

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Hanna Tagen**

**Halduskeskkonna erilahendus veebirakendusele**

**PärimusRada**

**Bakalaureusetöö (9 EAP)**

Juhendajad: Siim Karus  
Mari Sarv

Tartu 2019

## **Halduskeskkonna erilahendus veebirakendusele PärimusRada**

### **Lühikokkuvõte:**

Töö eesmärk on halduskeskkonna erilaheduse loomine veebirakendusele PärimusRada. PärimusRada veebirakendus on mobiilpositsioneerimisel põhinev mäng, mille kaudu on mängijal võimalik tutvuda pärimuspaikadega üle Eesti. Töö autor on ühtlasi ka üks PärimusRada mängu autoritest. Töös tutvustatakse põgusalt PärimusRada mängu, millele halduskeskkond luuakse ning sealhulgas ka bakalaureusetöö autori panust mängu loomisel. Samuti antakse ülevaade halduskeskkonna loomise vajadusest, võrreldes seda WordPress, Joomla! ja Drupal sisuhaldussüsteemidega. Võrdlus toimub neljas kategoorias, milleks on hind, süsteemi tugi, turvalisus ja erivajadused ning analüüsitakse nende põhjal erilahenduse valikut. Lisaks esitatakse funktsionaalsed ja mittefunktsionaalsed nõuded, millest esimesed on kirja pandud kasutaja- ja kasutuslugudena. Töös tutvustatakse halduskeskkonna loomisel kasutatud tehnoloogiaid ning töö arhitektuuri, mis on jaotatud tagarakenduseks, kasutajaliideseks ja andmebaasiks. Lõpus antakse ülevaade töö tulemist, tellija ja töö autori hinnang tehtud tööle ning projekti tulevikust. Töö eesmärk sai edukalt täidetud ja kulmineerus nõuetele vastava ning toimiva halduskeskkonnaga. Saadud kogemuste põhinedes võib väita, et halduskeskkondade erilahenduse loomise vajadus on aktuaalne, kuid suure mahu ning ajakulu tõttu tuleks võimalusel kasutada juba olemasolevaid süsteeme.

### **Võtmesõnad:**

Halduskeskkond, sisuhaldussüsteem, veebirakendus

**CERCS:** P170, Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

# **Custom Administrative Area Solution for Web Application PärimusRada**

## **Abstract:**

The goal of this thesis is to develop a custom CMS (Content Management System) for web application PärimusRada (Trail of Folklore). PärimusRada is a game based on mobile positioning to introduce Estonian sites of folklore to the game players. The author of this thesis is also one of the authors of the PärimusRada game. The thesis introduces the game PärimusRada, as well as the author's contribution to the game's development. Besides that, the thesis gives an overview of the reasoning behind creating a custom solution, which is based on comparison between WordPress, Joomla!, Drupal and the created custom solution. Those four systems are compared in four following categories: price, security, system support and special needs. The choice of making a custom CMS is then analyzed based on those four categories. Furthermore, the thesis gives an overview of functional and non-functional requirements, as well as user stories and use cases. Besides that, the thesis presents and introduces technologies used as well as the architecture of the project, which is divided into three parts: back-end application, front-end application and database. At the end, there is an overview of the result of the development, the client's and the author's evaluation of the created solution and the future of the project. The goal of this thesis was successfully fulfilled, which culminated in a functioning administrative area, thereby satisfying the requirements. Based on gained experience it can be stated that the necessity of the custom administrative area solutions is actual, but off-the-self solutions are preferred, where possible, for the great time expenditure and the volume of creating a custom solution from the scratch.

## **Keywords:**

Administrative area, CMS, Content Management System, web application

**CERCS:** P170, Computer science, numerical analysis, systems, control

## Sisukord

|     |  |    |
|-----|--|----|
| 1.  | Sissejuhatus .....   | 6  |
| 2.  | Mõisted.....   | 8  |
| 3.  | PärimusRada mängu taustainfo ja tutvustus .....            | 9  |
| 3.1 | PärimusRada mängu eesmärk ja sihtrühm .....                | 9  |
| 3.2 | PärimusRada mängu tutvustus ja juhend .....                | 9  |
| 3.3 | Autori panus eelprojekti ja klientrakendusse .....         | 10 |
|     | Disain .....   | 10 |
|     | Kasutajaliides .....                                       | 10 |
| 3.4 | Eelprojekti halduskeskkond .....                           | 11 |
| 3.5 | Eelprojekti andmebaas .....                                | 12 |
| 4.  | Halduskeskkond .....                                       | 14 |
| 4.1 | Võrdlus olemasolevate halduskeskkondade platvormidega..... | 14 |
|     | Hind.....  | 15 |
|     | Süsteemi tugi.....   | 15 |
|     | Turvalisus .....   | 15 |
|     | Erivajadused.....  | 16 |
|     | Võrdluse analüüs.....                                      | 16 |
| 5.  | Nõuded .....   | 17 |
| 5.1 | Kasutajalood.....  | 17 |
| 5.2 | Kasutuslood.....   | 18 |
| 6.  | Kasutatud tehnoloogiad.....                                | 21 |
| 6.1 | Spring Boot 2.1 .....                                      | 21 |
| 6.2 | PostgreSQL 42.2.5 .....                                    | 21 |
| 6.3 | REST API.....  | 21 |
| 6.4 | Vue.js 2.5.17 .....  | 22 |
| 6.5 | OpenLayers 5.3.0 .....                                     | 22 |
| 7.  | Arhitektuur .....  | 23 |
| 7.1 | Tagarakendus .....   | 23 |
| 7.2 | Kasutajaliides .....                                       | 24 |
| 7.3 | Andmebaas .....  | 25 |
| 8.  | Töö tulem ja hinnang .....                                 | 27 |
| 9.  | Kokkuvõte ja tulevik.....                                  | 29 |

|                              |    |
|------------------------------|----|
| 10. Viidatud kirjandus ..... | 30 |
| Lisad .....                  | 32 |
| I. Kasutajalood .....        | 32 |
| II. Kasutuslood .....        | 34 |
| III. Litsents .....          | 40 |

## 1. Sissejuhatus

Halduskeskkondade kasutamine aitab IT-kaugel inimestel oma veebilehti kergemini hallata. Sellega seoses tasub aga meeles pidada, et kõik veebilehed ei vaja eraldiseisvat sisuhaldussüsteemi, rääkimata erilahendusest. Halduskeskkonna rakendamise peale tasub mõelda siis, kui veebilehe sisu on tarvis tihti muuta. Hea näide on siinkohal blogid, kuhu lisatakse regulaarselt postitusi juurde. Kui tegu oleks veebilehena, kus ei ole sisuhaldussüsteem rakendatud, peaks blogija hakkama ise oma veebilehe lähtekoodi muutma. Blogijale, kellel on IT-haridus, ei pruugi see raskusi valmistada, kuid arvestama peab asjaoluga, et kõikidel inimestel ei ole IT-alaseid teadmisi varnast võtta. Sellisel juhul tuleks kindlasti lisada veebilehele halduskeskkond. Blogi oli küll hea näide halduskeskkonna vajalikkusest, kuid reeglina ei suuda see näide põhjendada halduskeskkonna erilahenduse loomise vajadust, mida käesolev bakalaureusetöö käsitleb.

Halduskeskkonna kasutamisele võtmise või mitte võtmise otsuse saab teha üsna kergesti, kuid hoopis keerulisem küsimus on millist halduskeskkonda kasutada. Juba kiire Google-i otsing annab vastuse, et valikus on ligi sadakond süsteemi, arvestamata erilahendusi. WordPress, Joomla!, Drupal on ühed valikus olevad sisuhaldussüsteemid, mida käesolev bakalaureusetöö ka oma erilahendusega võrdleb. Need platvormid, nagu ka paljud teised, lubavad oma stamplahendust modifitseerida ja sinna lisada erinevaid pistikprogramme. Taolised platvormid on mugavad inimestele, kellel on lihtsa loogikaga veebileht, nagu näiteks blogi või kasvõi veebipood, kuid ka suuremad süsteemid saavad taoliste platvormide kasutamisega edukalt hakkama. Juhul kui soovitatav lahendus läheb liiga keeruliseks, et olemasolevad sisuhaldussüsteemid seda rakendada saaksid, võib olla parim lahendus rakendada halduskeskkonna erilahendus.

Käesoleva bakalaureusetöö eesmärgiks on luua PärimusRada veebirakendusele halduskeskkonna erilahendus. PärimusRada veebirakendus on mobiilpositsioneerimisel põhinev mäng, mille kaudu on mängijal võimalik tutvuda pärimuspaikadega üle Eesti. Mängija saab punkte vastavalt punkti jõudmise eest ning ülesande lahendamise eest. Ülesanne põhineb kohasidusal pärimusainesel, mis pärineb Eesti Kirjandusmuuseumi Eesti Rahvaluule Arhiivist, mille esindaja on ühtlasi ka veebirakenduse tellija [1]. Projekt sai alustatud grupitööna 2018. aasta sügissemestril Tartu Ülikoolis, aine „Tarkvaraprojekt LTAT.05.005“ raames ning kandis sel ajal nime „Mobiilpositsioneerimisel põhinev veebirakendus pärimuskogude tutvustamiseks“.

Loodav erilahendus peab võimaldama radade, punktide, administraatorite, ülesannete, keelte ja sisutekstide haldust, ülesannete vastuste eksportimist ning mängijate nimekirja kuvamist ja olema sealjuures võimalikult kasutajasõbralik. Põhjalikum ülevaade nõuetest on välja toodud peatükis Nõuded ning esimeses lisas (I. Kasutajalood). PärimusRada projekt on hoiustatud GitHub koodihoidlas ja on avalikult kättesaadav aadressilt: <https://github.com/hannatagen/rada100/>.

Töö koosneb üheksast peatükist. Teises peatükis seletatakse töös kasutatavaid mõisteid. Kolmandas peatükis tutvustatakse lugejale mängu PärimusRada, millele halduskeskkond luuakse. Tuuakse välja bakalaureusetöö autori panus rakenduse loomisel, mängu eesmärk ja sihtrühm, mängu juhend ning muuhulgas ka kasutatud tehnoloogiad. Neljandas peatükis selgitatakse lugejale bakalaureusetöö eesmärki ehk miks on mängule just erilahendust tarvis ning võrreldakse erilahendust olemasolevate sisuhaldussüsteemidega. Viiendas peatükis käsitletakse rakenduse vajalikke funktsionaalsusi ning antakse lühiülevaade kasutaja- ning

kasutuslugudest. Kuuendas peatükis tuuakse välja töös kasutatud tehnoloogiad ning kuuendas peatükis antakse lugejale ülevaade rakenduse arhitektuurist. Kaheksandas peatükis viiakse lugeja kurssi töö tulemi ja nii tellija kui ka bakalaureusetöö autori hinnanguga tehtud tööle. Üheksandas peatükis antakse kokkuvõtte tehtud tööle ja tutvustatakse edasisi plaane. Lisades on põhjalikumalt välja toodud kasutajalugudena kirja pandud funktsionaalsed nõuded ning kasutuslood.

## 2. Mõisted

Käesolevas bakalaureusetöös on kasutatud nii mõndagi mõistet või terminit, mis ei pruugi olla kas üheselt mõistetav või kõigile teada. Et lugejat aidata, on siia peatükki kokku kogutud mõisted ja terminid ning nende selgitused.

**Sisuhaldussüsteem / haldussüsteem / halduskeskkond** – käesolev bakalaureusetöö käsitleb neid kolme mõistet samavõrdsena. Tegu on süsteemiga, mis võimaldab rakenduses haldustoimingute (sisu- ja kasutajatehaldus) teostamist.

**Pärimus** – inimeste suhtluse käigus edasiantud rahvalooming, tava uskumus või muu taoline [2].

**Pärimuspaik** – koht, mis on seotud mõne rahvaloomingu, tava, uskumuse või muu taolisega. Tüüpilised pärimuspaigad on: looduslikud pühakohad; ravipaigad; vaimolendite, hiidude ja ajalooliste isikutega seotud kohad; imeliste ilmingute avaldumiskohad; ajaloolised hooned, teed, rajatised; matusapaigad; kabeli- ja kirikukohad [3].

**Pärimusaines** – dokumenteeritud pärimus, nt Eesti Rahvaluule Arhiivis säilitatud pärimus.

**Eelprojekt** – PärimusRada veebirakenduse projekt, mis on valminud enne käesolevat bakalaureuse tööd, seega on eelprojekt bakalaureusetöö skoobist väljas ning ei kuulu kaitsmisele.

**Superadministraator** – administraator, kellel on lisaks tavaadministraatorile õigus ka lisada, muuta ja kustutada teisi administraatoreid.

**Klientrakendus** – PärimusRada veebirakenduse osa, mida saavad peale administraatorite ja superadministraatorite kasutada ka tavakasutajad ehk mängijad.

**Rada** – virtuaalne rada, mis on kujutatud PärimusRada mängu kaardil. Üks rada koosneb mitmest punktist ja iga punkt vastab ühele Eesti pärimuspaigale. Rajad ja nendes olevad punktid paiknevad koordinaatteljestikul üle Eesti. Raja läbimiseks tuleb mängijal füüsiliselt punkti kohale minna. Radade läbimist võib võrrelda orienteerumisega, kuid punkte võib läbida mis tahes järjekorras.

**Ülesanne** – kui mängija jõuab pärimuspunkti kohale, avaneb tal võimalus lugeda punktiga seonduvat kohasidusat pärimusainest. Põhinedes konkreetse punkti kohasidusal pärimusainesel, on mängijal soovi korral võimalik vastata ka küsimusele ning täiendada seeläbi Eesti Rahvaluule Arhiivi.

### 3. PärimusRada mängu taustainfo ja tutvustus

PärimusRada mängu autorid on Hanna Tagen, Kerttu Talts ja Age Roosi. Rakendus on arendatud koostöös Eesti Kirjandusmuuseumi Eesti Rahvaluule Arhiiviga, mille soov on saada mobiilpositioneerimisel põhinev veebirakendus pärismuskogude tutvustamiseks ja arhiivimaterjali täiendamiseks. PärimusRada mängu veebiaadress on <http://rajad.kirmus.ee/>. Käesoleva bakalaureusetöö esitamise hetkel tõsteti parasjagu veebirakendust uude serverisse üle, ning seetõttu võib lingi toimimine olla häiritud.

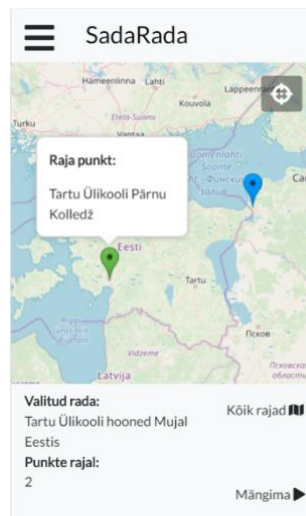
#### 3.1 PärimusRada mängu eesmärk ja sihtrühm

PärimusRada veebirakendus on mäng, mille kaudu saab Eestit ja eesti kultuuri paremini tundma õppida. Tellija eesmärgiks mängu loomisel on Eesti Rahvaluule arhiivimaterjali täiendamine kasutajate vastustega. Sihtrühmaks on kõik inimesed, kes on huvitatud kultuursest ajaviitest, aktiivsest puhkusest ja siseturismist.

#### 3.2 PärimusRada mängu tutvustus ja juhend

Mäng koosneb erinevatest radadest, mille moodustavad üle Eesti paiknevad pärimuspaigad. Ühe raja punkte saab läbida mis tahes järjekorras ehk punktid ei ole üksteisest sõltuvad.

Külastaja saab veebisaidile tulles vaadata külgmenüüst radade nimekirja, mängu tutvustust ja reegleid, korduma kippuvaid küsimusi ja haldajate kontaktandmeid. Samuti on põhiaknas näha kaart, millel on kuvatud kõik rajad. Ühe konkreetse raja kuvamiseks on kasutajal tarvis vajutada mis tahes punkti peale. Joonisel 1 on näha, et peale punkti valimist, kuvatakse kaardil ainult selle raja punktid, kuhu punkt kuulub.



Joonis 1. Mängija on valinud kaardilt ühe raja.

Klõpsates ühele punktile, avaneb alumises servas jalus (ingl *footer*), mis kuvab valitud raja nime ja rajal olevate punktide arvu. Kui kasutaja on teinud endale külgribal asuval alamlehel kasutajakonto, ja sellega ka sisse loginud, on alumises infoaknas lisaks raja infole nupp „Mängima“, millele vajutades saab mängija valitud rada läbima hakata. Kui mängija soovib vahepeal rada vahetada, siis saab ta käimasoleva mängu pausile panna, seejärel uue raja valida ning seda mängida. Tagasi poolelioleva mängu jätkamiseks on tarvis mängijal liikuda

külgribal alamlehele „Profiil“, kus on nähtavad konkreetse kasutaja poolel olevad mängud kui ka lõpetatud mängud ja kasutajanimi.

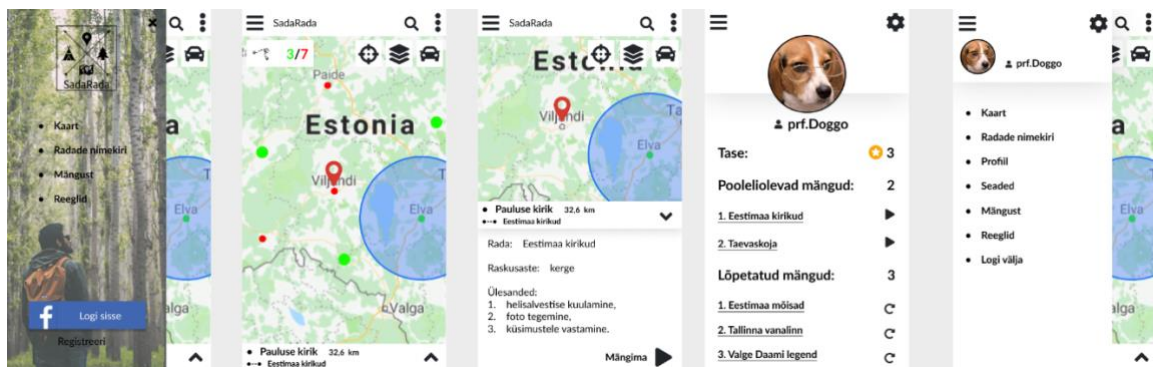
Punkti jõudes avaneb mängijale võimalus lahendada ülesannet, mis põhineb punktiga seotud Eesti Rahvaluule Arhiivi materjalile. Materjaliks on tekst, millele saab vabas vormis vastuse sisestada. Punkti jõudmise eest saab mängija ühe punkti ning ülesande lahendamise eest veel omakorda ühe lisapunkti.

### 3.3 Autori panus eelprojekti ja klientrakendusse

PärimusRada eelprojekti arendamisel oli tiimis üks inimene nõutavast neljaliikmelisest tiimist vähem, mis tähendas, et kolmeliikmelises tiimis langes igale liikmele suurem vastutus. Sellises olukorras tegelevad kõik tiimiliikmed enam-vähem kõigega. Bakalaureusetöö autori suuremateks ülesanneteks olid muuhulgas kujunduse loomine ja kasutajaliidese loomine.

#### Disain

Autori üheks esimeseks ülesandeks oli rakenduse ülddisaini välja mõtlemine. Parima lahenduseeni jõudmiseks tegid esmalt kõik tiimiliikmed paberi peal rakenduse voo skeemi. Skeeme võrreldi omavahel ning kolmest joonisest valiti välja parimad elemendid. Käisiti joonistatud skeemidest pani autor kokku esinduslikud maketid, et neid kliendile esitleda. Joonisel 2 kujutatud prototüüp on valmistatud veebirakendusel põhineva tööriistaga Figma.



Joonis 2. Rakenduse prototüüp.

Prototüübiga saab tutvuda aadressil:

<https://www.figma.com/proto/YlqVQUP9nsqxs9iIhofuEHhq/SadaRada?>

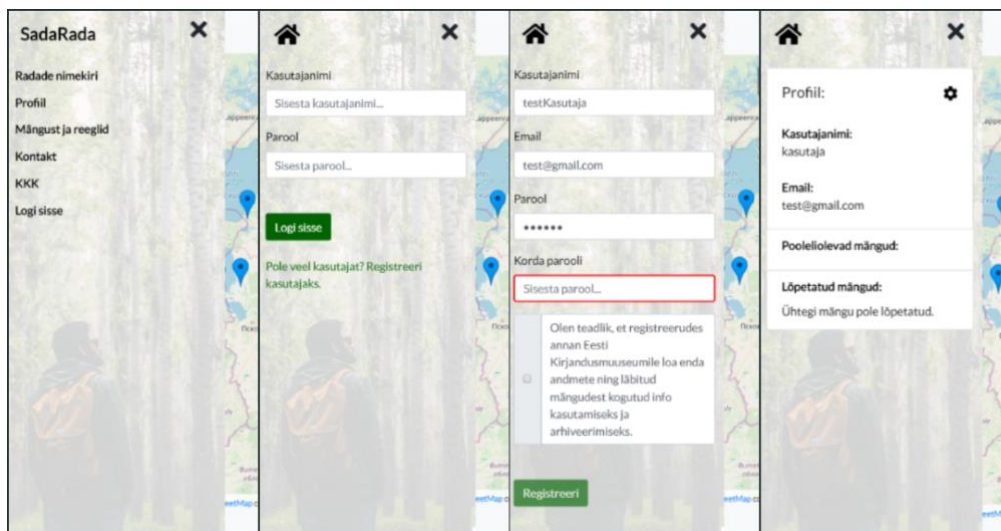
#### Kasutajaliides

Eelprojekti kasutajaliides on tehtud JavaScript-i raamistikuga Vue.js (2.5.17) ning CSS-i raamistikuga Bootstrap (4.3.1), mis on üks enimkasutatavaid CSS-i raamistikke tänu oma lihtsusele, dünaamilisusele ja platvormiülele toele (Android, iOS, Windows jne). Ikoonid on pärit FontAwesome-i (5.5.0) tööriistakomplekti tasuta versioonist, mis pakub peale kirjastiilide ka ikoonide tuge. Lisaks sellele kasutati Vue.js-i laiendit VueRouter-it (3.0.1), mis võimaldab erinevate komponentide vahel mugavalt liigelda. Vue.js raamistik kasutab veebilehe loomisel komponente, mis vähendab koodikordusi ning teeb erinevate leheosade taaskasutamise lihtsaks. Pealegi võimaldab see oma mallides kasutada JavaScript objekte, muutujaid ja meetodeid, mis teeb tingimusliku renderdamise mugavaks. Nendest

tehnoloogiast räägime põhjalikumalt aga peatükides Kasutatud tehnoloogiad ja Arhitektuur.

Kuna Vue.js hõlmab endas ka mitmeid poolfabrikaatkomponente, nagu näiteks Vue külgriba komponent (ingl *Vue SideBar Component*), siis esialgu ehitati külgmenüü selle abiga. Hiljem selgus, et Vue külgribal on suur miinus, mida ei olnud võimalik üle defineerida. Nimelt sulgus menüü automaatselt iga klõpsu peale. Kuna veebirakenduse enam kui pool loogikast on üles ehitatud just külgribale (sisse logimine, registreerimine, radade nimekiri, kontakt, KKK, profiil, seaded), siis ei olnud võimalik enam Vue külgriba komponenti kasutades edasi minna.

Autor asendas külgriba Bootstrap-i kasutades ise tehtud külgribaga ning lisas sellele VueRouter-i laiendi, mis muutis külgriba rakenduse tuumkomponendiks, kus saab erinevate komponentide ehk alamlehtede vahel liigelda. See tähendab ühtlasi ka seda, et rakenduse taustaks olev kaardivaade on staatiline. Joonisel 3 kujutatud rakenduse kuvatõmmistel on näha külgriba erinevaid alamlehti.



Joonis 3. Külgmenüü erinevate alamlehtede ekraanivaated.

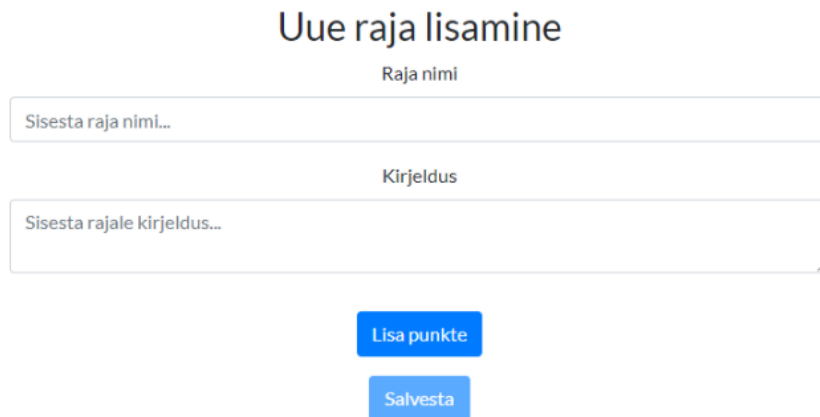
Rakenduse põhivaates olev kaart on loodud kasutades OpenLayers-i (5.3.0) kaardimootorit, mis omakorda kasutab OpenStreetMap-ist pärinevaid andmeid kaardi kuvamiseks. OpenLayers osutus teiste kaardimootorite seas (Google Maps, Leaflet) parimaks lahenduseks, kuna pakub rohkem võimalusi kaardile erinevaid vidinaid lisada ja nende rakendamine on kergem ning mugavam kui näiteks Google Maps-i kasutades. Samuti on OpenLayers-il parem vektoritugi, mis tähendab ka paremat punktide loomise tuge ning OpenLayers-i avatud lähtekood teeb lihtsamaks rakenduse koodi silumise.

### 3.4 Eelprojekti halduskeskkond

Kuna PärinusRada projekt sai alguse Tarkvaraprojekti aine raames, oli aine edukaks läbimiseks tarvis võimaldada tellijal ka radasid juurde lisada. Selle tingimuse täitmiseks lõi eelprojekti kaasautor Kerttu Talts rakendusele minimalistliku halduskeskkonna.

Joonisel 4 on näha esialgset halduskeskkonda, mis tegi ära absoluutse miinimumi. Tegu on lihtsa vormiga, millel on kaks välja, kuhu saab lisada raja nimetuse ning kirjelduse.

Vajutades nupule „Lisa punkte“ avaneb vaade punkti lisamiseks, mida on kujutatud joonisel 5. Punkti lisamise vormis tuleb sarnaselt raja lisamise vormile sisestada punkti nimi ning kirjeldus. Lisaks tuleb sisestada ka Google Maps URL punkti asukohast, mis sisaldab punkti koordinaate.



Uue raja lisamine

Raja nimi

Sisesta raja nimi...

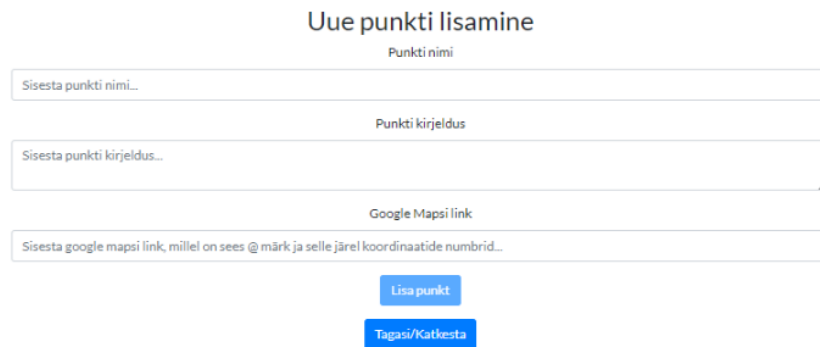
Kirjeldus

Sisesta rajale kirjeldus...

Lisa punkte

Salvesta

Joonis 4. Eelprojekti halduskeskkond – uue raja lisamine.



Uue punkti lisamine

Punkti nimi

Sisesta punkti nimi...

Punkti kirjeldus

Sisesta punkti kirjeldus...

Google Mapsi link

Sisesta google mapi link, millel on sees @märk ja selle järel koordinaatide numbrid...

Lisa punkt

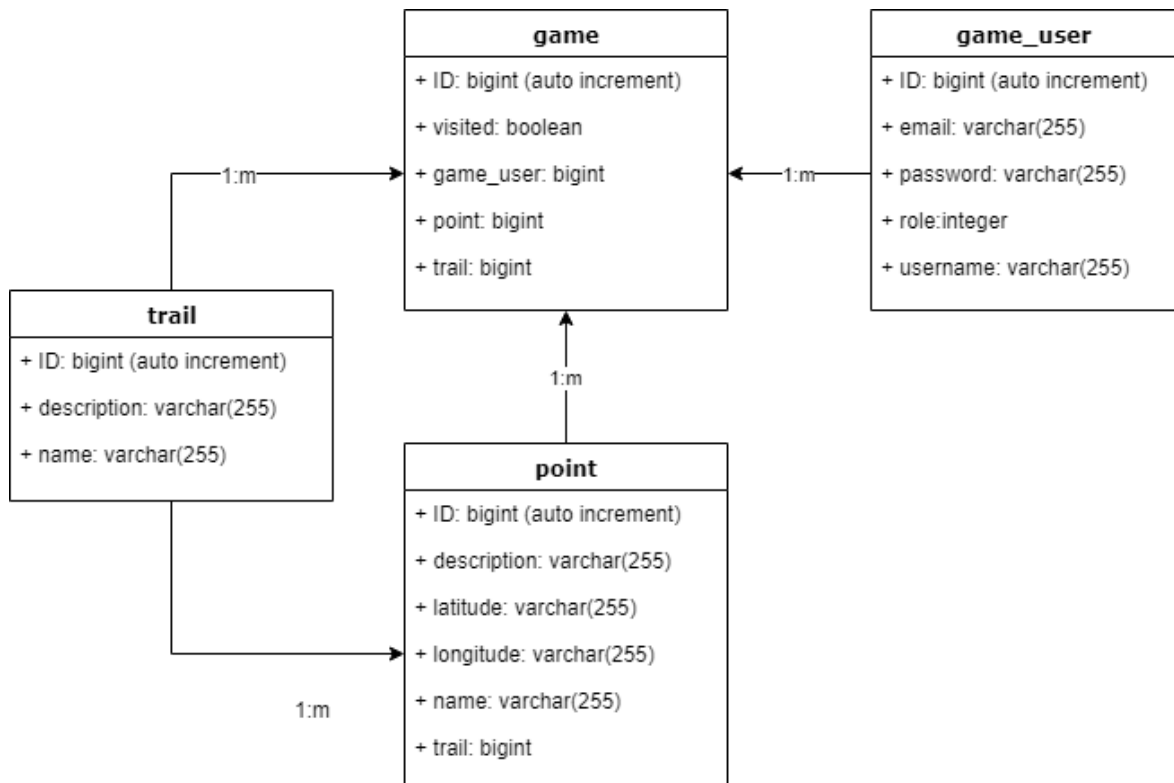
Tagasi/Katkesta

Joonis 5. Eelprojekti halduskeskkond – rajale punkti lisamine.

Esialgu oli idee punktide lisamisel kasutada koordinaate, kuid tellija leidis, et see on liiga keeruline, kuna peaks selleks kasutama spetsiaalseid veebirakendusi, mis punkti asukoha koordinaadid tagastavad. Kuna Tarkvaraprojekti aine raames oleks keerukam halduskeskkond olnud liialt mahukas jõudsid arendajad ja tellija omavahel kokkuleppele, et eelprojekti halduskeskkonnas toimub punktide lisamine Google Maps URL-i abil ning põhjalikum halduskeskkond valmis bakalaureusetöö autori lõputööna.

### 3.5 Eelprojekti andmebaas

PärimusRada eelprojekti ning eelkõige andmebaasi loomise ja haldamise eest vastutas enamjaolt eelprojekti kaasautor Age Roosi, kes ehitas andmebaasi sellised tabelid nagu *game*, *game\_user*, *trail* ja *point*. Joonisel 6 on näha eelprojekti valminud andmebaasi, selles olevaid tabeleid ja nende tabelite sõltuvusi. Kõikide tabelite ID-d on automaatselt genereeritud. Käesolev bakalaureusetöö kasutab oma töös halduskeskkonnaga suhtlemisel eelprojekti tehtud andmebaasi ja selles olevaid tabeleid (*game*, *game\_user*, *trail* ja *point*).



Joonis 6. Eelprojektis valminud andmebaas.

Lisaks Roosi tehtud tabelitele kasutab halduskeskkond töös ka bakalaureusetöö autori tehtud tabelleid *language*, *task* ja *task\_answer*, mille vajalikkus lisandus halduskeskkonna loomisel. Lisandunud tabelitest räägime lähemalt peatükis Arhitektuur.

## 4. Halduskeskkond

PärimusRada veebirakenduse üheks oluliseks osaks on mängu tarbeks radade loomine, mis peab halduskeskkonda kasutavale administraatorile olema võimalikult intuitiivne. Selleks soovib klient halduskeskkonda, kus administraator saab halduskeskkonnas olevale kaardile klõpsates lisada rajale punkte.

Ükski teadaolev sisuhaldussüsteemiplatvorm hetkel sellist võimalust ei paku, mistõttu sobib PärimusRada veebirakendusel põhinevale mängule halduskeskkonnaks eelkõige erilahendus. Üheks võimalikuks lahenduseks on olemasolevaid vabavaralisi platvorme erilahenduse koodiga integreerida, kuid tänu oma populaarsusele ja avatud lähtekoodile satuvad taolised platvormid ka tihti häkkerite rünnakute alla [4]. Arvestades, et tellijaks on Eesti Kirjandusmuuseumi Eesti Rahvaluule Arhiiv, mis on Eesti Haridus- ja Teadusministeeriumi hallatav riigi teadus- ja arendusasutus, vajab PärimusRada veebirakendus erilahendust, mis on mugavam, kiirem ja turvalisem vabavarana pakutavatest sisuhaldussüsteemide platvormidest [5].

Erilahendusena valmistatud halduskeskkonnad on kiiremad, kuna need on teenindavatele veebisaitidele optimeeritud, mis tähendab, et nad ei sisalda ebavajalikke komponente või loogikat [5]. Neid on võimalik luua täpselt veebirakenduse vajaduste järgi, kus ei ole midagi üleliigset ega puudu. Samuti on erilahendused turvalisemad – paljud sisuhaldussüsteemid on vabavaralised, mis tähendab, et kõigil inimestel on selle lähtekoodile ligipääs [6]. See tähendab aga seda, et lähtekoodi muudatusi tehes ei ole need muudatused kohalikud, vaid uuendamise käigus levib tehtud muudatus ka teiste klientideni, kes seda tarkvara kasutavad. Samuti teeb populaarsete sisuhaldussüsteemide kasutamise riskantsemaks nende tihe kasutajaskond. WordPress-i blogihaldamise süsteemi ja sisuhaldussüsteemi kasutas 2018. aasta seisuga 30.5% kõikidest veebilehtedest, mis tähendab, et võrreldes teiste sisuhaldussüsteemidega kasutavad veebilehed oma sisu haldamiseks WordPress-i tervelt 60.1% ulatuses [4]. Kuna WordPress haldab niivõrd palju veebilehti, langeb see rohkem ka häkkerite rünnakute alla, sest korraga saab tekitada suuremat kahju. Kuigi erilahendusena valminud halduskeskkonnad saavad seda riski oluliselt vähendada, ei saa alati väita, et üks või teine on absoluutselt parim valik. Järgmises alampeatükis võrdlemegi erilahendusi juba olemasolevate platvormidega ning toome välja analüüsi, miks otsustati luua just erilahendus.

### 4.1 Võrdlus olemasolevate halduskeskkondade platvormidega

Nagu eelmises lõigus mainitud, kasutab enamik veebilehtedest just WordPress sisuhaldussüsteemi, millele järgnevad populaarsuselt Joomla! ja Drupal. Sellest statistikast lähtuvalt võib julgelt öelda, et tegu on väärt platvormidega ja halduskeskkonna erilahendus ei ole alati parim valik. Otsus, millist haldussüsteemi valida, sõltub aga mitmest asjaolust ning mis kellegi jaoks on oluline, ei pruugi teiste jaoks oluline olla.

Järgmistes lõikudes võrdleme erilahendusena valmivat sisuhaldussüsteemi juba olemasolevate süsteemidega, milleks sedapuhku on WordPress, Joomla! ja Drupal. Kuna nende nelja süsteemi võrdlusest saaks eraldi bakalaureusetöö kirjutada, siis analüüsime neid süsteeme põgusalt neljas kategoorias, milleks on hind, süsteemi tugi, turvalisus ja erivajadused. PärimusRada halduskeskkonda võrreldakse sisuhaldussüsteemidega eelkõige seepärast, et bakalaureusetööna valminud haldussüsteem hõlmab muuhulgas ka sisuloomet

ja selle haldust. Olgu mainitud, et kuna käesolev bakalaureusetöö käsitleb arendajalt tellitud tööd, siis lähtume ka oma analüüsis sellesugusest näitest.

## **Hind**

Kui veebilehe omanik tellib arendajalt töö, mille käigus on vaja juba olemasolevale veebilehele üles seada sisuhalduskeskkond, siis pole see töö reeglina kunagi täiesti tasuta. Välja arvatud muidugi juhud, kui tegemist on samasuguse kokkuleppega, kus töö tehakse ära näiteks lõputöö raames. Antud võrdluses on arvestatud, et arendaja saab alati tehtud töö eest palka, et kujunevat lõpphinda paremini hoomata.

WordPress, Joomla! ja Drupal süsteemid on ise täiesti tasuta, kuid see 0-eurone hind võib kergesti kasvama hakata [7]. Esiteks lisandub sellele arendaja tasu, kes selle üles paneb. Kuna kõigil kolmel süsteemil on hinnasüsteem sarnane, siis võime teha üldistuse, et hind hakkab tänu lisatavatele pistikprogrammidele kasvama [7]. Näiteks kui tellija soovib, et halduskeskkonnas oleks ürituste korraldamise jaoks pistikprogramm, tuleb selle eest lisaraha välja käia. Nende kolme süsteemi vahel osutuks kalleimaks arvatavasti Joomla!, millel on teise kahe süsteemiga võrreldes vähim hulk tasuta lisasid [7].

Erilahenduse halduskeskkondade puhul kujundab lõpphinda eelkõige arendaja palk, mis ei pruugi olla odav. Arendamine võib võtta kaua aega, mis tähendab ka suuremat lõpphinda. Samas on reeglina aga kõikide lisafunktsionaalsuste lisamine ilma lisatasuta.

## **Süsteemi tugi**

Süsteemi tugi on erilahendusena valmivate halduskeskkondade üks tugev pluss. Võtame olukorra, kus loteriisid korraldava veebilehe omanik tahab halduskeskkonnas võidusumma 10000 euro pealt muuta 1000 euro peale, kuid avastab, et võidusumma muutmine ei toimi või on halduskeskkond sootuks maas. Halduskeskkondade erilahenduse puhul on veebilehe haldajal võimalik halduskeskkonna toimimise eest arendaja vastutusele võtta ja ideaalis saab halduskeskkond kiirelt töökorda tagasi. Kui selline probleem tekiks aga vabavaralise sisuhaldussüsteemiga, siis saab veebilehe omanik küll halduskeskkonna püsti pannud arendajaga ühendust võtta, kuid viimane ei pruugi olla võimeline viga kiirelt lokaliseerima või parandama.

Nii WordPress, Joomla! kui ka Drupal on kõik vabavaralised süsteemid ja olgugi, et neil on suur ja tugev kasutajatel põhinev tugigrupp olemas, ei pruugi see alati piisav olla [7].

## **Turvalisus**

Öeldakse, et iga sisuhaldussüsteem on turvaline kuni hetkeni, mil see interneti ühendatatakse [8]. WordPress, Joomla!, Drupal ega ka erilahendused ei ole siinkohal erandiks. Nagu eelpool mainitud, siis langeb WordPress tänu oma populaarsusele palju rünnakute alla. Samuti on turvalisuse probleeme Joomla!-l ja Drupal-il, kuid kuna nende kasutajaskond on väiksem, siis on ka avalikult registreeritud turvaauke vähem. Siinkohal tasuks mainida, et enim probleeme tekib kolmandate osapoolte pistikprogrammide installimisel. Lihtne lahendus oleks taoliseid lisasid mitte kasutada, kuid ei saa välja pruukida olukordi, kui seda pistikprogrammi on tõepoolest süsteemi vaja. Parem lahendus oleks uurida maad ja teha kindlaks kas ja milline teine süsteem sama pistikprogrammi pakub, kuid juhul kui ühtegi head varianti olemasolevatest süsteemidest ei leidu, siis on mõistlik ehitada halduskeskkond erilahendusena. Nii saab teha kindlaks, et ei kasutata kahtlase päritoluga lisasid ning kõik vajaliku saab luua täpselt tellija vajadustele vastavalt.

## **Erivajadused**

Rääkides tellija vajadustest, siis on see üks olulisemaid punkte, mis saab haldusüsteemiplatvormi valimisel otsustavaks. Lihtne blogi või veebisait ei vaja endale erilahendust ning võib julgelt väita, et WordPress on sellisel juhul üks parimaid valikuid. Blogi näite puhul on suure tõenäosusega halduskeskkonna ülespanijaks blogija ise ning sedapuhku on WordPress või Joomla! head valikud, kuna nende installimine on lihtne [8]. Drupal-i installimine on mõnevõrra keerukam ja IT-kauge inimene ei pruugi sellega hakkama saada [8]. Sama kehtib ka erivajaduslike funktsionaalsuste rakendamisel WordPress-i, Joomla! ja Drupal-i süsteemi. Drupal-i puhul on selleks vaja teadmisi PHP-st ja HTML-ist ning programmeerimisoskusi tuleb rakendada juba pelgalt veebilehe uuendamise tarvis [7].

Keerukamate süsteemide puhul, milleks võib lugeda ka PärimusRada mängu, on üheks võimaluseks võtta mõni olemasolev sisuhaldussüsteem ja luua sinna erilahenduse lisa. Üksikute erivajaduste puhul on see igati hea lahendus, kuid juhul kui taoliseid vajadusi tuleb rohkem, tasub mõelda halduskeskkonna erilahenduse peale.

## **Võrdluse analüüs**

PärimusRada mängu tellija soov oli saada veebirakendusele halduskeskkonna erilahendust ja antud juhul on see ka hea lahendus. Seda võime väita nende nelja kategooria põhjal, milles sisuhaldussüsteeme võrdlesime. Esiteks ei maksnud erilahendus tellijale midagi, kuna tegu on bakalaureusetööga, mis ei ole tasustatud. Teiseks võtab halduskeskkonna ja klientrakenduse edasiarendamise peale bakalaureusetöö valmimist üle Eesti Kirjandusmuuseumi IT-tiim, seega on erilahendusena valminud süsteemil olemas otsene haldaja, kes saab vajadusel probleeme jooksvalt lahendada. Kolmandaks ei kasuta töö käigus valminud halduskeskkond kolmandate osapoolte lisasiid, mis vähendavad riski rünnaku alla sattumiseks. Neljandaks on tellija sooviks saada mugav kasutajaliides, mis võimaldab kaardile klõpsates punktide asukohta määrata, radu lisada ja muuta, ülesandeid koostada ning nende vastuseid näha ning ekspordida, veebilehe sisu tõlkeid lisada ja muuta, mängijate nimekirja näha, administraatoreid ja nende rolle hallata ning uusi administraatoreid lisada. Väljatoodud funktsionaalsustest nõuab vähemalt kolm erilahendust, mille rakendamiseks on võimalik luua näiteks Drupal süsteemis lisavaateid, kuid kui tellijal on võimalik nullkuluga saada enda äranägemise järgi parim lahendus, siis on selleks kahtlemata erilahendusena valmiv halduskeskkond. Nendest asjaoludest tulenevalt vajab PärimusRada veebirakendus just halduskeskkonna erilahendust.

## 5. Nõuded

Loodud halduskeskkonna tellija eesmärgiks on Eesti Rahvaluule arhiivimaterjali täiendada mängijate sisenditega. Sellest tulenevalt peab halduskeskkond võimaldama mängijate sisestatud vastuste eksportimist arhiveerimiseks sobivas vormingus, milleks on JSON. Lisaks sellele on tellija soov, et rakendus võimaldaks eksportimist tavakasutajale mugavas CSV vormingus.

### Olulisemad funktsionaalsused halduskeskkonnas on:

1. uute administraatorite lisamine;
2. administraatorite (õiguste) haldamine – sh administraatorikontode kustutamine;
3. sisse loginud administraatori profiili kuvamine ja võimalus muuta konto seadeid;
4. radade koostamine, muutmine ja kustutamine;
5. ülesannete koostamine, muutmine ja kustutamine;
6. ülesannete lahenduste eksportimine JSON ja CSV formaadis.

### Sellele lisaks peab halduskeskkond võimaldama:

1. hallata sisutekste, sh lisada, muuta ja kustutada sisutekstide tõlkeid;
2. administraatoril näha loodud kasutajate nimekirja.

Mittefunktsionaalsete nõuete koostamine oli üheks eelprojekti osaks. Siinkohal toome välja neist mõned, mis puudutavad ka halduskeskkonda:

- Süsteemis saavad kasutajad (külastaja, mängija, administraator, superadministraator) ligi vaid vastavalt nende kasutajarollile mõeldud sisule [9].
- Veebirakendus peab olema dünaamilise veebidisainiga. Testimiseks kasutame Google DevTools' Device Mode [9].
- Arenduseks kasutatakse tehnoloogiad: Vue.js (vähemalt 2.5.17) + Bootstrap (vähemalt 4.1.3) + OpenLayers (vähemalt 5.2.0) [9].

Funktsionaalsed nõuded on kirja pandud kasutaja- (ingl *user story*) ja kasutuslugudena (ingl *use case*). Põgusalt tähendab kasutajalugu ühte lauset, mis kirjeldab, kes on arendatava funktsionaalsuse põhiline kasutaja, millist funktsionaalsust vaja on ja miks. Kasutuslood võtavad lähtepunktiks peakasutaja vaatenurga ja kirjeldavad kogu funktsionaalsuse kasutamiseks läbitud samme, sealhulgas ka alternatiivseid stsenaariume. Kasutaja- ja kasutuslugudest räägimegi järgmistes alampeatükkides lähemalt.

### 5.1 Kasutajalood

Arendajad, kes on harjunud infotehnoloogiaalaste terminite ja tehnoloogiatega igapäevaselt kokku puutuma, võivad lõppkasutajale mõtlemata luua rakendusi, mis ei pruugi IT-kaugetele inimestele intuiitiivne olla. Kasutajalood aitavad arendaja perspektiivi lõppkasutajal hoida ja seeläbi paremaid rakendusi luua [10].

Kasutajalugu on üks nõuete kirjapanemise viis. Kasutajalugudes rõhutatakse konkreetse kasutaja vajadusi ning see peab vastama küsimustele: kes, mida, miks [10]. Kasutajalugude vorm on järgmine: <roll>, <vajadus>, <põhjendus> [10].

Kasutajalugudes kasutatavad rollid:

- **Külastaja** – veebirakendust külastav isik, kellel on õigus:
  - vaadata olemasolevaid radasid ning kus nende punktid paiknevad;

- lugeda mängu kirjeldust ning reegleid;
- näha veebirakenduse haldajate kontaktandmeid;
- lugeda korduma kippuvaid küsimusi;
- registreerida endale konto;
- logida sisse.
- **Mängija** – külastaja, kes on endale konto teinud ning kellel on seeläbi lisaks külastaja õigustele võimalus:
  - mängida – läbida rada ja lahendada punkti jõudes ülesandeid;
  - näha enda profiili ja muuta konto seadeid.
- **Administraator** – administraator, kellel on lisaks külastaja võimalustele PärinusRada veebirakenduses ligipääs PärinusRada halduskeskkonda ning võib seal teha järgmist:
  - lisada, muuta ja kustutada radu ja nendega seotuid punkte;
  - lisada, muuta ja kustutada rakenduses kasutatavaid keeli;
  - lisada, muuta ja kustutada ülesandeid;
  - eksportida ülesannete lahendusi;
  - näha kõikide kasutajate nimekirja;
  - lisada, hallata, muuta ja kustutada rakenduse sisutekste.
- **Superadministraator** – administraator, kellel on lisaks tavaadministraatorile õigus ka lisada, hallata, muuta ja kustutada teisi administraatoreid.

Järgnev näide põhineb PärinusRada halduskeskkonna realsel kasutajalool. Kasutajalugu: „Administraatorina tahan muuta olemasolevaid radu ja nende punkte, et ma ei peaks iga kord uut rada tegema“. Kasutuslugude vormi järgi saab selle lause liigendada järgmiselt:

< Administraatorina >, < tahan muuta olemasolevaid radu ja nende punkte >, < et ma ei peaks iga kord uut rada tegema>.

Sellise meetodiga saab kontrollida, kas kõik vajalikud osad on olemas ja kui ei, siis vastavaid parandusi sisse viia. Rohkem funktsionaalseid nõudeid ja kasutajalugusid leiab töö esimesest lisast (I. Kasutajalood).

## 5.2 Kasutuslood

Kasutuslood on analüüsimismeetod, mis kirjeldab sündmuste käiku ning aitab nõudeid selgemini mõista [11]. Arendustiimil ja kliendil pole alati sarnane teadmistepagas, mis tähendab, et klient ei pruugi teada, kuidas enda soovetud IT-alaste terminitega väljendada. Arendaja peab aga oskama enda tööd kirjeldada ka inimesele, kes arendamisest mitte midagi ei tea. Kasutuslood, ja ka eelmises peatükis tutvustatud kasutajalood, aitavad mõlemal osapoolel üksteisest mõista.

Kasutuslooks valitakse esmalt välja üks rakenduse funktsionaalsus ning seejärel luuakse stsenaarium, kuidas see funktsionaalsus peab toimima ja mis samme kasutaja selleks kõige tõenäolisemalt läbima peab. Kasutuslugu sisaldab peale selle ka eeltingimust funktsionaalsuse kasutamiseks, peamist kasutajat ehk sihtrühma, funktsionaalsuse eesmärgi ehk miks seda üldse tarvis on, alternatiivseid stsenaariume ning järeleolekut ehk mis on funktsionaalsuse kasutamise oodatav lõpptulemus. Tabel 1 tutvustab ühte PärinusRada halduskeskkonna kasutuslugudest, kus käsitletakse uue raja tegemise funktsionaalsust.

Tabel 1. Kasutuslugu UC1 [12].

|                               |  |
|-------------------------------|--|
| <b>Kasutuslugu 1:<br/>UC1</b> | <b>Uue raja tegemine</b>   |
| Seotud nõuded:                | 1.1; 1.5; 1.6  |
| Eeltingimus:                  | Kasutaja on administraatorina sisse loginud halduskeskkonna lehele ja on navigeerinud Radade ja punktide halduse alamlehele.   |
| Peamine kasutaja:             | Administraator   |
| Eesmärk:                      | Administraator tahab luua uut raja.  |
| Peamine stsenaarium:          | <ol style="list-style-type: none"> <li>1. Administraator vajutab nupule “Lisa uus rada”;</li> <li>2. administraator täidab ära vajalikud väljad; <ol style="list-style-type: none"> <li>a. administraator paneb rajale nime; <ol style="list-style-type: none"> <li>i. kui pole nime pannud, siis saab teavituse, et nime peab sisestama;</li> </ol> </li> <li>b. administraator kirjutab rajale kirjelduse; <ol style="list-style-type: none"> <li>i. kui pole kirjeldust pannud, siis saab teavituse, et kirjelduse peab lisama;</li> </ol> </li> <li>c. administraator valib kaardilt punktid, mida soovib lisada rajale; <ol style="list-style-type: none"> <li>i. kui kaardile ei ole lisatud ühtegi punkti, siis saab teavituse, et ilma punktideta, ei saa uut rada teha;</li> </ol> </li> <li>d. administraator teeb iga punkti juurde mingi ülesande; <ol style="list-style-type: none"> <li>i. kui punkti juures ei ole ülesannet lisatud, siis saab teavituse, et punktile peab lisama ülesande;</li> </ol> </li> <li>e. administraator paneb igale punktile raadiuse, kus ülesanne avaneb; <ol style="list-style-type: none"> <li>i. kui raadiust ei ole lisatud, siis saab teavituse, et punktile peab lisama raadiuse;</li> </ol> </li> </ol> </li> <li>3. administraator vajutab nuppu “Loo rada”.</li> </ol> |
| Alternatiiv:                  | <ol style="list-style-type: none"> <li>1. Kui administraator soovib vastuste vaatamist katkestada, siis vajutab nuppu “Katkesta”;</li> <li>2. administraator navigeerib halduskeskkonna alamlehelt minema.</li> </ol>  |
| Järelolek:                    | Andmebaasi on salvestatud uus rada ja mängijad näevad mängus lisatud rada.   |

Joonisel 7 on kuvatõmmis, millel on kujutatud tabelis 1 kasutusloona kirjeldatud uue raja loomise funktsionaalsust. Täpsemalt kuvab joonis 7 halduskeskkonnas olevat radade nimekirja, mis on kasutusloo UC 1 eeltingimuseks. Joonisel 8 on näha uue raja loomist ja joonisel 9 punktide lisamist. Punktide asukohta saab määrata kaardile klõpsates.

Tänu kasutuslugudele saab klient valideerida, kas tema soovist on korrektselt aru saadud, mis vähendab olukorra tekkimise võimalust, kus rakendus tuleb suuremahuliselt ümber kirjutada. Mida hiljem vead avastatakse, seda kulukamaks nende parandamine muutub, seega on kasutuslugude kirjutamine ka säästlik meetod [13]. Põhjalikum ülevaade halduskeskkonna kasutuslugudest toodud välja töö teises lisas (II. Kasutuslood ja kuvatõmmised).

Radade ja punktide haldus

Radade nimekiri Lisa rada

Otsi...

| ID | Nimetus ↓                          | Kirjeldus ↓   | Muuda | Kustuta |
|----|------------------------------------|---|-------|---------|
| 1  | Tartu Ülikooli hooned Mujal Eestis | Sellel rajal saad näha Tartu ülikooli hooneid üle Eesti |       |         |
| 2  | Tartu Ülikooli hooned Tartus       | Sellel rajal saad läbi käia Tartu linna                 |       |         |
| 3  | Lubjaahjud                         | Eesti teadaolevad lubjapõletusahjud                     |       |         |

Joonis 7. Kasutusloo UC1 kuvatõmmis nr 1 – radade nimekiri [12].

Radade ja punktide haldus

Salvesta rada Katkesta

Nimetus:

Kirjeldus:

Punktide nimekiri:

Lisa punkt +

| ID | Nimetus ↓                    | Kirjeldus ↓  | Muuda | Kustuta |
|----|------------------------------|--|-------|---------|
| 1  | Tartu Ülikooli Narva Kolledž | Tartu ülikooli Narva kolledž pakub TÜ kvaliteedimärgiga kõrgharidust, hoiab ja arendab akadeemilisi traditsioone Kirde-Eestis ning on Eesti ühiskonna integratsiooni edendajaks. |       |         |
| 2  | Tartu Ülikooli Pärnu Kolledž | Tartu ülikooli Pärnu kolledž on Eestis teenuste juhtimise alase hariduse lipulaev juba mitu kümnendit.   |       |         |

Joonis 8. Kasutusloo UC1 kuvatõmmis nr 2 – uue raja loomine [12].

Radade ja punktide haldus

Salvesta punkt Katkesta

Nimetus:

Kirjeldus:

Punkti asukoht:  
 Punkti asukohta saad valida ja muuta klõpsates kaardile.

Joonis 9. Kasutusloo UC1 kuvatõmmis nr 3 – uue punkti loomine [12].

## 6. Kasutatud tehnoloogiad

PärimusRada halduskeskkonnas kasutatud tehnoloogiaid on kolme sorti – tagarakenduses kasutatud tehnoloogiad, mis vastutavad eelkõige andmebaasi ja kasutajaliidese vahel andmete vahetamise eest, kasutajaliidese tehnoloogiad ehk tehnoloogiad, mille abiga kasutajaliides valmis, ning andmebaasi loomiseks kasutatud tehnoloogiad.

### 6.1 Spring Boot 2.1

PärimusRada tagarakendus on kirjutatud Java Spring Boot raamistikus. Spring raamistik on Java platvorm, mis pakub Java rakenduste arendamiseks terviklikku infrastruktuurituge [14]. Spring Boot on Spring raamistiku laiendus, mis teeb mikroteenuste ja API-de kasutamise kergemaks ja kiiremaks [15]. Oma kerguse ja kiiruse tagab Spring Boot eelsätetud konfiguratsioonidega, mida on võimalik arendajal soovi korral muuta [15]. Näiteks autogeneraab Spring Boot POM faile, mida kasutatakse Maven tööriistaga [15]. Maven on aga omakorda Java projektihaldamise ning rakenduse ehitamise automatiseerimise tarkvaratööriist [16].

Tagarakendus võimaldab kasutajaliidesel suhelda andmebaasiga ja neil omavahel andmeid vahetada. Tagarakendus sai ulatuslikult üles ehitatud eelprojekti ajal ja selle autoriks on Age Roosi. Bakalaureusetöö autor lisas halduskeskkonna loomise käigus tagarakendusesse uute tabelitega suhtlemiseks vajalikke komponente, nagu näiteks *Model*, *DTO*, *Assembler*, *Controller*, *Repository* klasse, mis on vajalikud suhtlemaks tabelitega *language*, *task* ja *task\_answer*.

### 6.2 PostgreSQL 42.2.5

PostgreSQL on avatud lähtekoodiga relatsioonilise andmebaasi haldamise süsteem, mis tagastab tehtud päringutele vastuse JSON objektina [17]. Projekti andmebaas sai püsti juba eelprojekti valmimise käigus ja selle eest vastutas eelkõige Age Roosi. PärimusRada rakenduse tellija (Eesti Kirjandusmuuseum) kasutatud andmebaas põhineb samuti PostgreSQL tehnoloogial ning see sai ka andmebaasi valikul otsustavaks teguriks. Seda seetõttu, et projekt integreeritakse Eesti Kirjandusmuuseumi serverisse ning pannakse suhtlema tellija andmebaasiga, mis tõttu soovisid eelprojekti arendajad, et see üleminek oleks võimalikult lihtne. Halduskeskkonna loomise käigus kasutas autor juba eelprojekti ajal üles pandud andmebaasi, kuid täiendas seda mõne vajaliku tabeliga nagu näiteks *languages*, *task* ja *task\_answer*.

### 6.3 REST API

REST (*Representational State Transfer*) arhitektuurilaadi kirjeldas esmakordselt Roy Fielding oma 2000. aasta doktoritöös [18]. Veebiteenust, mis kasutab REST API-t, nimetatakse RESTful veebiteenuseks [18]. REST võimaldab luua erinevaid marsruute, millele päringuid sooritada. Näiteks võib marsruudiks olla „/api/trails/“ – seda nimetatakse ka ühtlasi üheks API otspunktiks (ingl *endpoint*). Otspunktile saab seejärel erinevaid päringuid esitada, nagu näiteks GET, POST, DELETE jms. Meeles tuleb pidada, et otspunkti vastuse saamiseks on vaja see otspunkt ja selle taga olev loogika (ehk mida selle otspunkti vastu tehtud päring teeb) tagarakenduses olevas *Controller* klassis eelnevalt ära defineerida.

## 6.4 Vue.js 2.5.17

Vue.js on progressiivne JavaScript-i raamistik kasutajaliideste loomiseks [19]. Vue.js-i projekti on lihtne üles seada, kasutades Vue veebipaketi malli (ingl *webpack*). Selleks on esmalt vajalik arvutisse tõmmata Vue.js käsurealiides `vue-cli` ning seejärel selle abil uue projekti baas genereerida [20]. Ka PärinusRada projekt (klientrakendus ja halduskeskkond) on üles ehitatud selle malli peale. Uue malli genereerimine käib järgmiselt:

```
$ npm install -g vue-cli
$ vue init webpack my-project
$ cd my-project
$ npm install
$ npm run dev
```

Vue.js kasutab veebilehe loomiseks komponente. Iga veebilehe alamleht on üks komponent ning selle alamlehe osad võivad samuti komponendid olla. Veebilehe osa on mõistlik komponendiks teha, kui seda kasutatakse rohkem kui ühes kohas, ja seejärel sellele muudes komponentides viidata. Lisaks Vue.js-ile kasutatakse PärinusRada projektis ka veel Vue.js-i laiendusi Vue-Router-it ja Vuex-i, millest räägitakse lähemalt järgmises peatükis.

## 6.5 OpenLayers 5.3.0

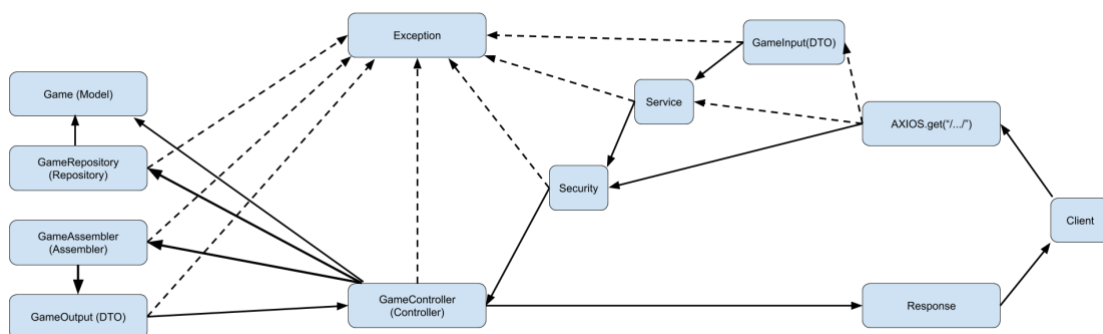
OpenLayers on JavaScript-i raamistik. Tegu on kaardimootoriga, millega saab luua nii staatilisi kui ka interaktiivseid kaarte. Näiteks on võimalik kaardile lisada vektorgraafikat nagu punkte, jooni, hulknurki jms, mis on seotud koordinaatteljestikuga [21]. Sellise funktsionaalsuse abil on võimalik näiteks mõõta läbitud maad või PärinusRada mägu näitel – koostada pärinuspaikadest koosnevaid radasid. Peale selle on võimalik kaardile lisada ka markereid, muuta aluskaarte, lisada kaardile kihte jmt [21].

## 7. Arhitektuur

Nagu eelmises peatükis mainisime, siis jaotub projekt kolme suuremasse rühma, milleks on tagarakendus, kasutajaliides ja andmebaas. Kasutajaliides on üles ehitatud Vue.js-i raamistikule, mille tööpõhimõtte seisneb rakenduse komponentideks jagamiseks. Tagarakenduseks on lihtne API (*Application Programming Interface*) ehk rakenduste programmeerimisliides, mis vahendab kasutajaliidese ja tagarakenduse vahel infot. Andmebaas kasutab relatsioonilist andmebaasi haldamise süsteemi, milleks on PostgreSQL. Nendest tehnoloogiatest ja lähenemistest räägimegi järgnevas alampeatükikes pikemalt.

### 7.1 Tagarakendus

Tagarakendus koosneb kaheksast erinevast osast, milleks on *Assembler*, *Controller*, *DTO*, *Exception*, *Model*, *Repository*, *Security* ja *Service* klassid. Joonisel 10 on kujutatud nende klasside omavahel suhtlemised *game* tabeli näitel.



Joonis 10. Tagarakenduse tööpõhimõtte tabeli *game* näitel.

Võtame näiteks olukorra, kus kasutajaliides tahab saada nimekirja kõikidest mängitud mängudest, mis PärinusRada andmebaasis olemas on. Selleks teeb kasutajaliides tagarakenduse API-le GET päringu. Kuna mängude saamiseks pole vaja päringusse midagi peale URL-i kaasa anda, siis selle päringu juures *GameInputDTO*-d ei kasutata. Lühidalt võtab *GameInputDTO* API päringule kaasa antud argumendi ja teisendab selle sobivale kujule. Antud juhul on selleks *ResourceSupport* alamklassi *GameInputDTO* objekt. Kui tegemist on sisselogimisega, siis toimub kasutajate sisselogimine läbi *Service*-i. Edasi võtab töö üle *Security* klassid, mis kontrollivad, kas sellist päringut on õigus teha ja kui päringuga on kaasa antud ka sisse loginud kasutajaga seotud identsustõend, siis kontrollitakse ka seda. Lisaks sellele tegelevad *Security* klassid ka näiteks parooli krüpteerimisega. Juhul kui *Security* klassides veateateid ei tulnud, siis liigutakse järgmisena *GameController*-i klassini, mis vastutab API URL laiendite loomise eest ja nende loogika eest. Näiteks kui tahame me API URL laiendit „/mängud/“, siis saame seda teha järgmiselt:

```
@GetMapping("/mängud/")
public @ResponseBody
List<GameOutputDTO> getPoints() {
    List<Game> games = gameRepository.findAll();
    return gametAssembler.toResources(games);
}
```

*GameController* pöördub *GameRepository* poolde, mis hoiustab kõiki punktide objekte ja küsib sellelt listi *GameOutputDTO* objektidest. Seejärel võtab *GameAssembler* selle nimekirja ja teisendab selle JSON objektiks. See on ühtlasi ka vastus mille kasutajaliides endale tagasi saab. Juhul kui mõnes etapis läheb midagi valesti, viskab *Exceptions* klass veateate.

## 7.2 Kasutajaliides

Kasutajaliides on ülesehitatud Vue.js-i raamistiku peal, mis on omakorda jaotatud komponentideks. Iga veebilehe alamleht on komponent, mis koosneb vajadusel veel teistest komponentidest. Vue.js-i komponentide loogikast räägime lähemalt aga juba järgmises lõigus.

Vue.js-i failid tunneb ära nende laienduse järgi, milleks on .vue. Vue.js on progressiivne JavaScript-i raamistik kasutajaliideste loomiseks, mis lööb veebilehe erinevad osad omavahel lahku [19]. Kusjuures need osad saavad endiselt üksteisega läbi *props* atribuudi või Vuex-i mälu suhelda. Vuex (3.0.1) on Vue.js-i laiendus, mille eesmärk on rakenduseüleselt hoiustada sellele ette antud olekuid. See tähendab seda, et Vuex ei salvesta lühiajaliselt kõike, mis rakenduses jooksvalt toimub, vaid seda, mida arendaja on Vuex-ile öelnud, et on vaja hoiustada. Tulles tagasi Vue.js-i tööpõhimõtte juurde, siis võib seda võrrelda objektorienteeritud programmeerimisega, mis on ka peamiseks põhjuseks, miks on kasutajaliides just nimelt Vue.js-i abil arendatud. Sarnaselt objektorienteeritud programmeerimise põhimõtetele kasutab Vue.js klassiobjektide asemel komponente, mida on võimalik taaskasutada ning millele on võimalik dünaamiliselt muutuvat sisendit ette anda. Näiteks on meil soov luua veebilehele tabel, mis kuvab tudengite nimekirja. Samal ajal oleks meil aga teisel alamlehel vaja tabelit, mis kuvaks õppejõudude nimekirja. Vue.js võimaldab selle lahendada kergelt ja lihtsalt, kus iga komponent algab JSX-i malliga. JSX meenutab oma olemuselt HTML-i, kuid tegelikult on tegemist JavaScript-i süntaksi laiendusega, mis võimaldab luua dünaamilist veebilehte. Vue.js-i JSX-i mallis kirjutatakse valmis komponendi kujundus ning toimub kasutajaliidese sidumine loogikaga. Nimelt suudab JSX mall suhelda otse komponendi skoobis olevate meetodite ja muutujatega. Lisaks mallile koosneb üks komponent CSS-ist, muutujatest ja meetoditest.

Kui komponendis olev CSS võib olla arendaja poolt märgitud kas komponendi skoobisiseks või rakenduseüleseks, siis muutujad ja kasutatud meetodid on ainult skoobisised. Seda välja arvatud juhul, kui neid ei talletata Vuex-i abil rakenduse mällu. Vue.js kasutab komponentides selle elutsüklis erinevaid etappe ja omadusi. Arendajal on võimalik läbi spetsiaalsete funktsioonide nende etappide poole pöörduda. Sellisteks etappideks on teiste seas näiteks `beforeCreate()`, `created()`, `mounted()`, `destroyed()` jmt. Oletame et meil on vaja tagarakenduses tehtud API-st saada kätte nimekirja tudengitest. Selleks kasutame me `mounted()` omadust, mille paremini mõistmiseks, räägime esmalt tingimuslikust renderdusest.

Vue.js võimaldab oma mallides kasutada elementide ja klasside tingimuslikku omistamist. Selleks on Vue.js oma JSX-i mallidesse lisanud võimalused nagu näiteks `v-if`, `v-else-if`, `v-else`, `:class`, `v-for`, `v-model` jpm. Need atribuudid jälgivad jooksvalt neile ette antud väärtuseid. Näiteks kui meil on element `<button/>`, mille atribuudiks on `v-if`kasNuppuVajutati`` ning meil on defineeritud muutuja `kasNuppuVajutati`, mille esialgne väärtus on väär, siis antud elementi ei laeta rakendusse. Kui aga mingis muus elemendis või meetodis muudetakse selle muutuja väärtus tõseks, siis laetakse kogu komponent uuesti ja

seekord kuvatakse ka element `<button/>`. Sellega seoses kutsutakse `created()` omadus välja ainult komponendi esmasel laadimisel, kuid `mounted()` omadust kutsutakse välja iga kord kui Vue kasutajaliidese uuesti genereerib. Arvestades, et element `<button/>` väljendab tudengite nimekirja uuesti laadimist, siis saamegi panna `mounted()` omadusesse päringu API-le, mis küsib igal laadimiskorral tudengite nimekirja uuesti.

Komponentidevahelise mugavaks liikumiseks kasutab kasutajaliides VueRouter-it (3.0.1), mis on taaskord JavaScript-i Vue.js-i raamistiku laiendus. VueRouter aitab arendajal defineerida komponendile URL laiendusi ning neid veebilehel olevates linkides ära kasutada. Näiteks kui meil on komponent KKK ehk korduma kippuvad küsimused, millele tahame seada URL laienduseks `/kkk`, siis VueRouter-ile kuuluvas `index.js` failis peaksime selle defineerima järgmiselt:

```
Vue.use(Router);
export default new Router({
  routes: [
    {
      path: '/kkk',
      name: 'KKK',
      component: KKK,
    },
  ],
});
```

Peale teekonna defineerimist saame seda mis tahes komponendis ära kasutada nii:

```
<router-link class="routerLink" :to="{ name : 'KKK'}">KKK</router-link>
```

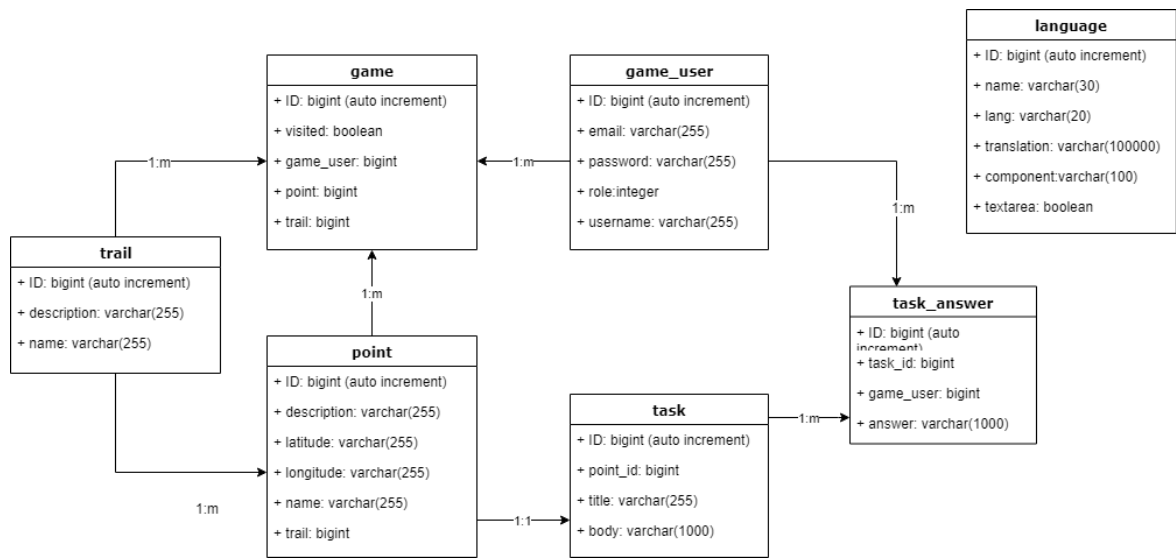
Pelgalt Vue.js raamistikust PärimusRada halduskeskkonna loomiseks siiski ei piisa. PärimusRada halduskeskkond vajab radade koostamise jaoks OpenLayers kaardimootori abi. OpenLayers on raamistik dünaamiliste kaartide koostamiseks [21]. OpenLayers kaardimootor sai teiste seast (Google Maps, Leaflet) valitud, kuna see pakub rohkem võimalusi kaardi modifitseerimiseks. Näiteks on sellel parem vektoritugi, mida on vaja just nimelt punktide koostamiseks ning teatavasti põhineb PärimusRada mäng just nimelt radadest ja need omakorda punktidest.

### 7.3 Andmebaas

PärimusRada projekti andmebaasis on kokku seitse tabelit, milleks on *game*, *game\_user*, *language*, *point*, *trail*, *task* ja *task\_answer*. Käesoleva bakalaureusetöö raames on andmebaasi lisatud tabelid *language*, *task* ja *task\_answer*, ülejäänud tabeleid küll kasutatakse halduskeskkonnas, kuid nende autor on Age Roosi.

Joonisel 11 on kujutatud PärimusRada projekti andmebaasi struktuuri. Nagu mainitud, siis bakalaureusetöö autor lisas andmebaasi kolm tabelit, milleks on *language*, *task* ja *task\_answer* ning pani need tabelid teiste tabelitega sõltuvusse.

Sõltuvused on tabelis märgitud lühenditega nagu „1:1“ ehk ühele tabeli A elemendile vastab tabelis B täpselt 1 element ning „1:m“ (ingl *one-to-many*) ehk ühele tabeli A elemendile vastab tabelis B rohkem kui üks elementi. Taoliseid sõltuvusi on tegelikult enam, kuid PärimusRada andmebaasis on kasutusel vaid need kaks.

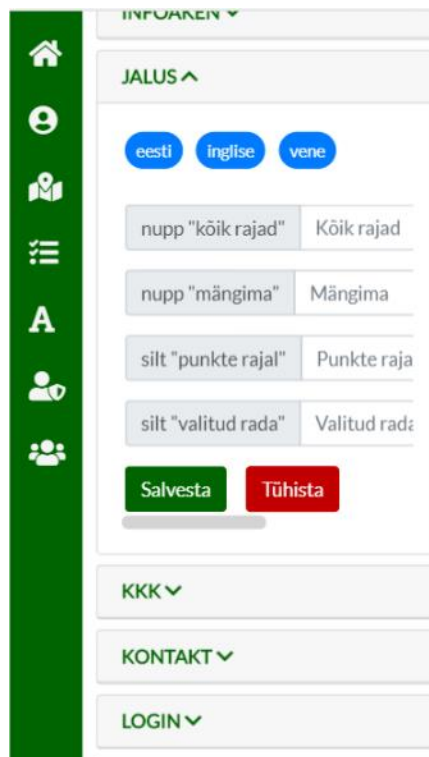


Joonis 11. Andmebaasi struktuur.

## 8. Töö tulem ja hinnang

Käesoleva bakalaureusetöö tulemiks on toimiv halduskeskkond. Kaetud said kõik vajalikud funktsionaalsused nagu radade, punktide, ülesannete, sisu, tõlgete haldus ja ülesannete eksportimise võimalus. Lisaks sellele ka mängijate nimekirja kuvamine ja administraatorite haldus.

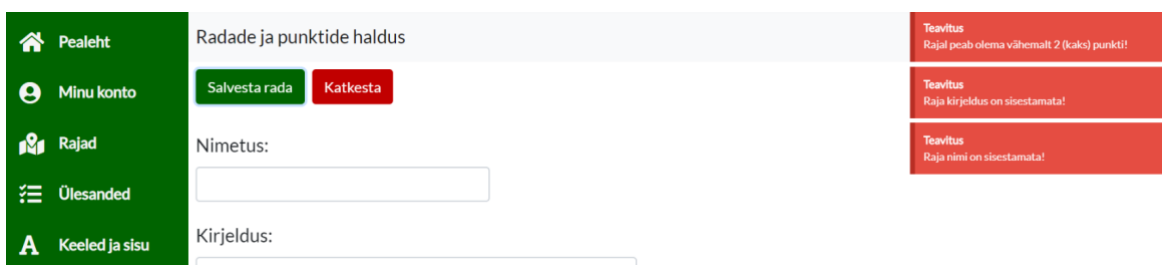
Joonisel 12 oleval Keele- ja sisuhalduse alamleheküljest tehtud kuvatõmmisel on näha väiksemate seadete jaoks tehtud menüüd, millel on ruumi kokkuhoiu eesmärgil nappudena ainult ikoonid. Peale selle on näha ka Keele- ja sisuhalduse alamlehe peamist loogikat, mis on lahendatud Bootstrap-i kaartidega (ingl *Card*). Iga PärimusRada mängu alamleht või komponent on eraldi kaart ja korruga saab avatud olla vaid üks kaart, mis tuleneb Bootstrap-i eelsätetestatud loogikast. Näiteks on eraldi kaart KKK, mis on alamleht, ja samuti ka infoaken (ingl *tooltip*), mis on pisike aknake, mis ilmub kui kasutaja klõpsab kaardil punktile. Joonisel on näha ka sinised keelevahetusenupud. Siinkohal tasub märkida, et „Salvesta“ nuppu vajutades salvestatakse ainult hetkel avatud kaart ja selles kaardis valitud keel. Ehk kui meil on avatud kaart JALUS ning inglise keel, siis salvestatakse ainult JALUS-e komponendis tehtud inglise keele tõlged. Suurem kuvatõmmis samast vaatest on joonisel 16, mis asub teises lisas (II. Kasutuslood). Mainitud joonisel on näha ka keele lisamise nuppu, mida saab teha rakenduseüleselt. See tähendab seda, et kui lisatakse uus keel, siis tehakse automaatselt andmebaasi koopiad kõikidest sisutekstidest. Nendele sisutekstidele määratakse keeleks sisestatud uus keel, kuid esialgu on sisutekstide väärtused eesti keeles, et vältida olukorda, kus mängija vahetab rakenduses keelt ja mõni sisutekst kaob sootuks ära. Taolisel kaardiloogikal on üles ehitatud ka Ülesannete alamleht.



Joonis 12. Keele- ja sisuhalduse alamlehe kuvatõmmis mobiilivaates.

Peatükis Nõuded on välja toodud rakenduse üks olulisim funktsionaalsus, milleks on punkti asukoha valimine kaardilt. Samas peatükis on olemas ka joonised 7, 8 ja 9, mis seda illustreerivad. Esmalt navigeerib kasutaja Radade halduse alamlehele (joonis 7), kus on võimalus vaadata radade nimekirja, kustutada radasid, muuta radasid ja nende punkte või lisada uus rada. Uue raja lisamisel peab kasutaja sisestama raja nime ja kirjelduse ning lisama vähemalt kaks punkti. Kahe punkti reegel tuleneb asjaolust, et rada, mis koosneb ühest punktist ei ole kuigi mõistlik. Klõpsates punkti muutmise nupul avaneb uus vaade (joonis 9), kus lehe allosas on kaart markeriga, mille asukoht võetakse esialgu andmebaasist. Punkti asukoha muutmiseks tuleb kasutajal lihtsalt mingile punktile kaardil klõpsata. Kui kõik väljad on täidetud ja asukoht määratud, saab kasutaja punkti salvestada.

Joonisel 13 on kujutatud kasutajale saadetud teavitusi, mida on kasutatud punktide ja radade salvestamisel ning ka Keele- sisuhaldus alamlehel. Teavitused annavad kasutajale märku, kas salvestamine oli edukas, ja kui ei, siis aitavad kasutajal vigadele tähelepanu pöörata.



The screenshot shows a web interface for managing routes and points. On the left is a green sidebar with navigation options: Pealeht, Minu konto, Rajad, Ülesanded, and Keeled ja sisu. The main content area is titled 'Radade ja punktide haldus' and contains a form with a 'Salvesta rada' button and a 'Katkesta' button. The form has two input fields: 'Nimetus:' and 'Kirjeldus:'. On the right side, there are three red error messages:

- Teavitus: Rajal peab olema vähemalt 2 (kaks) punkti!
- Teavitus: Raja kirjeldus on sisestamata!
- Teavitus: Raja nimi on sisestamata!

Joonis 13. Teavituste kuvatõmmis.

Nagu üks tellija nõuetest ette nägi, siis pidi halduskeskkond olema võimalikult kasutajasõbralik ja intuitiivne. Isegi kui need ei ole mõõdetavad ühikud, siis saab rakenduse tellija valideerida, kas neid nõudeid on täidetud või ei. Bakalaureusetöona valminud halduskeskkonna tellija sõnul on loodud lahendus kena ja arusaadav. Eriti toodi välja Keele- ja sisuhaldus alamlehekülg, kus tellijale meeldis enim keelte lisamise ja vahetamise süsteem ning võimalus kõiki tekste (sh ka nuppude silte) ise muuta.

Bakalaureusetöö autorina leian, et tulemus on hea, kuid samas olen saanud kogemusest veendunud, et kui on võimalik kasutada juba olemasolevaid platvorme, siis suure ajakulu tõttu erilahendust mõistlik teha ei ole. Siinkohal mainiksin ära, et taoliste erilahenduste kasutamine on juba hoopis teine jutt, sest PärimusRadade halduskeskkondade näitel võin öelda, et nende kasutamine on mugavam kui näiteks Drupal sisuhaldussüsteemi kasutamine. Seega võib öelda, et üldiselt olen siiski erilahenduste pooldaja, kuid kindlasti ei tasu iga veebilehe uut halduskeskkonda looma hakata.

## 9. Kokkuvõte ja tulevik

Käesoleva bakalaureusetöö eesmärgiks oli luua PärimusRada mängule halduskeskkond, mis võimaldab radade loomist ja muutmist, ülesannete vastuste nägemist ja eksportimist, keelevahetuse võimaldamist, mängijate nimekirja nägemist ning administraatorite haldamist. Töö eesmärk sai täidetud ning vastab eeltoodud tellija nõuetele. Lisaks mainitule saab radade loomisel punktide asukohta määrata interaktiivsel kaardil. See tähendab seda, et administraatoril on võimalik rajale punkte lisada klõpsates selle jaoks halduskeskkonnas oleval kaardil. Peale selle lisas autor rakendusele e-maili saatmise võimaluse nii mängijatele kui ka administraatoritele ja mängijate kustutamise võimaluse.

PärimusRada mängul on tulevikus tugev potentsiaal arendada nii Eesti rahva kui ka turistide teadmisi Eesti pärimuspaikade kohta, kelle huviks on aktiivne puhkus ja kultuur. Projekti tellija, Eesti Kirjandusmuuseumi Eesti Rahvaluule Arhiiv, plaanib peale projekti täielikku valmimist seda turustama hakata ning läbi mängijate sisendite Eesti Rahvaluule arhiivimaterjali täiendada, et pärimuskogule pealekasvu tekitada. Peale bakalaureusetöö käigus valminud halduskeskkonna, on plaanis PärimusRada mängu ka veel näiteks sotsiaalmeediaga sisse logimine lisada ja arendatakse edasi ülesannete formaatide võimalusi. Näiteks on tellijal plaanis lisada ka pilt- ja heliküsimusi ning samuti ka pilt- ja helivastuseid.

## 10. Viidatud kirjandus

- [1] A. R. K. T. Hanna Tagen, "GitLab," 15 september 2018. [Online]. Available: <https://gitlab.com/htagen/sadarada/wikis/Projekti-visioon>. [Accessed 20 märts 2019].
- [2] "e-keelenõu," [Online]. Available: <http://kn.eki.ee/?Q=p%C3%A4rimus>. [Accessed 4 mai 2019].
- [3] "EESTI KOHAPÄRIMUSE KESKUS," [Online]. Available: <http://www.ekoke.ee/kursused/tutvustus>. [Accessed 4 mai 2019].
- [4] J. Darrel, "WP Plugins," 17 aprill 2018. [Online]. Available: <https://wpplugins.tips/wordpress-security-statistics/>. [Accessed 12 märts 2019].
- [5] P. Czech, "New Possibilities Group," [Online]. Available: <https://www.npgroup.net/blog/5-reasons-to-choose-a-custom-cms-over-wordpress/>. [Accessed 13 märts 2019].
- [6] R. Kikas, "Analysis of issue and dependency management in open-source software projects," Tartu, 2018.
- [7] R. Mening, "WebsiteSetup," 1 juuni 2018. [Online]. Available: <https://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>. [Accessed 30 aprill 2019].
- [8] T. Canavan, CMS Security Handbook: The Comprehensive Guide for WordPress, Joomla, Drupal and Plone, 2011.
- [9] K. T. A. R. Hanna Täge, "GitLab," 2019. [Online]. Available: <https://gitlab.com/htagen/sadarada/wikis/Mittefunktsionaalsed-n%C3%B5uded>. [Accessed 1 mai 2019].
- [10] M. Rehkopf, "Atlassian," [Online]. Available: <https://www.atlassian.com/agile/project-management/user-stories>. [Accessed 14 märts 2019].
- [11] K. Bittner, 2002. [Online]. Available: <https://dl.acm.org/citation.cfm?id=557126>. [Accessed 17 märts 2019].
- [12] H. Tagen, A. Roosi and K. Talts, detsember 2018. [Online]. Available: [https://drive.google.com/file/d/13okaOYmPePZyU--wD\\_eMPohU3kzXC/view](https://drive.google.com/file/d/13okaOYmPePZyU--wD_eMPohU3kzXC/view). [Accessed 17 märts 2019].
- [13] M. Poppendieck and T. Poppendieck, "Options Thinking in Software Development," in *Lean Software Development: An Agile Toolkit: An Agile Toolkit*, 2003, p. 56.
- [14] "Spring," [Online]. Available: <https://docs.spring.io/spring/docs/3.0.x/spring-framework-reference/html/overview.html>. [Accessed 26 aprill 2019].
- [15] R. RV, "Using Spring Boot to build RESTful microservices," in *Spring Microservices*, Birmingham, UK, Packt Publishing, 2016, p. 56.
- [16] F. P. Miller, A. F. Vandome and J. McBrewster, Apache Maven, Alpha Press, 2010.
- [17] H.-M. Jaakson, "Toitumisnõustaja veebirakendus," Tartu, 2018.
- [18] M. Masse, REST, 2011, p. 5.
- [19] "Vue.js," [Online]. Available: <https://vuejs.org/v2/guide/>. [Accessed 30 aprill 2019].
- [20] "GitHub," [Online]. Available: <https://github.com/vuejs-templates/webpack>.

- [21] "OpenLayers," [Online]. Available: <https://openlayers.org/>. [Accessed 26 april 2019].
- [22] A. R. K. T. Hanna Tagen, "GitLab," 9 desember 2018. [Online]. Available: <https://gitlab.com/htagen/sadarada/wikis/Funktsionaalsed-n%C3%B5uded>. [Accessed 17 märts 2019].

# Lisad

## I. Kasutajalood

### 1. Radade haldamine

- 1.1. Administraatorina tahan lisada uusi radu, et muuta olemasolevate radade listi mitmekülgsemaks [22].
- 1.2. Administraatorina tahan muuta olemasolevaid radu ja nende punkte, et ma ei peaks iga kord uut rada tegema.
- 1.3. Administraatorina tahan kustutada radu, et andmebaas ei sisaldaks mittevajalikke radu.
- 1.4. Administraatorina tahan võimalust hallata radu (sh peita), et saaksin raja vajadusel mängust ajutiselt eemaldada.
- 1.5. Administraatorina tahan punkti loomisel valida punkti asukoht kaardilt, et punktide loomise protsessi kiiremaks teha.
- 1.6. Administraatorina tahan lisada radadele märksõnu, et külastajad saaksid radade nimekirjas radasid märksõnade järgi filtreerida [22].

### 2. Ülesannete haldamine

- 2.1. Administraatorina tahan lisada punktidele ülesandeid, et saada mängijatelt arhiivimaterjali tarbeks sisendit [22].
- 2.2. Administraatorina tahan muuta olemasolevaid ülesandeid, et ma ei peaks iga kord uut ülesannet looma.
- 2.3. Administraatorina tahan kustutada olemasolevaid ülesandeid, kui punkti kohta käiv ülesanne muutub üleliigseks.
- 2.4. Administraatorina tahan ülesannete lahendusi ekspordida, et saaksin arhiivimaterjali täiendada.

### 3. Sisutekstide haldamine

- 3.1. Administraatorina tahan muuta rakenduses kuvatavaid tekste, et need oleksid alati ajakohased.

### 4. Keelte haldamine

- 4.1. Administraatorina tahan lisada keelevelikuid (tõlkeid), et rohkem inimesi saaks mängu mängida.
- 4.2. Administraatorina tahan muuta tõlkeid, et ma ei peaks iga kord uut keelevelikut tegema.
- 4.3. Administraatorina tahan kustutada keelevelikuid, et rakendus ei kuvaks ebavajalikke keelevelikuid.
- 4.4. Administraatorina tahan peita keeleveliku kuvamist rakenduses, et saaksin keeleveliku ajutiselt rakendusest eemaldada.

### 5. Kasutajate nimekiri

- 5.1. Administraatorina tahan näha mängijate nimekirja, et mul oleks ülevaade rakenduse kasutajatest ja nende arvust [22].

## **6. Administraatorite konto haldamine**

- 6.1. Administraatorina tahan näha enda konto andmeid, et näha, mis andmetega ma registreerunud olen.
- 6.2. Administraatorina tahan muuta enda kontoga seonduvaid andmeid, et need oleksid alati ajakohased.
- 6.3. Administraatorina tahan kustutada oma konto, et eemaldada soovi korral enda seos PärinusRada mänguga.

## **7. Administraatorite haldamine**

- 7.1. Superadministraatorina tahan ma lisada administraatoreid, et rakenduse populaarsemaks muutumisel töökoormust jaotada [22].
- 7.2. Superadministraatorina tahan näha nimekirja olemasolevatest administraatoritest, et näha halduritest ülevaadet.
- 7.3. Superadministraatorina tahan muuta administraatorite õigusi (nt luua või eemaldada superadministraatoreid), et administraatorite õigused oleksid ajakohased.
- 7.4. Superadministraatorina tahan kustutada administraatorite kontosid, et vajadusel eemaldada konto omanikuga seos PärinusRada mänguga.

## II. Kasutuslood

Tabel 2. Kasutuslugu UC2 [12].

|                               |  |
|-------------------------------|--|
| <b>Kasutuslugu 2:<br/>UC2</b> | <b>Uute administraatorite lisamine</b>   |
| Seotud nõuded:                | 7.1; 7.3   |
| Eeltingimus:                  | Superadministraator on halduskeskkonda sisse loginud.  |
| Peamine kasutaja:             | Superadministraator  |
| Eesmärk:                      | Lisada uus administraator või superadministraator.   |
| Peamine stsenaarium:          | <ol style="list-style-type: none"><li>1. Superadministraator vajutab nupule “Lisa admin”;</li><li>2. superadministraator täidab ära vajalikud väljad;<ol style="list-style-type: none"><li>a. superadministraator valib, kas lisatav kasutaja on administraator või superadministraator;<ol style="list-style-type: none"><li>i. vaikumisi on valitud ADMIN kasutaja;</li></ol></li><li>b. superadministraator lisab administraatori nime;<ol style="list-style-type: none"><li>i. kui pole nime lisanud, siis saab teavituse, et nime peab lisama;</li></ol></li><li>c. superadministraator lisab administraatorile meili;<ol style="list-style-type: none"><li>i. kui meili pole lisatud, siis saab teavituse, et meili peab lisama;</li></ol></li></ol></li></ol> <p>Superadministraator vajutab nuppu “Saada kutse”.</p> |
| Alternatiivne stsenaarium:    | <ol style="list-style-type: none"><li>1. Kui superadministraator soovib vastuste vaatamist katkestada, siis vajutab nuppu “Katkesta”;</li><li>2. superadministraator navigeerib halduskeskkonna alamlehel minema.</li></ol>  |
| Järeldus:                     | Sisestatud meiliaadressi omanikule saadetakse kutse. Kutses on link konto loomise lehele, kus saab parooli sisestada. Kui kasutaja on parooli lisanud ning vajutanud nupule „Registreeru“, on lisatud uus administraator andmebaasi ja lisatud administraator pääseb halduskeskkonnale ligi.   |

Adminite haldus

Saada kiri Kustuta Lisa admin

Otsi...

| ID | Nimi ↓     | Roll ↓ | Email      |                          |
|----|------------|--------|------------|--------------------------|
| 1  | [redacted] | ADMIN  | [redacted] | <input type="checkbox"/> |
| 2  | [redacted] | ADMIN  | [redacted] | <input type="checkbox"/> |

Vali kõik

Joonis 14. Kasutusloo UC2 kuvatõmmis nr 1 – administraatorite nimekiri, milles olevad andmed on isikuandmete kaitseks mustaks värvitud [12].

Adminite haldus

Katkesta

**Lisa uus admin**

Uue adminkasutaja lisamiseks sisesta uue konto omaniku nimi, email, vali konto roll ning vajuta nuppu "Saada kutse".

Sisestatud meiliaadressile saadetakse link, läbi mille saab kutsutud isik endale konto teha.

Nimi:

Email:

Roll:

ADMIN

SUPERADMIN

Saada kutse

Joonis 15. Kasutusloo UC2 kuvatõmmis nr 2 – uue administraatori lisamine [12].

Tabel 3. Kasutuslugu UC3 [12].

|                               |   |
|-------------------------------|---|
| <b>Kasutuslugu 3:<br/>UC3</b> | <b>Veebirakenduse sisu muutmine</b>   |
| Seotud nõuded:                | 3.1; 4.1; 4.2; 4.3; 4.4   |
| Eeltingimus:                  | Administraator on halduskeskkonda sisse loginud.  |
| Peamine kasutaja:             | Administraator  |
| Eesmärk:                      | Muuta veebirakenduse sisu, et veebirakenduses kuvatav sisutekst oleks alati ajakohane.  |
| Peamine stsenaarium:          | <ol style="list-style-type: none"> <li>1. Administraator vajutab nupule “Keele- ja sisuhaldus”;</li> <li>2. administraator täidab ära vajalikud väljad; <ol style="list-style-type: none"> <li>a. administraator avab rakenduse komponendi „Footer“;</li> <li>b. administraator muudab välja „nupp „Kõik rajad““;</li> </ol> </li> <li>3. administraator vajutab nuppu “Salvesta”.</li> </ol> |
| Alternatiivne stsenaarium:    | <ol style="list-style-type: none"> <li>1. Kui administraator soovib vastuste vaatamist katkestada, siis vajutab nuppu “Tühista”;</li> <li>2. administraator navigeerib halduskeskkonna alamlehel minema.</li> </ol>   |
| Järeelseisund:                | Andmebaasi on sisestatud uus väärtus ja veebirakenduse sisu on uuendatud.   |

**Keele- ja sisuhaldus**

- Keeleelementide nägemiseks ja muutmiseks vali esmalt sobiv komponent.
- Tekstiväljade suurus, mis ulatuvad üle mitme rea, on võimalik paremalt alt nurgast muuta.
- Lisades uut keelt salvestatakse andmebaasi esmalt kõik uue keele keeleelementide tõlkeväärtused vaikekeele väärtusega. Antud juhul on selleks eesti keel. See on oluline, et vältida rakenduses (näiteks nuppude tekstides) tühjasid väärtuseid.
- Keelt vahetades, salvesta esmalt sisestatud info.

**Lisa keel +**  ✓

**FOOTER ^**

**eesti** **inglise** **vene**

nupp "kõik rajad" Kõik rajad

nupp "mängima" Mängima

silt "punkte rajal" Punkte rajal

silt "valitud rada" Valitud rada

**Salvesta** **Tühista**

**INFOAKEN** ▾

**KKK** ▾

**KONTAKT** ▾

**LOGIN** ▾

**PROFIIL** ▾

**PROFIILI SEADED** ▾

Joonis 16. Kasutusloo UC3 kuvatõmmis – keele- ja sisuhaldus [12].

Tabel 4. Kasutuslugu UC4.

|                               |   |
|-------------------------------|---|
| <b>Kasutuslugu 4:<br/>UC4</b> | <b>Ülesannete lahenduste eksportimine</b>   |
| Seotud nõuded:                | 2.6   |
| Eeltingimus:                  | Administraator on halduskeskkonda sisse loginud.  |
| Peamine kasutaja:             | Administraator  |
| Eesmärk:                      | Eksportida inimeste sisestatud ülesannete lahendusi/vastuseid.  |
| Peamine stsenaarium:          | <ol style="list-style-type: none"> <li>1. Administraator vajutab nupule “Ekspordi vastused”</li> <li>2. administraator täidab ära vajalikud väljad; <ol style="list-style-type: none"> <li>a. administraator valib raja; <ol style="list-style-type: none"> <li>i. kui rada pole valitud, siis saab teate, et peab valima;</li> </ol> </li> </ol> </li> <li>3. administraator vajutab nuppu “Laadi alla”;</li> <li>4. administraator näeb kindla raja kindla punkti ülesannetele vastatud vastuseid.</li> </ol> |
| Alternatiivne stsenaarium:    | <ol style="list-style-type: none"> <li>1. Kui administraator soovib vastuste eksportimist katkestada, siis vajutab nuppu “Katkesta”;</li> <li>2. administraator navigeerib halduskeskkonna alamlehel minema.</li> </ol>   |
| Järeleisund:                  | Administraator on saanud tutvuda mingi kindla raja vastustega ja saab need talletada arhiivi.   |

Tabel 5. Kasutuslugu UC5 [12].

|                               |   |
|-------------------------------|---|
| <b>Kasutuslugu 5:<br/>UC5</b> | <b>Kasutajate nimekirja vaatamine</b>   |
| Seotud nõuded:                | 5.1   |
| Eeltingimus:                  | Administraator on halduskeskkonda sisse loginud.  |
| Peamine kasutaja:             | Administraator  |
| Eesmärk:                      | Vaadata kasutajate nimekirja  |
| Peamine stsenaarium:          | <ol style="list-style-type: none"> <li>Administraator vajutab nupule “Mängijad”;</li> <li>administraator näeb kasutajate arvu ja nimesid;</li> <li>administraatoril on lisaks võimalus mängijatele saata meil;</li> <li>administraatoril on lisaks võimalus mängijaid kustutada.</li> </ol> |
| Alternatiivne stsenaarium:    | Administraator navigeerib alamleheküljelt minema.   |
| Järeelseisund:                | Administraator näeb kasutajate nimekirja, registreerunud kasutajate arvu ja nende kasutajanimedid.  |

| ID | Kasutajanimi ↓ | Email      |                                     |
|----|----------------|------------|-------------------------------------|
| 1  | [Redacted]     | [Redacted] | <input type="checkbox"/>            |
| 2  | [Redacted]     | [Redacted] | <input checked="" type="checkbox"/> |
| 3  | [Redacted]     | [Redacted] | <input type="checkbox"/>            |
| 4  | [Redacted]     | [Redacted] | <input type="checkbox"/>            |
| 5  | [Redacted]     | [Redacted] | <input type="checkbox"/>            |
| 6  | [Redacted]     | [Redacted] | <input type="checkbox"/>            |
| 7  | [Redacted]     | [Redacted] | <input type="checkbox"/>            |
| 8  | [Redacted]     | [Redacted] | <input type="checkbox"/>            |
| 9  | [Redacted]     | [Redacted] | <input type="checkbox"/>            |

Joonis 17. Kasutuslugu UC5 kuvatõmmis – mängijate nimekiri, milles olevad andmed on isikuandmete kaitseks mustaks värvitud [12].

### III. Litsents

#### Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Hanna Tagen**,

*(autori nimi)*

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose  
**Halduskeskkonna erilahendus veebirakendusele PärimusRada**,  
*(lõputöö pealkiri)*  
mille juhendajateks on Siim Karus ja Mari Sarv,  
*(juhendajate nimed)*  
reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

*Hanna Tagen*

**07.05.2019**