <https://www.postgresql.org/about/>. [Kasutatud 15 05 2017].
- [7] „DB-Engines,“ [Võrgumaterjal]. Available: <https://db-engines.com/en/ranking>. [Kasutatud 15 05 2017].
- [8] „Bootstrap,“ [Võrgumaterjal]. Available: <https://github.com/twbs/bootstrap>. [Kasutatud 15 05 2017].
- [9] „KeyCDN,“ [Võrgumaterjal]. Available: <https://www.keycdn.com/blog/front-end-frameworks/>. [Kasutatud 15 05 2017].
- [10] „Wikipedia,“ [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/HTML5>. [Kasutatud 15 05 2017].
- [11] „Puma,“ [Võrgumaterjal]. Available: <http://puma.io/>. [Kasutatud 15 05 2017].
- [12] „Heroku,“ [Võrgumaterjal]. Available: <https://www.heroku.com/about>. [Kasutatud 15 05 2017].