

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Karl Henri Kaljus 164663IABB

**ÜLIÕPILASORGANISATSIOONI
BEST-ESTONIA PUNKTISÜSTEEMI
TARKVARA UUE VERSIOONI
ARENDAMINE**

bakalaureusetöö

Juhendaja: Erki Eessaar
PhD

Tallinn 2021

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: Karl Henri Kaljus

31.12.2020

Annotatsioon

Käesoleva töö eesmärgiks on kavandada ja realiseerida üliõpilasorganisatsioon BEST-Estoniale punktide haldamise tarkvara uus versioon, milles andmed on salvestatud SQL-andmebaasisüsteemi abil loodud andmebaasi ja kasutajal on võimalik sinna läbi veebirakenduse uusi andmeid sisestada.

Töö kirjeldab olemasoleva tarkvara ülesehitust ja uue lahenduse otsimise põhjuseid. Seejärel analüüsitakse detailselt nõudeid, valitakse sobiv andmebaasisüsteem ning luuakse andmebaas ja veebirakendus andmete sisestamiseks.

Töö tulemusena on valminud andmebaas ja veebipõhine andmebaasirakendus, mille lähtekood anti BEST-Estoniale üle. Töö autor soovib valminud tarkvara edasi arendada, et saavutada kõik punktihalduse süsteemi eesmärgid.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 54 leheküljel, 7 peatükki, 22 joonist, 4 tabelit.

Abstract

Developing a New Version of the Student Organization BEST-Estonia Point Management Software

The purpose of this bachelor thesis is to design and develop a new version of the student organization BEST-Estonia point management software, in which data is being stored in an SQL database and where the system's users can enter new data through a web-based interface. Point management is a part of the wider BEST-Estonia's information system and the software implements its points management functional subsystem and register.

The thesis starts with a description of the current state of the existing software version and problems that led to the development of the new version. This is followed by the specification of requirements. The work specifies functional and non-functional requirements as well as requirements to data. After that the selection of the most suitable database management system and development tools takes place. The final step involves the creation of the database and a partial web application. The system is modeled in UML, the database is implemented in MySQL, and the web application is created in PHP and JavaScript.

As a result of this bachelor thesis the source code of the resulting software version has been handed over to student organization BEST-Estonia. Unfortunately, not all of the planned functionality was implemented at the time of submitting the thesis. The author of the thesis recommends continuing the development of the resulting software to meet all the objectives of the point management system.

BEST-Estonia has not started to use the software at the time of the thesis submission. However, a plan is to move forward with it as quickly as possible. A prerequisite of starting the use of the new system version is migration of data from the old (file-based) database to the new database and associating the web page that displays reports with the new database.

The work included collaboration with Stella Leego, who in her bachelor thesis identified the subsystems of BEST-Estonia's information system and specified the membership management part of the system in a detailed manner. This was an important input to the current work because in terms of database the points are associated with a membership and through this with a person.

The thesis is in Estonian and contains 54 pages of text, 7 chapters, 22 figures, 4 tables.

Lühendite ja mõistete sõnastik

| | |
|---------------|--|
| Ajax | Tehnoloogia, mis võimaldab teha asünkroonselt HTTP päringuid. |
| API | <i>Application Programming Interface</i> , rakendusliides. |
| BEST | <i>Boards of European Students of Technology</i> , Euroopa Tehnikatudengite Kogu. |
| BEST-Estonia | Euroopa Tehnikatudengite Kogu Eesti haru. |
| CASE | <i>Computer Aided System Engineering</i> , tarkvara mida kasutatakse tarkvara- ja infosüsteemide projekteerimiseks ja mis võib ka koodi genereerimise kaudu aidata kaasa realiseerimisele. |
| CSS | <i>Cascading Style Sheets</i> , veebilehe disaini kirjeldav keel. |
| DMTG | Disaini ja turundusega tegelev töögrupp BEST-Estonias. |
| EA | <i>Enterprise Architect</i> , CASE tarkvara. |
| FRTG | Sponsorlusega tegelev töögrupp BEST-Estonias. |
| Google Sheets | Pilvepõhine tabelarvutusprogramm. |
| HTML | <i>Hypertext Markup Language</i> , hüpertekst-märgistuskeel, veebidokumentide loomiseks mõeldud tekstivorming. |
| HTTPS | <i>Hypertext Transfer Protocol Secure</i> , turvaline hüperteksti edastamise protokoll. |
| JavaScript | <i>JavaScript</i> , vVeebirakenduste skriptides kasutatav programmeerimise keel. |
| jQuery | <i>JavaScripti</i> andmeteek, HTMLi ja <i>JavaScripti</i> kirjutamise lihtsustamiseks mõeldud rakendusliides |
| MTÜ | Mittetulundusühing. |
| MySQL | Avatud lähtekoodiga relatsiooniliste andmebaaside haldussüsteem, mis tugineb <i>Structured Query Language</i> kasutamisel. |
| PHP | <i>Hypertext Preprocessor</i> , skriptimiskeel. |
| POST-päring | HTTP meetod, mis saadab andmeid kliendilt serverile. |
| RV | Rahvusvahelise BESTiga tegelev töögrupp. |
| SQL | <i>Structured Query Language</i> . Standardiseeritud keel, mis sisaldab nii andmekäitluse, andmekirjelduse, tehingute juhtimise kui |

õiguste haldamise lauseid. Keel realiseerib relatsioonilist andmemudelit ja on kasutusel SQL-andmebaasisüsteemides.

TTG

Tehnikaga tegelev töögrupp BEST-Estonias.

UML

Unified Modeling Language, ühtne (unifitseeritud) modelleerimiskeel, mis võimaldab süsteeme kirjeldada visuaalselt ja mida saab kasutada paljude erinevate valdkondade modelleerimiseks.

XAMPP

Kõigile huvilistele kasutamiseks avaldatud vabavara versioon, mis sisaldab Apache, MySQL, PHP ja Perl tarkvara ning on mõeldud nendel põhinevate süsteemide arendamiseks.

Sisukord

| | |
|--|----|
| 1 Sissejuhatus | 12 |
| 1.1 Üldine taust ja töö lühikirjeldus..... | 12 |
| 1.2 Probleem ja töö eesmärk..... | 12 |
| 1.3 Töö edasine struktuur | 13 |
| 2 Tarkvara praegune olukord..... | 14 |
| 2.1 Punktide arvestus teiste riikide BESTi allorganisatsioonides | 19 |
| 3 Süsteemianalüüs | 20 |
| 3.1 Punktide saamiste funktsionaalne allsüsteem..... | 20 |
| 3.1.1 Eesmärgid | 20 |
| 3.1.2 Funktsionaalsed nõuded | 21 |
| 3.1.3 Mittefunktsionaalsed nõuded..... | 22 |
| 3.2 Punktide saamiste register | 23 |
| 3.2.1 Seosed teiste registritega | 23 |
| 3.2.2 Punkti saamise seisundidiagramm..... | 29 |
| 4 Disain..... | 31 |
| 4.1 Vahendite valik..... | 31 |
| 4.2 Kasutajaliides..... | 32 |
| 4.2.1 Personalijuhi vaade..... | 32 |
| 4.3 Rakendus | 39 |
| 4.4 Andmebaas | 42 |
| 5 Arendusprotsess..... | 45 |
| 5.1 Metoodika..... | 45 |
| 5.2 Tööriistade kirjeldus | 45 |
| 5.3 Tagasivaade arendusele | 46 |
| 6 Arendusvaade | 47 |
| 7 Kokkuvõte | 49 |
| Kasutatud kirjandus | 50 |
| Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks | 51 |

| | |
|--|----|
| Lisa 2 Tabelid Isik ja Roll BEST-Estonia infosüsteemi uues versioonis | 53 |
| Lisa 3 Tabelid Liikmelisus, Rolli_omamine ja Roll BEST-Estonia infosüsteemi uues versioonis..... | 54 |

Jooniste loetelu

| | |
|---|----|
| Joonis 1. Punkti saamisega seotud infosüsteem BEST-Estonias..... | 14 |
| Joonis 2. Google API kliendi päring tagarakenduses..... | 16 |
| Joonis 3. Andmebaasitabelid punktisüsteemi praeguses versioonis..... | 16 |
| Joonis 4. Üldine statistika https://punktid.best.ee/ lehel..... | 17 |
| Joonis 5. Punkti saamise individuaalne aruanne autori näitel..... | 18 |
| Joonis 6. Klassifikaatorite register..... | 24 |
| Joonis 7. Punkti saamise kontseptuaalne andmemudel..... | 25 |
| Joonis 8. Punkti saamise seisundidiagramm..... | 30 |
| Joonis 9. Kasutajaliidese avaleht..... | 33 |
| Joonis 10. Kasutajaliidese sisse logimise vaade..... | 34 |
| Joonis 11. Kasutajaliidese punktide sisestamise vaade..... | 35 |
| Joonis 12. Kasutajaliidese punktide eemaldamise vaade..... | 36 |
| Joonis 13. Kasutajaliidese punktide kustutamise vaade..... | 37 |
| Joonis 14. Kasutajaliidese statistika vaade..... | 37 |
| Joonis 15. Punkti lisamise vaade mobiiltelefonis (OnePlus 7T mobiiltelefonis)..... | 38 |
| Joonis 16. Andmebaasi ühenduse pärimise sõltuvus..... | 39 |
| Joonis 17. Punktide lisamise vaate andmebaasi päring..... | 40 |
| Joonis 18. Punkti saamise andmebaasi sisestamine..... | 41 |
| Joonis 19. jQuery Ajax välja kutsumine <i>JavaScriptis</i> | 41 |
| Joonis 20. Veergudele rakendatud kitsenduste koodi näited..... | 42 |
| Joonis 21. Punktide saamiste registri füüsilise disaini täpsusega andmebaasi diagramm | 43 |
| Joonis 22. Punktide saamistega seotud klassifikaatorite füüsilise disaini täpsusega andmebaasi diagramm..... | 44 |

Tabelite loetelu

| | |
|--|----|
| Tabel 1. Süsteemi kasutuslood..... | 21 |
| Tabel 2. Süsteemi mittefunktsionaalsed nõuded..... | 22 |
| Tabel 3. Olemitüüpide sõnalised kirjeldused..... | 25 |
| Tabel 4. Atribuutide sõnalised kirjeldused..... | 26 |

1 Sissejuhatus

Sissejuhatavas osas antakse lühiülevaade töö taustast, probleemidest, eesmärgist ning kirjeldatakse töö edasine struktuur.

1.1 Üldine taust ja töö lühikirjeldus

Üliõpilasorganisatsiooni BEST-Estonia punktisüsteem on loodud selleks, et motiveerida liikmeid aktiivselt organisatsiooni töös osalema. BEST-Estonia liikmed saavad erinevaid ülesandeid täites punkte ja vastavalt punktidele ka tunnustust ja auhindu. Kõik liikmed saavad punkte tegevuste eest, mida on võimalik mõõta.

Organisatsiooni vaatepunktist on tegemist tööriistaga, mida juhatus kasutab liikmete tunnustamiseks, motiveerimiseks ja nende töökoormuse hindamiseks – sellest lähtuvalt teavad nad, kas inimesi on tarvis organisatsiooni juurde värvata ning milliseid liikmeid suunata appi toimuvaid projekte korraldama. Selleks, et vältida andmete väärkasutust haldab punktisüsteemi juhatuses ainult personalijuht.

Eeltöö käigus koguti koostöös organisatsiooni personalijuhi ja IT koordinaatoriga nõudeid süsteemile, analüüsiti olemasoleva tarkvara olukorda ning koostati nimekiri tarkvaralt nõutavatest funktsionaalsustest. Taustatööna – võimalike probleemide ja lahenduste leidmiseks – uuriti ka seda, kuidas on lahendatud punktide kogumine mõningate lähiriikide BESTi gruppides.

Projekti käigus analüüsiti süsteemile esitatud nõudeid, disainiti ja loodi MTÜ BEST-Estonia punktide saamise infosüsteemi andmebaas ning realiseeriti tarkvara veebirakenduse näol.

1.2 Probleem ja töö eesmärk

MTÜ BEST-Estonia haldab punktidega seotud andmeid tabelarvutussüsteemis Google Sheets. Info punktide saamise kohta on koondatud kokku seitsmesse erinevasse Google Sheets tabelisse. Sinna sisestatud infot kuvatakse (andmete ülekande ja töötlemise järel)

veebilehel <https://punktid.best.ee/>, mis on ligipääsetav nii kõigile organisatsiooni liikmetele, kui ka laiemale avalikkusele.

Eeltöö käigus tuvastati koostöös personalijuhiga olemasoleva lahenduse mitmed probleemid, millest osad tulenevad Google Sheets piirangutest [1].

Eeltöö käigus tuvastati järgnevad olulisemad punktide haldusega seotud probleemid.

- Tarkvara on keeruline hallata ja muutusi sisse viia, kuna selle loonud inimesed ei osale enam aktiivselt organisatsiooni töös. Õnneks on siiski olemas üldine tarkvara kirjeldus [19].
- Uute punktide lisamine aeganõudev ja keeruline protsess, kuna Google Sheets pole mõeldud nii suure koguse andmete haldamiseks.
- Tarkvara pole töökindel ja lakkab sageli töötamast andmesisetajate poolt tehtud vigade tõttu.

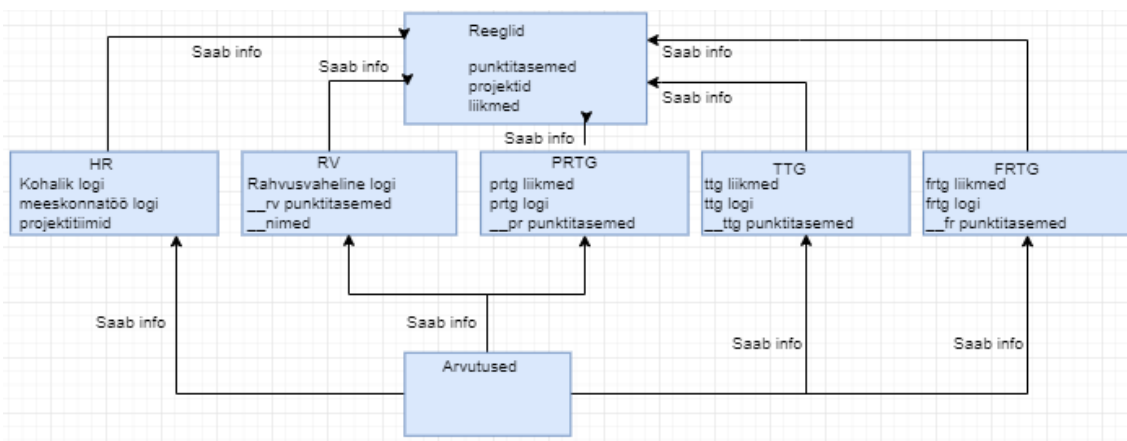
Käesoleva töö põhieesmärgiks oli parandada põhilised eeltöö käigus tuvastatud kasutajakogemusega seotud probleemid, millega süsteemi aruannete kasutajad ja andmesisetajad hetkel kokku puutuvad ning seeläbi optimeerida kogu organisatsiooni tööd.

1.3 Töö edasine struktuur

Töö on jaotatud viieks põhiosaks. Tarkvara praeguse olukorra all (peatükk 2) esitatakse senise tehnilise lahenduse detailsem kirjeldus. Süsteemianalüüsi osas (peatükk 3) esitatakse punktide saamise funktsionaalse allsüsteemi eesmärgid, mittefunktsionaalsed nõuded ning funktsionaalsed nõuded kasutuslugudena. Samuti esitatakse punktide saamise registri kontseptuaalne andmemudel ning seisundidiagramm. Disaini osas (peatükk 4) põhjendatakse kõigepealt kasutatud arendusvahendite valikut, esitletakse valminud veebirakenduse kasutajaliidest, kirjeldatakse rakenduse tehnilist lahendust ning esitatakse valminud andmebaasi tehnilist realiseerimise kirjeldav füüsilise disaini andmemudel. Arendusprotsessi osas (peatükk 5) kirjeldatakse kasutatud arendusmetoodikat ja analüüsitakse töö tegemise protsessi. Viimases töö põhiosa peatükis (peatükk 6) antakse soovitusi süsteemiarvenduse järgmiste sammude kohta.

2 Tarkvara praegune olukord

MTÜ BEST-Estonia kogub punktidega seotud andmeid pilve, tabelarvutussüsteemis Google Sheets loodud töölehtedele. Erinevaid punktisüsteemi tabeleid hoiustatakse Google Drive keskkonnas, kus nende redigeerimiseks on ligipääs antud ainult personalijuhile ning vastavate valdkondade juhtidele (edaspidi koordinaatorid). Süsteemiga seotud tabeleid on kokku seitse, kus igal koordinaatoril oli punktide lisamiseks oma tabel, mille täitmise reeglid ja vaikimisi punktid pärinevad üldisest reeglite andmetabelist (vt Joonis 1). Teiste sõnadega, iga valdkonna kohta on eraldi punktide tabel. Personalijuhi hallata on lisaks oma tabelile ka kõikide valdkondade punktidega seotud andmed.



Joonis 1. Info liikumine olemasolevas BEST-Estonia punktihalduse süsteemis.

Punktide kohta kogutud andmed edastatakse domeenide ja serverite pilveteenust pakkuva <https://www.zone.ee/et/> keskkonda loodud MySQL andmebaasi, kasutades Google Spreadsheet API lahendust (vt Joonis 2) [2]. Zone keskkonda on MySQL andmebaasi loodud kaks komplekti tabeleid (vt Joonis 3) [19]. Esimese komplekti tabelitel (koordinaatorite tabelitel; nende nimed algavad eesliitiga *points_*) on sarnane struktuur kui Google Sheets tabelitel ja sinna laetakse andmeid Google Sheets tabelitest iga viie minuti tagant Google Spreadsheet API abil. Uusi andmeid laetakse siis kui Google Sheets tabelis on võrreldes eelmise laadimise korraga andmed muutunud. Kui Google Sheets tabelis on vahepeal andmed muutunud, siis värskendatakse MySQL andmebaasis

koordinaatori tabelit täielikult, st loetakse Google Sheetist kõik tabeli andmed ja luuakse MySQL andmebaasis tabel uuesti.

Google Sheets ja vastavates MySQL tabelites on kõigis isikute nimed ja tegevuste nimed. See tähendab andmete liiasust (kordamist) üle erinevate tabelite. Kui näiteks isiku nimi muutub ja see muudetakse ühes tabelis kuid ei muudeta teises, siis tulemuseks on viga isiku detailsete punktide kuvamisel. Tabelites on isikute nimed, mis ei ole isikute unikaalsed identifikaatorid. Seega, kui BEST'i liikmete hulgas oleks mitu sama nimega inimest (mida *veel* ei ole juhtunud), siis nende punktid grupeeritaks kokku ühe nime alla ja jällegi oleks tulemuseks eksitav informatsioon. Kui isik palub organisatsioonil (tulevalt isikuandmete kaitse seadusest [20]) oma isikuandmed kustutada, siis tuleb muudatus teha kõigis Google Sheets tabelites. Osaline kustutamine tähendaks eksimist seaduse nõuete vastu.

Teise komplekti tabelitest (nende nimed algavad eesliitega *meta_*) moodustavad ette välja arvatud koondandmete päringute tulemused, mille alusel serveeritakse kasutajatele veebilehel punktide statistikat. Nendes tabelites on koondandmed. Sisuliselt on tegemist hetktõmmiste e materialiseeritud vaadetega, mida tarkvara samuti regulaarselt värskendab. Nende andmete värskendamine toimub koos koordinaatorite tabelite värskendamisega. Kogutud punktide kohta kuvatakse infot veebilehel <https://punktid.best.ee/>, mis on ligipääsetav kõigile organisatsiooni liikmetele, kui ka laiemale avalikkusele (vt Joonis 4). Veebilehel on kõigil võimalik vaadata läbi aegade koondpunktide edetabelit, jooksva kuu ja aasta aktiivsemaid, kui ka otsida kõigi BEST-Estonia liikmete kohta individuaalset aruannet saadud punktide ja valdkonnapõhise statistika kohta (vt Joonis 5). Kuvatavaid andmeid töödeldakse veebilehe rakenduse osas visuaalsele esitamisele sobivaks.

```

<?php
require_once('../vendor/autoload.php');
require_once('../util/global.php');
require_once('../util/SQLimport.php');

/**
 * Returns an authorized API client.
 * @return Google_Client the authorized client object
 */
function getClient($type)
{
    global $PUBLIC_PATH;

    putenv("GOOGLE_APPLICATION_CREDENTIALS=$PUBLIC_PATH/auth/cosmic-kiiln-2905215-4a52105421d7e.json");

    $client = new Google_Client();
    $client->setApplicationName('BEST punktisüsteem PHP backend');
    $client->setScopes([Google_Service_Sheets::SPREADSHEETS_READONLY, Google_Service_Drive::DRIVE_METADATA_READONLY]);
    $client->useApplicationDefaultCredentials();

    return $client;
}

```

Joonis 2. Google API kliendi päring tagarakenduses.

| Struktuur | SQL | Otsi | Päring | Ekspordi | Impordi | Tegevused | Funktsioonid | |
|--|-----|--------------|-----------|----------|---------|-----------|--------------|----------------------|
| <input type="checkbox"/> members_with_points | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 476 InnoDB |
| <input type="checkbox"/> meta_groups | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 8 InnoDB |
| <input type="checkbox"/> meta_personal_frtg | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 244 InnoDB |
| <input type="checkbox"/> meta_personal_global | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 567 InnoDB |
| <input type="checkbox"/> meta_personal_hr_local | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 509 InnoDB |
| <input type="checkbox"/> meta_personal_hr_projects | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 169 InnoDB |
| <input type="checkbox"/> meta_personal_hr_teamwork | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 351 InnoDB |
| <input type="checkbox"/> meta_personal_mtg | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 70 InnoDB |
| <input type="checkbox"/> meta_personal_prtg | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 213 InnoDB |
| <input type="checkbox"/> meta_personal_rv | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 144 InnoDB |
| <input type="checkbox"/> meta_personal_ttg | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 86 InnoDB |
| <input type="checkbox"/> points_frtg | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 1,432 InnoDB |
| <input type="checkbox"/> points_hr_local | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 9,193 InnoDB |
| <input type="checkbox"/> points_hr_projects | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 298 InnoDB |
| <input type="checkbox"/> points_hr_teamwork | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 2,068 InnoDB |
| <input type="checkbox"/> points_mtg | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 185 InnoDB |
| <input type="checkbox"/> points_prtg | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 1,191 InnoDB |
| <input type="checkbox"/> points_rv | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 500 InnoDB |
| <input type="checkbox"/> points_ttg | ★ | Vaata | Struktuur | Otsi | Lisa | Tühjenda | Kustuta | 173 InnoDB |
| 19 tabelit | | Summa | | | | | | 17,877 InnoDB |

Joonis 3. Andmebaasitabelid punktisüsteemi praeguses versioonis.

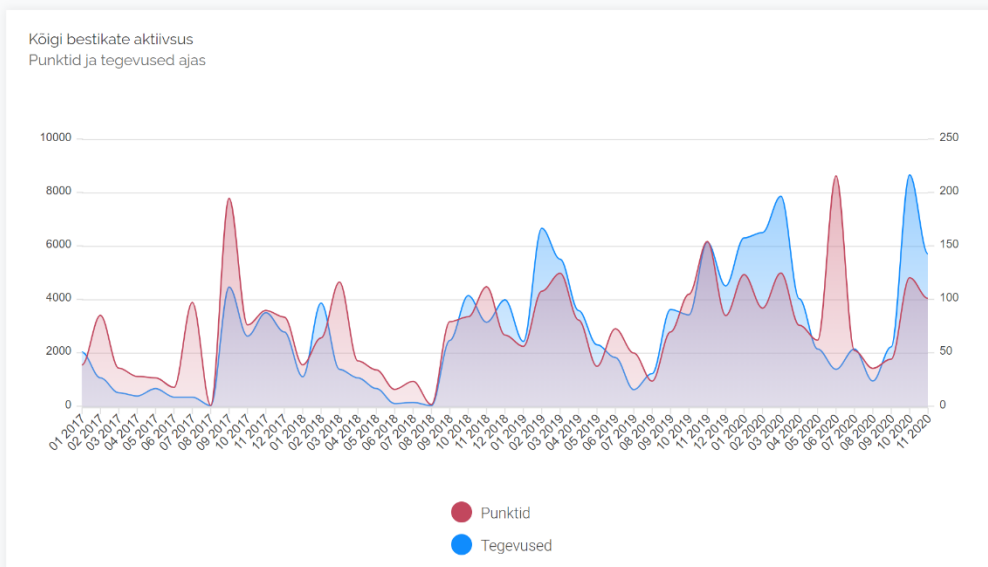
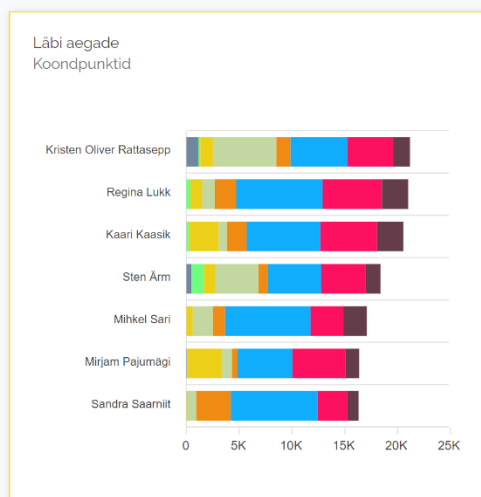


Nimi



Läbi aegade
Töörühvide tipud

| | | |
|-------------|--------------------------|------|
| TTG | Kristen Oliver Rattasepp | 1150 |
| MTG | Sten Ärm | 1190 |
| DMTG | Mirjam Pajumägi | 3280 |
| FRTG | Kristen Oliver Rattasepp | 6025 |
| RV | Sandra Saarniit | 3280 |
| LBG Tallinn | Sandra Saarniit | 8286 |
| Tiimitöö | Regina Lukk | 5645 |
| Projektid | Kristiina Kuusemets | 3050 |



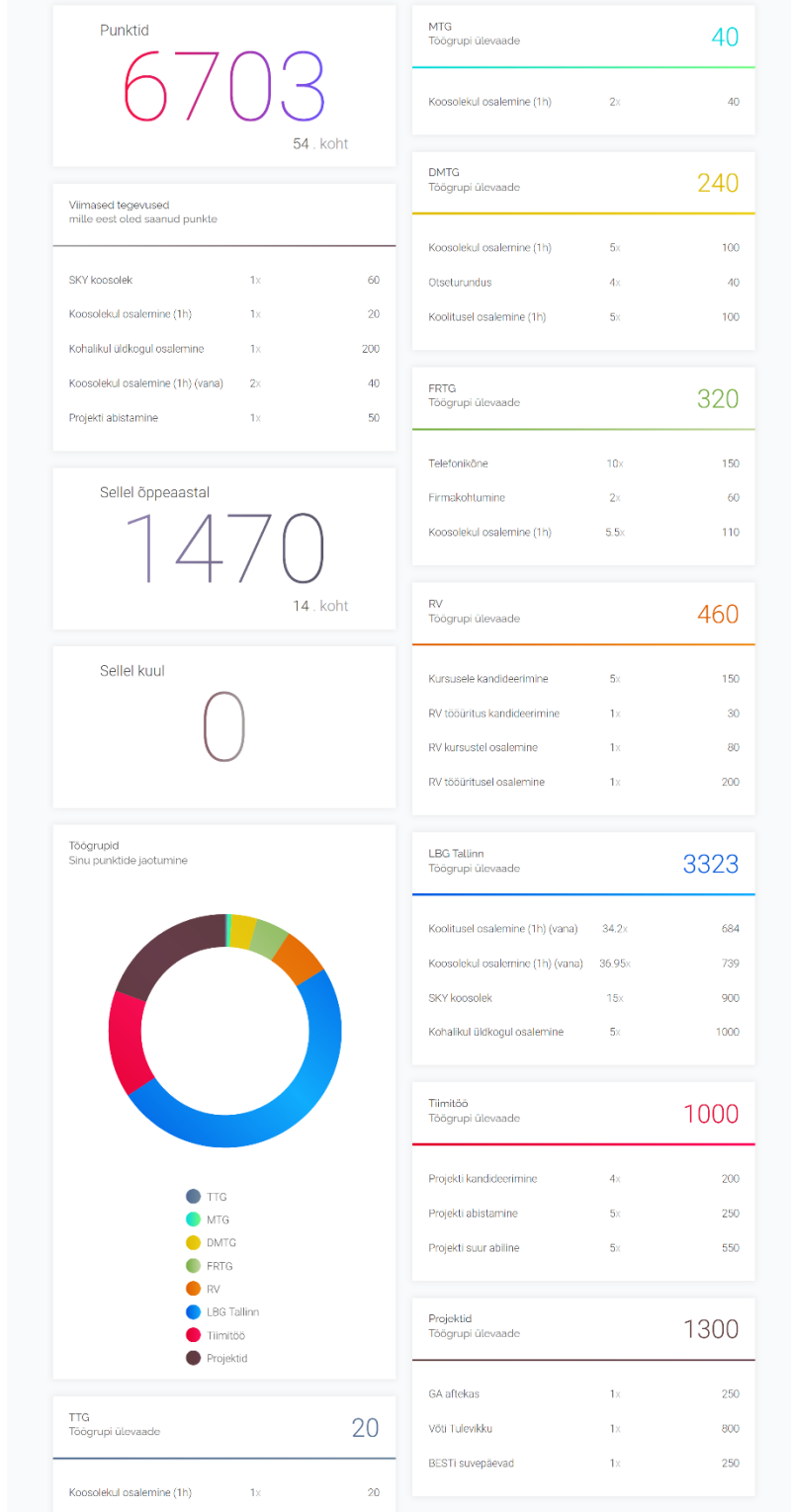
Kasutusjuhend
Arhitektuur

Tagasiside
GitHub

Joonis 4. Üldine statistika <https://punktid.best.ee/> lehel.

Karl Henri Kaljus

Täisliige



Joonis 5. Punkti saamise individuaalne aruanne autori näitel.

2.1 Punktide arvestus teiste riikide BESTi allorganisatsioonides

MTÜ BEST-Estonia on vaid üks allorganisatsioon kogu rahvusvahelisest BESTi organisatsioonist, kuhu kuulub kokku 93 samaväärset gruppi. Taustatööna – võimalike probleemide ja lahenduste leidmiseks – uuriti, kuidas on lahendatud punktide kogumine mõningate lähiriikide BESTi gruppides. Uuriti kohalikke BESTi gruppe Riias, Peterburis, Kaunases, Varssavis ning Lundis.

Sarnaselt BEST-Estoniale, oli Riias, Peterburis ning Varssavis asuvatel gruppidel liikmete motiveerimiseks kasutusele võetud punktisüsteem. Kõigis nimetatud allorganisatsioonides hallati punktisüsteemi tabelarvutussüsteemis Excel. Riias ja Peterburis kasutusel olev punktisüsteem ei ole liikmetele jooksvalt nähtav ning individuaalsete aruannete ja üldise statistika nägemiseks tuleb pöörduda personalijuhi poole. Varssavis kasutatav punktisüsteem on organisatsiooni liikmetele nähtav Exceli tabelite peale loodud statistikalehe näol, kuid individuaalset statistikat seal näha pole ning süsteemi haldaja sõnul pole antud lahendus töökindel ja pole seetõttu liikmete seas väga kasutuses.

Kaunases ja Lundis pole organisatsioonide eripära tõttu punktisüsteemi kasutusele võetud. Kaunases asuv BESTi grupp motiveerib liikmeid läbi erinevate auhindade ning Lundis asuvas allorganisatsioonis puudub igasugune liikmete motiveerimiseks mõeldud süsteem väheste aktiivsete liikmete arvu tõttu.

3 Süsteemianalüüs

Selles peatükis kirjeldatakse detailselt loodava süsteemi eesmärgid, mittefunktsionaalsed nõuded, funktsionaalsed nõuded kasutuslugudena (*user stories*) [5] ning punktide saamise registri konseptuaalne andmemudel ja punkti saamise seisundidiagramm.

3.1 Punktide saamise funktsionaalne allsüsteem

Arenduse keerukuse kontrolli all hoidmiseks (terve infosüsteem on korraga arendamiseks liiga suur) ja töö paremaks organiseerimiseks on BEST-Estonia hallatav infosüsteem jagatud alamosadeks e allsüsteemideks. Selle tööga samaaegselt BEST-Estonia infosüsteemi kohta tehtud Stella Leego lõputöö [21] kohaselt koosneb infosüsteem 31 funktsionaalsest allsüsteemist. Igale funktsionaalsele allsüsteemile vastab selle poolt teenindatav andmekeskne allsüsteem e register. Teenindamine tähendab andmete lugemist ja muutmist. Kõik infosüsteemi allsüsteemid on loetletud eelnimetatud lõputöös. Funktsionaalsed allsüsteemid ja registrid leiti selles põhiobjektide keskselt – igale põhiobjektile vastab funktsionaalne allsüsteem ja selle poolt teenindatav register.

Käesolev peatükk keskendub punktide saamise funktsionaalsele allsüsteemile ehk punktisüsteemile ja selle poolt tööks vajatavale andmebaasi alamosale. See alamosa hõlmab punktide saamise registrit, kuid funktsionaalse allsüsteemi toimimiseks on vaja ka andmeid isikute, liikmelisuste ja klassifikaatorite registritest.

3.1.1 Eesmärgid

Punktide saamise funktsionaalse allsüsteemi eesmärgid on järgmised.

- Võimaldada koguda ja hallata (lisada, muuta, kustutada) andmeid punktide saamise kohta.
- Võimaldada saada ülevaade registreeritud punktidest ja nende saajatest.
- Võimaldada lisada uusi sissekandeid punktide saamise sündmustest (punktide saamisega seotud seisundimuudatustest).
- Võimaldada pärida ajakohast statistikat kasutaja poolt määratud ajavahemikus, et tunnustada aktiivseid organisatsiooni liikmeid.

3.1.2 Funktsionaalsed nõuded

Punktisüsteemi uue versiooni arendamisel koguti esimese asjana kokku esialgne hulk kasutuslugusid (sellel ajal kõige olulisemana tundunud funktsionaalsuste kohta) ning lisati need Exceli tabelisse (Tabel 1). Nõuded leiti koostöös praeguse süsteemi kasutajaga ehk organisatsiooni personalijuhiga.

Tabel 1. Süsteemi kasutuslood

| ID | Kasutuslugu |
|-----|--|
| 100 | Personalijuhina tahan liikmele lisada ettemääratud tegevuse eest punkte, et tunnustada liiget aktiivselt organisatsiooni tööst osavõtmise eest. |
| 101 | Personalijuhina tahan eemaldada liikmele lisatud punkte selliselt, et punktid süsteemis säilivad, kuid neid ei arvestata punktisumma arvutamisel, et eemaldada vale põhimõtte järgi antud punktid ja tagada õiglane punktiarvestus, kuid säilitada see info tulevikus vigadest õppimiseks. |
| 102 | Personalijuhina tahan kustutada liikmele lisatud punkte, kui liikmele on lisatud punkte ekslikult, et punktiarvestus oleks täpne. |
| 104 | Personalijuhina tahan näha punktide pingerida omalt poolt määratud ajavahemikus, et premeerida aktiivseid liikmeid (näiteks kuu või aasta aktiivsemaid). |
| 105 | Personalijuhina tahan punktide saamise statistikat filtreerida nime ja punktisumma järgi, et lihtsustada otsingut. |
| 106 | Koordinaatorina tahan lisada punkte oma valdkonnaga seotud tegevuste eest, et tunnustada töögrupis aktiivseid liikmeid. |
| 107 | Koordinaatorina tahan enda poolt hallatavasse valdkonda lisatud punkte kustutada, kui mingi tegevuse eest on ekslikult punkte lisanud, et punktiarvestus oleks täpne. |

| ID | Kasutuslugu |
|-----|--|
| 108 | Koordinaatorina tahan saada statistikat oma töögrupi aktiivsematest liikmetest, et premeerida neid valdkonnas aktiivse tegutsemise eest. |
| 109 | Koordinaatorina tahan näha oma töögrupis punkte saanud liikmete arvu, et omada ülevaadet valdkonna seisukorrast. |

3.1.3 Mittefunktsionaalsed nõuded

Eraldi tabeli kujul pandi kirja kõik arendatava punktide saamiste süsteemi mittefunktsionaalsed nõuded (Tabel 2). Nendele tuleb pöörata tähelepanu kogu arenduse vältel.

Tabel 2. Süsteemi mittefunktsionaalsed nõuded.

| Tüüp | Nõude kirjeldus |
|-----------------|--|
| Andmekvaliteet | Rakendus peab salvestama automaatselt informatsiooni kirjete sisestamise ja kustutamise/eemaldamise aja kohta |
| Arhitektuur | Andmebaasis, SQL-skriptides ning rakenduses tuleb kasutada UTF-8 (Unicode) kodeeringut [22]. |
| Dokumentatsioon | Süsteemi kasutajaliides ja dokumentatsioon peavad olema eesti keeles. Süsteem tuleks üles ehitada nii, et ei oleks väga raske lisada kasutajaliidesesse uusi keeli (inglise keel). |
| Kasutajaliides | Kasutajaliides peab olema veebipõhine. |
| Kasutajaliides | Rakendus peab kasutajale andmeid kuvama visuaalselt arusaadavalt ja selgelt, et rakendus oleks kasutajasõbralik. |
| Kasutajaliides | Kasutajaliides peab olema kasutatav enamlevinud veebilehitsetajatel (Firefox, Google Chrome, Microsoft Edge, Opera), sealhulgas nutiseadmetel (Android ja iOS). |

| Tüüp | Nõude kirjeldus |
|------------------|--|
| Serveri tarkvara | Süsteem peab andmete hoidmiseks kasutama SQL-andmebaasisüsteemi abil loodud andmebaasi. |
| Standardid | Nii andmebaasis kui rakenduses peab olema tagatud koodipuhtus ja korrektne stiil. |
| Turvalisus | Punktide saamiste rakendusse sisenemiseks peab olema loodud kasutajakontode süsteem, et vältida kõrvaliste isikute poolt punktide lisamist ja kustutamist. |
| Turvalisus | Rakendus peab kasutama andmebaasist andmete küsimiseks POST-päringuid, mitte GET-päringuid, et tagada andmete suurem turvalisus [4]. |
| Turvalisus | Andmebaasis hoitakse parooli räsiväärtus, mis on leitud selle parooli ja soola põhjal. Räsiväärtuse leidmisel ei tohi kasutada räsifunktsioone MD5 ja SHA-1, mille puhul on algset parooli jõuga liiga lihtne teada saada. |

3.2 Punktide saamiste register

Punktide saamiste registri eesmärk on säilitada informatsiooni punktide saamiste kohta sellises mahus, et oleks tagatud punktide saamiste funktsionaalses allsüsteemis defineeritud eesmärkide täitmine. Punktide saamiste registri andmeid teenindab (loeb ja muudab) punktide saamiste funktsionaalne allsüsteem.

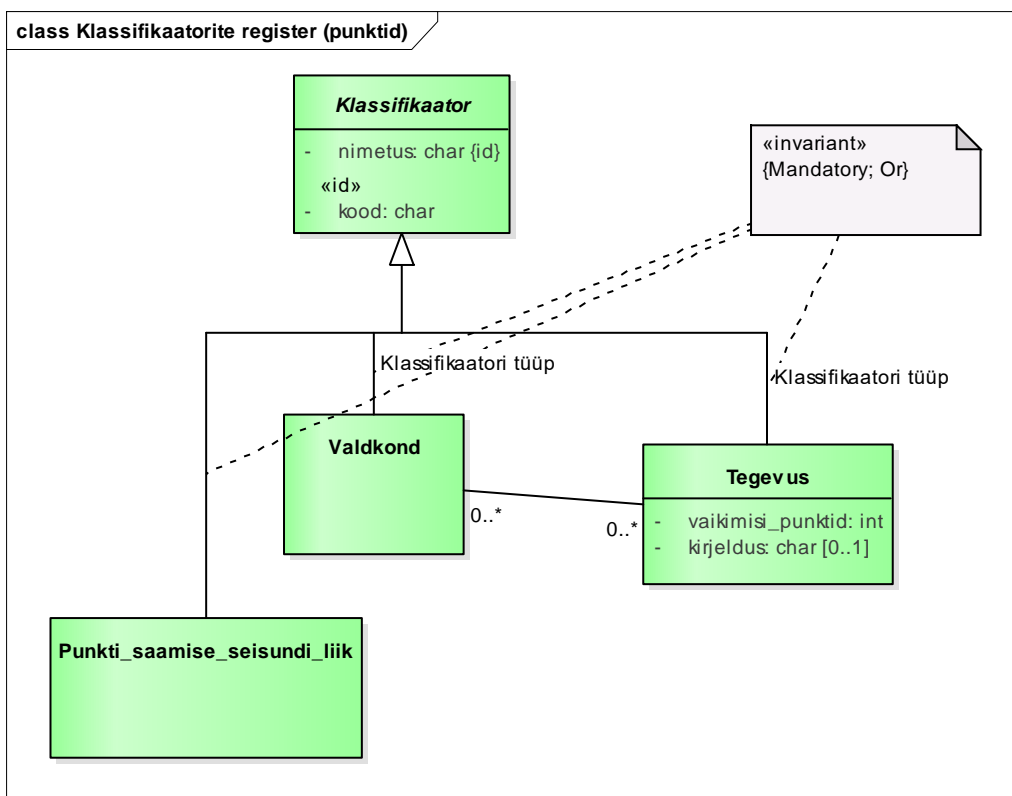
3.2.1 Seosed teiste registritega

Liikmelisuste register – Liikmelisuste registriga on punkti saamine seotud olemitüübi *Liikmelisus* kaudu. Saadud punktid omistatakse kindlale organisatsiooni liikmele. Igal isikul on süsteemis null või rohkem liikmelisust, kusjuures samaaegselt võib olla vaid üks lõppemata liikmelisus. Punktide saamine on seotud liikmelisusega ja selle kaudu isikuga.

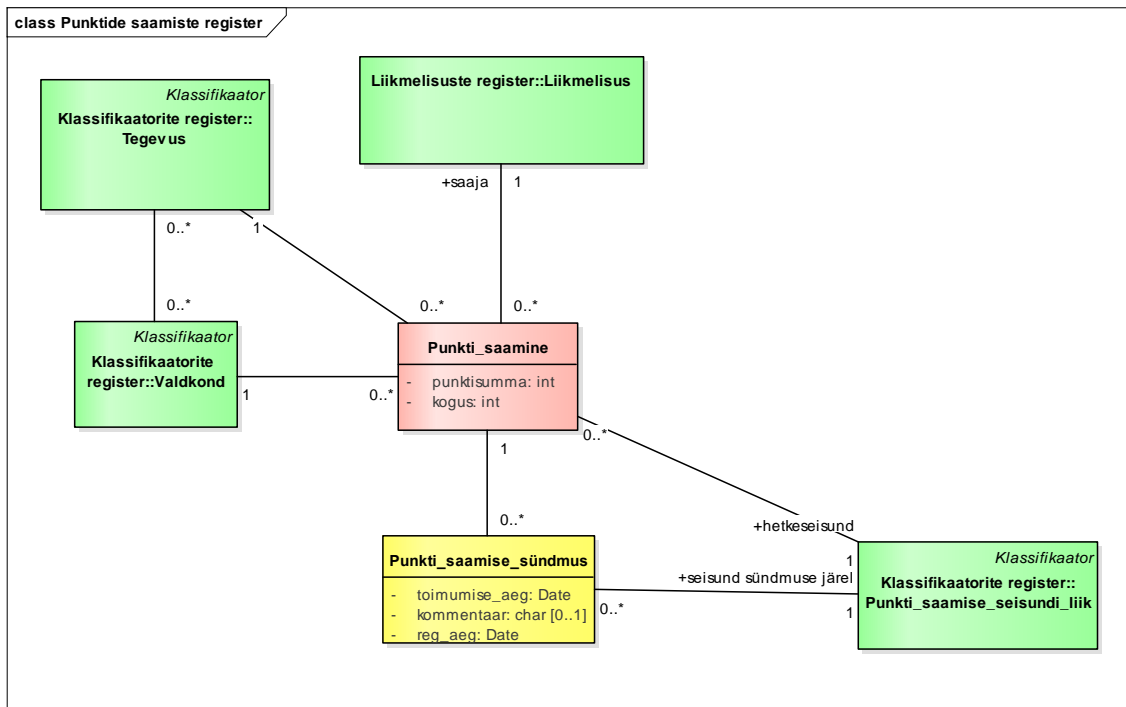
Klassifikaatorite register – Klassifikaatorite registriga on punktide saamine seotud olemitüüpide *Tegevus*, *Valdkond* ja *Punkti_saamise_seisundi_liik* kaudu. Punkte saadakse kindlate tegevuste eest ettemääratud valdkonnas ning neile määratakse registreerimisel seisund „aktiivne“.

Joonis 6 ja Joonis 7 esitatakse kontseptuaalne andmemudel, mis koosneb olemi-suhte diagrammidest ning olemitüüpide ja atribuutide sõnalistest kirjeldustest (vt Tabel 3 ja Tabel 4). Olemi-suhte diagrammidel on värvidel järgmine tähendus.

- Punasega on tähistatud *Punktide saamise registri* põhiobjekt.
- Kollasega on tähistatud *Punktide saamise registrisse* kuuluvad mitte-põhiobjektid.
- Rohelisega on tähistatud teistesse registritesse kuuluvad objektid, mida on antud juhul vaja punktide saamise funktsionaalse allsüsteemi toimimise tagamiseks.



Joonis 6. Klassifikaatorite register.



Joonis 7. Punktide saamise registri kontseptuaalne andmemudel.

Tabel 3. Olemitüüpide sõnalised kirjeldused.

| Olemitüübi nimi | Registri nimi | Definitsioon |
|------------------------------|----------------------------|--|
| Klassifikaator | Klassifikaatorite register | Andmebaasis olevate andmete liigituskeem. |
| Liikmelisus | Liikmelisuste register | Liikmelisus väljendab organisatsiooni liikmeks olemist ja selle staatust. Korraga saab isikule omistatud olla ainult üks lõppemata liikmelisus. |
| Punkti saamine | Punktide saamise register | Punkti saamine väljendab liikme tunnustamist punktide näol organisatsiooni tööst osavõtmisel. Punkte määratakse kindla tegevuse eest ning punkti saamine võib olla seotud mitme tegevusega, valdkonnaga ja liikmega. |
| Punkti saamise seisundi liik | Klassifikaatorite register | Punkti saamise hetkeseisundi iseloomustus. Näited on aktiivne ja eemaldatud. |

| Olemitüübi nimi | Registri nimi | Definitsioon |
|------------------------|----------------------------|--|
| Punkti saamise sündmus | Punktide saamiste register | Punkti saamise sündmus on punkti saamise seisundi muutus infosüsteemis. Näiteks punktid on määratud valedel põhimõtetel ning eemaldatakse ning selle kohta registreeritakse sündmus. |
| Tegevus | Klassifikaatorite register | Tegevus väljendab liikme poolt toime pandud tegevust organisatsiooni töös osa võtmisel, mille eest teda tunnustatakse. Tegevus võib olla seotud mitme valdkonnaga ning liige võib saada punkte erinevate tegevuste eest. |
| Valdkond | Klassifikaatorite register | Valdkond väljendab tegevusala, millega BEST-Estonia töögrupp tegeleb. Iga valdkond on seotud ühe või rohkema tegevusega. Näited on tehnika ja sponsorlus. |

Tabel 4. Atribuutide sõnalised kirjeldused.

| Olemitüübi nimi | Atribuudi nimi | Definitsioon | Näiteväärtus |
|-----------------|----------------|---|--------------|
| Klassifikaator | kood | Klassifikaatori väärtusele viitav kood, mida saab kasutada selle väärtuse lühidalt esitamiseks. Kood võib olla arvuline või tekstiline väärtus. { Klassifikaatori unikaalne identifikaator, mis on | 16 |

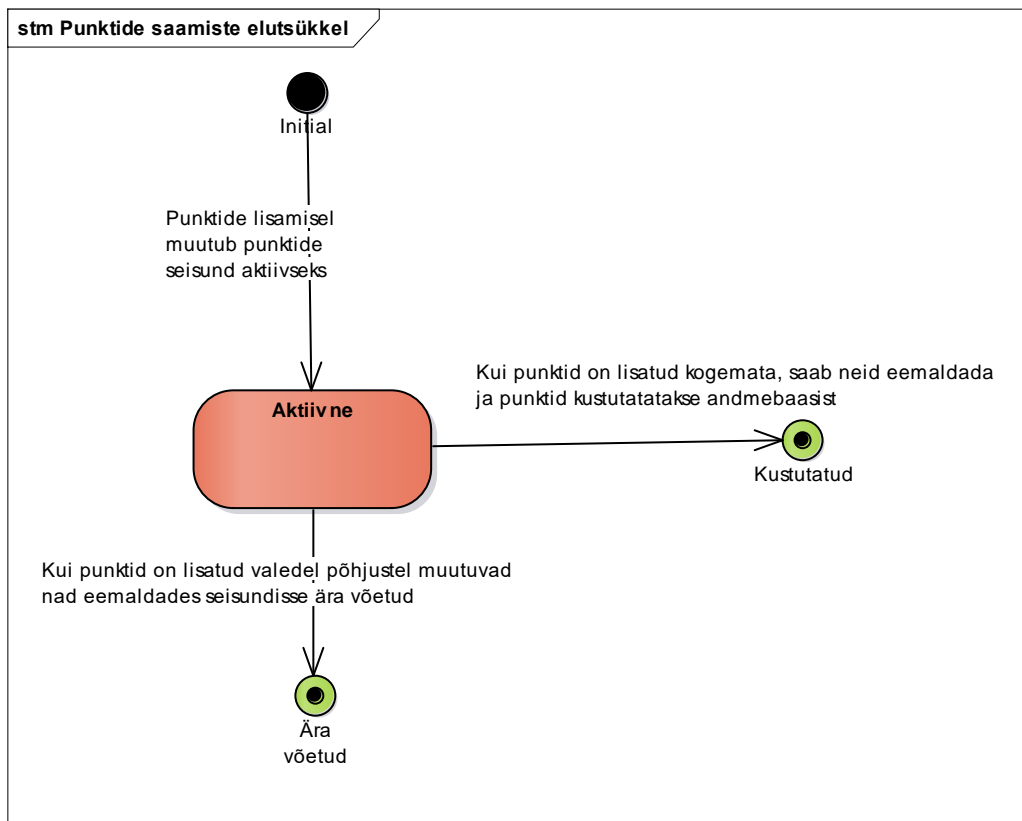
| Olemitüübi nimi | Atribuudi nimi | Definitsioon | Näiteväärtus |
|-----------------|----------------|---|--------------|
| | | unikaalne klassifikaatori tüübi piires. Peab olema registreeritud.} | |
| Klassifikaator | nimetus | Klassifikaatori väärtuse ametlik nimetus. {Klassifikaatori unikaalne identifikaator, mis on unikaalne klassifikaatori tüübi piires. Peab olema registreeritud. Nimetus ei tohi olla tühi string ja ainult tühimärkidest koosnev string.} | Valdkond |
| Punkti saamine | kogus | Tegevuste arvu näitav kogus, mille järgi arvutatakse sissekande punktisumma. Vaikimisi väärtus 1. {Registreerimine on kohustuslik. Peab olema vahemikus 1 kuni 99, otspunktid kaasa arvatud.} | 3 |
| Punkti saamine | punktisumma | Punkti saamise eest liikmele omistatav punktisumma, mille alusel teda tunnustatakse. Punktisumma tuleneb tegevuse vaikimisi | 400 |

| Olemitüübi nimi | Atribuudi nimi | Definitsioon | Näiteväärtus |
|------------------------|----------------|---|---------------------------------|
| | | määratud punktidest ja tegevuse kogusest – need korrutatakse. {Registreerimine on kohustuslik. Peab olema vahemikus 1 kuni 1999, otspunktid kaasa arvatud. } | |
| Punkti saamise sündmus | kommentaar | Sündmuse vabatekstiline kirjeldus. {Ei tohi olla tühi string ja ainult tühimärkidest koosnev string. } | Võti tulevikku projektimeeskond |
| Punkti saamise sündmus | reg_aeg | Punkti saamise sündmuse süsteemi lisamise aeg. {Peab olema registreeritud. Registreerimisaeg sisestatakse süsteemi poolt automaatselt. Registreerimise aeg peab olema suurem võrdne toimumise ajast. } | 2021-01-04 13:58:19 |
| Punkti saamise sündmus | toimumise_aeg | Millal toimus tegevus, mille eest punkte saadakse või ära võetakse. {Peab olema registreeritud. Toimumise aeg ei saa olla peale registreerimise aega. | 2021-01-02 17:52:46 |

| Olemitüübi nimi | Atribuudi nimi | Definitsioon | Näiteväärtus |
|-----------------|-------------------|---|---|
| | | Toimumise aeg peab olema vahemikus 1. jaanuar 1991 ja 31. detsember 2199, otspunktid kaasa arvatud. } | |
| Tegevus | vaikimisi_punktid | Tegevuse eest vaikimisi määratud punktide väärtus. Organisatsiooni poolt fikseeritud väärtus. {Peab olema registreeritud. Peab olema suurem kui 0. } | 120 |
| Tegevus | kirjeldus | Tegevuse vabatekstiline kirjeldus. {Ei tohi olla tühi string ja ainult tühimärkidest koosnev string. } | Koosoleku läbiviimisesse panustamine peakorraldajana. |

3.2.2 Punkti saamise seisundidiagramm

Joonis 8 esitab seisundidiagrammi, mis kirjeldab Punktide saamise registri põhiobjekti *Punkti saamine* kõikvõimalikke elutsükleid.



Joonis 8. Punkti saamise seisundidiagramm.

4 Disain

Selles peatükis põhjendatakse kõigepealt kasutatud arendusvahendite valikut. Seejärel tutvustatakse kasutajaliidese, rakenduse ja andmebaasi disaini.

4.1 Vahendite valik

Arendusvahenditest oli andmebaasisüsteemi valik üks olulisemaid, mida disainile eelnevalt tuli otsustada. Andmebaasisüsteemi valikul tuli arvesse võtta nii autori enda võimalusi, kui ka organisatsiooni nõudeid ja võimalusi. Neid nõuded silmas pidades tehti eeltöö populaarsemate andmebaasisüsteemide kohta, peale mida jäid sõelale kaks andmebaasisüsteemi: MySQL [23] ja PostgreSQL [24]. Nende kahe vahel valiku tegemisel võeti valikul arvesse järgmised parameetrid.

- Kas on tasuta: Jah/Ei?
- Kas on avatud lähtekoodiga: Jah/Ei?
- Kas on ühilduv BEST-Esonia olemasoleva infosüsteemiga: Jah/Ei?
- Kas autoril on varasem kogemus: Jah/Ei?
- Kas on populaarne andmebaasisüsteem e leidub infot internetis: Jah/Ei?

PostgreSQL puhul on tegemist tasuta ja avatud lähtekoodiga andmebaasisüsteemiga, mis jälgib hästi SQL standardit ja pakub arendajatele palju võimalusi. Samuti oli tegemist 2021. aasta jaanuari seisuga populaarsuselt neljanda andmebaasisüsteemiga [17], mille kohta leidub internetis palju töö raames kasulikku infot ning millega on palju kogemust ka juhendajal. Autoril selle süsteemiga kogemust ei ole.

MySQL puhul on tegemist samuti tasuta ja avatud lähtekoodiga andmebaasisüsteemiga, millega on ka autoril vähesel määral kogemust. MySQL oli 2021. jaanuari seisuga populaarsuselt teine andmebaasisüsteem [17]. Internetis leidub selle kasutuse ja rakendamise kohta väga palju infot.

BEST-Estonia infosüsteemi hoitakse hetkel Zone repositooriumis, mis PostgreSQLi ei toeta, kuid MySQLi toetab. Olemasolev punktide haldamise lahendus kasutab MySQLi. Seetõttu langes andmebaasisüsteemi valik MySQLi kasuks.

Arendusvahendi valiku alla kuulus ka veebirakenduse programmeerimiskeele valik. Kuna töö eesmärgiks oli arendada punktide saamise funktsionaalse allsüsteemi tarkvara uus versioon ning olemasolev versioon tugines PHP programmeerimiskeelele, siis tulevikus süsteemi haldamise ning edasiarendamise lihtsustamiseks valiti ka töö raames arenduseks PHP. Nii nagu olemasolevas versioonis kasutatakse ka uues PHP *Yii* raamistikku.

4.2 Kasutajaliides

Kasutajaliidese loomisel lähtuti olemasoleva veebilehe <https://punktid.best.ee/> visuaalsete elementide taaskasutusest, kus on kuvatud BESTi logo ning elemente BESTi visuaalsest identiteedist [13]. Veebilehe kujundamisel on lähtutud ES6 standarditest [16]. Kasutajaliides on eestikeelne, et see annaks kasutajale võimalikult täpset informatsiooni nii väljade, kui ka teadete näol.

4.2.1 Personalijuhi vaade

Järgnev kasutajaliidese kirjeldus põhineb kasutajaliidese osal, mis on mõeldud personalijuhile.

Kasutajaliidese navigatsiooni loomisel võeti aluseks punktide saamise funktsionaalse allsüsteemi olulisemad funktsionaalsed nõuded. Nende põhjal võimaldab kasutajaliides personalijuhil valida nelja funktsionaalsuse vahel (vt Joonis 9).

- Punktide lisamine – kuvab vaate punkti sisestamiseks vajalikkest väljadest.
- Punktide statistika – kuvab vaate sisestatud punktide koondstatistika nägemiseks.
- Punktide eemaldamine – kuvab vaate liikmetele sisestatud punktide eemaldamiseks, kui need on saadud valel põhimõtetel.
- Punktide kustutamine – kuvab vaate kogemata sisestatud punktide andmebaasist kustutamiseks.



Siit saad liikmetele punkte anda

[Punktide lisamine](#)

Siit saad punktide statistikat

[Punktide statistika](#)

Siit saad liikmetele antud punkte kustutada

[Punktide kustutamine](#)

Siit saad liikmetele antud punkte eemaldada

[Punktide eemaldamine](#)



Kasutusjuhend
Arhitektuur

Tagasiside
GitHub

Joonis 9. Kasutajaliidese avaleht.


Kasutajaliidese avalehele ligi pääsemisele eelnevalt peab kasutaja sisse logima talle määratud kasutajanime ning parooliga (vt Joonis 10). Samuti peab ta näitama rolli, mille alt soovib siseneda.

Kasutajanimetid, paroole ning rolli omamisi talletatakse andmebaasis. Täpsemalt on kasutajanimena kasutusel meiliaadress, mis on tabelis *Isik* ja paroolina parool, mis on tabelis *Konto* (vt Lisa 2). Tabelis *Konto* hoitakse parooli väärtusi parooli räsiväärtustena. Räsiväärtused on leitud bcrypt funktsiooni abil ja nende arvutamisel on kasutatud soola. Info rolli kohta (kas tegemist on personalijuhi või koordinaatoriga) on leitav tabelitest *Liikmelisus* ja *Rolli_omamine* (vt Lisa 3). Nende tabelite kirjeldus on lisades kuna nende kavandamine jäi väljapoole käesoleva lõputöö skoopi ja need loodi Leego [21] lõputöö tulemusena.

Sisse logimine

 Kasutajanimi

 Parool

-- Vali roll -- 

Logi sisse

Joonis 10. Kasutajaliidese sisse logimise vaade.

Valede kasutajatunnuste sisestamise korral kuvatakse kasutajale teateks „Vale informatsioon!“. Eduka sisselogimise korral määratakse kasutajale sessioonivõti, millega pääseb ligi veebirakenduse funktsionaalsusele ning suunatakse kasutajaliidese avalehele.

Punktide lisamise vaates on kasutajal võimalik valida andmebaasi sisestatud aktiivsete liikmete seast inimene, kellele punkte lisada ning valida seejärel valdkond ning tegevus enda poolt määratud valdkonnast. Isiku täpseks tuvastamiseks samade nimede korral kuvatakse lisaks ka isiku email, mis on unikaalne. Kasutajal tuleb ka määrata tegevuste kogust, et hoida aega kokku üksikute sisestamiste pealt (Joonis 11). Näiteks, selle asemel, et sisestada ükshaaval kolm firmakohtumist, millest igaühe eest saab 10 punkti saab ta sisestada koguse 3 ja selle alusel luuakse punktide saamine kus punktisumma on 30. Punktisumma arvutamiseks mõeldud punktid üksiku tegevuse eest võetakse tabelist *Tegevus*.



Lisa punkte

BESTikas

Valdkond

Tegevus

Kogus

Joonis 11. Kasutajaliidese punktide sisestamise vaade.

Punkti eemaldamise vaates on kasutajal võimalik valida aktiivsete liikmete loetelust inimene, mille järel kuvatakse loeteluna valitud inimesega seotud aktiivsed punkti saamised. Seejärel on kasutajal võimalus valida punkti saamine, mille eest punktid eemaldatakse e muudetakse andmebaasis punkti saamise seisundi liigiks „Eemaldatud“ (vt Joonis 12).



Eemalda punkte
Kasuta punktide eemaldamist, kui punkte on saadud ebaausalt

BESTikas
Peeter Pets - peeter@gmail.com

Tegevus
Firmakohtumine 2020-12-31 03:16:53

Eemaldan punktid



Kasutusjuhend
Arhitektuur

Tagasiside
GitHub

Joonis 12. Kasutajaliidese punktide eemaldamise vaade.

Punktide kustutamise vaates kuvatakse sarnaselt eemaldamisega nimekiri aktiivsetest BEST-Estonia liikmetest. Isiku valides kuvatakse temaga seotud punktide saamised, mille valimisel on kasutajal võimalus punkte kustutada (vt Joonis 13). Punktide kustutades eemaldatakse punktide saamine andmebaasist.



Kustuta punkte

BESTikas

Mari Maasikas - mari@gmail.com

Tegevus

Mentor 2021-01-02 19:25:18

Kustutan punktid



Kasutusjuhend
Arhitektuur

Tagasiside
GitHub

Joonis 13. Kasutajaliidese punktide kustutamise vaade.

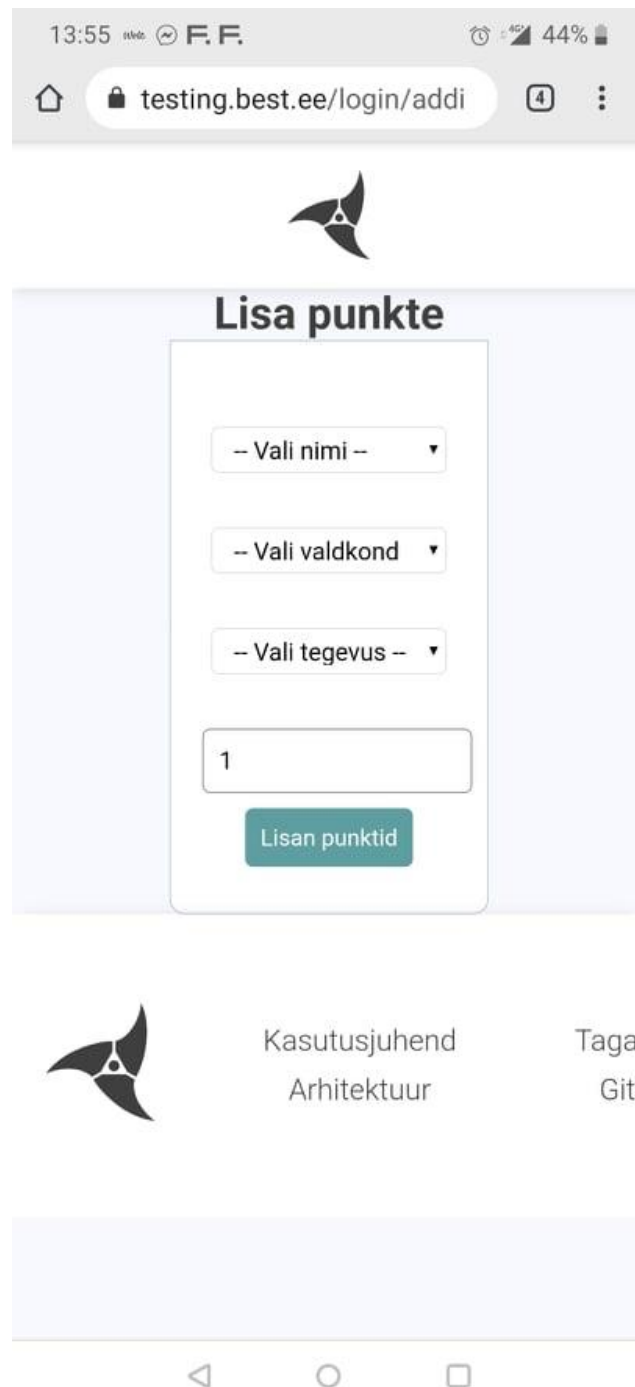
Kasutajaliidese statistika vaates kuvatakse kasutajale tabeli näol koondstatistika kõikide sisestatud punktide saamise kohta ning järjestatakse see punktisumma alusel (vt Joonis 14).

Statistika

| Eesnimi | Perenimi | Email | Punktisumma |
|---------|------------|----------------------|-------------|
| Peeter | Pets | peeter@gmail.com | 3017 |
| Mari | Maasikas | mari@gmail.com | 2700 |
| Kalev | Poeg | kalevipoeg@gmail.com | 2301 |
| Priit | Viljandist | priidik@gmail.com | 60 |

Joonis 14. Kasutajaliidese statistika vaade.

Nii andmete kuvamine kui ka sisestamine toimivad kasutajasõbralikult ja mobiililt (Joonis 15).



The screenshot shows a mobile browser interface. At the top, the status bar displays the time 13:55, signal strength, Wi-Fi, and battery level at 44%. The address bar shows the URL 'testing.best.ee/login/addi'. Below the address bar is a logo consisting of three curved shapes meeting at a central point. The main content area is titled 'Lisa punkte' (Add points). It contains a form with three dropdown menus: '-- Vali nimi --', '-- Vali valdkond --', and '-- Vali tegevus --'. Below these is a text input field containing the number '1'. A green button labeled 'Lisan punktid' (Add points) is positioned below the input field. At the bottom of the page, there is a footer with the same logo on the left, and the text 'Kasutusjuhend Arhitektuur' and 'Tagas GitH' on the right. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Joonis 15. Punkti lisamise vaade mobiiltelefonis (OnePlus 7T mobiiltelefonis).

4.3 Rakendus

Rakenduse struktuur tugineb enamuses eelmise punktisüsteemi versiooni realiseerimisel kasutatud PHP raamistikule *Yii*. Rakenduse arenduses on lähtutud puhta koodi põhimõtetest [6] ja [18]. Põhilised puhta koodi põhimõtted, millele arenduses teiste seast suuremat rõhku pöörati.

- Funktsioonide korduste vältimine.
- Nimed peavad olema arusaadavad ja vastama kontekstile.
- Lokaalsed atribuudid tuleb defineerida võimalikult lähedale nende kasutusele.
- Klass tohiks teada ainult oma otseseid sõltuvusi.

Veebirakendusse sisenedes päritakse andmebaasi ühendus globaalselt defineeritud ühenduse failist, et luua andmebaasiga ühendus vaid päringu hetkeks [7] (vt Joonis 16). Headest programmeerimise tavadest tulenevalt aitab selline lähenemine paremini hallata loodud ühendusi ja väldib samaaegselt andmebaasiga mitme ühenduse loomist [8].

```
$INC_DIR = $_SERVER["DOCUMENT_ROOT"]. "/util/";
include_once($INC_DIR. "global.php");

global $username;
global $password;
global $servername;
global $POINTSdatabaseName;

try {
    $conn = new PDO(
        "mysql:host=$servername; dbname=$POINTSdatabaseName",
        $username, $password
    );
    $conn->setAttribute(PDO::ATTR_ERRMODE,
        PDO::ERRMODE_EXCEPTION);
}
```

Joonis 16. Andmebaasi ühenduse pärimise sõltuvus.

Kasutajatunnuse valideerimise skript pärib andmebaasist registreeritud kasutajate andmed ning valideerib vormist POST päringuga sisestatud andmete korrektsust. Andmebaasist tulnud parool on räsi kujul ning selle valideerimiseks kasutatakse PHP

sisse kirjutatud funktsiooni *password_verify()*. Õigete kasutajatunnustega kasutajale määratakse sessiooni võti, millega pääseb ligi veebirakenduse funktsionaalsustele.

Punktide sisestamise funktsionaalsuse lehel kuvatakse isikud, kes on andmebaasis registreeritud aktiivsete liikmetena. Valdkonna nimekiri päritakse andmebaasist *query* funktsiooniga ning kasutaja poolt valitud valdkonnast tulenevalt kuvatakse tegevuste nimekirjas valitud valdkonnaga seotud tegevuste loetelu (vt Joonis 17).

```
</select name="valdkond_kood" onchange =ValkondAjutine
(this.valdkond_kood)>
</div>
<div>
Valdkond:
<select name="valdkond_kood" onchange="ValkondAjutine(this.value)">
  <option disabled selected>-- Vali valdkond --</option>
  <?php
    $records = mysqli_query($db, "SELECT Valdkond.nimetus,
Valdkond.kirjeldus, Valdkond.valdkond_kood FROM Valdkond"); // Pärib
andmebaasist andmeid

    while($data = mysqli_fetch_array($records))
    {
      echo "<option value='". $data['valdkond_kood'] .'-'>". $data
['nimetus'] .'-' . $data['kirjeldus'] . "</option>"; // Näitab päringuid
dropdownis
```

Joonis 17. Punktide lisamise vaate andmebaasi päring.

Kasutaja poolt sisestatud valikud sisestatakse seejärel andmebaasi punkti saamise tabelisse ning samuti registreeritakse punkti saamise sündmus (vt Joonis 18). Sisestamise aja märkimise järgi saab kasutaja hiljem vaadata statistikat.


```

//Sisestab punkti saamise
$insertPoints = mysqli_query($db,"INSERT INTO `Punkti_saamine` (`liikmelisus_id`,
`tegevus_kood`, `valdkond_kood`, `punktisumma`, `tegevuse_arv`) VALUES
($liikmelisus_id, $tegevus_kood, $valdkond_kood, $punktisumma, $kogus)");

//Sisestab punkti saamise sündmuse punkti saamise viimase id järgi
$insertEvent = mysqli_query($db,"INSERT INTO `Punkti_saamise_syndmus`
(`punkti_saamine_id`, `seisund_syndmuse_jarel`, `registreerija_id`) VALUES
(LAST_INSERT_ID(), 1, $reg_id)");

if(!$insertPoints){
    echo mysqli_error($db);
}
if(!$insertEvent){
    echo mysqli_error($db);
}
else{
    echo "Punktid edukalt sisestatud.";
}
}

```

Joonis 18. Punkti saamise andmebaasi sisestamine.

Punkti saamise, kustutamise ning eemaldamise funktsionaalsused tuginevad kõik jQuery Ajax tehnoloogial, mis võimaldab asünkroonselt päringuid teha lehel ilma lehte värskendamata ning mille abil on loodud kõik sõltuvad loetelud (vt Joonis 19).

```

</html>
<script type="text/javascript">
    $(document).ready(function(){
        // Valdkonna muutudes kutsub välja
        $("#valdkondId").on("change",function(){
            var valdkond = $(this).val();
            if (valdkond) {
                $.ajax({
                    url : "actionAddPoints.php",
                    type:"POST",
                    cache:false,
                    data:{valdkond:valdkond},
                    success:function(data){
                        $("#tegevusId").html(data);
                    }
                });
            }else{
                $('#tegevusId').html('<option value="">Vali tegevus</option>');
            }
        });
    });
</script>

```

Joonis 19. jQuery Ajax välja kutsumine *JavaScriptis*.

4.4 Andmebaas

Andmebaasi füüsilise disaini loomisel võeti aluseks kontseptuaalne andmemudel. Olemitüüpidest tekkisid tabelid, atribuutidest veerud, seosetüüpidest välisvõtme veerud ning kitsendused ja unikaalsete identifikaatorite määrangutest primaarvõtme ning unikaalsuse kitsendused.

Füüsilise disaini andmebaasi diagrammid on toodud välja joonistel 21 ja 22. Diagrammidel on esitatud ka üldistatud kujul CHECK kitsendused.

Atribuutide alusel loodud veergudele jõustatud kitsendused on kirjeldatud üldisel kujul ning iga tingimuse kohta on toodud koodinäide (Joonis 20).

Teksti hoidmiseks mõeldud veergude kitsendused.

- Ei tohi olla tühi string.
- Ei tohi olla ainult tühimärkidest koosnev string.

Temporaalsete (ajaandmete) hoidmiseks mõeldud veergude kitsendused.

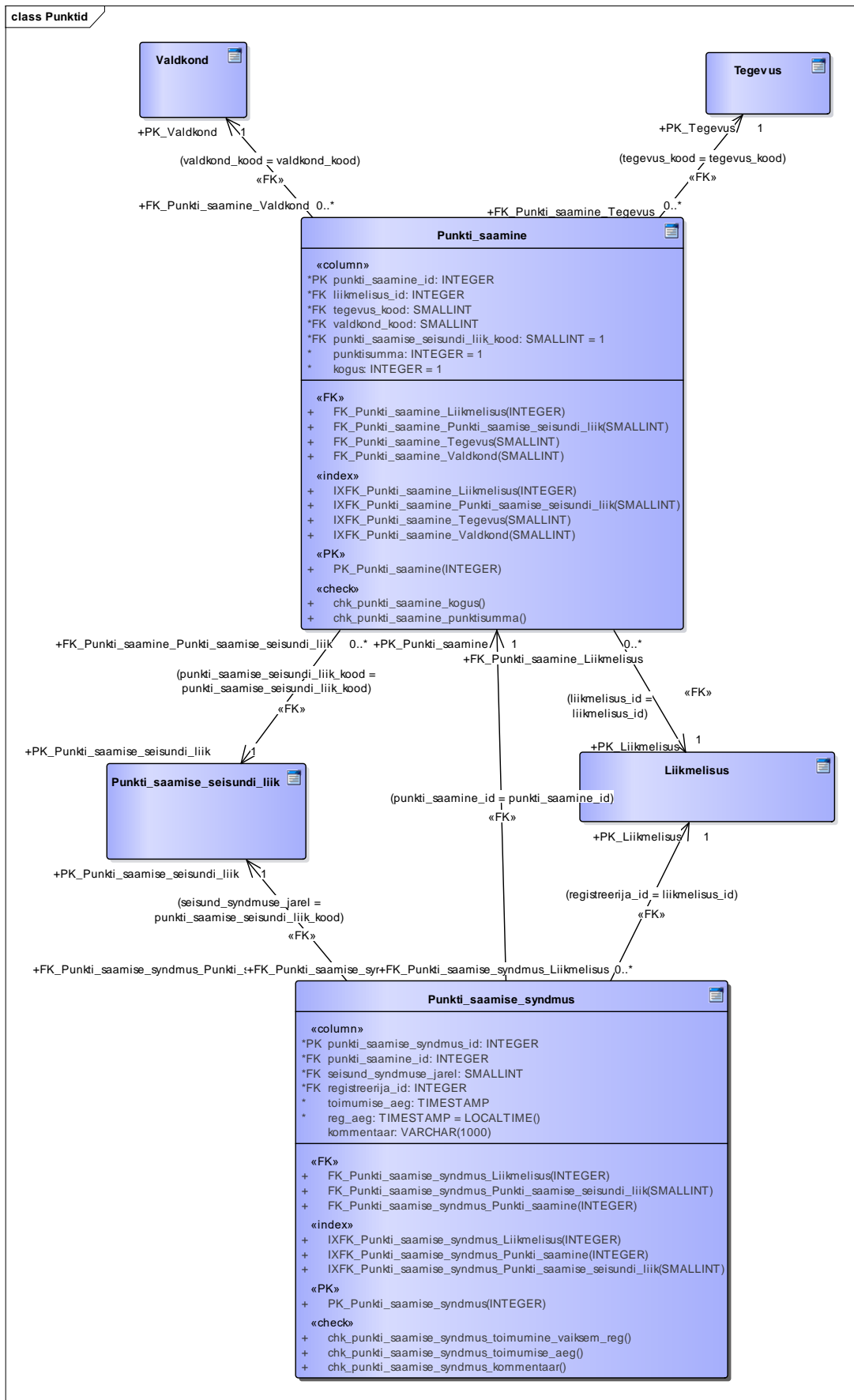
- Peab olema mingis kindlas vahemikus (alguse- ja lõpu aeg kaasa arvatud).
- Ei tohi olla varasem selle atribuudi väärtusest, millele tegevus kronoloogiliselt järgneb.

Arvväärtuste hoidmiseks mõeldud atribuutide kitsendused.

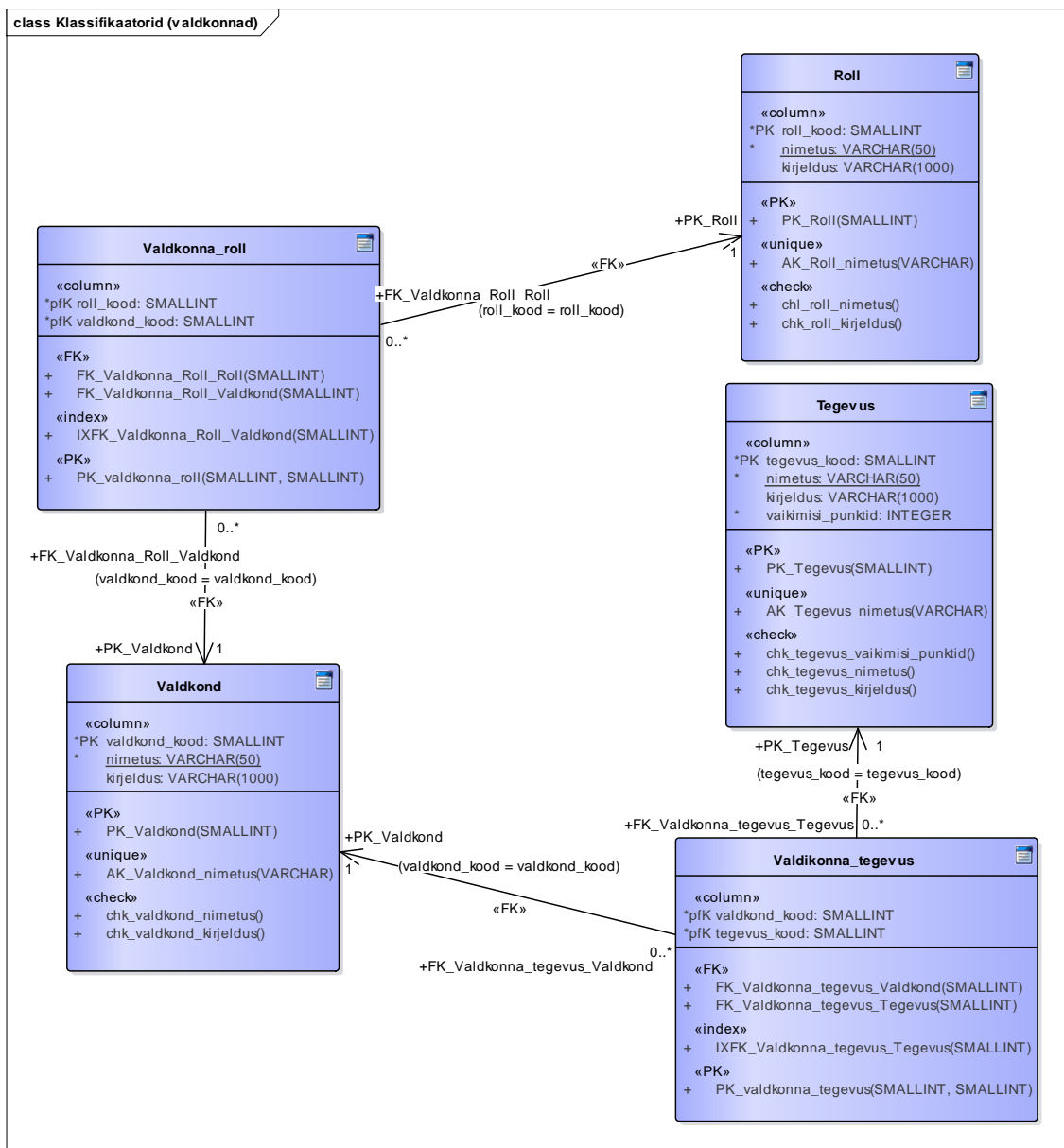
- Peab olema mingis kindlas vahemikus (mis on organisatsiooni vaatepunktist realistlik)

```
CONSTRAINT chk_tegevus_nimetus CHECK (nimetus NOT REGEXP '^[:space:]*$')
CONSTRAINT chk_tegevus_kirjeldus CHECK (kirjeldus NOT REGEXP '^[:space:]*$')
CONSTRAINT chk_punkti_saamise_syndmus_toimumise_aeg CHECK
(toimumise_aeg>=str_to_date('1991-01-01 00:00:00', '%Y-%m-%d %k:%i:%s') AND
toimumise_aeg<str_to_date('2200-01-01 00:00:00', '%Y-%m-%d %k:%i:%s'))
CONSTRAINT chk_punkti_saamise_syndmus_toimumise_aeg_vaiksem_reg CHECK toimumise_aeg <= reg_aeg
CONSTRAINT chk_punkti_saamine_punktisumma CHECK punktisumma >= 1 AND punktisumma < 2000
CONSTRAINT chk_punkti_saamine_kogus CHECK kogus >= 1 AND kogus < 100
```

Joonis 20. Veergudele rakendatud kitsenduste koodi näited.



Joonis 21. Punktide saamiste registri füüsilise disaini täpsusega andmebaasi diagramm



Joonis 22. Punktide saamistega seotud klassifikaatorite füüsilise disaini täpsusega andmebaasi diagramm.

Andmebaasi loodi järgmised indeksid.

- Indeksid primaar- ja unikaalsuse kitsendustega hõlmatud veergudele loodi indeksid automaatselt andmebaasisüsteemi poolt.
- Indeksid välisvõtmetele lisati autori poolt. Välisvõtme veergude indekseerimisel jälgiti seda, et lisatud indeksid ei hakkaks dubleerima automaatselt loodud indeksid.

Kõik loodud andmebaasiobjektid paigutati ühte skeemi koos Stella Leego poolt loodud liikmelisusega seotud registritega. Loodud andmebaasis on kokku 16 baastabelit, milles on kokku 72 veergu.

5 Arendusprotsess

Antud peatükis kirjeldatakse arendusprotsessi kulgu, selgitatakse kasutatud metoodikaid ja tööriistu ning analüüsitakse tehtud arendustööd.

5.1 Metoodika

Süsteemi arendati koskstiilis [9] koos tagasipöördumisega varasematesse etappidesse selles mõttes, et kõigepealt allsüsteemi analüüsi, siis disainiti ja siis asuti ehitamise juurde. Kui mingis etapis tekkis täiendusi, mis mõjutavad eelmise etapi tulemusi (näiteks disainimise käigus avaldusid uued nõuded), siis pööruti tagasi eelmisesse etappi, et teha täiendusi selle etapi tulemustesse.

Eesmärkideni jõudmiseks analüüsi kõigepealt olemasolevat punktisüsteemi tarkvara ja olemasolevat kirjandust. Süsteemianalüüs oli andmekeskne selles mõttes, et kõigepealt loodi kontseptuaalne andmemudel ning allsüsteemi põhiobjekti seisundidiagramm ning sealt saadud info oli üks sisend funktsionaalsete nõuete kirjapanemisele.

Töö ühe tulemusena valmisid CASE-vahendis andmebaasi füüsilise disaini diagrammid, mille põhjal CASE vahend genereeris SQL lauseid. Loodud andmebaas realiseeriti domeenide ja serverite pilveteenust pakkuvas <https://www.zone.ee/et/> keskkonnas. Loodud andmebaasi andmete sisestamiseks loodi PHP, JavaScript ja HTML programmeerimiskeeli kasutades veebirakendus.

Loodavate mudelite valimisel ning loomisel kasutati näitena Tallinna Tehnikaülikooli õppeaine „Andmebaasid II“ näiteprojekti „Ülikooli infosüsteemi vastuvõtuaegade allsüsteem“ [14] ja andmebaasi ning andmebaasirakenduse kavandamise ja realiseerimise (analüüsist ehitamiseni) kohta käivat mustripõhist juhendit [15].

5.2 Tööriistade kirjeldus

Infosüsteem modelleeriti visuaalses üldotstarbelises modelleerimiskeeles UML (*Unified Modeling Language*) [10], luues järgnevat tüüpi diagramme.

- Olemi-suhte diagrammid ja andmebaasi disaini diagrammid klassidiagrammidena.

- Põhiobjekti võimalikud elutsüklid olekudiagrammina (seisundidiagrammina).

CASE vahendina oli kasutusel Enterprise Architect (EA). EA võimaldas genereerida kontseptuaalsest andmemudelist andmebaasi füüsilise disaini mudeli esialgse versiooni. Peale selle täiustamist sai sellest omakorda genereerida baastabelite loomiseks mõeldud SQL laused.

Videokoosolekuteks nii juhendaja, kui ka BEST-Estonia infosüsteemi liikmelisuse allsüsteemi arendaja Stella Leegoga kasutati Microsoft Teams teenust.

Andmebaasi ja koodi repositooriumina ning realiseerimiseks oli kasutusel Zone pilveteenust pakkuv keskkond.

Andmebaasisüsteemina oli kasutusel MySQL (5.7). Veebirakenduse tegemiseks oli kasutusel PHP (7.4).

Rakenduse arendamise algusfaasis kasutati XAMPP platvormi [11]. Kuna süsteemi praegune versioon on samuti loodud XAMPP platvormil, võimaldas selle kasutus eeltöö käigus saada paremini aru tagarakenduse struktuurist ja arhitektuurist andes sisendit uue versiooni arendamisele.

5.3 Tagasivaade arendusele

Iga projekti juures on oluline hiljem ka tagasi vaadata sellele, mida töö raames tehti ning õppida vigadest, et järgmine kord paremini teha.

Tagasivaatavalt oleks ajaplaneerimine pidanud olema parem ning tööga oleks tulnud alustada varem. Arendamiseks oleks võinud kasutada T. Normani poolt välja pakutud agiilse väljalaske planeerimise meetodit [3], mille kohaselt arendus (uue väljalaskeni jõudmine) on jaotatud arenduse pisitsükliteks e iteratsioonideks.

Töö esitamise ajaks jõuti paraku realiseerida vaid personalijuhiga seotud kasutuslood, kuid mitte koordinaatoriga seotud kasutuslood. Samuti jäi realiseerimata üldise avaliku statistikalehe sidumine loodud andmebaasiga.

Halva ajaplaneerimise tõttu ei jõudnud ka loodud lahendust piisavalt testida, sh luua ühikteste.

6 Arendusvaade

Töö tulemusena valmis andmebaas ning osaliselt rakendus, kuid BEST-Estonia töös ei ole neid veel kasutusele võetud. Plaan on siiski loodud tarkvara päriselt kasutusele võtta, kuid selleks tuleb veel tööd teha ning esmajärjekorras realiseerida koordinaatorite funktsionaalsus, kanda andmed vanast (failipõhisest) andmebaasist uude ning siduda üldise statistika lehekülg uue andmebaasiga.

Käesoleva lõputöö autori hinnangul tuleks valminud tarkvara edasi arendada, et saavutada kõik punktide saamiste funktsionaalse allsüsteemi eesmärgid. Järgmistes väljalasetes oleks autori hinnangul mõistlik keskenduda kõigepealt olemasoleva rakenduse parandamisele, mitte kohe uue funktsionaalsuse lisamisele. Võttes arvesse lisaks oma mõtetele ka organisatsiooni personalijuhi tagasisidet eelmise versiooni ja uue versiooni kohta, pakub autor välja järgmised ideed.

- Hetkel kasutuses olevatest Google Sheets töölehtedest tuleks andmed migreerida uue versiooni andmebaasi, et süsteemi uues versioonis oleks näha ka vana versiooni kaudu lisatud andmed. Enne andmete migreerimist tuleks teha põhjalik eeltöö sobivate abivahendite kohta ning mõelda läbi, milliseid andmeid on mõistlik andmebaasis pikaajaliselt säilitada ja milliseid mitte. Migreerimise käigus võib olla vaja andmeid teisendada, et need sobituksid uue versiooni andmebaasiga.
- Veebilehel <https://punktid.best.ee/> kuvatav üldine statistika lehekülg tuleks panna kasutama uue versiooni andmebaasis olevaid andmeid, säilitades samas BESTi visuaalsest identiteedist tulenevaid visuaalseid elemente.
- Korrastada Zone koodirepositooriumis paiknev rakenduse kood ning eemaldada sealt vanadest versioonidest pärinevad failid ja nende sõltuvused, et muuta tarkvara tulevikus kergemini hallatavaks ja arusaadavamaks.
- Loodud veebirakenduse statistika kuvamise leht muuta visuaalselt atraktiivsemaks ning kasutaja jaoks arusaadavamaks, kasutades CSS (*Cascading Style Sheets*) skripte [12].

- Katta loodud veebirakendus ühik- ja integratsioonitestidega, et muuta loodud tarkvara kvaliteetsemaks ja töökindlamaks.
- Kui kõik eelnevad parandused on tehtud, siis võib keskenduda uue funktsionaalsuse tuvastamisele ja realiseerimisele.

7 Kokkuvõte

Käesoleva töö eesmärgiks oli kavandada ja realiseerida MTÜ BEST-Estonia punktisüsteemi uus tarkvara versioon. Lõputöö tulemusena valmis andmebaasi ja osalise rakenduse esimene versioon. Töö tulemusena realiseeriti andmebaas MySQL andmebaasisüsteemi kasutades ning loodi osaliselt seda andmebaasi kasutatav veebipõhine andmebaasirakendus. Töö esitamise ajaks ei ole loodud lahendust veel kasutusele võetud, kuid soov on seda võimalikult kiiresti teha. Eelnevalt tuleb realiseerida koordinaatorite funktsionaalsus, kanda andmed vanast (failipõhisest) andmebaasist uude ning siduda üldise statistika lehekülg uue andmebaasiga.

Punktisüsteemi tarkvara on üks osa laiemast BEST-Estonia infosüsteemist. Valminud punktihalduse tarkvara realiseerib selle infosüsteemi äriarhitektuuris ettenähtud punktide saamiste funktsionaalset allsüsteemi ja punktide saamiste registrit.

Töö käigus uuriti eeltööna olemasoleva tarkvara ülesehitust ja uue lahenduse otsimise põhjuseid. Taustatööna – võimalike probleemide ja lahenduste leidmiseks – uuriti ka seda, kuidas on lahendatud punktide kogumine mõningate lähiriikide BESTi gruppides. Seejärel koguti nõudeid, valiti sobiv andmebaasisüsteem ning loodi punktide saamiste allsüsteemi andmebaas ja veebirakendus andmete sisestamiseks.

Töö käigus tehti koostööd üliõpilane Stella Leegoga, kelle samal ajal valminud lõputöö nimetas muuhulgas BEST-Estonia infosüsteemi allsüsteemid ning käsitles detailselt liikmelisuste funktsionaalset allsüsteemi ja registrit. See oli oluline sisendinfo käesoleva töö tegemiseks, sest andmebaasi mõttes on punkti saamine seotud liikmelisusega ja selle kaudu isikuga.

Kasutatud kirjandus

- [1] Database vs Spreadsheet [WWW] <https://365datascience.com/sql-why-databases/> (30.12.2020)
- [2] Comprehensive access to spreadsheets [WWW] <https://developers.google.com/sheets/api> (01.01.2021)
- [3] Agile Release Planning 101, T.Norman [WWW] <http://tommynorman.blogspot.com/2012/09/agile-release-planning-101.html> (01.01.2021)
- [4] GET vs. POST [WWW] <https://www.diffen.com/difference/GET-vs-POST-HTTP-Requests> (30.12.2020)
- [5] User-stories [WWW] <https://www.mountaingoatsoftware.com/agile/user-stories> (31.12.2020)
- [6] How to write clean code? Lessons learnt from „The Clean Code“ – Robert C. Martin [WWW] <https://medium.com/mindorks/how-to-write-clean-code-lessons-learnt-from-the-clean-code-robert-c-martin-9ffc7aef870c> (02.01.2021)
- [7] Connecting to and querying a MySQL database [WWW] <https://phpbestpractices.org/#mysql> (01.01.2021)
- [8] Understanding Dependency Injection [WWW] <https://php-di.org/doc/understanding-di.html> (02.01.2021)
- [9] SDLC – Waterfall Model [WWW] https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm (01.01.2021)
- [10] Unified Modeling Language. [WWW] <https://www.uml-diagrams.org/> (01.01.2021)
- [11] XAMPP [WWW] <https://en.wikipedia.org/wiki/XAMPP>
- [12] CSS: Cascading Style Sheets [WWW] <https://developer.mozilla.org/en-US/docs/Web/CSS> (02.01.2021)
- [13] Differences between branding, brand and visual identity [WWW] <https://www.antevenio.com/usa/differences-between-branding-brand-and-visual-identity/> (03.01.2021)
- [14] Eessaar, E. Ülikooli infosüsteemi vastuvõtuaegade allsüsteem. Näiteprojekt õppeaines “Andmebaasid II”. [Online] TalTech Tarkvarateaduse instituut (28.12.2020)
- [15] Eessaar, E. Iseseisva töö tegemise mustripõhine juhend. Näidisjuhend õppeaines “Andmebaasid I”. [Online] TalTech Tarkvarateaduse instituut (01.01.2021)
- [16] ECMAScript [WWW] <https://en.wikipedia.org/wiki/ECMAScript> (02.01.2021)
- [17] DB-Engines Ranking [WWW] <https://db-engines.com/en/ranking> (01.01.2021)
- [18] The Must-Know Clean Code Principles [WWW] <https://medium.com/swlh/the-must-know-clean-code-principles-1371a14a2e75> (01.01.2021)

- [19] BESTi punktisüsteemi arhitektuur [WWW] <https://punktid.best.ee/arhitektuur> (02.01.2021)
- [20] Isikuandmete kaitse seadus [WWW] <https://www.riigiteataja.ee/akt/104012019011> (03.01.2021)
- [21] Leego., S. Üliõpilasorganisatsiooni BEST-Estonia liikmelisuse arvestuse tarkvara kavandamine. Tallinna Tehnikaülikooli Tarkvarateaduse instituut. Bakalaureusetöö. (2021).
- [22] UTF-8 [WWW] <https://en.wikipedia.org/wiki/UTF-8> (31.12.2020)
- [23] MySQL System Properties [WWW] <https://db-engines.com/en/system/MySQL> (02.01.2021)
- [24] PostgreSQL System Properties [WWW] <https://db-engines.com/en/system/PostgreSQL> (02.01.2021)

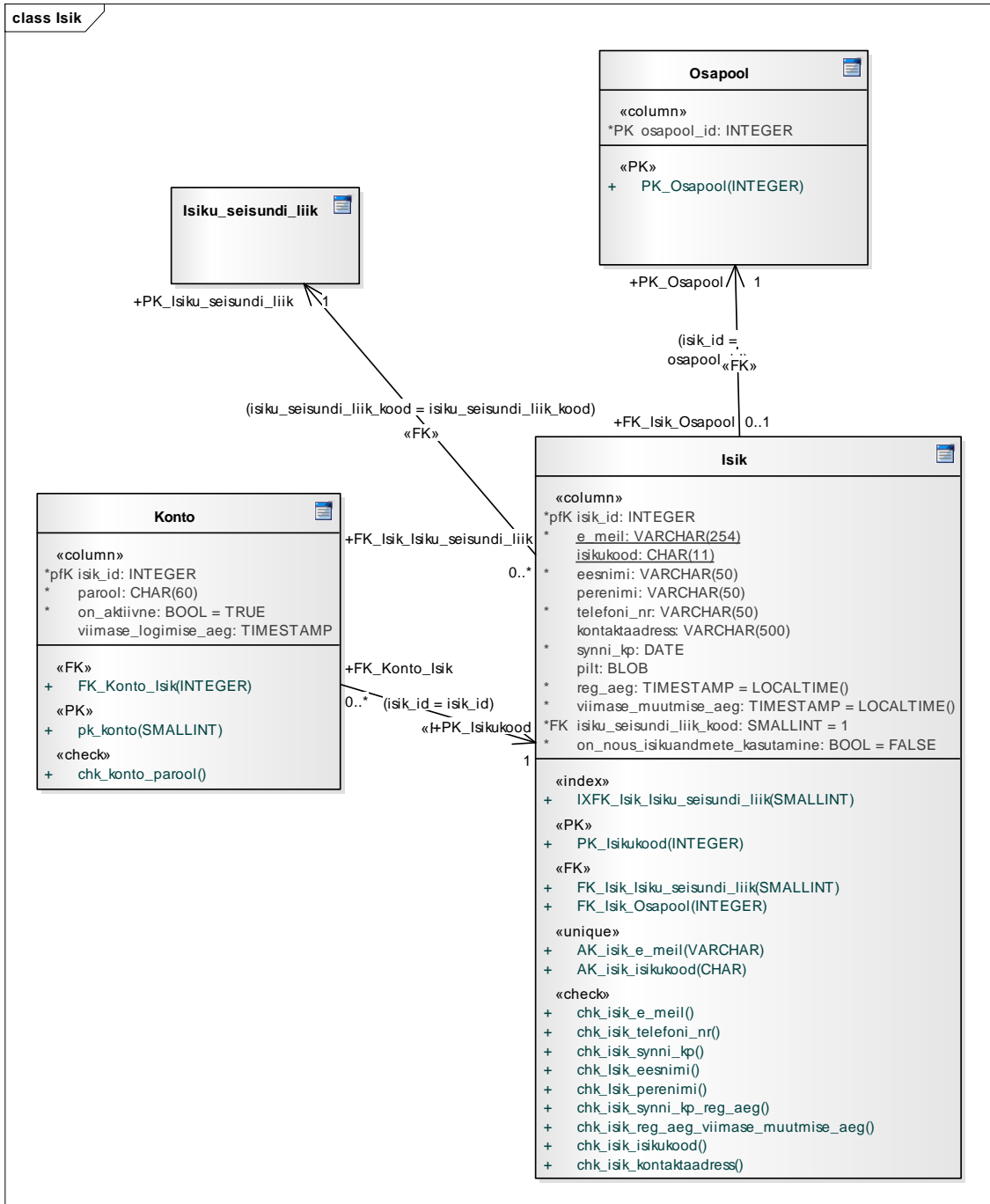
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Karl Henri Kaljus

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "[Lõputöö pealkiri]" , mille juhendaja on [Juhendaja nimi]
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

31.12.2020

Lisa 2 Tabelid Isik ja Roll BEST-Estonia infosüsteemi uues versioonis



Lisa 3 Tabelid Liikmelisus, Rolli_omamine ja Roll BEST-Estonia infosüsteemi uues versioonis

