

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Hannes Härm 185718IADB

**Üliõpilasorganisatsiooni BEST-Estonia liikmelisuse
ja punktisüsteemi arvestuse tarkvara arendamine**

Bakalaureusetöö

Juhendaja: Erki Eessaar

Tallinn 2023

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: Hannes Härm

16.05.2023

Annotatsioon

Käesoleva töö eesmärk on üliõpilasorganisatsiooni BEST-Estonia liikmelisuse ja punktihaldussüsteemi tarkvara loomine. Olemasolev tehniline lahendus liikmete ja punktide andmete haldamiseks on ajale jalgu jäänud ja vajab väljavahetamist. Varasemastes selleteemalistes lõputöödes on kogutud nõudeid tarkvarale, kavandatud ja realiseeritud uus andmebaas ja loodud esialgne kasutajaliidese prototüüp. Samuti on loodud skriptid andmete ülekandmiseks uude andmebaasi. Selleks, et uus süsteem oleks võimalik BEST-Estonias kasutusele võtta, on vaja realiseerida veebipõhine andmebaasirakendus.

Selleks uuendatakse nõudeid, analüüsitakse populaarsemate raamistike sobivust rakenduse arendamiseks ning antakse ülevaade kasutajaliidese parimatest praktikatest. Seejärel kirjeldatakse süsteemi täiendused, planeeritakse rakenduse arenduse iteratsioonid ning luuakse uus rakendus. Lõpuks analüüsitakse töö tulemust kasutatavuse valideerimise näol.

Töö tulemusena on valminud uus veebipõhine andmebaasirakendus, mille lähtekood on BEST-Estoniale üle antud. Rakendus on kirjutatud kasutades PHP keelt ja Laraveli raamistikku. Rakenduse kasutusele võtmiseks tuleb uuendada üldiste punktide statistika veebilehte, et see andmeid uuest andmebaasist võtaks ning teha koostööd eelmise lõputöö autoriga andmete migratsiooni osas, sest tema poolt tehti andmete migratsiooni skriptid.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 88 leheküljel, 11 peatükki, 23 joonist, 1 tabelit.

Abstract

Developing a Membership Management and Point Management Software of the Student Organization BEST-Estonia

The goal of this bachelor thesis is to develop a new web application for points and membership management for student organization BEST-Estonia. The existing software is old and needs replacement. In previous theses, software requirements have been collected, a new database has been designed and implemented, and an initial user interface prototype has been created. Scripts for data transfer to a new database have also been created. To be able to use the new system in BEST-Estonia, it is necessary to create a new web-based database application.

The thesis starts with a summary of the current situation and the problems described in the previous theses. To update the requirements the BEST-Estonian Vice President or Human Resources was closely included in the process of developing the new web application. After that an analysis was made to find the best hosting environment and programming framework for the PHP web application. Zone was chosen as the hosting environment and partly due to that choice Laravel was selected as the programming framework because it requires minimal efforts from the client side. The system uses MariaDB database management system. To develop a usable and efficient web application, the best practices in developing a good user interface and experience were listed.

To start the development of the new web application, it was necessary to update the already done work with the system development. Updates were made in use cases, in the database design, and in the previously developed user interface prototypes. The prototypes were also updated to meet the accessibility requirements that were set in this thesis. After these preparations the development process started with planning the iteration process. The development process was divided into three iterations, each having roughly the same complexity. After that, the web application was developed according to the iteration plan. The thesis also describes the core architecture of the developed web application.

As a result of the work, the source code of the developed application has been handed over to BEST-Estonia. To use the new web application, some work still must be done. The points statistics web page needs to be updated to use the new database. After that, a cooperation with the author of the previous thesis is required to adapt the data migration scripts to the current state of the new database and migrate the data to the new database. With this the web application should be usable and BEST-Estonia should be able to write off the old systems for member management and point management.

The thesis is in Estonian and contains 88 pages of text, 11 chapters, 23 figures, 1 table.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> . Rakendusliides
BEST-Estonia	<i>Boards of European Students of Technology</i> , rahvusvahelise tudengiorganisatsiooni Eesti haru
CSS	<i>Cascading Style Sheets</i> , veebilehe disaini kirjeldav keel
CQRS disain	<i>Command and Query Responsibility Segregation</i> – muster, mis eraldab andmete lugemis- ja värskendamistoimingud
Discord	Suhtlusplatvorm, mis võimaldab pidada hääle-, video- ja tekstivestlusi
Git	Versioonikontrolli tarkvara, mida põhiliselt kasutatakse lähtekoodi versioneerimiseks
GitHub	Majutusteenus tarkvara arendamiseks ja versioonikontrolliks, kasutades Git'i
Google Forms	Pilvepõhine andmete kogumiseks mõeldud sisestusvormide koostamise tarkvara
Google Sheets	Pilvepõhine tabelarvutusprogramm
HTTP-päring	<i>Hypertext Transfer Protocol</i> protokolliga tehtav internetipäring
IDE	<i>Integrated development environment</i> – integreeritud arenduskeskkond; programm, mis lihtsustab lähtekoodi kirjutamist
JavaScript	Programmeerimiskeel, mida põhiliselt kasutatakse kasutajaliidese arendamisel.
KISS	<i>Keep it simple, stupid</i> – hea programmeerimise tava
Modaal	Veebilehe element, mis kuvatakse kogu muu lehe sisu peal ja kasutatakse üldiselt teavituse edastamiseks, kinnituse küsimiseks, vormi- või detailvaadete jaoks.
MTÜ	Mittetulundusühing
MVC	<i>Model-View-Controller</i> – tarkvara arhitektuurimuster, mille eesmärgiks on hoida eraldi rakenduse äri loogika, vaated ning andmemudelid
MariaDB	Andmebaasisüsteem, mis põhineb MySQL-il, SQL- andmebaasisüsteem
ORM	<i>Object-Relational-Mapping</i> – vahelüli objektorienteeritud programmeerimiskeele ja SQL-andmebaasi vahel
PhpStorm	JetBrains arendatud IDE, mis on mõeldud veebirakenduste arendamiseks kasutades põhiliselt PHP programmeerimiskeelt
PHP	Programmeerimiskeel veebirakenduste loomiseks

POLA	<i>The principle of least astonishment</i> , hea programmeerimise tava
Powers of 2	Scrumis kasutatav kasutuslugude keerukuse hinnete jada, kus hinded kasvavad eksponentsiaalselt: 1, 2, 4, 8, 16...
Programmiviga	Vigane koht lähtekoodis, mille tõttu ei käitu rakendus korrektselt
Refaktoreerimine	Lähtekoodi korral selle struktuuri muutmine ja korrastamine, ilma uute funktsionaalsuste lisamiseta
Scrum	Tarkvara arendusmetoodika, mis kuulub paindmetoodikate hulka
SKY koosolek	BEST-Estonia igakuine koosolek
SOLID printsiip	Mnemooniline akronüüm viiele objektorienteeritud programmide disainipõhimõttele: <i>Single-responsibility principle</i> , <i>Open-closed principle</i> , <i>Liskov substitution principle</i> , <i>Interface segregation principle</i> , <i>Dependency inversion principle</i>
SQL	<i>Structured Query Language</i> , andmebaasikeel
SQL päring	SQL andmebaasist andmete küsimise lause, SELECT lause
T-särkide suurused	<i>T-Shirt sizes</i> . Scrumis kasutatav kasutuslugude keerukuse hinnete jada, kus keerulisemad ülesanded on näidatud ära suurema "suurema" suuruse järgi: S, M, L, XL...
YAGNI	<i>You Aren't Gonna Need It</i> , hea programmeerimise tava

Sisukord

1 Sissejuhatus	13
1.1 Töö taust ja lühikirjeldus	13
1.2 Ülevaade töö struktuurist	14
1.3 Töö tegemise protsess	14
2 Probleem ja töö eesmärk.....	16
3 BEST-Estonia olemasoleva tarkvara olukorra analüüs	17
3.1 Kokkuvõtte eelnevate tööde põhjal.....	17
3.2 BEST-Estonia 2022/2023. perioodi personalijuhi kokkuvõtte hetkeolukorrast	18
3.2.1 Liikmete andmete haldus	18
3.2.2 Uued nõuded	19
3.2.3 Loodava funktsionaalsuse probleemid	22
4 Sobivaima veebirakenduse raamistiku valimine	24
4.1 Arvestamine BEST-Estonia võimekustega.....	24
4.2 PHP raamistiku valimine	25
4.2.1 Laravel.....	25
4.2.2 CodeIgniter.....	26
4.2.3 Symfony	26
4.3 Kasutatava raamistiku valimine	27
5 Kasutatavad kasutajaliidese parimad praktikad.....	28
5.1 Kasutatavus	28
5.2 Kasutajaliidese järjepidevus ja tuttavlikkus	29
5.3 Informatsiooni struktureeritus.....	30
5.4 Värvikontrastid	31

5.5 Ligipääsetavus.....	32
6 Varem tehtud töö täiendused enne rakenduse arendusega alustamist	34
6.1 Muutused funktsionaalsetes nõuetes	34
6.2 Täiendused kasutajaliidese kavandites	37
6.2.1 Taaskasutatavad komponendid	37
6.2.2 Liikmehaldus	38
6.2.3 Punktihaldus	44
6.2.4 Mobiiltelefoniga kasutatavus	49
6.3 Andmebaasi täiendused	50
7 Iteratsioonide planeerimine	52
7.1 Prioriteetide määramine	52
7.2 Keerukuse hindamine.....	53
7.3 Funktsionaalsete nõuete jaotamine iteratsioonide vahel.....	53
8 Veebirakenduse arendus	55
8.1 Arhitektuur	55
8.1.1 Laraveli MVC arhitektuur	55
8.1.2 Rakenduse tõlkimine	56
8.1.3 CQRS disaini kasutamine.....	57
8.1.4 Klassifikaatorite andmebaasist pärimise vältimine	60
8.2 Järgitud arendustavad.....	61
8.3 Kasutatud teegid.....	63
9 Valideerimine	65
10 Arendusvaade ja rakenduse edasi arendamine	68
11 Kokkuvõte	70
Kasutatud kirjanduse loetelu	71

Lisa 1 – Lihtlitsents	74
Lisa 2 – Olemasolevas kasutajaliidese prototüübis kasutatavate värvide kontrastisuhted ...	75
Lisa 3 – Mõned mobiilivaate kasutajaliidese prototüübid.....	76
Lisa 4 – Tabelid Liikmelisus ja Liikmelisuse_huviala BEST-Estonia andmebaasi uues versioonis.....	78
Lisa 5 – Tabelid Isik, Konto ja Osapool BEST-Estonia andmebaasi uues versioonis	79
Lisa 6 – Uuenenud liikmelisuse seisundidiagramm	80
Lisa 7 – Muudetud kasutusjuhtude diagramm.....	81
Lisa 8 – Funktsionaalsuste hindamine	82
Lisa 9 – Käsu realisatsioon tegevuse kustutamise näitel	85
Lisa 10 – Enumeration klassi definitsioon	86
Lisa 11 – Mõned kuvatõmmised valminud rakenduse vaadetest	87

Jooniste loetelu

Joonis 1. Liikme huvialade nimekiri loendina.....	30
Joonis 2. Liikme huvialade nimekiri komaga eraldatuna.	31
Joonis 3. Taaskasutatavad komponendid.....	37
Joonis 4. Liikmete nimekiri.	39
Joonis 5. Liikmeandmete detailvaade.....	40
Joonis 6. Uue liikmemuutuse sündmuse lisamise modaal.....	41
Joonis 7. Isikuandmete muutmise vaade.	42
Joonis 8. Liikmeavalduse vorm uuele liitujale.	43
Joonis 9. Liikmeavalduse otsustamise modaal.	44
Joonis 10. Punktikannete vaade.....	45
Joonis 11. Punktide lisamise modaal.	46
Joonis 12. Punktide statistika vaade.	47
Joonis 13. Liikme punktide kannete detailvaade.....	48
Joonis 14. Tegevuste nimekirja vaade.....	49
Joonis 15. Näide eestikeelse tõlkesõnastiku faili sisust.....	56
Joonis 16. Tõlkeressursi kasutamise näide.	57
Joonis 17. Päringu keele seadistamine.	57
Joonis 18. Käsuobjekti kasutamise tegevuse kustutamise näitel.	59
Joonis 19. Käsu vahendaja meetod.....	59
Joonis 20. Klassifikaatori klassi definitsioon <i>Isiku_seisundi_liik</i> tabeli põhjal.	60
Joonis 21. Tühja konstruktori kasutamine uute liikmeavalduste olemasolu kontrolli näitel.	62
Joonis 22. Veebirakenduse kontrollimine WAVE tööriistaga.....	67
Joonis 23. Rakenduse kiiruse testimine.....	67

Tabelite loetelu

Tabel 1 Uutest nõuetest loodud kasutuslood.	34
--	----

1 Sissejuhatus

Sissejuhatuses antakse ülevaade töö taustast ning struktuurist ning kirjeldatakse käsitlevat probleemi ja töö eesmärki.

1.1 Töö taust ja lühikirjeldus

BEST-Estonia on üliõpilasorganisatsioon, mis on Euroopa tehnikatudengite organisatsiooni (BEST) Eesti allharu.

Iga organisatsioon vajab oma tööks infosüsteemi. Kaasaegne infosüsteem on arvutisüsteemi toega. Ka BEST-Estonial on selline süsteem, kuid selle pakutav IT-tugi võiks olla olemasolevast palju parem. Üliõpilasorganisatsiooni BEST-Estonia kolme liikme – Karl Henri Kaljuse, Stella Leego ja Jorgen Ööpiku lõputööde tulemusel on valminud alamosa infosüsteemi uuest andmebaasist koos migratsiooni skriptidega vanadest andmebaasidest uude [1]. Uue süsteemi eesmärgiks on parandada ja kiirendada organisatsiooni liikmelisuse [2] ja punktide haldust [3]. Stella Leego ja Karl Henri Kaljuse lõputööde tulemusel valmis ka uue süsteemi funktsionaalsete nõuete nimekiri ja kasutajaliidese prototüüp.

Hetkel toimub andmete kogumine samal viisil, mis on kirjeldatud Jorgen Ööpiku lõputöö sissejuhatuses: K.H. Kaljuse lõputöö raames valminud punktisestusrakenduse prototüüpi ei kasutata ja punktiandmeid sisestatakse siiani Google Sheets kaudu ning liikmete andmeid hoiustatakse punktisüsteemist eraldi MariaDB andmebaasis *zone.ee* virtuaalserveris. Sellest tulenevalt on andmete kvaliteet tänase päevani väga madal ning andmete sisestamine ja uuendamine on keerukas.

Käesoleva töö käigus analüüsitakse uusi ärivajadusi, mis jäid erinevatel põhjustel eelnevatest lõputöödest välja, disainitakse ja arendatakse BEST-Estonia liikmehalduse ja punktisüsteemi andmebaasirakendust ning analüüsitakse edasisi vajalikke tegevusi seoses süsteemi arendamisega.

1.2 Ülevaade töö struktuurist

Käesoleva töö raames antakse ülevaade liikmehalduse tarkvara hetkeolukorrast eelnevate lõputööde põhjal ning ka 2022/2023. perioodi BEST-Estonia personalijuhi antud tagasiside abil, et võrrelda hetke ärivajadusi eelnevates töödes välja toodutega. Lõputöö tulemusena on valminud uus andmebaasirakendus, mille kasutuselevõtt nõuab punktide statistika lehe uuendamist ning koostööd Jorgen Ööpikuga andmete migreerimise osas.

Peatükis 2 kirjeldatakse töö eesmärk ning probleem, tehes eelnevate tööde põhjal olemasolevatest probleemidest kokkuvõtte. Peatükis 3 kirjeldatakse tarkvara praegust olukorda, tehakse sellest eelnevate tööde põhjal kokkuvõtte ning kirjeldatakse uuenenud nõudeid süsteemile. Peatükis 4 leitakse parim keskkond rakenduse hostimiseks ja raamistik, millel rakendus arendada. Peatükis 5 kirjeldatakse parimad tavad, mida järgida kasutajaliidese kavandamisel ning kasutatavuse tagamisel. Peale seda kirjeldatakse muudatused süsteemis vastavalt muutunud ärinõuetele (Peatükk 6), sh koostatakse uued kasutajaliidese mallid. Peatükis 7 planeeritakse iteratsioonid, prioritseerides selleks realiseerimist vajavaid funktsionaalsuseid.

Arendusprotsessi esitavas osas (Peatükk 8) tuuakse välja kasutatud arendusmetoodika ja analüüsitakse protsessi, kuidas tööd tehti. Töö viimase osana analüüsitakse töö tulemust, valideerides rakenduse kasutatavust ning efektiivsust (Peatükk 9) ning kirjeldatakse uue süsteemi kasutusele võtmiseks vajalikud tegevused ja antakse soovitusi süsteemi edasiarendamiseks (Peatükk 10).

1.3 Töö tegemise protsess

Rakendus arendatakse PHP keeles Laravel 10 raamistikul. Projekt arendatakse PhpStorm IDE'ga ning lähtekood hoiustatakse GitLab keskkonnas. Kasutajaliidese prototüübid disainitakse tööriista Figma abil.

Rakendus arendatakse iteratiivselt, kus funktsionaalsuste arendamised jaotatakse erinevate iteratsioonide vahel laiali. See aitab kiiremini saada tehtud tööle tagasisidet ning lihtsustab vajadusel muudatuste tegemist.

Iga iteratsiooni alguses koostatakse antud iteratsiooni jaoks vajalikud kasutajaliidese prototüübid Figma tarkvaraga, sest sama tööriistaga on tehtud ka olemasolevad kasutajaliidese prototüübid. Antud prototüüpidele kogutakse BEST-Estonia esindaja tagasisidet ning vastavalt vajadusele prototüüpe muudetakse. Seejärel vaadatakse koostatud kasutajaliideseid ja kasutuslood üle ning pannakse ülesanded väiksemate osadena GitHub ülesannetena (*issue*) kirja. See võimaldab töö autoril keskenduda paremini ühele konkreetsele ülesandele ning annab ülevaate käesoleva iteratsiooni sisust ja seisust.

Kvaliteetse koodi kirjutamiseks tehakse iteratsioonide jooksul mitmeid koodi puhastamisi ja refaktoreerimisi, et lihtsustada uute funktsionaalsuste lisamist. Selle abil on samuti võimalik koodist leida mitmeid peidetud programmivigu.

Iga iteratsiooni lõpus vaadatakse valminud funktsionaalsused koos BEST-Estonia esindajaga üle ning planeeritakse vajadusel järgmisesse iteratsiooni uued vajaminevad funktsionaalsused või muudatused.

Rakenduse kasutatavuse ja töökiiruse testimiseks kasutatakse veebilehitseja pistikprogramme.

2 Probleem ja töö eesmärk

MTÜ BEST-Estonia liikmete andmete ja ligipääsude haldus on hajutatud erinevate süsteemide vahel ja vajab palju käsitööd. Lisaks ei ole süsteemides realiseeritud andmete kvaliteedi kontrollimist ja tagamist, mistõttu ei ole olemasolevate andmete kvaliteet piisavalt hea. Seetõttu ei ole näiteks võimalik vaadata täpsemaid andmeid liikmete kohta mingi konkreetse ajahetke seisuga.

Stella Leego töös leiti, et olemasoleva süsteemi põhiliseks probleemiks on andmete hajutamine, mille tagajärjel on väga raske isikuandmete regulatsioone täita. Karl-Henri Kaljuse töös leiti analüüsi tulemusena, et olemasolev süsteem punktide sisestamiseks ja haldamiseks on väga problemaatiline, kusjuures osad probleemid tulenevad Google Sheets tabelarvutussüsteemi piirangutest.

Andmete kvaliteedi langus tuleneb põhiliselt erinevate süsteemide kasutamisest. Sellest tekib andmete dubleeritus ja kontrollimatu andmete liiasus, mis tähendab, et andmemuudatust ühes kohas ei kanta automaatselt üle koopiatesse. See tingib igas süsteemis andmete käsitsi kontrollimise ja parandamise vajaduse. Lisaks tekitab probleeme andmetele seatud piirangute kontrollimise puudumine olemasolevates süsteemides. Seetõttu saab näiteks punktisüsteemis sisestada erinevas formaadis kuupäevi.

Selleks, et lahendada eelnimetatud probleemid, on vaja valmis teha uus andmebaasirakendus, milles on võimalikult palju toiminguid lihtsustatud ja automatiseeritud. Kuna üliõpilasorganisatsiooni infosüsteemi tarkvara oleks tervikuna väga mahukas ning nõuaks tuhandeid arendustunde, et lahendada ära kõik olemasolevad probleemid, keskendutakse antud töö raames ainult liikmehaldust ja punktisüsteemi ühendava rakenduse valmimisele. Samal ajal hoitakse tähelepanu all ka vajadust lisada tulevikus juurde uusi funktsionaalsuseid.

3 BEST-Estonia olemasoleva tarkvara olukorra analüüs

Selles peatükis kirjeldatakse BEST-Estonia kasutatava tarkvarasüsteemi seisukorda enne käesoleva arenduse algust.

3.1 Kokkuvõtte eelnevate tööde põhjal

Praegu BEST-Estonias kasutusel olevas andmehalduses on mitmeid probleeme. Suurim probleem on andmete hajutatus mitme süsteemi vahel, mille tagajärjel on organisatsioonil keeruline vastu tulla andmehaldussüsteemidele kehtivatele regulatsioonidele. Lisaks puudub süsteemne lähenemine liikmelisuste haldamisele. Olenemata olemasolevast liikmehaldustarkvarast on praegu kasutatav liikmehaldustarkvara vananenud, ebatõhus ning ei vasta täpselt organisatsiooni ärivajadustele. See toob kaasa ebakõlasid ja vähendab andmete kvaliteeti. Iga personalijuht käsitleb liikmelisust erinevalt ja ilma standardmeetodita pole süsteem jätkusuutlik.

Karl-Henri Kaljuse koostatud tööst selgus, et BEST-Estonia haldab punktidega seotud andmeid Google Sheets tabelarvutussüsteemis. Praeguses lahenduses leiti mitmeid probleeme, millest osa tulenevad Google Sheetsi piirangutest.

- Lahendust on raske hallata ja uuendada, sest selle tegijad ei võta enam organisatsiooni tööst aktiivselt osa.
- Uute kirjade lisamine võtab aega, sest Google Sheets ei tule hästi toime suurte andmemahtudega.
- Lahendus ei ole töökindel ning sageli lakkab töötamast andmete sisestamisel tekkinud vigade tõttu.

Jorgen Ööpiku töö tõi lisaks eelnevatele probleemidele välja andmete valideerimise meetmete rakendamise puudumise. Tarkvara ei saa kontrollida andmete õigsust, kuid süsteemianalüüsi tulemusel on võimalik kindlaks teha reeglid, millele ebaõiged andmed kindlasti ei vasta. Nende reeglite jõustamise kaudu saab vähemalt osa ebaõiged andmeid juba andmete registreerimise käigus üles leida ja sundida andmete registreerijaid sellele

reageerima. Sellised meetmed takistaksid ilmselgelt ebaõigete väärtuste süsteemis registreerimist. Valideerimismeetmete puudumise tagajärjel sisestasid andmesisestajad puudulikke või vigastes formaatides andmeid, mille tõttu kannatab andmete terviklikkus ja õigsus.

BEST-Estonia seisab silmitsi oluliste väljakutsetega, mis on seotud andmekaitse nõuete täitmise, andmete terviklikkuse tagamise ja säilitamise, liikmehalduse ja punktihaldusega. Nende probleemide lahendamiseks ja organisatsiooni protsesside tõhustamiseks on hädasti vaja ühtset, usaldusväärset ja tõhusat lahendust.

3.2 BEST-Estonia 2022/2023. perioodi personalijuhhi kokkuvõtte hetkeolukorrast

BEST-Estonia uue liikmehaldus- ja punktihaldustarkvara arenduse kohta kirjutatud eelnevad lõputööd [1] [2] [3] on juba analüüsinud organisatsioonis antud teemaga seoses esinevaid probleeme ja puuduseid. Enne uue tarkvara arendamist võttis autor ühendust organisatsiooni personalijuhiga, et vajadusel saada vastused eelnevalt vastamata jäänud küsimustele äriloogika kohta. Suhtlemise käigus tulid välja mitmed lisaprobleemid hetkel kasutatava süsteemi juures, mistõttu leiti, et oluline oleks saada personalijuhilt uus tagasiside hetkel kasutatava süsteemi kohta. See on vajalik, et olla kursis BEST-Estonia liikmehalduse nõuete muutustega. Selle jaoks sai töö autor personalijuhiga kokku ning vestles temaga. Vestluse käigus palus töö autor personalijuhil detailsemalt kokku võtta kõige suuremad probleemid liikmete haldamise juures.

3.2.1 Liikmete andmete haldus

Igal aastal liitub BEST-Estoniaga umbes 100 uut liiget. Liikmeavaldus on koostatud tööriista Google Forms abil, mille tulemused võetakse liitumisperioodi lõpus kokku ning iga kirje sisestatakse käsitsi süsteemi. Keskmiselt võtab ühe liikme sisestamine aega umbes kolm minutit ning see teeb kokku umbes viis tundi käsitsi andmesisestamist igal sügisel. Kokkuvõtvalt on tegemist väga pika ja korduva tööga, mida võiks automatiseerida.

Organisatsioon kasutab erinevaid liikmestaatusi, nagu “Beebiliige” ja “Uusliige”, mis annavad organisatsioonisiselt erinevaid sisemisi õiguseid. Samas puudub olemasolevas andmebaasirakenduses lihtne viis nende staatuste muutmiseks, kuna näiteks on puudu muudetava liikme otsing. Lisaks salvestab andmebaas aegunud ja ebaolulisi andmeid, näiteks MSN kasutajatunnus.

Kasutusel on mitmeid eraldi süsteeme, mis nõuavad kasutaja õiguste ja juurdepääsu haldust: punktisüsteem, liikmete andmebaas, meililist, korraldatud projektide kokkuvõtete haldamine ja Discord suhtluskanal, millest viimased kaks on organisatsiooni poolt kasutusele võetud viimase kahe aasta jooksul. See nõuab igas süsteemis muutuste eraldi haldamist, mis on aeganõudev ja muudab isikuandmete regulatsioonide järgimise keerulisemaks. Lisaks kuvab best.ee veebileht hetkel juhatuses olevate liikmete nimesid. Ka selle muutmine käib käsitsi.

Puudub ka võimalus teha vahet mitteaktiivsete ja aktiivsete liikmete vahel. Näiteks on vaja andmeid "SKY" koosoleku jaoks, et tähistada aktiivsete liikmete sünnipäevi ja premeerida aktiivsemaid liikmeid. Praegu eeldab selle info hankimine programmeerimisalaseid teadmisi, kuna kuu kümme aktiivsemat liiget küsitakse SQL-päringu abil. Projektkorraldusmeeskondadele väljastatakse ka aktiivsete liikmete nimekirjad, et aidata neil paremini leida enda projekti korraldamisele abilisi. Lisaks toimub lõpetavate liikmete jälgimine käsitsi. BEST-Estonia reeglite kohaselt ei loeta liiget enam kuus kuud pärast ülikoolist lahkumist aktiivseks.

3.2.2 Uued nõuded

Vestluse käigus leiti muutunud nõuded, kuna eelnevate tööde tegemisest kulunud aja jooksul on organisatsiooni läbiviidavad protsessid muutunud.

BEST-Estonia on peale Stella Leego töös tehtud analüüsi võtnud kasutusele mõned uued süsteemid, näiteks Discord suhtluskanali ja projektide kokkuvõtte andmebaasirakenduse. Need süsteemid vajavad igaüks eraldi rollide haldamist, mis lisab liikmete haldusele veelgi keerukust.

Liikmeavalduse vormis palutakse liitujal näidata ära enda huvialad. Sellega soovib BEST-Estonia teada saada, kuidas paremini uut liiget organisatsiooni kaasata. Näiteks on liitujal võimalik märkida üheks huvialaks „Disain“. Selle järgi oskab organisatsiooni juhatus suunata antud liiget erinevatele disaini- ning turunduskoosolekutele ning töötubadesse. Stella Leego töös antud võimalus liitumise avalduse vormil ei ole, mistõttu on see funktsionaalsus vaja ka lisada.

Lisaks mainiti varasemates töödes, et vastavalt organisatsiooni reeglitele tuleks akadeemiliste puhkuste ajaks liikmelisus peatada. Siiski on avastatud, et see ei pruugi olla vajalik, kuna paljud akadeemilisel puhkusel olevad liikmed jätkavad aktiivselt projektides osalemist ja organisatsiooni panustamist.

Personalijuhi hinnangul on vaja ka lisada võimekus muuta punktisüsteemi tegevusi. Sellel on kaks ärilist põhjust. Esiteks leiab BEST-Estonia tihtipeale uusi projekte ning ülesandeid, mille eest saaks liikmetele punkte lisada. Teine põhjendus tuleneb olemasolevate projektide mahu muutusest ning vahel ka projektide ärajäämisest, mistõttu on vaja tegevusi muuta või lausa eemaldada.

Personalijuht juhtis ka tähelepanu isikuandmete kättesaadavusele teistele BEST-Estonia liikmetele, seal hulgas ka rakenduses kuvamine. Tema hinnangul tuleks isikuandmete kättesaadavust piirata nii palju kui võimalik, ehk ainukene kasutaja, kes isikuandmeid saab näha, on personalijuht ise. Seetõttu otsustati hetkel piirata rakendusele ligipääs ainult personalijuhile ja koordinaatoritele. Kuna valmival rakendusel on esialgu põhieesmärgiks tuua kokku punktihaldus ja lihtsustada liikmehaldust, siis ei nähtud vajadust anda rakenduses ligipääsu ja tegevusi teistele liikmetele, näiteks ise oma isikuandmete uuendamine või omavoliliselt organisatsioonist lahkumine. Need andmesisestused teeb vajadusel personalijuht ise. Põhikasutajaks jäävad hetkel ainult personalijuht ning koordinaatorid. Sellest tulenevalt ei anta hetkel rakendusele ligipääsu ka alumni juhatusel.

Andmebaasi disainis on arvestatud ka võimalusega lahkunud liikmel organisatsiooniga uuesti liituda ning olemasolevates kasutuslugudes on olemas ka vastav funktsionaalsuse nõue. Seevastu vanu liikmeid, kes on juba korra aktiivne liige olnud ning soovivad uuesti organisatsiooni protsessidest osa saada, näiteks alumniliiige, ametlikult tihtipeale uuesti

liikmeks ei määrata, sest otsest vajadust ei ole olnud. Vajadus tekib ainult siis, kui vastav liige soovib uuesti hääleõiguslikuks liikmeks saada. Sellisel juhul määratakse ta andmebaasis uuesti aktiivseks liikmeks, kuid ka siin ametlikku protsessi ei ole ja liikmeavaldust täita ei lasta. Isikuandmete uuendamised teeb vajadusel personalijuht ning huvialade märkimine siin enam mingisugust tähtsust ei oma.

Hetkel hoiab personalijuht liikmete nimekirja lisaks liikmehaldustarkvarale ka tabelarvutusprogrammi abil loodud töövihikutes. Üks kasutus sellele on aktiivsete liikmete nimekirja väljastamine projektikorraldusmeeskondadele, et lihtsustada neil projektikorraldusprotsessile abiliste leidmist. Lisaks on personalijuhil vaja koostada nimekirju aktiivsetest ja mitteaktiivsetest liikmetest, et vajadusel rohkem kaasata vähemaktiivseid liikmeid organisatsiooni protsessidesse või üldsegi nende liikmelisus organisatsiooni töös mitteosalemise tõttu lõpetada. Nimekirjade kasutamise kasutusjuhtusid on veelgi. Samuti on olemasolevas rakenduses võimalik andmeid eksportida, kuid kuna punktisüsteem ei ole hetkel kasutatava rakendusega seotud, siis ei ole võimalik kaasa anda punktide statistikat. Sellega on võimalik näha, millised liikmed on aktiivsemad ja millal keegi viimati organisatsioonis midagi kaasa tegi.

Käesoleval aastal on BEST-Estonias liikmete nimekirjas vähemalt kaks liiget, kellel on sama eesnimi ja perenimi, millest tekkis vajadus eristada samanimelisi liikmeid. Selle jaoks otsustati lisada võimalus märkida liikme isikuandmete juurde ka täpsustuse väli, mille sisu saab olemasolul kuvada näiteks liikmete otsingutes või nimekirjades, võimaldades eristada samanimelisi liikmeid. Üks võimalus, mismoodi antud välja saab kasutada, on kasutada antud välja hüüdnime hoiustamisel.

Liikmenimekirja kuvamine BEST-Estonia kodulehel võeti samuti uuesti analüüsi alla küsimusega, kas selle näitamine annab organisatsioonile väärtust juurde, või pigem tekitab asjatuid probleeme. Selle jaoks kaasati arutellu ka BEST-Estonia revisjoni läbiviija, kellelt samuti antud teema kohta arvamust küsiti. Arutelu tulemusena otsustati hetkel liikmenimekirja kodulehel mitte kuvada. Nähti, et ainult beebi- ja täisliikmete kuvamine võib organisatsiooni tõsiseltvõetavust halvendada, sest organisatsiooni nähakse väljaspool kui üle saja aktiivse liikmega organisatsiooni. Aastaringelt on BEST-Estonias beebi- ja täisliikmeid

mitte üle viiekümne, seega lehel kuvataks kaks korda vähem nimesid, kui tegelikult organisatsioonis aktiivseid on. Uusliikmete kuvamist ei saa lubada, sest see sunnib liituja ennast organisatsiooniga kohe avalikult siduma, mistõttu võib tekkida oht, et liige ei pruugi organisatsiooniga alguses üldsegi liituda. Selle probleemi lahendamiseks oleks võimalus tekitada vormile vastav valik, mis võimaldaks nime näitamisest hoiduda, kuid sellega võib kaasneda oht, et kõik värsked liitujad otsustavad nime lehel mitte kuvada, säilitades algse probleemi. Lisaks tekitab liikme kuvamise funktsionaalsus lisakoormust personalijuhile, tekitades vajaduse igalt liikmelt eraldi uurida beebiliikmeks saamisel, kas antud liige on nõus nime kuvamisega. See koormus tekib tänu otsusele mitte lisada võimalust saada ligipääsu rakendusele tavaliikmel (kes ei ole näiteks koordinaator või personalijuht). Antud otsus ei mõjuta juhatuse koosseisu kuvamist BEST-Estonia lehel.

Kokkuvõtvalt on uuenenud nõuded järgnevad:

- Integratsioon Discordiga;
- Integratsioon projektide kokkuvõtte rakendusega;
- Liikmeavalduses huvialade märkimine;
- Peatatud liikmelisuse staatus pole oluline;
- Punktisüsteemi tegevuste lisamine ja muutmine;
- Rakendust kasutavad ainult koordinaatorid ja personalijuht;
- Kunagise liikme uuesti liikmeks võtmine;
- Liikmeandmete eksport;
- Hüüdnime väli isikuandmete juurde;
- Liikmenimekirja kuvamise asemel kuvada organisatsiooni kodulehel ainult juhatuse koosseis.

3.2.3 Loodava funktsionaalsuse probleemid

Andmeekspordi funktsionaalsuse lisamine tõstab oluliselt andmete lekkimise ohtu. Tegemist on isikuandmetega, mille kasutamisele on sätestatud selged piirid: neid võib kasutada ainult organisatsioonisiseste tegevuste jaoks. Probleem tekib täpsemalt sellest, et eksport võimaldab tekitada suurtes kogustes faile, mille sisuks on organisatsiooni liikmete

isikuandmed, ning antud failid võivad kontrollimatult ringlema hakata. See nõuab BEST-Estonia juhatuselt lisatähelepanu, sest isikuandmete kaitse regulatsioonide täitmise kohustuse kontrollimine on organisatsiooni juhatusel. Olgugi, et andmete ekspordi funktsionaalsuse kasutamise võimalus on ainult organisatsiooni personalijuhil, on lekkimise oht ikkagi oluline.

Lisaks ei ole andmete ekspordi võimaluse olemasolul süsteemis enam ühte kesket andmebaasi, kus kõiki andmeid ja õiguseid saab hallata. Säilib vajadus pöörata tähelepanu failihaldusele. Seetõttu säilib osaliselt olemasolev probleem, milleks on andmete hajutatatus.

Idealis võiks antud töös arendataval rakendusel olla võimalus koostada erinevaid nimekirju ja lisada vajadusel iga liikme juurde märkmeid, samal ajal hoides kõiki andmeid ühes kohas. Lisaks võiks olla ka võimalus anda teistele liikmetele ligipääs koostatud nimekirjadele, et ei peaks andmeid rakendusest üldse väljapoole tooma, hoides ära võimaliku andmete lekkimise ohu.

Teisest küljest ei lisa andmete eksportimine organisatsioonile juurde tähelepanu vajavaid lisaprobleeme, sest organisatsiooni tööprotsessis kasutatakse juba erinevate nimekirjade koostamisel tabelarvutusprogramme. Lisaks on tabelite koostajaks hetkel ainult personalijuht ning uues rakenduses eksportimise funktsionaalsuse kasutamise võimalus on samuti ainult personalijuhil.

Andmeekspordi funktsionaalsuse lisamise eeliseks on selle lihtsus. See ei nõua erilist lisaanalüüsi ega keerulisemat arendust, sest sama võimalus on juba vanas rakenduses olemas ning jätab andmeeksportijale vabad käed, mis moodi andmeid edaspidi töödelda. Samuti võimaldab eksportimise funktsionaalsuse lisamine kasutada andmete töötlemiseks olemasolevaid tööriistu, nagu näiteks Google Sheets või MS Excel.

Andmeekspordi funktsionaalsuse lisamine on töö autori arvates hetkel ainus hea lahendus. Tegemist on nii-öelda kompromisslahendusega, ehk funktsionaalsus on piisavalt lihtne, mis ei lisa käesolevale tööle lisamahtu, säilitab personalijuhil võimaluse töödelda andmeid vabas vormis, kuid otseselt ei lahenda andmete hajutatuse probleemi.

4 Sobivaima veebirakenduse raamistiku valimine

Selles peatükis kirjutatakse BEST-Estonia veebipõhise andmebaasitarkvara realiseerimiseks sobivaima vahendi valmisest.

4.1 Arvestamine BEST-Estonia võimekustega

Arvestades, et BEST keskendub peamiselt rahvusvahelise mõtteviisi edendamisele [4], on oluline tunnistada, et BEST-Estonia IT-alased pädevused on piiratud. Lisaks sellele on ka enamasti liikmeid majandus- või inseneriateaduskondadest [5]. Sellest tulenevalt on veebirakenduse arendamisel ülioluline valida programmeerimisraamistik ja hostimiskeskond, mis tagab arendamise ja hooldamise lihtsuse nii praegu kui ka tulevikus.

Kuigi BESTi praegune hostimisteenuse pakkuja Zone.ee pakub piiratud hostimisvõimalusi, mis on suunatud sellistele sisuhaldussüsteemidele nagu WordPress, võimaldab see siiski kasutada programmeerimiskeeli nagu PHP, JavaScript ja Python. Kuigi need piirangud mõjutavad programmeerimisvahendite valikut, on antud süsteemi puhul juba lahendatud kaks kriitilist probleemi: kuluefektiivsus ning hoolduse ja turvalisuse tagamise lihtsus. Viimane on piiratud IT-teadmistega organisatsiooni jaoks ülioluline.

Võimalus osta Zone.ee-st lisateenuseid konteineriseeritud veebirakenduse majutamiseks on küll olemas, kuid sellel on kaks puudust: vajadus tulevase kompetentsi järele sellise seadistuse haldamisel ja selle valikuga kaasnevad lisakulud. Lisaks killustaks selline variant organisatsiooni veebihaldust, muutes süsteemi hoolduse ja haldamise keerulisemaks.

Arvestades BESTi spetsiifilisi vajadusi ja piiranguid, on oluline valida sobivaim programmeerimisraamistik, mis võimaldab jääda kasutama praegust hostimiskeskonda. Selline lähenemine ei taga mitte ainult kulutõhusust ja hoolduse lihtsust, vaid säilitab ka keske asukoha kõikidele BESTi veebirakendustele ja -lehtedele, võimaldades organisatsioonil jätkata keskendumist oma peamistele eesmärkidele ilma, et organisatsiooni veebipoole halduse tehniline keerukus kasvaks.

Töö raames otsustati PHP kasuks kahel põhjusel. Töö autoril on olemas varasem töökogemus PHP programmeerimiskeelega ning BEST-Estonia enamus olemasolevatest veebirakendustest on kirjutatud samuti PHP keeles. See tagab kiirema arendusprotsessi, eemaldades lisaajakulu uue keele tundmaõppimiseks. Samuti võimaldab see lihtsamat rakenduse hooldamist.

4.2 PHP raamistiku valimine

PHP on üks populaarsemaid veebirakenduse arenduse keeli, kus üle 21% arendajatest hindavad ennast professionaalseks PHP arendajaks [6]. 2023. aasta mais oli PHP TIOBE programmeerimiskeelte populaarsuse indeksis kaheksandal kohal [7], Seetõttu on loogiline ka järeldada, et PHP-le leidub mitmeid erinevaid raamistikke. Kõige lihtsam viis sobiva raamistiku leidmiseks on vaadata selle populaarsust, sest mida populaarsem on raamistik, seda kergem on saada lahendust arendamise käigus ette tulevatele probleemidele.

Käesolevas töös analüüsitakse kolme populaarseima PHP raamistiku sobivust BEST-Estonia veebirakenduse realiseerimiseks. Kolm populaarsemat raamistikku loodeti saada *Stack Overflow developer survey 2022* järgi, kuid seal olid näidatud ainult kaks – *Laravel* ja *Symfony*. *Laraveli* professionaalseks arendajaks hindas ennast 10% kõigist vastanud arendajatest ning *Symfony* puhul 4%. Nimekirjas on ka raamistik *Drupal*. *Drupal* on mõeldud veebisisu haldamiseks, mistõttu ei ole see antud töö raames sobilik.

Populaarsust on kogumas ka *CodeIgniter* raamistik, millel oli Githubis 290 projekti panustajat [8], mis oli *Laraveli* raamistikuga võrreldes veidi üle kümne korra väiksem, sest *Laraveli* panustas 3115 arendajat [9]. Need vaatlused panustajate arvu kohta on tehtud 23. aprilli 2023 seisuga.

4.2.1 Laravel

Laravel on 2023. aasta kevadel kõige populaarsem PHP raamistik, omades Githubis veidi üle 3000 panustaja. Taylor Otwelli 2011. aastal välja töötatud [10] *Laraveli* eesmärk on pakkuda tööriistu uustulnukatele ja vanemarendajatele. Raamistikul on suur

dokumentatsioonikogu ning ka erinevaid kõrgtaseme tööriistad nagu sõltuvuste süstimine, testimine, järjekorrad, sündmused jne [11].

Laravel järgib Model-View-Controller (MVC) arhitektuurimustrit, mille eesmärk on eraldada äri loogika vaadetest ja andmemudelitest. Lisaks on Laravelil Eloquent ORM, eesmärgiga lihtsustada ja standardiseerida andmebaasitoiminguid, eemaldades vajaduse kasutada koodis puhast SQL koodi. See annab võimaluse vajadusel andmebaasisüsteem välja vahetada ilma lähtekoodis muudatusi tegemata või kasutada mitut erinevat andmebaasisüsteemi, tagades igaihe puhul ühesuguse viisi andmete küsimiseks ja manipuleerimiseks.

Vaadete visualiseerimiseks on Laravelil välja töötatud mallimootor Blade, mida kasutatakse vaadete loomiseks ja kasutajaliidese komponentide organiseerimiseks. See lihtsustab dünaamilise sisu renderdamist, võimaldades arendajatel HTML-i ja PHP-koodi omavahel puhtama süntaksiga segada.

4.2.2 CodeIgniter

CodeIgniter on teine PHP-põhine MVC raamistik, mis sobib hästi väikese ja keskmise suurusega projektide jaoks. Tegemist on Laravelist väiksema raamistikuga, mille komponente ja osasid on väga lihtne asendada või laiendada. Eesmärk on pakkuda lihtsasti kasutatavat raamistikku [12].

Kuna CodeIgniter on väiksemat sorti raamistik, siis ei ole sellel nii paljusid sisemiselt arendatud funktsionaalsuseid, mistõttu nõuab keerulisema projekti arendus baaskomponentide arendamise arvelt rohkem tööd. Samas on neid komponente tihtipeale võimalik asendada väliste teekidega. Siin tekib oht, et uuel arendajal on projekti süvenedes raskem koodist aru saada, sest võidakse kasutada mitte-standardseid teeke.

4.2.3 Symfony

Symfony on 2022 Stackoverflow developer survey järgi populaarsuselt teine PHP raamistik, kus 4% kõigist vastanud arendajatest hindasid ennast professionaalseks Symfony arendajaks. Symfony on mõeldud pigem suuremate ning suure jõudlusega veebirakenduste loomiseks. Seda on näha näiteks Symfony ORM Doctrine puhul, millel on juures võimekus panna

erinevad SQL käsud järjekorda, mistõttu saab ühe korraga mitu SQL käsku käivitada, hoides kokku rakenduse ja andmebaasi ressursi [13].

Lisaks on Symfonyl mitmeid erinevaid konfigureerimisviise, näiteks autentimiseks ja operatsioonide töökiiruse optimeerimiseks, mis teeb sellest ideaalse kandidaadi suuremate ja äärmiselt keerukamate projektide jaoks. Sarnaselt Laravelile on ka Symfonyl aktiivne kogukond, mis pakub arendajale tuge.

4.3 Kasutatava raamistiku valimine

Kuigi CodeIgniter oleks võinud olla sobiv valik, ei sisalda see nii palju eelnevalt väljatöötatud komponente. Eelehitatud funktsioonide puudumine nõuaks nende oluliste komponentide rakendamiseks lisa-aega ja -vaeva, muutes selle Laraveliga võrreldes BEST-Estonia vajaduste jaoks vähem ideaalseks.

Symfony on mõeldud keerukamate süsteemide jaoks, mistõttu ei sobi see organisatsiooni oluliste nõuetega. BEST-Estonia ei vaja liiga keerulist raamistikku ja Symfony kasutuselevõtt võib arendusprotsessi tarbetut keerukust lisada.

Seega valiti kasutatavaks raamistikuks Laravel. Laravel valiti veebirakenduse programmeerimisraamistikuks mitmel põhjusel. Esiteks on Laravelil sisseehitatud komponendid, nagu autentimine, mis vähendab oluliselt esialgse seadistamise aega ja -vaeva. See eelis võimaldab arendusprotsessi sujuvamaks muuta, võimaldades arendajatel keskenduda rakenduse muudele kriitilistele aspektidele. Lisaks on Laravel tuntud raamistik, millel on suur ja aktiivne kogukond. See laialdane tunnustus ja ulatuslik kasutajakogukond pakuvad arendajatele rohkelt ressursse, õpetusi ja tuge. See omakorda tagab, et arendusprotsessi käigus tekkinud väljakutseid saab kiiresti ja tõhusalt lahendada.

5 Kasutatavad kasutajaliidese parimad praktikad

Antud projekti kasutajaliidese põhjaks võeti Stella Leego töös tehtud kasutajaliidese prototüübid. Need prototüübid on tehtud koostöös 2020/2021. perioodi BEST-Estonia personalijuhiga ning nende loomisel on juba arvestatud erinevate heade tavade ja reeglitega. Samas on muutunud nõuete ja puuduolevate prototüüpide tõttu vaja kohandada olemasolevad ja koostada uued prototüübid.

5.1 Kasutatavus

Kasutajaliidese ja kasutajakogemuse disaini üks peamisi eesmärke on muuta toode kasutatavaks paljudele kasutajatele. Nielsen järgi on kasutatavus defineeritav viie omaduse järgi: õpitavus, tõhusus, meeldejäätvus, vigade ennetamine ja rahulolu [14] :

- Õpitavus viitab sellele, kui hõlpsalt saavad kasutajad süsteemi abil soovitud ülesandeid täita.
- Tõhusus näitab kiirust, millega kasutajad saavad neid ülesandeid täita, olles juba liideselega tuttavad.
- Meeldejäätvus näitab kasutajaliidese ja selle funktsioonide meeldejätmise lihtsust, muutes tootega uuesti tutvumise lihtsamaks.
- Vigade ennetamine hõlmab süsteemi kavandamist selliseks, et see vähendab kasutajate tehtud vigu.
- Rahulolu mõõdab üleüldist toote kasutamise kogemust.

Kasutatavust aitab oluliselt tõsta liikmehaldus- ja punktihaldussüsteemi analüüs. Vaatamata sellele, et eelnevates töödes on süsteemianalüüs juba tehtud, on vaja paralleelselt rakenduse arendamisega antud analüüsiga jätkata, et saada võimalikult palju infot vajamineva funktsionaalsuse kohta ning kasutajamugavuse parandamise jaoks. Üks parimaid viise selle tegemiseks on terve arendusprotsessi jooksul kasutajaga kontakti hoidmine, et võimaldada leida erinevaid vigu võimalikult kiiresti. Nielsen soovib info efektiivsemaks kogumiseks paluda kasutajal näidata konkreetseid näiteid oma töötegevustest [14]. Sellega on võimalik vältida üleliigseid üldistusi ülesannete täitmise kirjeldamisel.

Kasutatavuse parandamiseks on soovitatav kasutada mõnda olemasolevat tööriista, mis vaatab lehe erinevaid parameetreid ning annab vastavalt sellele ka kasutatavuse hinnangu. Võimalusel võiks see sama tööriist korjata statistikat ka jooksvalt, mis annaks parema ülevaate rakenduse kasutajate käitumisest. Sellest saadava infoga on võimalik rakenduse kasutatavust oluliselt parandada.

Lisaks tuleb kasutatavuse tõstmiseks tagada veebilehe kiire laadimiskiirus. Google soovitab hoida lehe laadimise kiirust alla kolme sekundi, sest mida kiiremini laeb leht ära, seda suurema tõenäosusega jääb kasutaja antud lehte edasi kasutama [15].

5.2 Kasutajaliidese järjepidevus ja tuttavlikkus

Järjepidevus hõlmab disainielementide ja interaktsioonide puhtuse säilitamist kogu tootes, mis aitab kasutajatel paremini mõista ja ennustada, mida mingisuguses vaates teha saab. Järjepidevus lihtsustab oluliselt kasutajaliidese õppimist, kuna kasutajal ei ole vaja talletada pikaajamällu mitmeid erinevaid viise, kuidas sarnaseid tegevusi teha. Tuttavlikkus viitab disainimustrite ja tavade kasutamisele, mida kasutajad üldiselt juba teavad, mis vähendab oluliselt õppimiskõverat.

Järjepidevuse järgimine nõuab terves tootes ühtse disaini kasutamist sarnase eesmärgiga elementidel [16]. Erinevatel lehtedel asuvad sarnase käitumisega juhtimiselemendid ja andmete vaated, võiksid olla samal asukohal, sama kujuga, sama värviga jms. See võimaldab kasutajal kergemini märgata nende vaadete eesmärke.

Rakenduse kasutajaliidese ja funktsionaalsuste kavandamisel tuleb järgida ka tuttavlikkust. See tähendab, et tuleb kasutada kasutajale tuttavat terminoloogiat, graafikat, värve, jms, mis hoiab kasutaja tähelepanu hetke ülesande täitmisel, ehk kasutajal pole vaja välja mõelda, mis moodi ta saaks mõnda ülesannet teha. Näitena tõi Jeff Johnson enda raamatus [16] sessiooni aegumise teate: *“Your session has expired. Please reauthenticate.”* Kui seda näeb IT terminoloogiat tundev inimene, siis saab ta kohe aru, et sellega palutakse kasutajal uuesti sisse logida. Terminoloogia mitteteadjale võib sõna *“reauthenticate”* jääda võõraks, sundides kasutajat tööhoost välja tulema ning välja mõtlema, mida antud sõnum võiks tähendada.

Järjepidevuse ja tuttavlikkuse reeglite järgimine on liikmelisuse ja punktisüsteemi halduse rakenduse puhul väga tähtis aspekt, sest rakenduse kasutajaskond vahetub igal aastal. Lisaks, kui rakendusele soovitakse arendada juurde muud ärioluliselt vajalikud funktsionaalsused, mis antud töö skoobist välja jäävad, võib kasutajaskonna maht tõusta, mistõttu muutub nende reeglite järgimine veelgi tähtsamaks. Järjepidevuse reeglit on juba olemasolevas liikmehalduse tarkvara kasutajaliidese prototüüpides arvestatud, kasutades organisatsiooni kodulehel kasutatavaid värve ja fonte.

5.3 Informatsiooni struktureeritus

Kasutajad ei loe veebilehti sirvides või rakendusi kasutades hoolikalt igat seal leiduvat sõna, vaid otsivad pigem lehte pilguga skaneerides asjakohast teavet [16]. Hea kasutajakogemuse üheks tähtsaks komponendiks on informatsiooni struktureeritud esitus. Kui tähtis informatsioon on mingil viisil peidetud, näiteks täislausetesse, siis selle leidmine on oluliselt raskendatud. Kuid kui informatsiooni näitamine on hästi struktureeritud, siis leiab kasutaja vajaliku info kiiresti, mistõttu ei ole vaja süveneda info otsimisse. Selle asemel on kasutajal võimalik keskenduda täielikult oma käesoleva ülesande täitmisele.

Üks viis, kuidas informatsiooni kergemini leitavamaks teha, on struktureerida see võimalusel tabelitesse või nimekirjadesse. Nimekirja elemendid võiksid samuti olla üksteisest eristatavad, asudes näiteks eraldi ridadel. Järgnevalt on toodud kaks erinevat viisi liikme huvialade näitamiseks (Joonis 1, Joonis 2).

Muu info

Huvialad

- Disain
- Muusika
- Tehnika
- Reisimine

Kogutud punktid

6855

Joonis 1. Liikme huvialade nimekiri loendina.

Muu info

Huvialad	Disain, Muusika, Tehnika, Reisimine
Kogutud punktid	6855

Joonis 2. Liikme huvialade nimekiri komaga eraldatuna.

Toodud näidete vaate üheks eesmärgiks on informeerida kasutajat, millised huvialad on antud liige endale märkinud, et leida antud liikmele sobiv mentor ja sobivad tegevused organisatsiooni sisseelamiseks. Siin aitab selle info kergemini kättesaadavaks teha elementide grupeerimine ja sümmeetria. Elemendid, mis on komaga eraldatud ja ühel real (Joonis 2) on raskemini loetavad, sest seal puudub sümmeetria, olenemata sellest, et need on grupeeritud.

Sümmeetriliselt üksteise all esitatud nimekirjast on oluliselt lihtsam vajalikku informatsiooni välja lugeda (Joonis 1). Siin on iga element piisavalt eraldatud, et silmaga üle skaneerides leiab kasutaja vajaliku info kiiresti üles, kuid samas on elemendid grupeeritud täpikutidega, mis teeb ilmselgeks, et tegemist on ühte ja samasse gruppi kuuluva informatsiooniga.

Antud rakenduse arendamisel tuleb kasutajaliidese disainimisel võtta arvesse informatsiooni kokkukuuluvust ning need omavahel grupeerida. See parandab oluliselt kasutajakogemust ning kiirendab informatsiooni leidmist.

5.4 Värvikontrastid

Värvikontrast mängib ligipääsetavuse juures väga tähtsat rolli, tagades selge erinevuse komponentide vahel, millel on vaja teha vahet, näiteks tekst ja selle taust. Suure kontrasti tagamine teksti ja taustavärvide ning interaktiivsete elementide vahel lihtsustab oluliselt sisu tarbimist kõigi tarbijate jaoks, andes positiivsema kasutajakogemuse. Arendatava rakenduse juures on väga tähtis kontrollida värvide kasutatavust, et rakenduse sisu oleks kergesti arusaadav.

Tase AA WCAG 2.1 standardis soovib saavutada minimaalse kontrastisuhte 3:1 suurema teksti puhul, 4.5:1 muudel elementidel [17]. Kontrast ei ole oluline kohtades, kus antud element ei mängi informatiivset rolli.

Olemasolevate kasutajaliidese disainide suhtes tehti kasutatavate värvide kontrastide analüüs (Lisa 2 – Olemasolevas kasutajaliidese prototüübis kasutatavate värvide kontrastisuhted) ning sellega leiti probleeme. Madalama nägemisteravusega kasutajatega arvestamiseks võiks värvikontrastide suhe olla informatiivsetes kohtades vähemalt 4.5:1, kuid paljudes kohtades see nii ei ole. Näitena on sisestusväljade raamide kontrastisuhe võrreldes valge taustaga 1.49:1, mis võib olla raskesti nähtav isegi hea nägemisega kasutajatele. Probleeme võib tekitada ka liikmestaatuse näitamine erinevates tabelites, kus näiteks beebiliikme staatuse teksti ja tausta vahel on kontrastisuhteks 2.99:1, selle raamil koguni 1.45:1. Täisliikmestaatuse, uusliikmestaatuse ja väljavisatud liikmestaatuse puhul on olukord veidi parem, kuid endiselt on kontrastisuhe alla 4.5:1.

Kontrastisuhete probleeme leidub veelgi. Sellest tulenevalt tuleb olemasolevad kasutajaliidese prototüübid üle vaadata ning võimalusel kontrastiprobleemid ära parandada.

5.5 Ligipääsetavus

Ligipääsetavus on rakenduse kujundamise oluline aspekt, mis võimaldab suuremal kasutajate ringil rakendust kasutada, sõltumata nende võimekusest. Kõige tavalisem standard, millega veebilehe ligipääsetavust hinnatakse, on WCAG, mille värskem versioon töö kirjutamise hetkel (2023. aasta kevad) on 2.1. WCAG on soovitude kogum veebisisu juurdepääsetavamaks muutmiseks ka puuetega inimestele. Need soovitused hõlmavad erinevaid valdkondi, nagu tekstilised alternatiivid mittetekstilise sisu jaoks, klaviatuuri juurdepääsetavus, piisav värvikontrastsus ja ühilduvus erinevate tehnoloogiatega sisu tarbimiseks [17].

Antud töös arendatava rakenduse puhul arvestatakse WCAG soovitustega osaliselt, sest seda hakkavad kasutama väga vähesed kasutajad ning antud rakenduse eesmärk on olla töövahend BEST-Estonia personalijuhile ning koordinaatoritele. Seetõttu ei ole rakenduse arendusel

kriitilise tähtsusega arvestada ekraanilugejate toetusega. Samas on WCAG soovitude seas ka soovitused, mis oluliselt parandavad kasutajakogemust ka puueteta kasutajatele, näiteks klaviatuuriga ligipääsetavus või värvikontrastid. Värvikontrastide kasutusest antud töös arendatavas rakenduses räägiti lähemalt jaotises 5.4 Värvikontrastid.

Kuna rakendus on mõeldud igapäevaseks kasutamiseks, siis on oluline võimaldada rakenduse kasutamist ka nutitelefonide kaudu. See võimaldab kasutajal teha väiksemad andmesisestused liikvel olles. Lisaks on oluline võimaldada rakendust kasutada ka klaviatuuri abil, et oluliselt lihtsustada korduvaid tegevusi.

6 Varem tehtud töö täiendused enne rakenduse arendusega alustamist

Selles peatükis kirjutatakse muudatustest nõuetes, kasutajaliidese kavandites ja andmebaasis, mida tehti MTÜ BEST-Estonia personalijuhiga vestlemise järel.

6.1 Muutused funktsionaalsetes nõuetes

MTÜ BEST-Estonia personalijuhiga vestluse käigus tulid välja veel mõned nõuded, mida antud töö raames on rakenduse arendusega vaja arvestada. Selleks on punktisüsteemi reeglite muutmine ja vaatamine, huvialade märkimine liituja poolt ja vaatamine personalijuhi poolt, andmete eksport ning lahkunud liikme uuesti aktiivseks määramine, millele vastavalt on koostatud ka kasutuslood (Tabel 1). Tabelis ei ole ära näidatud teiste rakendustega liidestamist, sest see funktsionaalsus jääb antud töö skoobist välja. Koostatud kasutuslugude numbriliste identifikaatorite määramist jätkati Karl Henri Kaljuse lõputöös punktisüsteemide tarkvara analüüsi käigus koostatud kasutuslugude põhjal.

Tabel 1 Uutest nõuetest loodud kasutuslood.

ID	Kasutuslugu
110	Punktisüsteemi tegevuste lisamine, muutmine ja kustutamine – Personalijuhina tahan lisada uusi, muuta või kustutada olemasolevaid tegevusi, mille põhjal saaksid koordinaatorid ja muud liikmed teistele liikmetele punkte lisada.
111	Liikmete andmestiku eksport – Personalijuhina tahan eksportida organisatsiooni liikmete nimekirja koos isikuandmetega ja punktistatistikaga, et andmeid vabas vormis töödelda ning vajadusel teiste organisatsiooni liikmetega jagada. Samal ajal tahan hoida täielikku kontrolli, millised andmed teisteni jõuavad.
112	Huvialade märkimine liikmeavalduses – BEST-Estoniasse liitudes soovin märkida enda huvialasid, et organisatsiooni juhtkonnal oleks kergem mind mulle huvi pakkuvatesse tegevustesse kaasata.
113	Huvialade nägemine isikuandmete juures – Personalijuhina soovin näha liitunud liikme isikuandmete juures, millised huvialad on liitunud liige endale märkinud, et aidata tal organisatsioonis leida erinevaid huvipakkuvaid tegevusi. Huvialad peavad olema standardiseeritud, et oleks võimalik grupeerida erinevaid liikmeid vastavalt nende huvialadele.

ID	Kasutuslugu
114	Kunagise liikme uuesti liikmeks võtmine – Personalijuhina tahan kunagise liikme uuesti liikmeks võtta, luues talle uue liikmelisuse, et anda taasliitujale hääleõigus, samal ajal säilitades infot eelmise liikmelisuse kohta, näiteks eriala.

Stella Leego töös koostatud kasutusjuhtudes on kasutusjuht „Peata liikmelisus“. Kuna BEST-Estonia ei järgi liikmelisuse peatamise reeglit, siis sellel kasutusjuhul väärtust ei ole, mistõttu selle nõudega rakenduse arendamisel ei arvestata. Vastavalt muutus ka liikmelisuse seisundidiagramm (Lisa 6 – Uuenenud liikmelisuse seisundidiagramm). Lisaks ei peeta hetkel oluliseks teistele liikmetele rakendusesse ligipääsu andmist, mistõttu ei oma tähtsust sellised kasutuslood, kus kasutaja saab oma andmeid vaadata või oma enda liikmelisust lõpetada või peatada. Järgnevalt on välja toodud eemaldatavate kasutusjuhtude sisud, mis on leitavad ka Stella Leego tööst:

- **Kasutusjuht:** Peata liikmelisus
Tegutsejad: Personalijuht
Kirjeldus: Tegutseja vaatab seisundis *Beebi* või *Täis* liikmelisusi ja valib need, mis tuleb viia seisundisse *Peatatud*. Valitud liikmelisuste seisundiks saab *Peatatud*.
- **Kasutusjuht:** Lahku omal soovil
Tegutsejad: Liige
Kirjeldus: Tegutseja soovib organisatsioonist lahkuda. Tegutseja vaatab enda kehtivat (lõppemata) liikmelisust ja annab korralduse lahkuda. Liikmelisuse seisundiks saab Lahkunud.
- **Kasutusjuht:** Vaata enda liikmelisust
Tegutsejad: Liige, Alumni juhatus, Juhatus liige, Koordinaator, Personalijuht, Liikmelisuseta kasutaja
Kirjeldus: Tegutseja vaatab enda liikmelisusi. Süsteemi poolt kuvatakse praegune ja varasemalt omatud liikmelisused, sh sündmused, mis on nende liikmelisustega seotud.

Hetkel ei ole vajadust tekitada võimalust vanal liikmel teistkordsele liikmelisuse taotlemisele. See funktsionaalsus asendatakse hetkel kasutuslooga „Kunagise liikme uuesti liikmeks võtmine“ (Tabel 1). Järgnevalt on välja toodud eemaldatav kasutusjuht:

- **Kasutusjuht:** Taotle liikmelisust vähemalt teist korda
Tegutsejad: Liikmelisuseta kasutaja
Kirjeldus: Tegutseja vaatab enda lõppenud liikmelisusi. Tegutseja taotleb uut liikmelisust. Tekib liikmelisus seisundiga Taotletud. Süsteemi poolt saadetakse personalijuhile e-kiri, andes märku, et on lisatud uus liikmelisuse taotlus.

Isikuandmete kättesaadavuse piiramiseks on vaja muuta kolme kasutusjuhu puhul tegutsejaid: „Vaata aktiivseid liikmeid“, „Vaata kõiki liikmelisusi“ ning „Määra liige alumniks“, jättes kõigile alles tegutsejateks ainult Personalijuhi. Kusjuures viimast kasutusjuhtu muudeti nii, et tegevust ei peaks teostama liikmenimekirja vaates. Järgnevalt on näidatud uuendatud kasutusjuhud:

- **Kasutusjuht:** Vaata aktiivseid liikmeid
Tegutsejad: Personalijuht
Kirjeldus: Tegutseja vaatab aktiivseid liikmeid (isikuid, kellel on lõppemata liikmelisus). Näidatakse beebi ja täisliikmeid ning nende kontaktandmed.
- **Kasutusjuht:** Vaata kõiki liikmelisusi
Tegutsejad: Personalijuht
Kirjeldus: Tegutseja vaatab liikmeid (isikuid, kellel on vähemalt üks seotud liikmelisus). Süsteem kuvab liikmelisuse andmed ja seotud isikuandmed, sh sündmused, mis on nende liikmelisustega seotud.
- **Kasutusjuht:** Määra liige alumniks
Tegutsejad: Personalijuht
Kirjeldus: Tegutseja valib liikme, kellele soovib määrata alumniks ning määrab talle staatuse *Alumniliige*. Valida saab ainult *Beebi* ja *Täis* liikmelisusega liikmeid.

Liikmenimekirja otsustati BEST-Estonia kodulehel mitte kuvada. Küllaga otsustati alles jätta juhatare koosseisu kuvamine. Seetõttu muutub kasutusjuhu „Vaata aktiivsete liikmete nimesid“ sisu:

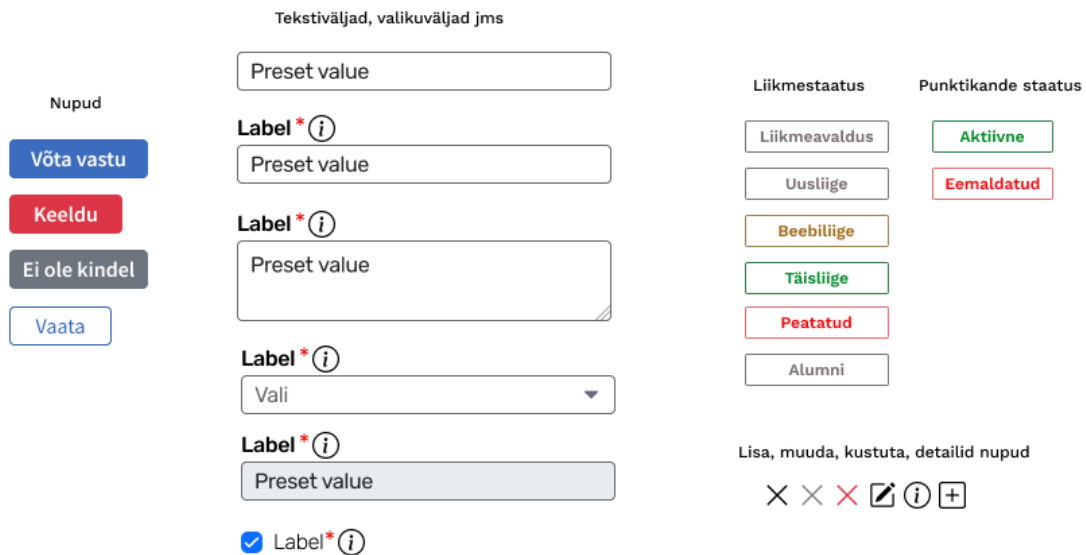
- **Kasutusjuht:** Vaata juhatuse koosseisu
Tegutsejad: Uudistaja
Kirjeldus: Tegutsejal on võimalik juhatuse koosseisu näha ainult kodulehe kaudu.

6.2 Täiendused kasutajaliidese kavandites

Kasutajaliidese kavandite põhjaks võeti Stella Leego töös koostatud kavandid. Kuna paljudele funktsionaalsustele polnud kasutajaliidese kavandit või leidis vastuolusid kasutajaliidese parimate praktikatega, siis tuli olemasolevad kavandid üle käia ning luua ka uusi. Lisaks tuleb luua kavandid punktisüsteemile, sest K.H. Kaljuse koostatud prototüüpi ei ole kasutajaliidese disaini ühtluse huvides mõtet eeskujuna kasutada.

6.2.1 Taaskasutatavad komponendid

Kasutajaliidese lihtsamaks disainimiseks tekitati mõned taaskasutatavad komponendid (Joonis 3).



Joonis 3. Taaskasutatavad komponendid.

Liikmestaatuse komponendid olid kasutusel ka olemasolevates kavandites, kuid nendele tehti värvimuudatusi, et tulla vastu värvikontrastile seatud eesmärgile. Järgnevalt on toodud

liikmestaatuste komponentide värvimuudatused, kusjuures nende raam ei ole informatiivne osa, mistõttu raamide puhul ei ole kontrastisuhe oluline. Küll aga on nende värve muudetud, et disaini ühtlustada. Iga värvikoodi järgi on lisatud ka värvikontrasti suhe võrreldes valge taustaga.

- Rohelised liikmestaatused:
 - Raam: #338149 - 4.79:1
 - Tekst: #008A27 - 4.5:1
- Hallid liikmestaatused:
 - Raam: #767676 – 4.54:1
 - Tekst: #776E6E – 4.91:1
- Kollased liikmestaatused:
 - Raam: #A38252 – 3.57:1
 - Tekst: #A16914 – 4.61:1
- Punased liikmestaatused:
 - Raam: #E66B6D – 3.13:1
 - Tekst: #EA1014 – 4.58:1

Muudatusi tehti ka andmesisestusväljadel, kus põhiliseks probleemiks oli sisestusvälja raami madal kontrastisuhe (1.49:1) võrreldes valge taustaga. Raami uueks värviks valiti #69696D, mille tulemusena on uus suhe valge taustaga 5.46:1. See värvivalik parandab ka oluliselt väljalülitatud tekstivälja raami kontrastisuhet sisendi taustaga, saades tulemuseks 4.6:1. Lisaks eemaldati kasutusest rohelist värvi nupp, mida kasutati näiteks täidetud vormi saatmisel. Selle nupu kontrastisuhe valge taustaga oli halb: 2.18:1. See asendati terves rakenduses läbivalt sinise nupuga, millel on kontrastiskoor oluliselt parem.

6.2.2 Liikmehaldus

Muudatust nõuab ka küljepaneelil asuv teavitus, mille eesmärk on teavitada personalijuhti uutest liikmetaotlustest. Antud kohas on värvikontrastisuhe väga halb, raskusi võib olla selle nägemisel isegi hea nägemisega kasutajatel. Selle parandamise teeb keeruliseks olukord, kus taust on sinine ja tekst on valge. Ideaalis võiks teavitusikooni värv olla selline, mis sobib mõlemaga. Selle jaoks valiti uus valge värviga infoikoon ning liigutati see valgest tekstist

natuke eemale, et seda oleks selgelt näha (Joonis 4). Lisaks eemaldati meiliaadressi, telefoni ja rolli näitamine liikmete nimekirjast, et mitte näidata kõigi liikmete kontaktandmeid, mis tulenes soovist võimalikult vähe näidata delikaatseid isikuandmeid ning asendati sünnikuupäeva ja eriala näitamisega (Joonis 4). Rolli näitamine ei andnud personalijuhile lisaväärtust.

BEST-Estonia liikmed

Otsing

Nimi: Liikmestaatus: Aktiiv Eriala: Otsi

Sünnikuu: Valli

Otsi

Liikmete nimekiri

Nimi	Liikmestaatus	Sünnipäev	Eriala
Juku Juurikas (Juss)	Täistlige	13 detsember 1996	Maa süsteemid, kliima ja tehnoloogiad
Peeter Maasikas	Täistlige	03 juuli 1997	IT süsteemide administreerimine
Tiina Kaasik	Beebilige	31 juuli 1996	Rahvusvaheline ärikorraldus
Mari Mustikas	Uustlige	10 juuni 2001	Küberturbe tehnoloogiad

Joonis 4. Liikmete nimekiri.

Liikmeandmete detailvaadet uuendati samuti, et koondada kokku kõik antud liikme isikuandmed ning rollide ja liikmestaatuste muutmise tegevused (Joonis 5). Lisaks on antud vaates vaadatava liikme isikuandmed vaikimisi peidetud, et võimaldada personalijuhil teha ka tööd rahvarohketes kohtades. Liikmestaatuste lisamiseks, muutmiseks ja kustutamiseks järgiti rakenduses põhimõtet, kus mingisuguse kande lisamiseks tuleb vajutada pluss nuppu, muutmiseks pliiatsi ikooniga nuppu ning kustutamiseks risti ikooniga nuppu. Nendele nuppudele vajutamine kuvab kasutajale vastava tegevuse modaali (Joonis 6). Selline lähenemine võimaldab rakenduse kasutajal rakendusega kiiremini tutvavaks saada, sest igal

pool on kirje lisamiseks, muutmiseks ning kustutamiseks samasugused nupud ning neil on ühesugune käitumine. Lisaks ei ole vaja iga erineva liikmestaatuse lisamiseks tekitada eraldi nuppu, mis võib kasutajat segadusse ajada.

Kui liige on organisatsioonist lahkunud, näiteks tema staatuseks on „Alumni“, lisandub isiku detailvaates juurde nupp „Määra liige aktiivi“ (Joonis 5), mille kaudu on võimalik personalijuhil lisada isikule uus liikmelisus, kui antud isik on näiteks otsustanud jätkata õpinguid, ning muuta sellega antud isik uuesti hääleõiguslikuks liikmeks.

Stella Leego
Personalijuht

Priit Viljandist

Isikuandmed

Eesnimi Priit
Perenimi Viljandist
Hüüdnimi -
Sünnipäev 10 mai 1998

Näita rohkem Uuenda isikuandmeid

Muu info

Huvialad

- IT
- Muusika
- Tehnika
- Reisimine

Kogutud punktid 6855

Liikmestaatuste ajalugu

Määra liige aktiivi

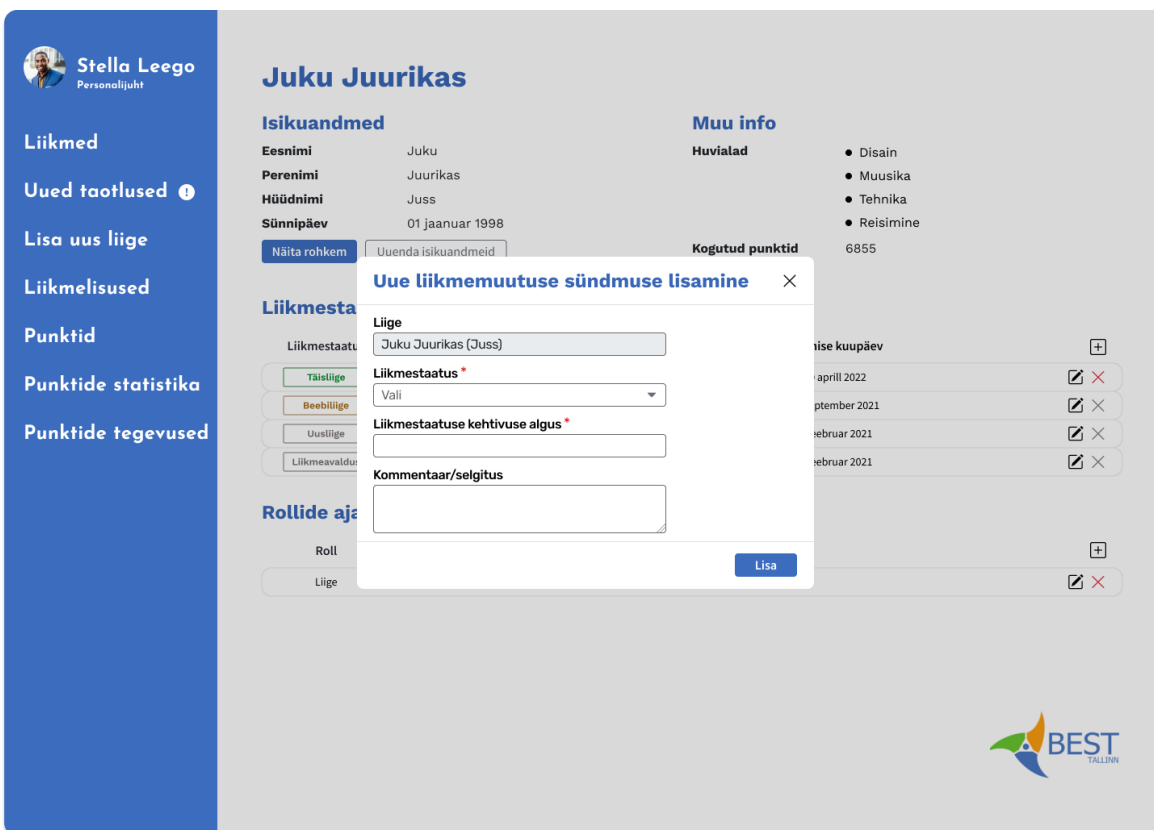
Liikmestaatus	Saamise põhendus/kirjeldus	Saamise kuupäev	
Alumni		23 aprill 2023	✍ ✕
Täisliige	Suurprojekti projektijuht	20 aprill 2022	✍ ✕
Õeeliige	Suurprojekti Väikeprojekti projektijuht	16 september 2021	✍ ✕
Uusliige	Liikmeavaldus rahuldatud	2 veebruar 2021	✍ ✕
Liikmeavaldus	Liikmeavaldus best.ee lehelt	1 veebruar 2021	✍ ✕

Rollide ajalugu

Roll	Kehtivuse periood	
Liige	2 veebruar 2021 - 23 aprill 2023	✍ ✕

BEST
TALLINN

Joonis 5. Liikmeandmete detailvaade.



Joonis 6. Uue liikmemuutuse sündmuse lisamise modaali.

Isikuandmete muutmiseks otsustati mitte kuvada modaali, sest isikuandmete seas on mitmeid erinevaid väljasid (Joonis 7). Modaale kasutati kohtades, kus polnud eriti palju vaja andmeid sisestada, hoides modaali suhteliselt väiksena.

Stella Leego
Personalijuht

- Liikmed
- Uued tootlused 1
- Lisa uus liige
- Liikmelisused
- Punktid
- Punktide statistika
- Punktide tegevused

Muuda isikuandmeid

Eesnimi *
Juku

Perenimi
Juurikas

Hüüdnimi ⓘ
Juss

Email *
Juku.Juurikas@gmail.com

Telefoni number
+372 59106343

Kontaktaadress
Tänav 24 korter 5 Tallinn

Isikukood *
50012030116

Eriala *
IT-süsteemide arendus

Sisseastumise aasta
2020

Salvesta

Selle välja sisu kuvatakse antud liikme nime kuvamisel, näiteks liikmete nimekirjas või punktsisestamisel liikmete otsingus. Välja eesmärgiks on aidata eristada isikuid, kellel on sama nimi.

Joonis 7. Isikuandmete muutmise vaade.

Liikmetelt küsitakse liitumisel ka nende huvialasid. Selle võimalusega pole eelnevates töodes arvestatud, sest BEST-Estonia protsessid on alates sellest ajast muutunud. Lisaks otsustati liigutada rakenduse arendamise lihtsamaks muutmise nimel liikmeavalduse vorm liikmehaldusrakendusesse. Eelnevalt oli antud vorm planeeritud asuma BEST-Estonia kodulehel. Vastavalt sellele uuendati liikmeavalduse vormi kavandit, mis on sobitatud rakenduse välimusega ning kuhu on lisatud huvialade valik (Joonis 8).

BEST-Estonia liikmeavaldus

Eesnimi *

Perenimi

E-posti aadress *

Telefoni number

Isikukood * 

Eriala *

Kooli sisseastumise aasta

Kontaktaadress

Miks soovid saada liikmeks?

Olen nõus isikuandmete töötlemisega *

Huvialad:

- Disain
- IT
- Juhtimine
- Koolitused
- Müük
- Rahvusvahelised projektid
- Reisimine
- Tehnika
- Turundus
- Ürituste korraldamine

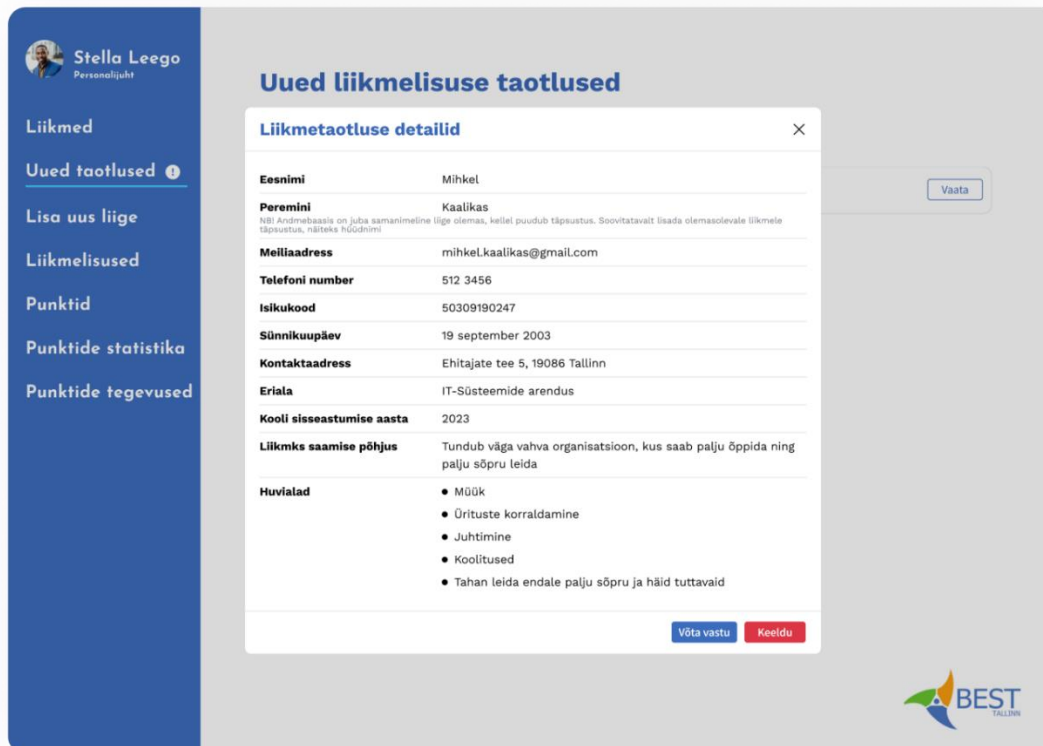
Muu

ESITA LIIKMEAVALDUS



Joonis 8. Liikmeavalduse vorm uuele liitujale.

Liikmeavalduste vaatamiseks ja vastuvõtmise otsuse tegemiseks on valmistatud kasutajaliidese kavand (Joonis 9). Otsustamise vaade kuvatakse modaalina. Muuhulgas lisati otsustamise vormi juurde ka info, mis teavitab personalijuhti juba andmebaasis samanimelise liikme olemasolust. Teavitus kuvatakse, kui samanimeline liige on olemas.




Joonis 9. Liikmeavalduse otsustamise modaali.

Antud töös otsustati BEST-Estonia liikmete nimekirja organisatsiooni kodulehel mitte kuvada. Vastavalt sellele alustati olemasoleva kasutajaliidese disaini muutmisega, kuid see disain jäi lõpetamata. Vaja on teha suurem muudatus organisatsiooni kodulehe üldises struktuuris, sest hetkel on koduleht üles ehitatud nii, et seal on eraldi leht liikmenimekirja kuvamiseks.

6.2.3 Punktihaldus

Kuna antud töös ei kasutatud kasutajaliidese väljanägemise ühtluse huvides K. H. Kaljuse töö raames koostatud punktihalduse tarkvara prototüüpi, siis tuli erinevate tegevuste jaoks uued kasutajaliidese mallid disainida. Kõigepealt alustati punktikannete vaatest, kuhu tekitati lisamise, muutmise, kustutamise, otsimise ning detailvaate funktsionaalsused (Joonis 10). Siin lähtuti varem paika pandud põhimõttest, kus kirjade lisamiseks, kustutamiseks ja muutmiseks kasutatakse samu käitumisi ning samade ikoonidega nuppe. Punktide lisamiseks

kuvatakse modaal (Joonis 11). Selle juures on tähelepanu väärt võimalus valida punktide lisamisel mitu liiget korraga.

 **Stella Leego**
Personalijuht

- Liikmed
- Uued taotlused 1
- Lisa uus liige
- Liikmelisused
- Punktid**
- Punktide statistika
- Punktide tegevused

Punktid


Kannete otsing

Nimi **Perioodi algus** **Perioodi lõpp**

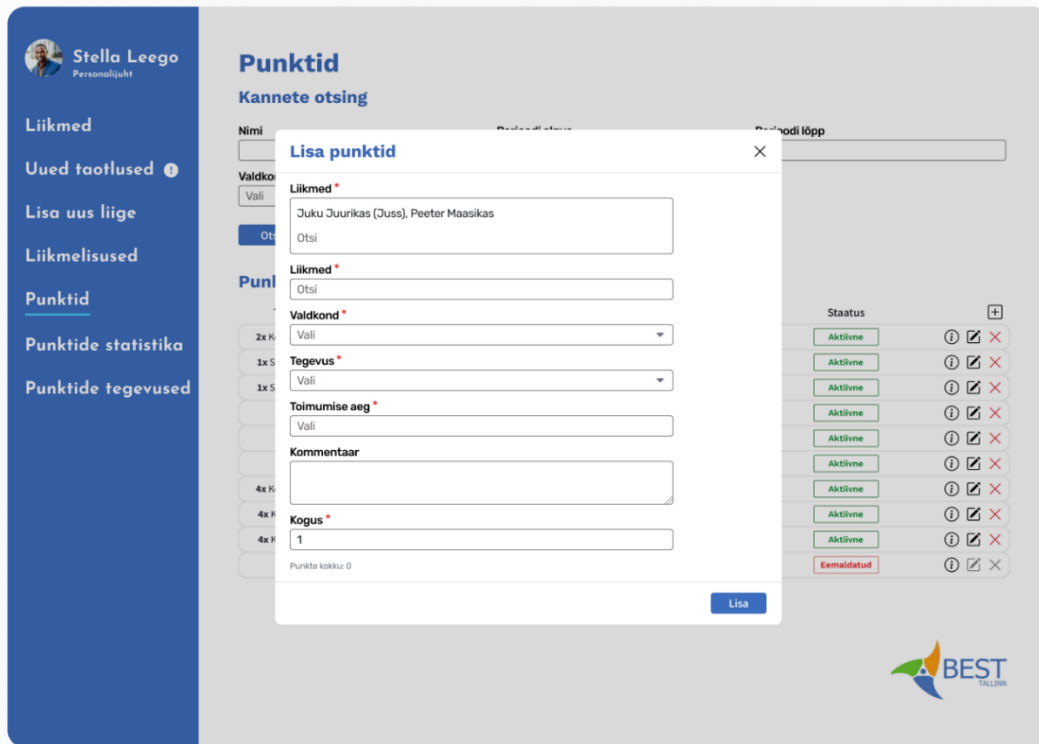
Valdkond

Punktide kanded

Tegevuse nimetus	Liige	Valdkond	Saamise aeg	Staatuse	
2x Koosolekul osalemine (1h)	Peeter Maasikas	FRTG	02 september 2022	Aktiivne	? <input checked="" type="checkbox"/> <input type="checkbox"/>
1x Sky koosolekul osalemine	Peeter Maasikas	Üldine	07 september 2022	Aktiivne	? <input checked="" type="checkbox"/> <input type="checkbox"/>
1x Sky koosolekul osalemine	Jüri Jürison (JJ)	Üldine	07 september 2022	Aktiivne	? <input checked="" type="checkbox"/> <input type="checkbox"/>
2x Firmakohtumine	Jüri Jürison (JJ)	FRTG	10 september 2022	Aktiivne	? <input checked="" type="checkbox"/> <input type="checkbox"/>
3x Firmakohtumine	Peeter Maasikas	FRTG	10 september 2022	Aktiivne	? <input checked="" type="checkbox"/> <input type="checkbox"/>
1x Firmakohtumine	Priit Vijandist	FRTG	10 september 2022	Aktiivne	? <input checked="" type="checkbox"/> <input type="checkbox"/>
4x Koosolekul osalemine (1h)	Peeter Maasikas	Tiimitöö	12 september 2022	Aktiivne	? <input checked="" type="checkbox"/> <input type="checkbox"/>
4x Kooolitusel osalemine (1h)	Peeter Maasikas	Üldine	20 september 2022	Aktiivne	? <input checked="" type="checkbox"/> <input type="checkbox"/>
4x Kooolitusel osalemine (1h)	Jüri Jürison (JJ)	Üldine	20 september 2022	Aktiivne	? <input checked="" type="checkbox"/> <input type="checkbox"/>
100x Telefonikõne	Peeter maasikas	FRTG	29 september 2022	Eemaldatud	? <input checked="" type="checkbox"/> <input type="checkbox"/>




Joonis 10. Punktikannete vaade.



Joonis 11. Punktide lisamise modaal.

Nõuetes on kirjeldatud soov näha punktide statistikat, mida oleks võimalik kitsendada kuupäevade järgi ja valdkonna järgi. Samuti lisati siia võimalus otsida statistikat nime järgi. Vastavalt sellele on koostatud punktide statistika vaade erinevate filtreerimisvõimalustega (Joonis 12). Kasutajale valitud liikme detailsemate punktikannete näitamiseks on samuti tekitatud vastav vaade, kus kuvatakse soovitud perioodi ning valdkonna kohta punktikanded (Joonis 13).

 **Stella Leego**
Personalijuht

- Liikmed
- Uued taotlused 1
- Lisa uus liige
- Liikmelisused
- Punktid
- Punktide statistika**
- Punktide tegevused

Punktide statistika

Kitsendamise valikud


Nimi Perioodi algus Perioodi lõpp

Valdkond


Otsi

Tulemused

Nimi	Punktisumma	
Peeter Maasikas	340	Detailid
Mari Mustikas	250	Detailid
Jüri Jürison (JJ)	220	Detailid
Priit Viljandist	60	Detailid
Tiina Kaasik	0	Detailid
Juku Juurikas (Juss)	0	Detailid



Joonis 12. Punktide statistika vaade.

 **Stella Leego**
Personalijuht

Liikmed

Uued taotlused 1

Lisa uus liige

Liikmelisused

Punktid

Punktide statistika

Punktide tegevused

Peeter Maasikas punktide statistika


Kitsendamise valikud

Perioodi algus:
 Perioodi lõpp:
 Valdkond:

Otsi

Punktide saamise ajalugu

Tegevuse nimetus	Valdkond	Saamise aeg	Punktisumma	Staat	
2x Koosolekul osalemine (1h)	FRTG	02 september 2022	40	Aktiivne	? ✓ ✗
1x Sky koosolekul osalemine	Üldine	07 september 2022	60	Aktiivne	? ✓ ✗
3x Firmskohtumine	FRTG	10 september 2022	90	Aktiivne	? ✓ ✗
4x Koosolekul osalemine (1h)	Tiimitöö	12 september 2022	80	Aktiivne	? ✓ ✗
4x Koollitusel osalemine (1h)	Üldine	20 september 2022	70	Aktiivne	? ✓ ✗
100x Telefonikõne	FRTG	29 september 2022	1500	Eemaldatud	? ✓ ✗



Joonis 13. Liikme punktide kannete detailvaade.

Uute nõuete seas on ka vajadus tekitada funktsionaalsus tegevuste lisamiseks, muutmiseks ning kustutamiseks. Vastavalt sellele on koostatud tegevuste nimekirja vaade, kus samuti järgitakse rakenduses seatud kirjade lisamise, muutmise ja kustutamise põhimõtet, kasutades läbivalt samu ikoone (Joonis 14).

Stella Leego
Personalijuht

Punktide tegevused

Nimekiri

Nimetus	Punktid	Valdkonnad, kus näidatakse punktisestamiset:	
Koosolekul osalemine (1h)	20	FRTG, DMTG, MTG, Projektid, Tiimitöö	<input checked="" type="checkbox"/> <input type="checkbox"/>
Koollituset osalemine (1h)	20	FRTG, DMTG, MTG, Projektid, Tiimitöö	<input checked="" type="checkbox"/> <input type="checkbox"/>
Telefonikõne	15	FRTG, Projektid	<input checked="" type="checkbox"/> <input type="checkbox"/>
Firmakohtumine	30	FRTG, Projektid	<input checked="" type="checkbox"/> <input type="checkbox"/>
Tegevus, millel pole sisestusi	60	FRTG, Projektid	<input checked="" type="checkbox"/> <input type="checkbox"/>

BEST
TULIJAH

Joonis 14. Tegevuste nimekirja vaade.

6.2.4 Mobiiltelefoniga kasutatavus

Ligipääsetavuse tõstmiseks on oluline tagada rakenduse (hea) kasutatavus ka mobiilivaadetes. Selle jaoks koostati mõned kasutajaliidese prototüübid (Lisa 3 – Mõned mobiilivaate kasutajaliidese prototüübid), kõikidele olemasolevatele vaadete prototüüpidele telefoni vastet ei koostatud, sest selle jaoks ei nähtud mõtet. Koostatud mobiilivaate prototüüpide eesmärk oli olla eeskujuks teiste vaadete loomisele, sest väga paljud vaated on üpris samasugused.

Mobiilivaate prototüüpide koostamisel arvestati asjaoluga, et sõrmedega on väikeseid nuppe raskem tabada, mistõttu tehti erinevad nupud ja andmesisestusväljad suuremaks. Lisaks pandi nende vahele ka suuremad vahed.

6.3 Andmebaasi täiendused

Muutunud nõue näidata „Peatatud“ liikmestaatus, mille eesmärk on näidata, kas antud tudeng on antud hetkel näiteks õppepuhkusel, andmebaasi disaini ei muuda. Muutub ainult liikmelisuse seisundidiagramm, mis on uuendatud kujul ära näidatud peatükis Lisa 6 – Uuenenud liikmelisuse seisundidiagramm.

Personalijuhiga vestluse käigus tuli välja vajadus küsida liitujate käest huvialasid. Hetkel disainitud andmebaas seda infot salvestada ei võimalda, mistõttu on see vaja lisada. Tarkvara peab huvialade valiku välja pakkuma. Samas on vaja jätta ka võimalus liikmeavalduses märkida „muu“ huviala, et anda võimalus avalduse täitjatel märkida huviala, mida valikus olemas ei ole.

See nõuab ühe tabeli juurde lisamist: *Liikmelisuse_huviala* (Lisa 4 – Tabelid Liikmelisus ja Liikmelisuse_huviala BEST-Estonia andmebaasi uues versioonis), mis võimaldab andmebaasi salvestada liikme huvialasid vabas vormis. Standardiseeritud huvialad salvestatakse rakenduse lähtekoodi. Alternatiivse variandina oleks saanud tekitada kaks uut tabelit, kus üks tabel hoiab huviala variante, ehk klassifikaatori tabel, ning teine tabel oleks seose tabel *Liikmelisus* tabeli ja *Huviala* tabeli vahel. Siin tekib probleem „muu huviala“ salvestamisega. Variant oleks tekitada lisaväli *Liikmelisus* tabelisse, kuid see variant ei takista „muu huviala“ väljale salvestamast sama huviala, mis juba on klassifikaatorina ette antud. Teine variant oleks salvestada „muu huviala“ uue variandi tekkimisel kohe klassifikaatori väärtusena, kuid siis on vaja märkida ära variantidena ette antavad huvialad, ehk kokkuvõttes nõuab veel ühte muutujat, tekitades liiga keerulise süsteemi.

Lisaks tehti muudatus ka kõikide tabelite primaarvõtmete veergudele, kus lisati juurde kitsendus *UNSIGNED*, mis eemaldab võimaluse lisada primaarvõtme veergu negatiivseid arve. See otsus tehti, sest ei nähtud ette olukorda, kus oleks vaja lisada primaarvõtmete veergu negatiivseid arve. See järeldus tehti ka andmebaasis olemasolevate andmete põhjal, kus primaarvõtmete väärtused algasid alati arvust „1“. Lisaks võimaldab see muudatus kasutada *AUTO_INCREMENT* omadust, mis iga uue rea lisamisel võimaldab genereerida lisatud rea jaoks uue unikaalse täisarvulise väärtuse. See lihtsustab oluliselt andmebaasi uute

ridade lisamise funktsionaalsust, eemaldades vajaduse küsida iga kord enne uue rea lisamist suurimat tabelis registreeritud võtmeväärtust. Oluline on ka siinkohal mainida, et MySQL ei toeta negatiivsete väärtuse lisamist veergu, millele on määratud *AUTO_INCREMENT* omadus [18].

Uutes nõuetes tuli välja vajadus tekitada funktsionaalsus tegevuste lisamiseks, muutmiseks ja kustutamiseks. Kuna hetkel ei ole *Tegevus* tabelil määratud primaarvõtme veerule *AUTO_INCREMENT* omadust (eelneva süsteemianalüüsi tulemuse kohaselt oli tegemist klassifikaatoriga, mille koodid pannakse paika inimkasutaja poolt), nõuaks see uue tegevuse lisamisel kasutaja poolt koodi sisestamist. Täiendav süsteemianalüüs näitas, et kasutajalt ei ole mõistlik koodi küsida, sest antud kood ei tähenda kasutajale mitte midagi. Seetõttu määratakse tabeli *Tegevus* primaarvõtme veerule *AUTO_INCREMENT* omadus.

Samanimeliste liikmete eristamiseks lisati *Isik* tabelisse uus väli nimega *hyydnimi*, et võimaldada rakenduses eristada samanimelisi liikmeid hüüdnime abil. Vastav muudatus on näha peatükis Lisa 5 – Tabelid Isik, Konto ja Osapool BEST-Estonia andmebaasi uues versioonis.

7 Iteratsioonide planeerimine

Iga iteratsiooni alguses tehakse valmis üks või mitu kasutajaliidese kavandit. Nende põhjal kogutakse tagasisidet ning tehakse vajalikud muudatused. Peale seda realiseeritakse kavandid koodis.

Lähtetöodes on esitatud funktsionaalsed nõuded kasutusmallidena (Stella Leego töö) ja kasutuslugudena (K. H. Kaljuse töö). Ka käesoleva töö tulemusena leitud täiendavad funktsionaalsed nõuded esitati kasutuslugudena. Käesolevas töös lähtutakse nendest nõuetest. Iteratsioonide käigus kasutajatele ekraanivormi kavandite näitamine võib neid nõudeid täpsustada.

7.1 Prioriteetide määramine

Prioriteetide määramisel lähtuti kõigepealt sellest, millised funktsionaalsused on rakenduse põhja jaoks vajalikud ning ilma milleta ei oleks antud arendusel üldse mõtet. Kõige olulisemaks funktsionaalsusteks peeti autentimist, liikmete andmete lisamist ja muutmist, liikmelisuse lisamist ja muutmist, punktide lisamist ja muutmist ning rollide määramist. Automatiseeritud tegevused, nagu näiteks TalTechi õppeinfosüsteemist üliõpilasstaatus kontrollimist, ei ole töö autori hinnangul niivõrd tähtsad, sest ilma selleta toimib rakendus endiselt edasi ning täidab endale seatud eesmärgid.

Samuti võeti prioriteetide määramise lähtepunktiks personalijuhi vaatenurk, eesmärgiga kõigepealt lihtsustada ja kiirendada tema tööd. Seetõttu saavad madalama prioriteedi funktsionaalsused, mille eesmärgiks on liikmete nimekirja kuvamine BEST-Estonia kodulehel ning sisselogitud kasutaja isikuandmete muutmise ning vaatamine.

Küll aga lihtsustaks BEST-Estonia kodulehel juhatuse koosseisu kuvamine siiski mingil määral personalijuhi tööd, võttes ära vajaduse iga uue juhatuse aasta alguses käsitsi juhatuse koosseisu andmeid muuta. Samas koosseisu andmete muutmise ei nõua hetkel töö autori ja organisatsiooni personalijuhi hinnangul nii palju tööd, et seda peaks tähtsaks pidama peaks.

Samas leiti, et seda funktsionaalsust ei ole väga keeruline lisada, mistõttu võiks selle võimalusel siiski realiseerida.

7.2 Keerukuse hindamine

Keerukuse hindamine on rakenduse funktsionaalsuste lisamise planeerimisel tähtis osa. Sellega üritatakse saada umbkaudset hinnangut, kui kaua mingisuguse arendusega võiks aega minna. Tihtipeale on suuremate ülesannete hinnangud ebatäpsed, sest inimene oskab hästi hinnata asju, mis jäävad ühte suurusjärku [19].

Kasutuslugude ja –mallide hindamiseks on erinevaid viise, millest populaarsemad on järgnevad.

- *Fibonacci*, kus keerukuse hinnangud järgivad Fibonacci jada: 1, 2, 3, 5, 8, 13...
- T-särgi suurus, kus keerukuse hinnangute variantideks on näiteks S, M, L ja XL
- *Powers of 2*, kus keerukuse hinnangute jada kasvab eksponentsiaalselt: 1, 2, 4, 8...

Hindamismeetodi valimisel tuleb arvestada asjaolu, et mida keerulisem on ülesanne, seda ebatäpsem on ka hinnang, mistõttu võiks järgmine keerukusaste olla palju suurem võrreldes eelmisega. Seepärast ei ole lineaarsed hinnangud nii atraktiivsed, mistõttu ei ole ka T-särgi suuruse hindamismeetod niivõrd efektiivne, sest keerukamate ülesannete korral jätab see välja teadmatusega arvestamise. Efektiivseks hindamiseks on vaja mõnda viisi, milles raskusastmete kasv ei ole lineaarne. Kaheks valikuks on siin *Fibonacci* jada ja *Powers of 2*, kus esimene on populaarsem, olles kasutusel näiteks planeerimispokkeri planeerimismeetodis. Seetõttu otsustas töö autor valida *Fibonacci* meetodi, sest see on kõige populaarsem valik ning lahendab ära ka teadmatuse hindamise probleemi.

7.3 Funktsionaalsete nõuete jaotamine iteratsioonide vahel

Eelnevatest töödest on võetud kokku olemasolevad kasutuslood ja mallid, mis kõik on koondatud tabelisse (Lisa 8 – Funktsionaalsuste hindamine). Lisatud on ka uued kasutuslood, mis tekkisid muutunud ärivajaduste tagajärjel ning eemaldatud on kasutuslood, mis hetkel ei

anna ärilist kasu. Tabelisse on lisatud juurde ka prioriteedi ja keerukuse hinnangund. Vastavalt nendele on planeeritud ka iteratsioonid.

Tabeli järgi on tulemuseks saadud perioodide keerukused järgmised:

- Periood 1: 34
- Periood 2: 32
- Periood 3: 30

Perioodide planeerimisel üritati jälgida, et iga periood oleks umbes sama keerukuse mahuga ning iga periood võiks keskenduda seotud või sarnastele funktsionaalsustele. Perioodide planeerimisel võeti arvesse ka seda, et esimese perioodi jooksul tuleb tegeleda rakenduse põhja valmistamisega, sealhulgas arhitektuuri kavandamisega ning rakenduse hostimisega. Need keerukused on peidetud kasutaja tuvastamise alla. Kasutaja tuvastamise funktsionaalsuse lisamine ise pole keeruline, sest see funktsionaalsus on Laraveli raamistikul juba olemas, kuid samas paralleelselt selle funktsionaalsuse sobitamisega olemasoleva andmebaasiga tegeletakse ka rakenduse hostimise ja arhitektuuri kavandamisega.

Lisaks võeti arvesse ka seda, et viimase perioodi jooksul on vaja tegeleda erinevate parandustega ning lisaks kohandatakse viimase perioodi jooksul rakendus mobiilivaate jaoks.

Iga perioodi pikkuseks planeeriti kolm nädalat.

8 Veebirakenduse arendus

Selles peatükis kirjutatakse loodud veebirakendusest. Valminud veebirakenduse kasutajaliidest on tehtud mõned kuvatõmmised, mis on leitavad Lisa 11 – Mõned kuvatõmmised valminud rakenduse vaadetest.

8.1 Arhitektuur

Rakendus on arendatud monoliitses stiilis, ehk kasutajaliides ehk *front-end* on samas koodibaasis kui tagarakendus ehk *back-end*.

8.1.1 Laraveli MVC arhitektuur

Laraveli arhitektuur on üles ehitatud MVC stiilis, kus mudel ehk *Model* esindab rakenduse andmestruktuure ja ärioloogikat. Mudelid suhtlevad andmebaasidega, töötlevad andmeid ja määravad seoseid tabelite vahel. Vaade ehk *View* esindab kasutajaliidese ja esitluskihi poolt. Laravel kasutab vaadete jaoks Blade malle, mis vastutavad andmete kasutajale kuvamise eest. Blade võimaldab arendajatel luua korduvkasutatavaid komponente, muutes rakenduse arenduse lihtsamaks ja koodi korduvkasutatavamaks. Kontroller ehk *Controller* toimib vahendajana mudelite ja vaadete vahel, käsitledes kasutajate sisestusi ja hallates nende vahelist andmevoogu. Kontrollerid võtavad vastu HTTP-päringuid, töötlevad sisendit, suhtlevad mudelitega toimingute tegemiseks ja tagastavad vastuseid vaadete renderdamise kaudu.

Laravel võimaldab väga kiiresti valmistada rakenduste prototüüpi, sest see pakub paljusid valmisfunktsionaalsuseid. Standardne viis on registreerida valideerimine kontrollerites, kusjuures valideerimise loogika seotakse tihedalt ka õigustega [20]. Väiksemate rakenduste puhul või olukordades, kus on vaja kiiresti prototüüpi arendada, võimaldavad etteantud lahendused kiiresti arendada ja tulemuseni jõuda.

8.1.2 Rakenduse tõlkimine

Jorgen Ööpiku töös tõsteti esile vajadust kasutada rakendust nii inglise kui ka eesti keeles. Laravel raamistikul on tõlgete lisamiseks väga hea süsteem, kus iga uue keele lisamiseks on ainult vaja lisada uus sõnastikufail. Tõlkefailid on php skriptid, mis hoiavad endas ühte sõnastiku-formaadis massiivi, võimaldades grupeerida tõlkeid erinevate võtmete kaupa (Joonis 15).

```
<?php
return [
    'auth' => [
        'login' => [
            'header' => 'Logi sisse',
            'button' => [
                'login' => 'Logi sisse',
                'forgotPassword' => 'Unustasid salasõna?'
            ],
            'messages' => [
                'accountDisabled' => 'Kasutaja ei ole aktiivne. Palun võtta ühendust
süsteemi administraatoriga.',
                'invalidCredentials' => 'Sisselogimine ebaõnnestus. Palun proovige
uuesti.'
            ]
        ],
        'forgotPassword' => [
            'header' => 'Unustasid salasõna?',
            'email' => 'Meiliaadress',
            'button' => [
                'send' => 'Saada',
                'nevermind' => 'Salasõna tuli meelde! Logi sisse'
            ]
        ]
    ],
];
```

Joonis 15. Näide eestikeelse tõlkesõnastiku faili sisust.

Võtmed on valitud võimalikult informatiivsed, eesmärgiga anda vajalikku infot edasi ka juhul, kui kasutatava keele tõlge on puudu. Laravel võimaldab kasutada võtmetena ka tõlgitavaid fraase, kuid otsustati valida hierarhiline võtmete struktuur, sest see võimaldab vajadusel tõlkeid väga lihtsalt muuta. Fraaside tõlke muutmine võib tuua kaasa vajaduse muuta tõlkevõtit iseennast.

Tõlgete kasutamiseks on Laravelil olemas globaalne funktsioon „__“. Järgnevalt on toodud tõlkeressursi kasutamise näide (Joonis 16). Enne toodud näitega tagastatakse antud

tõlkeressursi võtme puhul fraas „Unustasid salasõna?“. Juhul, kui tõlkeressurssi soovitud keelele pole, siis tagastatakse võti ise.

```
<a class="btn btn-link forgot-password" href="{{route('auth.forgotpassword')}}">
    {{ __('pages.auth.login.button.forgotPassword') }}
</a>
```

Joonis 16. Tõlkeressursi kasutamise näide.

Laravelil puudub vaikimisi keele valimise funktsionaalsus, kuid tegemist pole keerulise probleemiga. Töös arendatavas rakenduses on keele valiku salvestamine lahendatud selle veebilehitseja küpsisesse salvestamisega. Seejärel kontrollitakse iga HTTP-päringu alguses küpsise olemasolu ning sätitakse vastavalt keel. Kui küpsist ei ole, siis valitakse vaikekeel (Joonis 17). Vaikekeel saadakse rakenduse konfiguratsioonifailist, kus on vaikekeeleks sätitud eesti keel.

```
$locale = Cookie::get('lang') ?: config('app.locale');
app()->setLocale($locale);
```

Joonis 17. Päringu keele seadistamine.

8.1.3 CQRS disaini kasutamine

Rakenduse arhitektuur on rakenduse arenduse käigus oluliselt muutunud. Alguses oli äriloogika rohkem seotud veebiliidese poolega, näiteks tegeles andmete valideerimisega kontrollid. Lisaks ei olnud alguses planeeritud kasutada käsu vahendaja kihti, sest see tundus olevat rakenduse arenduse alguses liiga keeruline arendus, minnes vastuollu YAGNI printsiibiga. Rakenduse arhitektuuri esialgsel kavandamisel üritati vältida olukorda, kus muututakse arhitektuuri astronautiks. Arhitektuuri astronaut on rakenduse arhitektuuri kavandaja, kes loob väga abstraktse ja keerulise arhitektuuri, vältides tegelikke ärilisi nõudeid [21]. Rakenduse arhitektuuri kavandamisel on mõistlik järgida ka erinevaid häid programmeerimise tavasid, näiteks YAGNI, mis otseselt ütleb, et pole mõtet arendada funktsionaalsust ja teha disaini asjadele, mida pole vaja [22]. Rakenduse kasvades arendati arhitektuuri edasi, et lihtsustada uute funktsionaalsuste lisamist ja parandada koodi arusaadavust, eraldades omavahel erinevate kohustustega kihid.

Töö autor nägi Laraveli arhitektuuris probleeme, nimelt mõned kihid vastutasid mitme tegevuse eest, mis käib vastu SOLID printsiibile *single-responsibility* (iga tarkvaraelement peab vastutama ühe kindla asja eest). Laravelis on mudelid üles ehitatud vastutama mitme asja eest: äriloogika, andmeedastus vaadetele ja andmete küsimine ning muutmine andmebaasis. Väiksemate rakenduste puhul, kus on vähe funktsionaalsust, on antud disain aktsepteeritav, sest pole mõtet kavandada arhitektuuri või kirjutada koodi, mida pole vaja. Kuna BEST-Estonia liikmehalduse ja punktiarvestuse rakenduses on funktsionaalsuseid pigem mitmeid ning neid tekib tulevikus kindlasti juurde, siis tuleb tekitada võimalus eraldada äriloogika, andmebaasiga suhtlus ning andmeedastus erinevatesse kihtidesse.

Eelnimetatud kihtide lahku löömiseks valiti CQRS lähenemine. Tegemist on muustriga, mis eraldab lugemis- ja kirjutamistoimingud, soodustades selget vahet käsu (andmete kirjutamine) ja päringu (andmete lugemine) mudelite vahel. Neid kohustusi jagades on kergem igat mudelit eraldi optimeerida, hooldada ja edasi arendada, võttes arvesse antud mudeli spetsiifilisi vajadusi, samal ajal jättes teised mudelid täielikult mõjutamata. Selle tulemuseks on lihtsustatud ja fokuseeritud kood. Käskude ja päringute objektide nimed nimetatakse täpselt vajaliku äriloogika järgi, joondades rakenduse arhitektuuri valdkonnas kasutatava keelega, kus käsud ja päringud väljendavad otseselt ärilisi vajadusi.

CQRS eeliseks on võimalus anda igale käsule või päringule oma valdkond, näiteks teise andmebaasiga suhtlemine, suhtlus välise API-ga vms. Selline disain võimaldab igat käsku ja päringut kasutada igal pool samamoodi, olenemata erinevast sisust. See vastab ka KISS reeglile, olles peale vaadates väga lihtne ja arusaadav. Näitena toodud (Lisa 9 – Käsu realisatsioon tegevuse kustutamise näitel) käsu käitleja nimega *DeleteActivityHandler* tegeleb tegevuse kustutamisega. Antud meetodi ainus ülesanne on vastutada andmebaasist rea leidmise ja selle kustutamise eest.

CQRS disainiga kaasneb üldjuhul ka käskude vahendaja, mis oskab siduda etteantud käsku vastava käsu käitlejaga. Näitena tuuakse tegevuse kustutamine. Kontrolleris defineeritakse käsu objekt, antakse kaasa vajalikud parameetrite väärtused ning antud objekt saadetakse käskude vahendajale (Joonis 18). Antud näite puhul on käsuobjekti definitsiooni nimeks *DeleteActivityCommand* (Lisa 9 – Käsu realisatsioon tegevuse kustutamise näitel).

Rakenduses tegeleb käsu käitleja ehk *Mediator* käsu parameetrite väärtuste valideerimisega, käsu täitmiseks vajalike õiguste kontrollimisega ning käsu täitja ehk *Handler* leidmisega. Igal käsu täitjal on võimalus tagastada ka vastuse objekt, kuid antud näite puhul ei ole seda vaja. Antud tegevuste täitmiseks kutsutakse välja meetod *send*, mille parameetri oodatavaks väärtuseks on käsu või pärimise objekt (Joonis 19). Valideerimisreeglid defineeritakse samuti eraldi klassina, antud näite puhul nimega *DeleteActivityValidator* (Lisa 9 – Käsu realisatsioon tegevuse kustutamise näitel). See võimaldab eemaldada ärioloogikat veebiliidese poolt, mille tulemusena tegelevad kontrollid ainult HTTP-päringute vastuvõtmisega ja seovad selle vastava käsu või päringuga.

```
public function delete(int $id)
{
    $this->mediator->send(new DeleteActivityCommand($id));
    return ResponseHelpers::deleteSuccessJson();
}
```

Joonis 18. Käsuobjekti kasutamise tegevuse kustutamise näitel.

```
/**
 * @template TResult
 * @param MediatorMessage<TResult> $message
 * @return TResult
 * @throws Exception
 */
public function send(MediatorMessage $message)
{
    $handler = $this->getHandler($message);

    if ($handler == null) {
        throw new Exception(sprintf("Handler not found for %s", get_class($message)));
    }

    $requiredPermission = $handler->getRequiredPermission();
    $this->permissionService->throwIfCurrentUserNotAllowed($requiredPermission);

    $this->throwIfMessageInvalid($message);

    return $handler->invoke($message);
}
```

Joonis 19. Käsu vahendaja meetod.

Lisaks eelnimetatule võimaldab CQRS rakendust väga lihtsalt edasiarendada. Olenemata sellest, et antud töös on rakenduse funktsionaalsuste skoop piiratud punktihaldusele ja

liikmehaldusele, nähakse vajadust arendada juurde funktsionaalsused, mis on välja toodud töö sissejuhatuses ja muudetud nõuete peatükis.

8.1.4 Klassifikaatorite andmebaasist pärimise vältimine

Rakenduses on nõuete tagajärjel andmebaasi defineeritud mitmed klassifikaatorid, näiteks *Isiku_seisundi_liik*. Selleks, et ei peaks antud klassifikaatoreid iga kord andmete andmebaasist pärimisel juurde kaasama, tekitati rakenduses baasklass nimega *Enumeration* (Lisa 10 – Enumeration klassi definitsioon). Arendaja saab antud klassi kasutada klassifikaatori defineerimiseks koodis, et võimaldada kiirem väärtustamine, ilma andmebaasist andmete pärimiseta. Iga klassifikaator omab vähemalt kahte väärtust: kood ja nimetus, kuid parameetreid on võimalik juurde lisada vastavalt arendaja vajadustele. Iga klassifikaatori väärtus salvestatakse staatilise omadusena (*property*). Selleks, et klassifikaatorile tekiks väärtused, tuleb HTTP-päringu alguses kutsuda välja vastava klassi meetod *initValues*.

Järgnevalt on toodud klassifikaatori klassi definitsioon tabeli *Isiku_seisundi_liik* põhjal.

```
class PersonStatusType extends Enumeration
{
  public static PersonStatusType $active;
  public static PersonStatusType $inactive;

  public function __construct(int $id, string $name)
  {
    parent::__construct($id, $name);
  }

  public static function initStaticFields()
  {
    PersonStatusType::$active = new PersonStatusType(
      1,
      __('enumerations.personStatusType.active'));
    PersonStatusType::$inactive = new PersonStatusType(
      2,
      __('enumerations.personStatusType.inactive'));
  }
}
```

Joonis 20. Klassifikaatori klassi definitsioon *Isiku_seisundi_liik* tabeli põhjal.

Sellisel arhitektuurilisel lähenemisel on lisaks andmebaasist pärimise vajaduse kadumise eelisele ka eelis tõlgete lisamise lihtsuse osas. Selline lähenemine ei nõua uue tõlke tekkimisel andmebaasi struktuuri muutmist läbi uue veeru lisamise, vaid arendajal on võimalus tekitada uus tõlkefail ning vastavad tõlked sinna sisse kanda.

Sellist lähenemist on mõistlik kasutada ainult juhul, kui klassifikaatori väärtuseid muudetakse väga harva, sest väärtuste muutumisel on vaja teha lähtekoodis muutuseid.

8.2 Järgitud arendustavad

Antud projekti arendamisel järgiti mitmeid häid programmeerimise tavasid, millest enamus on kirjeldatud raamatus Clean Code [23]. Häid tavasid järgiti eesmärgiga võimaldada tulevikus teistel arendajatel lihtsamini kirjutatud koodist aru saada ja seda edasi arendada.

Mõned järgitud tavad Clean Code raamatust:

- Eesmärki näitavad nimetused
- Erindite väljastamine veakoodide tagastamise asemel
- Kõrvalnähtude vältimine
- Kordamise vältimine
- Kood peab suutma selgitada ennast ilma kommentaarideta
 - Kommentaarid ei ole halvad, näiteks kui selgitavad ärivajadust
- Vältida koodi, mida pole vaja

Punkti „Kood peab suutma selgitada ennast ilma kommentaarideta“ juurde tuleb lisada mõned erandid, mis on lahti selgitatud ka raamatus Clean Code. Antud rakenduses kommenteeriti kohti, mis nõudsid konteksti selgitamist, näiteks mõnda spetsiifilist ärivajadust, mida muud moodi koodiga ei saanud selgitada või koodiga selgitamine oleks halvendanud koodi loetavust.

Punkti „Vältida koodi, mida pole vaja“ osas tehti rakenduse arendamisel mõnes kohas erand. Probleem tekib tühjade käsu- või päringuobjektide defineerimisel, millel pole konstruktoris parameetreid. PhpStorm, millega rakendust arendati, sisaldab kiirklahvi funktsiooni nimega

„Go to declaration or usages“, mille abil on võimalik leida definitsiooni (näiteks muutuja, klass) kasutust või deklareerimiskohta. Kui klassi konstruktor on defineeritud, saab selle abil kiiresti leida antud klassi konstruktori abil kohti, kus kohas antud klass väärtustatakse. Näitena on toodud allolev joonis (Joonis 21), kus klassi nime definitsioonil on näha viis kasutuskohta (*usages*) kuid konstruktori kasutusi on kaks. Kui konstruktorit ei ole defineeritud, siis peab kasutama klassi nime kiirklahvi funktsiooniga ning see teeb klassi väärtustamise kohtade leidmise keeruliseks, sest siis antakse kasutajale ette valik minna lisaks klassi väärtustamise kohtadele ka näiteks failide päistesse, kus antud klassi kasutamine defineeritakse. Selle lahendamiseks on võimalik defineerida tühi konstruktor.

```
5 usages  ± HanY2B
readonly class HasNewMembershipApplicationsQuery implements Query
{
    2 usages  ± HanY2B
    public function __construct()
    {
    }
}
```

Joonis 21. Tühja konstruktori kasutamine uute liikmeavalduste olemasolu kontrolli näitel.

Rakenduse arenduse käigus järgiti ka enamlevinuid programmeerimise häid tavasid, millest mõned on need:

- KISS
- YAGNI
- POLA
- SOLID

Eelnimetatud põhimõtete eesmärk on parandada koodi kvaliteeti, hooldavust ja loetavust. KISS (*Keep It Simple Stupid*) julgustab kirjutama arusaadavat koodi [24]. YAGNI (*You Aren't Gonna Need It*) keskendub mittevajalike funktsioonide vältimisele ja liigsele projekteerimisele [22]. POLA (*Principle of Least Astonishment*) rõhutab järjepidava ja

vastavalt eeldustele käitüva koodi tähtsust [25]. SOLID on juhiste kogum koodi struktureerimiseks, et soodustada paindlikkust ja hooldatavust [26].

Nende põhimõtete kasutamiseks seati rakenduse arendamisel esikohale lihtsus, reaalsus ja järjepidevus. Nende saavutamiseks kirjutati selget, kokkuvõtvat koodi, mis järgib tavalisi mustreid ja tavasid, vältides tarbetut keerukust või funktsionaalsust, tagades etteaimatavalt käitüva koodi. Rakendus üritati arendada võimalikult modulaarseks ja kergesti hooldatavaks, et vähendada vigade tekkimise ja tehnilise võla kasvamise riski ning soodustada tulevikus paremat koostööd teiste arendajatega, kes võivad antud projekti edasi arendada.

8.3 Kasutatud teegid

Rakenduse arendamise lihtsustamiseks kasutati mitmeid olemasolevaid teeke. Põhiliselt kasutati väliseid teeke rakenduse esipoole arendamiseks, mistõttu on kõik lisatud teegid *JavaScript* või *CSS* teegid. Põhiliseks probleemiks on veebilehitsejate ebaühtlus erinevate funktsionaalsuste toetamise suhtes [27].

Kasutajaliidese lihtsamaks koostamiseks otsustati kasutada *Bootstrap 5 CSS* raamistikku. Tegemist on populaarse avatud lähtekoodiga *CSS* raamistikuga, mis on loodud veebirakenduste kasutajaliidese lihtsamaks ja kiiremaks arendamiseks. Sellel on olemas valmiskomponendid, näiteks navigeerimisriba, modaalid ja nupud [28]. Seetõttu ei nõua kasutajaliidese valmistamine eriti palju tööd.

Lisaks *Bootstrap* raamistikule võeti kasutajaliidese lihtsamaks arendamiseks kasutusele *jQuery* teek. See muudab HTML-dokumentide manipuleerimise, sündmuste käitlemise ja animatsioonid palju lihtsamaks, võimaldades kirjutatud koodil toimida kindlalt kõikides veebilehitsejates.

Kuupäevade andmeväljade jaoks võeti kasutusele *flatpickr v4* [29]. Tegemist on teegiga, mille eesmärk on ühtlustada kuupäeva ja kellaaja valiku andmevälja välimust ja funktsionaalsust kõikide veebilehitsejate vahel. Teek võeti kasutusele põhiliselt seepärast, et osades levinumates veebilehitsejates on ainult osaline või lausa koguni puudub vaikimisi toetus kuupäeva ja kellaaja valiku väljale [30].

Lisaks võeti kuupäevadega töö tegemise lihtsustamiseks kasutusele teek *moment.js* [31]. Tegemist on populaarse teegiga, mida kasutatakse kuupäevade ja kellaegade valideerimiseks, manipuleerimiseks ja kuvamiseks erinevates formaatides. Selle kasutamine lihtsustab erinevates vormingus kuupäevaandmetega töötamist. Töö arendusprotsessi lõpus juhiti töö autori tähelepanu asjaolule, et alates 2020. aastast on tegemist pärandprojektiga, mida enam aktiivselt ei arendata. Seetõttu on vaja võimalusel antud teek millegi muu vastu välja vahetada.

Andmeväljade, kus on võimalik valida korraga mitu valikut, kasutamiseks võeti kasutusele *select2* teek [32]. See funktsionaalsus on samuti ebakvaliteetne ning erineb erinevate veebilehitsejate vahel. *Select2* toetab otsimist, märgistamist, kaugandmekogumeid, tulemuste lõputut kerimist ja paljusid muid laialdaselt kasutatavaid valikuid. Antud projektis kasutati antud teeki põhiliselt mitme valikuvõimalusega valikväljade jaoks.

9 Valideerimine

Antud töö raames uuriti valminud rakenduse kasutatavust võrreldes vana, hetkel kasutuses oleva süsteemiga. Selle jaoks kohtuti organisatsiooni koordinaatorite ja personalijuhiga ning lasti valminud rakendust kasutada: koordinaatorid katsetasid punktisisestust ning personalijuht katsetas liikmehalduse funktsioone ja samuti punktisisestust. Peale seda lasti igal kasutajal võrrelda uut rakendust vana süsteemiga ning koguti kokku nende arvamused. Sisuliselt on tegemist A/B testimisega [33].

Info kogumise ajal hoiti tähelepanu alla valideerimise probleemidele, milleks on kogutud info usaldusväärsus ja kehtivus. Usaldusväärse probleem tekib, kui sama testi korrates ei tule samad vastused. Kehtivus näitab testi vastuste asjakohasust, ehk kas saadud vastused peegeldavad päriselt olemasolevaid probleeme, millele tahetakse vastust saada. Need probleemid tekivad testkasutajate suurte individuaalsete erinevuste tõttu [14]. Olenemata saadavate vastuste kvaliteedi probleemidest on testimine endiselt oluline, sest vigadega informatsioon on siiski alati parem, kui puuduv informatsioon [14].

Kõigepealt lasti personalijuhil kasutada arendatud rakenduse kõiki funktsioone ning uuriti personalijuhilt, mis on tema arvamus uue rakenduse kohta pealt võrreldes praegu kasutusel oleva süsteemiga. Tulemused olid positiivsed, valminud rakendust on oluliselt mugavam kasutada, näeb kaasaegsem välja ja kasutajale näidatakse ainult seda informatsiooni, mida on vaja. Kõige olulisemaks kasuteguriks hinnati võimalust kasutada sama rakendust liikmeavalduste täitmiseks, mille tulemusel ei pea personalijuht enam käsitsi andmeid sisestama.

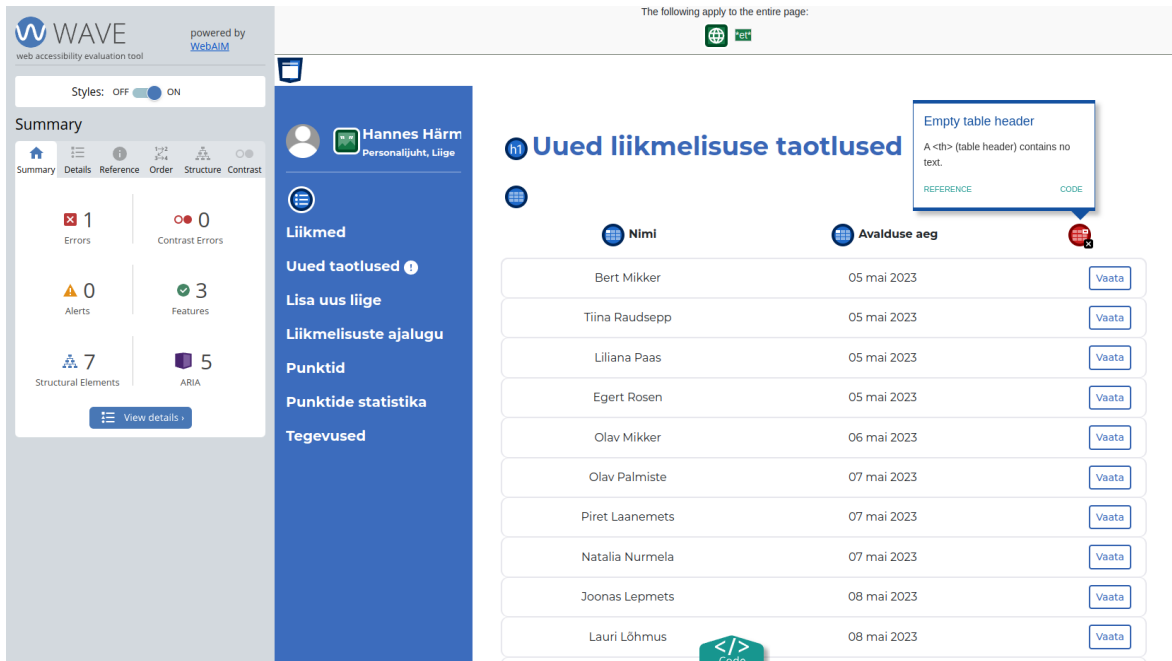
Seejärel paluti personalijuhil ja koordinaatoritel proovida punktisisestust praeguses süsteemis ning uuriti nende arvamust uue rakendus kohta võrreldes vana süsteemiga. Siin on samuti tulemused positiivsed. Rakenduse kasutatavust hinnati oluliselt mugavamaks võrreldes vana süsteemiga, kusjuures kõige tähtsamaks peeti hirmu kadumist süsteemi katkitemise ees. Lisaks võimaldab punktide statistika nägemine saada hea ülevaate töögrupis toimuva kohta.

Kokkuvõttes on arendatud rakendus oluliselt kasutajasõbralikum ning parandab mõned olulised probleemid hetkel kasutatavates süsteemides.

Kasutatavuse jälgimiseks integreeriti rakendus Google Analytics tööriistaga. See tööriist kogub rakenduse kasutajate käitumise kohta statistikat, mille põhjal saab vajadusel rakenduses teha muudatusi kasutajamugavuse ja efektiivsuse tõstmiseks [34]. Olgugi, et antud tööriist on pigem mõeldud avalike veebilehtede, nagu näiteks veebipoodide või blogide jälgimiseks, on tegemist siiski väga tõhusa tööriistaga, millega saab koguda statistikat ka privaatsete rakenduste kohta.

Kahjuks pole võimalik antud töö raames Google Analyticsist kasu saada, sest statistikat saab hakata koguma alles peale rakenduse kasutuselevõttu.

Rakenduse kasutatavuse testimiseks kasutati rakenduse arendamise ajal WAVE veebilehitseja pistikprogrammi [35], mille põhieesmärgiks on kontrollida veebilehe vastavust WCAG soovitudele. Funktsionaalsuse arendamise jooksul kontrolliti lisandunud vaateid selle tööriistaga ning vigade tekkel need parandati koheselt. Eesmärgiks seati parandada kõik vead, hoiatused üritati samuti ära parandada, kuid neid ei peetud niivõrd oluliseks. Järgnevalt toodud joonis näitab ühte parandamist vajavat viga, mis ütleb, et tabeli päises ei tohiks olla tühja lahtrit (Joonis 22).



Joonis 22. Veebirakenduse kontrollimine WAVE tööriistaga.

Rakenduse töökiiruse testimiseks kasutati veebilehitseja pistikprogrammi *Page load time*, mille ainuke eesmärk on näidata veebilehe laadimise kiirust [36]. Laadimise kiirus näidatakse sekundites veebilehitseja ülemises paremas nurgas. Näitena on toodud liikmete lehe laadimise kiirus, milleks on 0.28 sekundit, kui andmebaasis on 2000 liikme kirjet (Joonis 23). Eesmärgiks seati, et kõik rakenduse lehed laeksid alla kahe sekundi, ideaalis alla ühe sekundi. Kui leidis mõni leht, mille laadimise kiirus ei olnud sobilik, prooviti seda optimeerida. Enamasti oli probleemiks ebaefektiivne SQL-päring.



Joonis 23. Rakenduse kiiruse testimine.

10 Arendusvaade ja rakenduse edasi arendamine

Töö tulemusena on valminud uus andmebaasirakendus punktide arvestamiseks ja liikmehalduseks, kuid BEST-Estonia töös ei ole seda veel kasutusele võetud. Enne ei saa uut rakendust kasutusele võtta, kui üldise punktide statistika leht (punktid.best.ee) pole uue andmebaasiga seotud. Peale seda on vaja teha koostööd Jorgen Ööpikuga, et vaadata üle tema koostatud andmete migratsiooni skriptid, neid vajadusel kohandada ning lõpuks vanast andmebaasist andmed uude andmebaasi üle tõsta. Peale seda on võimalik uut rakendust hakata BEST-Estonia tööprotsessides kasutama. Rakendus ise on kasutusvalmis ning ei vaja uusi arendusi. Küll aga on vaja jälgida pärast rakenduse kasutuselevõttu selle kasutamise statistikat ning vajadusel vastavalt sellele rakenduse kasutatavust parandada.

Eesmärk on valminud rakendus võtta kasutusele enne 2023. aasta septembrikuud, et eemaldada vana süsteemi kasutamisega tekkivaid probleeme uute liikmete liitumisel. Selle jaoks alustatakse 2023. aasta juunis punktide statistika lehe uuendamisega ning tehakse paralleelselt koostööd Jorgen Ööpikuga andmete migratsioonide skriptide kohandamisega. See tagab piisavalt pika ajavahemiku, mille jooksul saab tekkivad probleemid enne sügist ära lahendada.

Arendamata jäi üks funktsionaalsus, milleks on Tallinna Tehnikaülikooli õppeinfosüsteemist BESTi liikmete üliõpilasstaatus kontrollimine. Põhjuseks on pikem suhtluse vajadus, et teha kindlaks kuidas seda oleks võimalik teha. Töö autor tegeleb sellega peale töö esitamist aktiivselt edasi.

Rakenduse arenduse käigus jäi lahendamata andmeekspordi funktsionaalsusega kaasnevad probleemid. Kõige olulisem probleem on isikuandmete lekkimise oht, lisaks pole selle funktsionaalsuse olemasolul rakendus enam keskne andmebaasirakendus, kus andmeid ja õiguseid hallata. Andmeekspordiga kaasnevad probleemid jäid antud töö raames lahendamata, mistõttu tuleksid need esmajärjekorras ära lahendada.

Olenemata sellest, et rakendus ise antud töö skoobi suhtes uusi arendusi ei vaja, tuleks käesoleva töö autori hinnangul valminud rakendust edasi arendada, et lahendada ära antud töös lahendamata jäänud probleemid, millest kõige olulisem on isikuandmete kaitse

põhimõtete järgimine. Lisaks on vaja analüüsida olemasolevate süsteemide integreerimise võimalusi. Näiteks võiks rakendus olla seotud Discordi suhtluskanali või projektide kokkuvõtte rakendusega, et vähendada liikmehalduse juures korduvaid tegevusi, millest põhilised on õiguste haldamine.

Selle jaoks soovib töö autor BEST-Estonial osta sisse arendusteenust, millega eelnimetatud probleemid ära lahendada. Tegemist on väga suure probleemide hulgaga, mida ei ole võimalik vabatahtlike abil lihtsalt lahendada. Personalijuhhi hinnangul ei leidu vähemalt järgmise kahe aasta jooksul organisatsioonis piisavate programmeerimisalaste teadmistega üliõpilast, kes kasvõi lõputöö raames võiks mõned olemasolevad probleemid ära lahendada. Pealekauba on tegemist kiiresti lahendust vajavate, keeruliste ja sügavamalt analüüsi vajavate probleemidega, mida professionaalset arendusteenust pakkuv ettevõtte suudab palju efektiivsemalt ja korrektsemalt lahendada. Sellise lähenemise probleemiks on arenduse maksumus.

Lisaks eelnimetatule soovib töö autor BEST-Estonial üle vaadata organisatsiooni koduleht, et kohandada see paremini juhatuse koosseisu kuvamisele.

11 Kokkuvõte

Käesoleva töö eesmärk oli arendada üliõpilasorganisatsioonile BEST-Estonia uus liikmelisuse ja punktisüsteemi andmebaasirakendus. Töö sisuks oli teha kindlaks uuenenud nõuded rakendusele, valida sobiv raamistik ja hostimiskeskond rakenduse arendamiseks, kirjeldada järgitavad kasutajaliidese head praktikad, täiendada süsteemi arendamiseks juba tehtud tööd vastavalt uuenenud nõuetele ning teha valmis uus rakendus.

Töö lähtepositsiooniks on kolm eelnevat lõputööd, mille raames on tehtud kindlaks nõuded süsteemile ning on loodud uus andmebaas ja andmete migreerimise skriptid eesmärgiga kanda andmeid vanadest andmebaasidest uude andmebaasi.

Töö tulemusena on valminud kasutusvalmis veebirakendus, mille kasutuselevõtuks on vaja veel uuendada üldise punktistatistika veebilehte, et see andmeid uuest andmebaasist võtaks ning teha koostööd eelmise selle süsteemi arendamise teemalise lõputöö autoriga andmete uude andmebaasi ülekandmise osas, sest tema lõputöö raames valmisid andmete ülekandmise skriptid. Töö raames jäi lisamata funktsionaalsus, mille abil saab pärida Tallinna Tehnikaülikooli õppeinfosüsteemist üliõpilase staatust, eesmärgiga automatiseerida mitteaktiivsete liikmete andmete uuendamist. Ülejäänud eesmärgid on töö autori hinnangul täidetud.

Kasutatud kirjanduse loetelu

- [1] J. Ööpik, „Üliõpilasorganisatsiooni BEST-Estonia uude andmebaasi andmete ülekandmise ettevalmistamine,“ Tallinna Tehnikaülikooli Tarkvarateaduse instituut, 2022.
- [2] S. Leego, „Üliõpilasorganisatsiooni BEST-Estonia liikmelisuse arvestuse tarkvara kavandamine,“ Tallinna Tehnikaülikooli Tarkvarateaduse instituut, 2021.
- [3] K. H. Kaljus, „Üliõpilasorganisatsiooni BEST-Estonia punktisüsteemi tarkvara uue versiooni arendamine,“ Tallinna Tehnikaülikooli Tarkvarateaduse instituut, 2021.
- [4] Board of European Students of Technology, „About BEST,“ [Võrgumaterjal]. Available: <https://www.best.eu.org/aboutBEST/welcome.jsp>. [Kasutatud 20 4 2023].
- [5] C. Juurik, Interviewee, *Privaatne suhtlus*. [Intervjuu]. 19 4 2023.
- [6] „Stack Overflow Developer Survey,“ [Võrgumaterjal]. Available: <https://survey.stackoverflow.co/2022/#technology>. [Kasutatud 2023 4 23].
- [7] „TIOBE Index for May 2023,“ [Võrgumaterjal]. Available: <https://www.tiobe.com/tiobe-index/>. [Kasutatud 14 5 2023].
- [8] CodeIgniter, „CodeIgniter,“ GitHub, [Võrgumaterjal]. Available: <https://github.com/codeigniter4/CodeIgniter4>. [Kasutatud 23 4 2023].
- [9] Laravel, „Laravel,“ GitHub, [Võrgumaterjal]. Available: <https://github.com/laravel/framework>. [Kasutatud 23 4 2023].
- [10] „Laravel,“ Wikipedia, 28 3 2023. [Võrgumaterjal]. Available: <https://en.wikipedia.org/wiki/Laravel>. [Kasutatud 24 4 2023].
- [11] „Installation,“ Laravel, [Võrgumaterjal]. Available: <https://laravel.com/docs/10.x#why-laravel>. [Kasutatud 24 4 2023].
- [12] „Welcome to CodeIgniter4,“ CodeIgniter, 26 3 2023. [Võrgumaterjal]. Available: https://codeigniter.com/user_guide/intro/index.html. [Kasutatud 24 4 2023].
- [13] „Laravel vs. Symfony: A Side by Side Comparison,“ Adevait, [Võrgumaterjal]. Available: <https://adevait.com/laravel/laravel-vs-symfony-comparison>. [Kasutatud 24 4 2023].
- [14] J. Nielsen, Usability Engineering, 1993.

- [15] „Find out how you stack up to new industry benchmarks for mobile page speed,“ [Võrgumaterjal]. Available: <https://www.thinkwithgoogle.com/marketing-strategies/app-and-mobile/mobile-page-speed-new-industry-benchmarks/>. [Kasutatud 15 5 2023].
- [16] J. Johnson, Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines, 2010.
- [17] „Web Content Accessibility Guidelines,“ 5 6 2018. [Võrgumaterjal]. Available: <https://www.w3.org/TR/WCAG21/>. [Kasutatud 24 4 2023].
- [18] „MySQL 8.0 Reference Manual - Numeric Type Attributes,“ [Võrgumaterjal]. Available: <https://dev.mysql.com/doc/refman/8.0/en/numeric-type-attributes.html>. [Kasutatud 26 4 2023].
- [19] M. Cohn, Agile Estimating and Planning, 2005.
- [20] „Laravel - Validation,“ Laravel, [Võrgumaterjal]. Available: <https://laravel.com/docs/10.x/validation>. [Kasutatud 8 5 2023].
- [21] J. Spolsky, „Don’t Let Architecture Astronauts Scare You,“ 21 4 2001. [Võrgumaterjal]. Available: <https://www.joelonsoftware.com/2001/04/21/dont-let-architecture-astronauts-scare-you/>. [Kasutatud 8 5 2023].
- [22] M. Fowler, „Yagni,“ 26 5 2015. [Võrgumaterjal]. Available: <https://martinfowler.com/bliki/Yagni.html>. [Kasutatud 7 5 2023].
- [23] R. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, 2008.
- [24] „KISS principle,“ Wikipedia, 17 3 2023. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/KISS_principle. [Kasutatud 24 4 2023].
- [25] „Principle of least astonishment,“ Wikipedia, 19 3 2023. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Principle_of_least_astonishment. [Kasutatud 23 4 2023].
- [26] „SOLID,“ Wikipedia, 18 1 2023. [Võrgumaterjal]. Available: <https://en.wikipedia.org/wiki/SOLID>. [Kasutatud 10 5 2023].
- [27] „Introduction to cross-browser testing,“ 16 4 2023. [Võrgumaterjal]. Available: https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Cross_browser_testing/Introduction. [Kasutatud 11 5 2023].

- [28] „Get started with Bootstrap,“ [Võrgumaterjal]. Available: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>. [Kasutatud 11 5 2023].
- [29] „Flatpickr Introduction,“ [Võrgumaterjal]. Available: <https://flatpickr.js.org/>. [Kasutatud 11 5 2023].
- [30] „Can I use Date and time input types,“ [Võrgumaterjal]. Available: <https://caniuse.com/input-datetime>. [Kasutatud 11 5 2023].
- [31] „Moment.js Documentation,“ [Võrgumaterjal]. Available: <https://momentjs.com/docs/>. [Kasutatud 11 5 2023].
- [32] „Select2,“ [Võrgumaterjal]. Available: <https://select2.org/>. [Kasutatud 11 5 2023].
- [33] M. Nemberg, „Suur A/B testimise ülevaade – mida testida, kuidas testida, kuidas planeerida,“ 2 12 2015. [Võrgumaterjal]. Available: <https://blog.twn.ee/et/suur-ab-testimise-ulevaade-mida-testida-kuidas-testida-kuidas-planeerida>. [Kasutatud 14 5 2023].
- [34] „How Google Analytics works,“ Google, [Võrgumaterjal]. Available: <https://support.google.com/analytics/answer/12159447>. [Kasutatud 11 5 2023].
- [35] „WAVE Browser Extensions,“ [Võrgumaterjal]. Available: <https://wave.webaim.org/extension/>. [Kasutatud 14 5 2023].
- [36] „GitHub - page-load-time,“ [Võrgumaterjal]. Available: <https://github.com/alex-vv/page-load-time>. [Kasutatud 15 5 2023].

Lisa 1 – Lihtlitsents

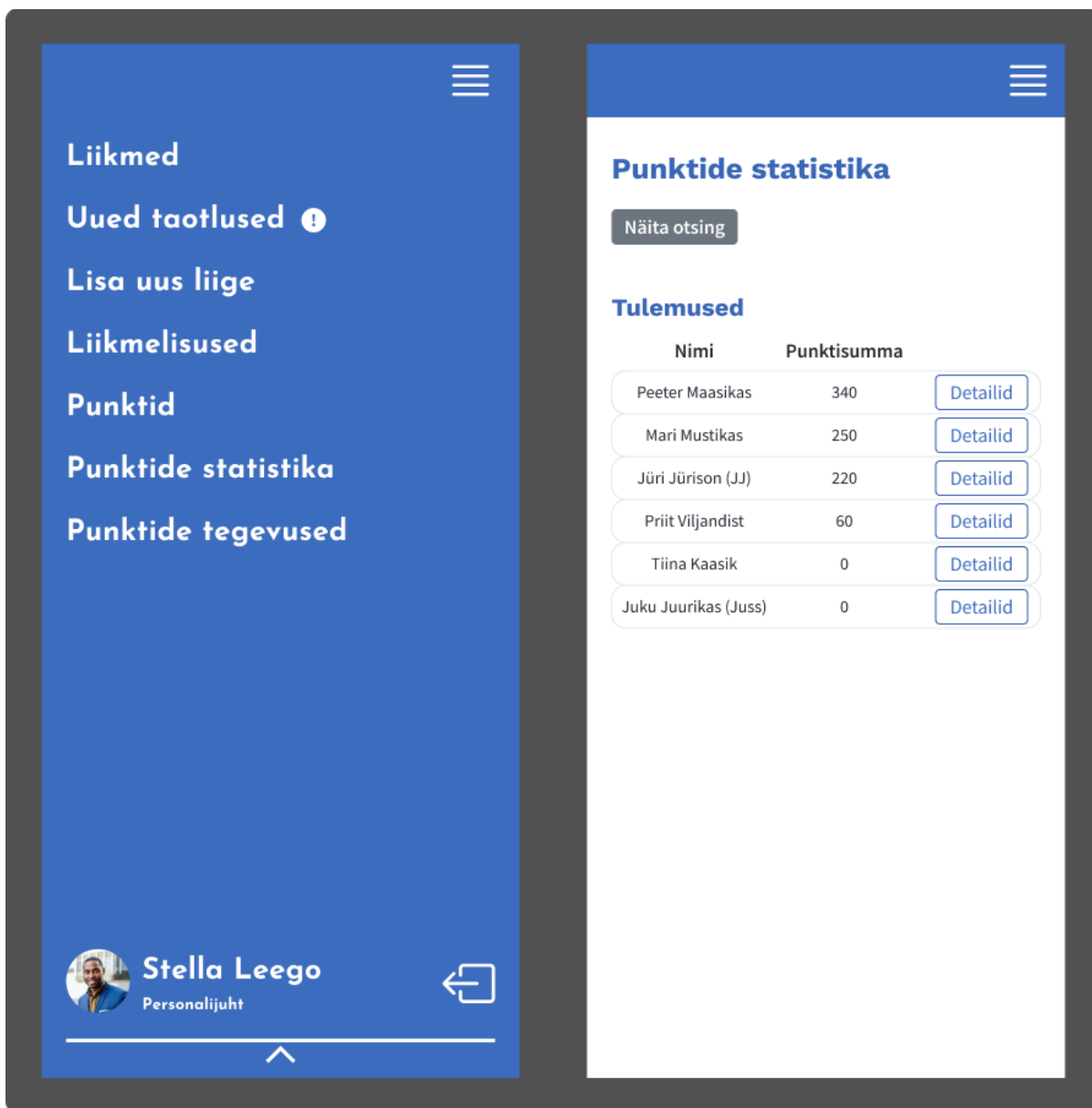
Mina, Hannes Härm

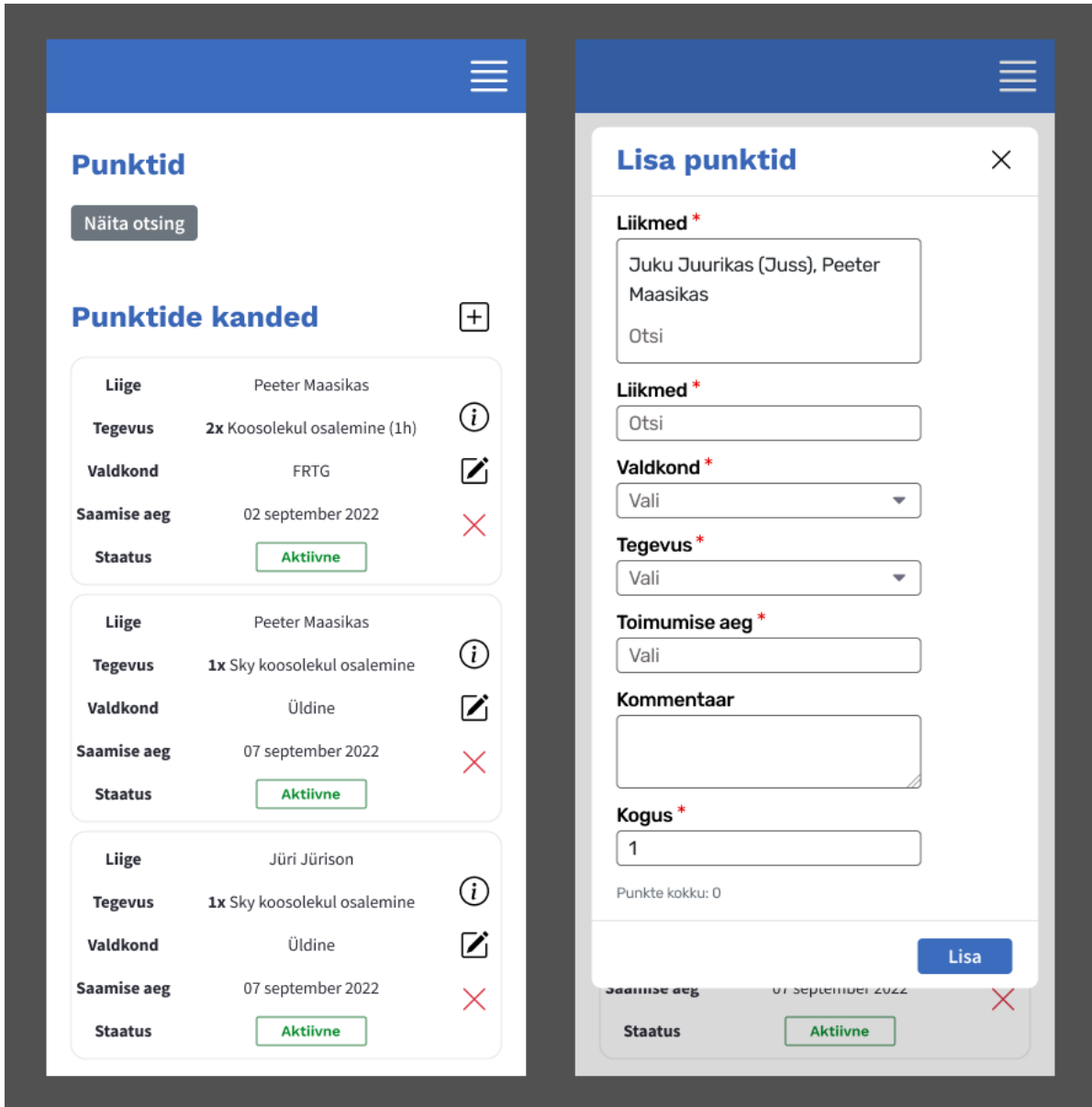
1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose Üliõpilasorganisatsiooni BEST-Estonia liikmelisuse ja punktisüsteemi arvestuse tarkvara arendamine", mille juhendaja on Erki Eessaar
 - a. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - b. 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.

Lisa 2 – Olemasolevas kasutajaliidese prototüübis kasutatavate värvide kontrastisuhted

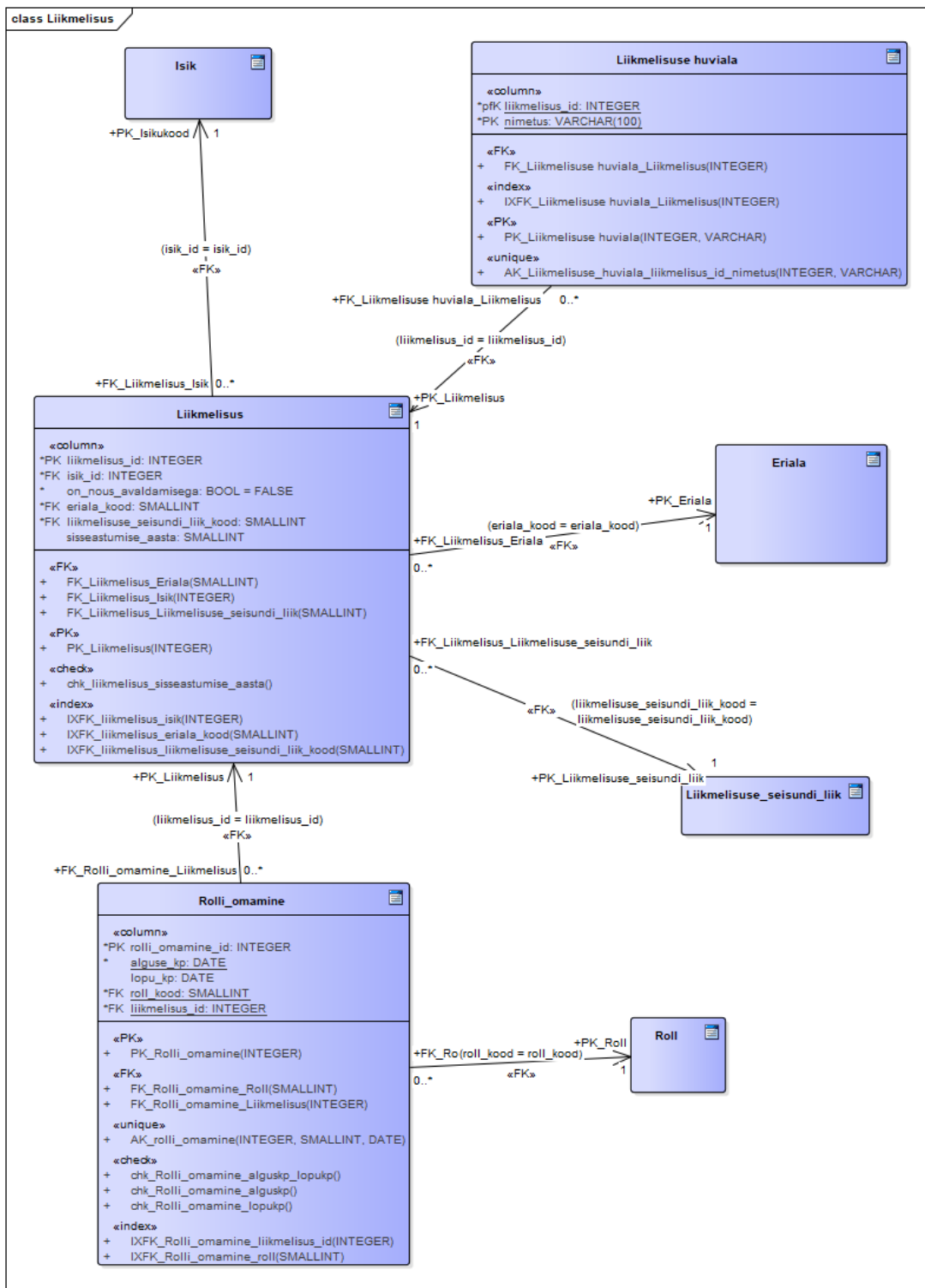
Esiplaan	Taust	Kontrasti suhe	Näide kohast, kus kasutatakse
FFFFFF	3C6CBD	5.15:1	Küljepaneeli lingid, põhinupp
98D4A9	FFFFFF	1.70:1	Roheline liikmestaatus raam
039D2F	FFFFFF	3.57:1	Roheline liikmestaatus tekst
EAD3B0	FFFFFF	1.45:1	Kollane liikmestaatus raam
C6892D	FFFFFF	2.99:1	Kollane liikmestaatus tekst
BFBFBF	FFFFFF	1.83:1	Hall liikmestaatus raam
8B8282	FFFFFF	3.74:1	Hall liikmestaatus tekst
EE9596	FFFFFF	2.24:1	Punane liikmestaatus raam
F13235	FFFFFF	4.00:1	Punane liikmestaatus tekst
808080	FFFFFF	3.94:1	Kasutamatu nupp
DC3545	FFFFFF	4.52:1	Kustuta nupp
000000	FFFFFF	21.00:1	Muuda ja lisa nupp, must tekst
FF0000	3C6CBD	1.28:1	Hüüumärgi sümbol küljepaneelil liikmeavalduste lingi juures
67C52F	FFFFFF	2.18:1	Roheline nupp
3966B6	FFFFFF	5.60:1	Pealkiri
E7E7ED	FFFFFF	1.23:1	Tabeli äärejoon
212529	FFFFFF	15.42:1	Tekstivälja sisendi tekst
CED4DA	FFFFFF	1.49:1	Tekstivälja äärejoon
E9ECEF	#CED4DA	1.26:1	Väljalülitatud tekstivälja sisu värv
212529	E9ECEF	13.00:1	Sisendtekst väljalülitatud tekstiväljal
6C757D	FFFFFF	4.68:1	Hall nupp

Lisa 3 – Mõned mobiilivaate kasutajaliidese prototüübid

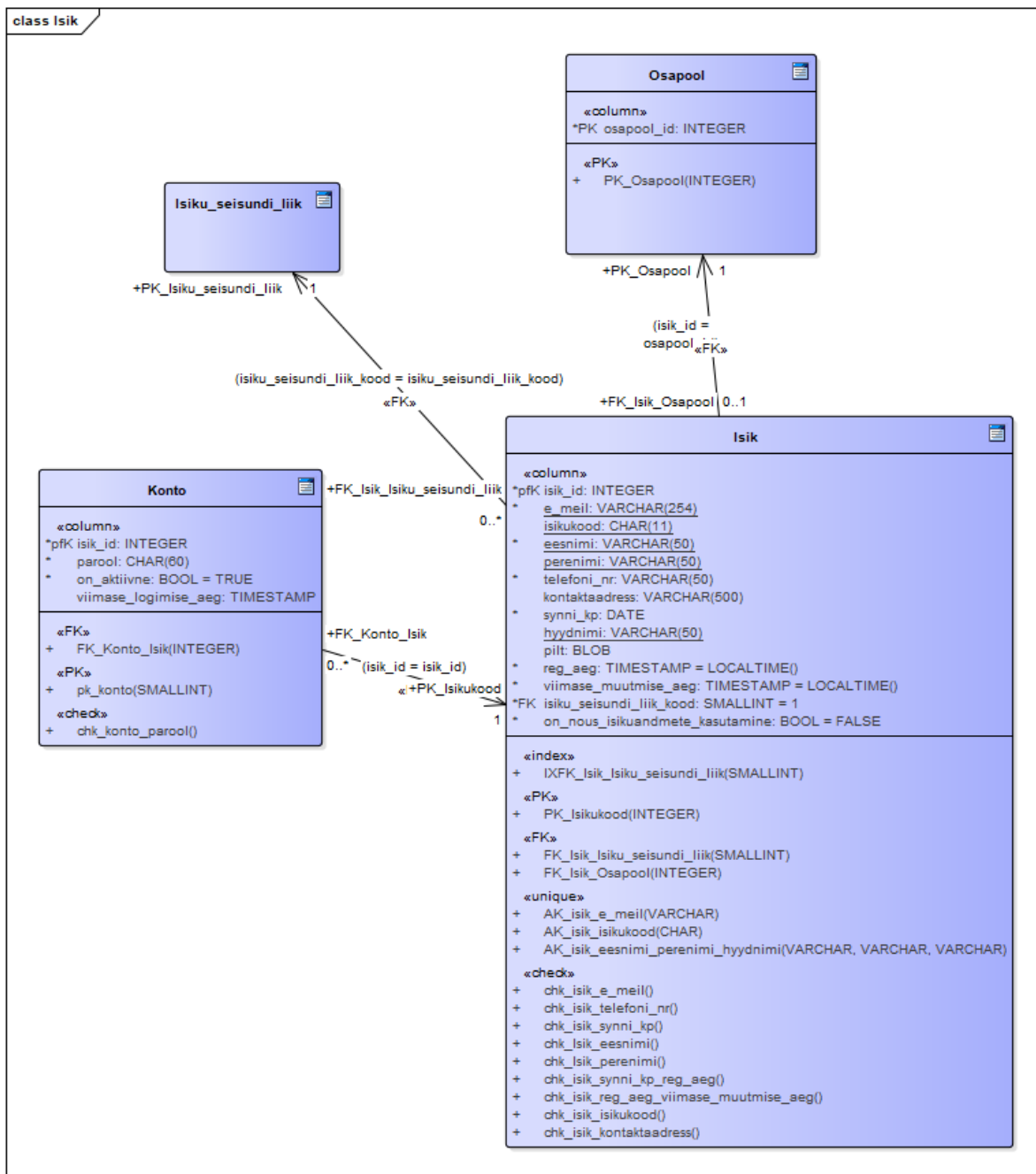




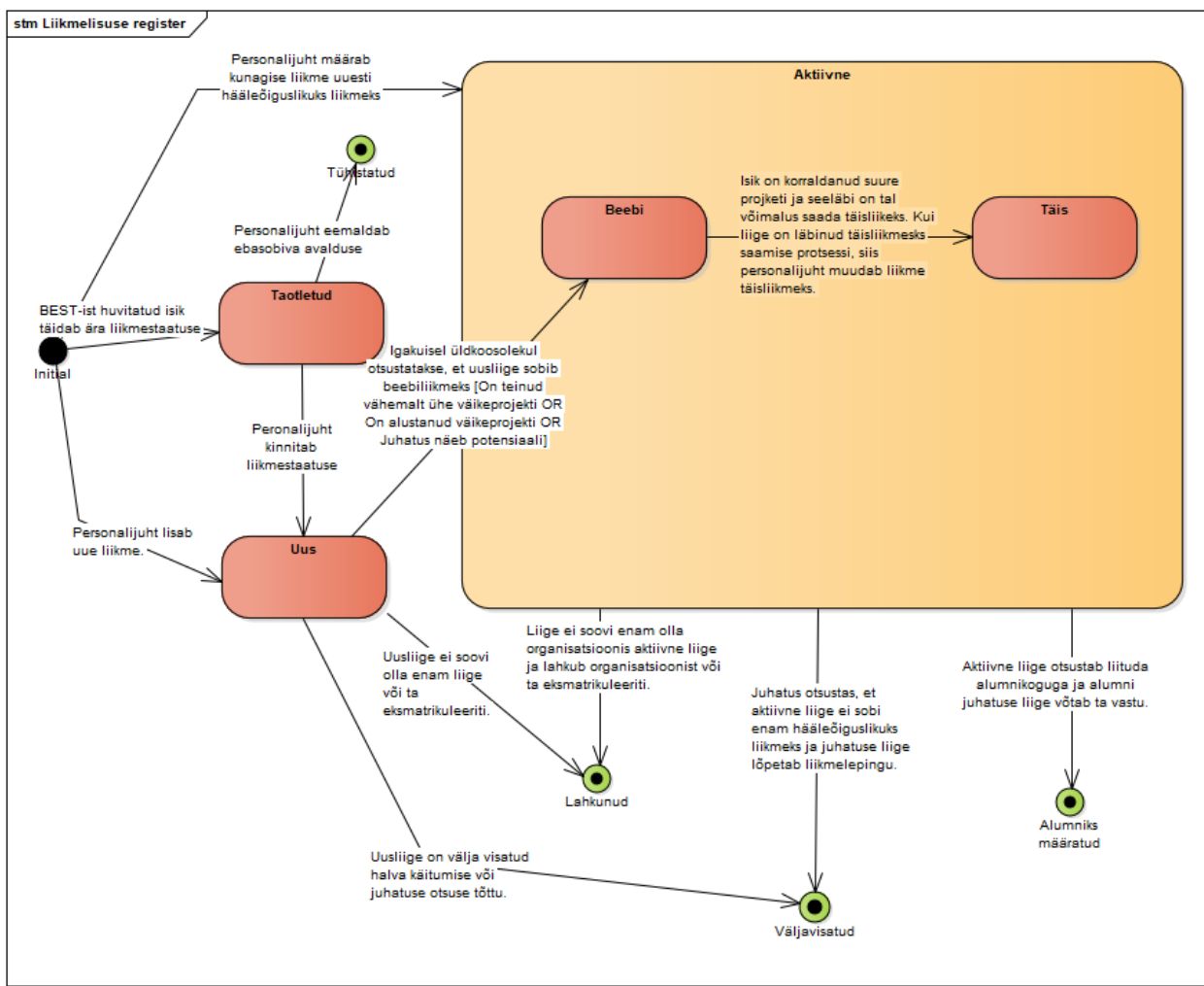
Lisa 4 – Tabelid Liikmelisus ja Liikmelisuse_huviala BEST-Estonia andmebaasi uues versioonis



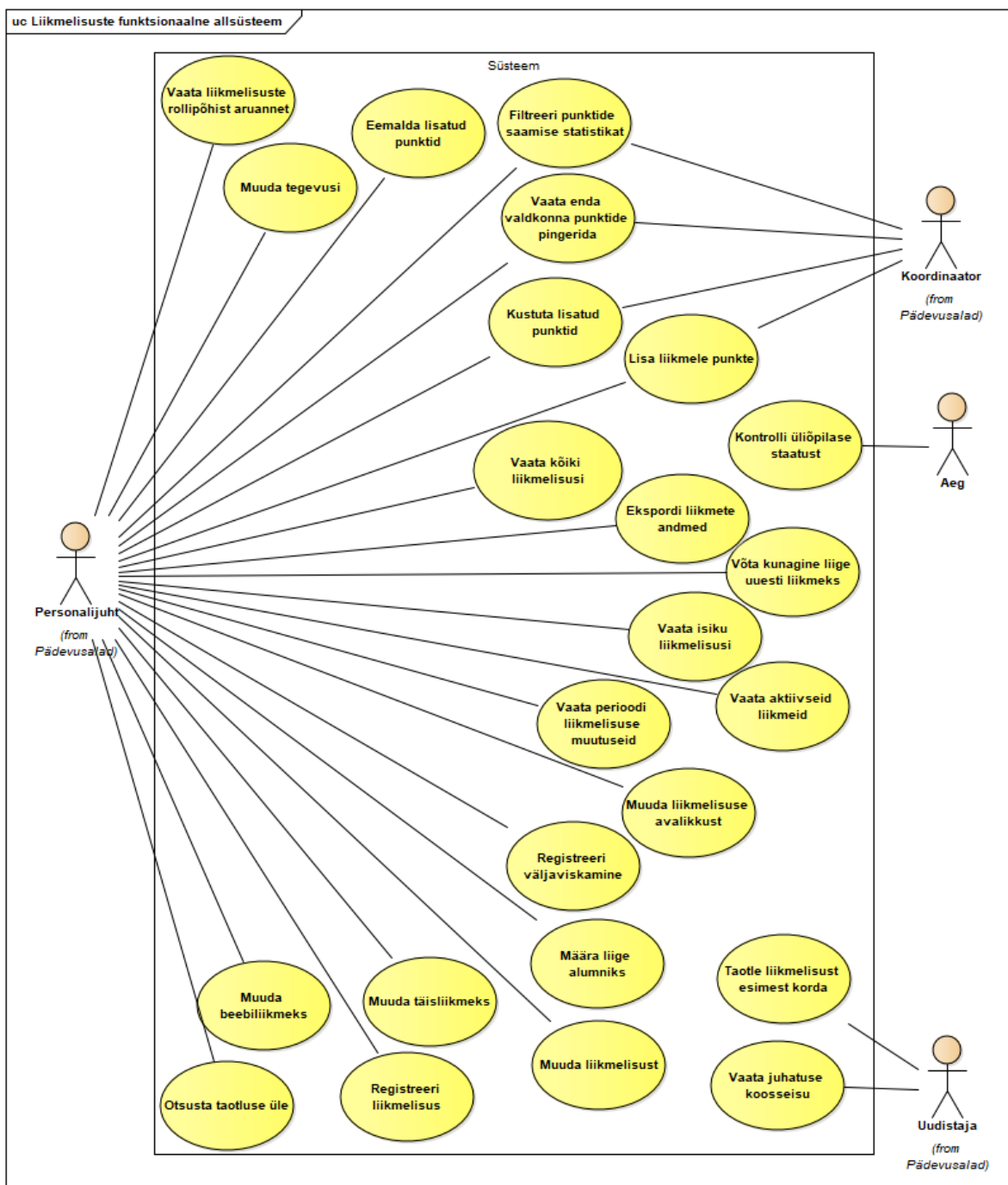
Lisa 5 – Tabelid Isik, Konto ja Osapool BEST-Estonia andmebaasi uues versioonis



Lisa 6 – Uuenenud liikmelisuse seisundidiagramm



Lisa 7 – Muudetud kasutusjuhtude diagramm



Lisa 8 – Funktsionaalsuste hindamine

ID	Kasutusmall või kasutuslugu	Prioriteet	Keerukus	Periood
1	Tuvasta kasutaja	1	8	1
2	Taotle liikmelisust esimest korda	2	3	1
4	Otsusta taotluse üle	4	3	1
5	Registreeri liikmelisus	5	5	1
6	Muuda beebiliikmeks	6	3	1
7	Muuda täisliikmeks	7	0.5	1
10	Muuda liikmelisust	8	0.5	1
11	Registreeri väljaviskamine	9	0.5	1
12	Määra liige alumniks	10	0.5	1
13	Muuda liikmelisuse avalikkust	29	1	3
14	Kontrolli üliõpilase staatust	30	13	3
15	Vaata aktiivseid liikmeid	11	3	1
16	Vaata juhatuse koosseisu	31	5	3
18	Vaata kõiki liikmelisusi	12	3	1
19	Vaata isiku liikmelisusi	13	3	1
20	Vaata liikmelisuste rollipõhist aruannet	14	1	1
21	Vaata perioodi liikmelisuse muutuseid	31	1	3
100	Personalijuhina tahan liikmele lisada ettemääratud tegevuse eest punkte, et tunnustada liiget aktiivselt organisatsiooni tööst osavõtmise eest.	15	5	2
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.	16	5	2
102	Personalijuhina tahan kustutada liikmele lisatud punkte, kui liikmele on lisatud punkte ekslikult, et punktiarvestus oleks täpne.	17	1	2

ID	Kasutusmall või kasutuslugu	Prioriteet	Keerukus	Periood
104	Personalijuhina tahan näha punktide pingerida omalt poolt määratud ajavahemikus, et premeerida aktiivseid liikmeid (näiteks kuu või aasta aktiivsemaid).	18	5	2
105	Personalijuhina tahan punktide saamise statistikat filtreerida nime ja punktisumma järgi, et lihtsustada otsingut.	19	3	2
106	Koordinaatorina tahan lisada punkte oma valdkonnaga seotud tegevuste eest, et tunnustada töögrupis aktiivseid liikmeid.	20	1	2
107	Koordinaatorina tahan enda poolt hallatavasse valdkonda lisatud punkte kustutada, kui mingi tegevuse eest on ekslikult punkte lisanud, et punktiarvestus oleks täpne.	21	1	2
108	Koordinaatorina tahan saada statistikat oma töögrupi aktiivsematest liikmetest, et premeerida neid valdkonnas aktiivse tegutsemise eest.	22	1	2
109	Koordinaatorina tahan näha oma töögrupis punkte saanud liikmete arvu, et omada ülevaadet valdkonna seisukorrast.	23	1	2
110	Punktisüsteemi tegevuste lisamine, muutmine ja kustutamine - Personalijuhina tahan lisada uusi, muuta või kustutada olemasolevaid tegevusi, mille põhjal saaksid koordinaatorid ja muud liikmed teistele liikmetele punkte lisada, et vähendada korduvaid tegevusi.	24	5	2
111	Liikmete andmestiku eksport – Personalijuhina tahan eksportida organisatsiooni liikmete nimekirja koos isikuandmetega ja punktistatistikaga, et andmeid vabas vormis töödelda ning vajadusel teiste organisatsiooni liikmetega jagada. Samal ajal tahan hoida täielikku kontrolli, millised andmed teisteni jõuavad.	25	2	2
112	Huvialade märkimine liikmeavalduses – BEST-Estoniasse liitudes soovin märkida enda huvialasid, et organisatsiooni juhtkonnal oleks kergem mind mulle huvi pakkuvatesse tegevustesse kaasata.	26	1	2

ID	Kasutusmall või kasutuslugu	Prioriteet	Keerukus	Periood
113	Huvalade nägemine isikuandmete juures – Personalijuhina soovin näha liitunud liikme isikuandmete juures, millised huvalad on liitunud liige endale märkinud, et aidata tal organisatsioonis leida erinevaid huvi pakkuvaid tegevusi. Huvalad peavad olema standardiseeritud, et oleks võimalik grupeerida erinevaid liikmeid vastavalt nende huvaladele.	27	1	2
114	Kunagise liikme uuesti liikmeks võtmine – Personalijuhina tahan kunagise liikme uuesti liikmeks võtta, luues talle uue liikmelisuse, et anda taasliitujale hääleõigus, samal ajal säilitades infot eelmise liikmelisuse kohta, näiteks eriala.	28	1	3

Lisa 9 – Käsu realisatsioon tegevuse kustutamise näitel

```
/**
 * @template-implements Command<void>
 */
readonly class DeleteActivityCommand implements Command
{
    public function __construct(
        public int|null $id
    )
    {
    }
}

class DeleteActivityValidator extends BaseValidator
{
    protected function rules(): array
    {
        return array(
            'id' => [
                'required',
                'int',
                new ActivityExistsRule($this),
                function (string $attribute, mixed $value, Closure $fail) {
                    if (PunktiSaamine::query()->where('tegevus_id', $value)->exists()) {
                        $fail(__('validation.activity.hasPointEntries'));
                    }
                }
            ]
        );
    }
}

class DeleteActivityHandler implements Handler
{
    function getRequiredPermission(): Permission|null
    {
        return Permission::ManageActivities;
    }

    /**
     * @param DeleteActivityCommand $message
     * @return void
     * @throws NotFoundException
     */
    function invoke(MediatorMessage $message): void
    {
        $activity = Tegevus::query()->find($message->id);
        if ($activity == null)
            throw new NotFoundException();
        $activity->delete();

        Log::info('Deleted activity', ['activity' => $message->id]);
    }
}
```

Lisa 10 – Enumeration klassi definitsioon

```
abstract class Enumeration
{
    public function __construct(
        public int $code,
        public string $name
    )
    {
    }

    public function __toString(): string
    {
        return $this->name;
    }

    /**
     * @return array<static>
     */
    public static function getAll(): array
    {
        $reflection = new ReflectionClass(static::class);
        $res = array();
        /** @var Enumeration $value */
        foreach ($reflection->getStaticProperties() as $value) {
            $res[] = $value;
        }

        return $res;
    }

    public static function getByKey(int $key): static|null
    {
        $reflection = new ReflectionClass(static::class);

        /** @var Enumeration $value */
        foreach ($reflection->getStaticProperties() as $value) {
            if ($key == $value->code) {
                return $value;
            }
        }
        return null;
    }

    public abstract static function initValues();
}
```

Lisa 11 – Mõned kuvatõmmised valminud rakenduse vaadetest

Hannes Härm
Personalijuht, Liige

- Liikmed
- Uued taotlused !
- Lisa uus liige
- Liikmelisuste ajalugu
- Punktid
- Punktide statistika
- Tegevused

BEST-Estonia liikmed

Otsi

Nimi Liikmestaatus Eriala

Sünnikuu

Otsi

Liikmete nimekiri ↓

1 - 30 kuvatud 182-st < 1 2 3 ... 7 >


Nimi	Liikmestaatus	Sünnipäev	Eriala
Adeele Orav	Beeblliige	25 juuni 1997	Elektroenergeetika ja mehhatroonika
Adeele Rõõm	Beeblliige	18 oktoober 1994	Rakenduslik majandusteadus
Alma Palm	Beeblliige	04 juuni 1995	Küberturbe tehnoloogiad
Alma Sammal	Taisliige	24 veebruar 2001	Rahvusvaheline ärikorraldus
Bert Nugis	Beeblliige	26 veebruar 1998	Küberturbe tehnoloogiad
Birgit Leemets	Uusliige	28 mai 1998	IT-Süsteemide arendus

Hannes Härm
Personalijuht, Liige

- Liikmed
- Uued taotlused !
- Lisa uus liige
- Liikmelisuste ajalugu
- Punktid
- Punktide statistika
- Tegevused

Uued liikmelisuse taotlused

Nimi	Avalduse aeg
Roland Pärtel	08 mai 2023 Vaata
Eerik Rõõm	08 mai 2023 Vaata
Peeter Raudsepp	12 mai 2023 Vaata
Alo Nõmme	13 mai 2023 Vaata
Ivar Rosen	13 mai 2023 Vaata
Mart Vaher	13 mai 2023 Vaata
Sirje Laanemets	14 mai 2023 Vaata



Hannes Härm
Personalijuht, Liige

Liikmed

Uued taotlused 1

Lisa uus liige

Liikmelisuste ajalugu

Punktid

Punktide statistika

Tegevused



Punktide statistika

Otsi

Nimi Perioodi algus Perioodi lõpp

valdkond

Otsi

Tulemused

1 - 30 kuvatud 206-st

< 1 2 3 ... 7 >

Liige	Punktisumma	
Kadri Lõbu	13980	Detailid
Madis Rosen	9060	Detailid
Matis Kriisa	9055	Detailid
Birgit Leemets	8890	Detailid
Nelli Mets	8005	Detailid
Imbi Laats	7490	Detailid

Hannes Härm
Personalijuht, Liige

Liikmed

Uued taotlused 1

Lisa uus liige

Liikmelisuste ajalugu

Punktid

Punktide statistika

Tegevused



Punktide tegevused

Nimekiri

Tegevuse nimetus	Punktid	Valdkonnad, kus valitav	
Firmakohtumine	30	FRTG, Tiimitöö	✍ ✕
Juhatuselise liige	1200	Üldine	✍ ✕
Koolituselise osalemine (1h)	20	Üldine	✍ ✕
Koordinaator	500	Üldine	✍ ✕
Koosolekul osalemine (1h)	20	DMTG, FRTG, MTG, Projektid, RV, TTG, Tiimitöö, Üldine	✍ ✕
Mentor	500	Üldine	✍ ✕
Protomine	10	DMTG, FRTG, MTG, Projektid, RV, TTG, Tiimitöö, Üldine	✍ ✕
SKY koosolek	60	Üldine	✍ ✕
Telefonikõne	15	FRTG, Tiimitöö	✍ ✕