

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Sander Puntso  
142753IAPB

**ELEKTROONILISE HUVIKOOLIDE  
PÄEVIKU VEEBITEENUSE ANALÜÜS JA  
PROTOTÜÜP**

Bakalaureusetöö

Juhendaja: Tarvo Treier  
MSc

Tallinn 2017

## **Autorideklaratsioon**

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

Autor: Sander Puntso

22.05.2017

## **Annotatsioon**

Käesoleva lõputöö põhieesmärgiks on analüüsida huvihariduse pakkujate elektroonilise päevikupidamise vajadusi ning luua prototüüpveebiteenus. Töös antakse ülevaade veebiteenuste üldistest tehnoloogiatest ja põhimõtetest ning sõnastatakse tutvustatud tehnoloogiate ja konkurentsianalüüsi tulemuste põhjal loodava prototüüpteenuse eesmärgid. Püstitatud eesmärkide ja esitatud huvikoolide infovajaduste analüüsi põhjal viiakse läbi veebiteenuse analüüs, kus püstitatakse realiseeritavale prototüübile nõuded. Analüüsi põhjal realiseeritakse veebiteenuse prototüüp, mis täidab kõik esitatud funktsionaalsed nõuded.

Realiseeritud veebiteenuse kõikide operatsioonide kohta luuakse põhjalik dokumentatsioon, mis annab ülevaate operatsioonide võimalustest ning toimib veebiteenuse kasutusjuhendina. Veebiteenuse realiseeritud operatsioonide funktsionaalsuse testimiseks luuakse ressursipõhised testid ja testkollektiivid. Valminud veebiteenusega saavutatud tulemusi hinnatakse ning analüüsitakse püstitatud eesmärkide saavutamist.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 39 leheküljel, 6 peatükki, 16 joonist.

## **Abstract**

### **The analysis and prototype of a web service for an electronic journal for hobby education**

The aim of this thesis is to analyze the need for electronic journal for hobby education providers in Estonia and to develop a hobby education journal software prototype as a web service in order to provide a solution to hobby education providers' communication needs.

The constructed analysis of hobby education providers' communicational practices and competitive analysis of software providing assistive tools to manage hobby education communication needs are the basis of the analysis of the web service prototype. The aforementioned web service analysis provides the functional demands of the web service.

This thesis provides an introduction to web service development by explaining main protocols and tools for developing diverse web service to provide functionality for wide range of clients. Based on introduced technologies and analysis of hobby education electronic journal functional demands, this thesis provides realisation of a software prototype as a web service with full documentation and functionality tests. At the end of the thesis an analysis of results is formulated to evaluate the fulfillment of goals.

The thesis is in estonian and contains 39 pages of text, 6 chapters, 16 figures.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> ehk rakendustarkvara liides [1]
EHIS	Eesti Hariduse Infosüsteem
EMTA	Eesti Maksu- ja Tolliamet
JSON	<i>Javascript Object Notation</i> ehk Javascripti objekti esitlusviis
PAI	Pearaha Arvestamise Infosüsteem
REST	<i>Representational State Transfer</i> ehk kujutava olekuedastuse protokoll
SaaS	<i>Software as a Service</i> ehk tarkvara teenusena
SOAP	<i>Simple Object Access Protocol</i> ehk lihtne objekti juurdepääsu protokoll [1]
token	Ligipääsuvõti päringu tegija autoriseerimiseks
URL	<i>Uniform Resource Locator</i> ehk universaalne ressursi lokaator [1]
WSDL	<i>Web services description language</i> ehk veebilehtede teenuste kirjelduskeel [1]
XML	<i>Extensible markup language</i> ehk laiendatav märgistuskeel [1]

## Sisukord

1 Sissejuhatus .....	9
1.1 Taust ja probleem .....	9
1.2 Ülesande püstitus .....	10
1.3 Metoodika.....	11
1.4 Ülevaade tööst .....	11
2 Veebiteenused ja tehnoloogiad.....	13
2.1 Veebiteenused.....	13
2.2 WSDL.....	15
2.3 SOAP ja REST .....	16
2.4 SaaS .....	16
2.5 Swagger .....	18
2.6 Postman .....	19
3 Info kogumine ja konkurentsianalüüs .....	20
3.1 Huvikoolide pävikupidamise hetkeseis .....	20
3.2 Vajadused .....	21
3.3 Sihtrühmad.....	22
3.4 Konkurentsianalüüs .....	23
3.4.1 VEMI.....	24
3.4.2 Sportlyzer .....	25
3.4.3 Excel.....	26
3.4.4 eKool ja Studium.....	27
3.5 Konkurentsieelised .....	28
4 Analüüs.....	30
4.1 Veebiteenuse analüüs .....	30
4.1.1 Funktsionaalsed nõuded .....	30
4.1.2 Mittefunktsionaalsed nõuded.....	31
4.1.3 Andmeressursside kirjeldus.....	32
4.1.4 Pävikuteenuse huvigrupiga seotud kasutusjuhud .....	35
4.1.5 Pävikuteenuse ühe põhiprotsessi tegevusdiagramm .....	36

4.1.6 Veebiteenuse turvalisus .....	37
5 Realisatsioon.....	38
5.1 Veebiteenuse realisatsioon .....	38
5.1.1 InstitutionService WSDL .....	39
5.1.2 Operatsioonide realisatsioon .....	41
5.2 Veebiteenuse dokumentatsioon ja testimine.....	43
6 Tulemuste analüüs .....	46
6.1 Realiseeritud eesmärgid.....	46
6.2 Elektroonilise päevikuteenuse tulevik .....	47
Kokkuvõte .....	48

## Jooniste loetelu

Joonis 1. Teenusepõhise arhitektuuri mudel [4].....	14
Joonis 2. WSDL põhiosade suhted [6] .....	15
Joonis 3. Pävikuteenuse klassidiagramm.....	34
Joonis 4. Pävikuteenuse huvigrupi kasutusjuhtude diagramm .....	35
Joonis 5. Huvigrupi kustutamise tegevusdiagramm .....	36
Joonis 6. WSDL kirjeldusfaili grupi globaalne tüüp .....	39
Joonis 7. WSDL kirjeldusfaili grupi globaalne tagastustüüp .....	39
Joonis 8. WSDL kirjeldusfaili grupi lisamise päringu ja vastuse elemendid .....	40
Joonis 9. WSDL kirjeldusfaili grupi lisamise sõnumielemendid .....	41
Joonis 10. WSDL kirjeldusfaili grupi lisamise operatsioon porditüübis.....	41
Joonis 11. WSDL kirjeldusfaili grupi lisamise operatsioon sidumise elemendis .....	41
Joonis 12. Swagger dokumentatsiooni kolm andmeressurssi.....	43
Joonis 13. Swagger grupi alamressursside dokumentatsioon.....	44
Joonis 14. Postman keskkonnamuutujate salvestamine .....	45
Joonis 15. Postman päringu tagastusväärtuse kontrolltest.....	45
Joonis 16. Postman kollektsioonitestide tulemuste raport.....	45



# 1 Sissejuhatus

Tänapäeva ühiskonnas on liikumas aina suurem osa kommunikatsiooniprotsesse internetti, mis toob endaga kaasa nõudmised keerukamaid infoprotsesse võimaldavate tehnoloogiate ja rakenduste järele. Isegi võimekamad tarkvarapakkujad ei suuda sageli pakkuda lahendust erinevate kasutajate nõudmistele, mis ühest küljest suunab tarkvaraarendust kiiremini evolutsioneeruma, teisest küljest piirab aga inimeste infovahetust käesoleval ajahetkel. Pakkumaks keerukamate süsteemide väljatöötamiseks vajalikke ehituskomponente on veebiteenused kaasaegse tarkvaraarenduse üheks olulisemaks aluseks.

## 1.1 Taust ja probleem

Huvihariduse pakkujaid on Eestis ja maailmas väga palju ning vaatamata pakutavate teenuste erinevustele on korralduslikud tööprotsessid enamikes huvikoolides samad. Huvikoolide eesmärk on edendada mingi kindla valdkonna spordi- või huvitegevust ja pakkuda nii noortele kui ka täiskasvanutele eesmärgipärast vaba aja sisustamist. Noorte huvitegevuse tellija vormis on reeglina lapsevanemad, kes soovivad oma laste arengu ja käitumise kohta regulaarset tagasisidet ning ülevaadet teenust pakkuva huvikooli tegemistest. Nii on kujunenud kõikides huvikoolides kindlad harjumused, kuidas ja mis kanaleid pidi pakkuda nii potentsiaalsetele uutele kui ka olemasolevatele klientidele ülevaadet ja tagasisidet oma pakutavast teenusest.

Käesoleva töö autor puutub töö kirjutamise ajal huviharidusega kokku igapäevaselt spordikooli noortetreenerina. Isikliku kogemuse põhjal teenusepakkuja rollis ja varasemalt kliendi rollis on autor puutunud kokku kehvasti korraldatud kommunikatsiooni tõttu negatiivsete mõjudega huvihariduse kvaliteedile. Autori hinnangul on huvihariduse pakkujatele ja klientidele kõige suuremaks kommunikatsiooniprobleemiks muutuste kohta kõige värskema info kohaletoometamine kõikidele vajalikele osapooltele. Huvikoolide ja klientide seisukohalt on oluline võimaldada kõikidele osapooltele alati ligipääsu kõige värskemale korralduslikule infole

või kontaktidele, et vältida info edastamisel topelttööd või teabe puudujääke. Kontaktandmete muutumisest tingitud eksimused ja inimlikud vead infoedastusel võivad lõppeda huvihariduse pakkuja poolt kvaliteedilangusega, sest tegelema peab korralduslike probleemidega, või klientide poolt rahulolematusega kui huvikool ei taga kliendile vajalikku teavet.

Kuna aina suurem osa kommunikatsioonist professionaalsel tasemel toimub interneti vahendusel, siis on huvikoolidel tekkinud üldhariduskoolide elektroonilise päevikupidamise eeskujul vajadus elektroonilise päevikupidamise ehk universaalse tööriista järgi, mis pakuks võimalust kliendil ja teenusepakkujal mugavalt üksteisega ühendust hoida ja vastastikku vahetada üksteisele olulist infot. Käesolev töö püüab analüüsida antud hetkel enamkasutatavaid lahendusi huvikoolide tegevuse kohta info talletamisel ja õppetöö päeviku pidamisel ning üritab pakkuda veebiteenusena lahenduse, mis rahuldaks üldiseid huvikoolide infoprotsessidest tingitud vajadusi ning looks liidestamisvõimalusega eeldused mitmekülgselt kasutamiseks.

## **1.2 Ülesande püstitus**

Bakalaureusetöö põhieesmärk on analüüsida huvihariduse kommunikatsiooni- ja päevikuprotsesside hetkeseisu ja vajadusi ning analüüsida ja realiseerida kogutud info põhjal elektroonilise päeviku veebiteenuse prototüüp.

Töö põhieesmärgi täitmiseks püstitatud alameesmärgid on sõnastatud järgmiselt:

1. Anda ülevaade veebiteenuste programmeerimise põhitehnoloogiatest ning üldistest praktikatest
2. Anda ülevaade ning analüüsida huvitegevuse päevikupidamise vajadusi ja hetkeolukorda
3. Analüüsida konkureerivate teenusepakkujate võimalusi huvikoolide päevikupidamise vajadusi arvestades ning hinnata, miks üks või teine lahendus ei ole kujunenud ainsaks eelistatud võimaluseks
4. Viia läbi huvikoolide infovajadusi arvesse võttes ning konkurentsianalüüsile tuginev elektroonilise päevikuteenuse analüüs

5. Realiseerida, testida ja dokumenteerida elektroonilise pävikupidamise funktsionaalust võimaldav veebiteenuse prototüüp

### **1.3 Metoodika**

Bakalaureusetöö valmib järgmistes etappides:

- Veebiteenuste arendamistehnoloogiate üldiste põhimõtete uurimine
- Huvikooli tegevusest lähtuva elektroonilise pävikupidamise hetkeolukorra analüüsimine ning huvikoolide pävikupidamise vajaduste sõnastamine
- Elektroonilise päeviku funktsionaalsust võimaldavate teenusepakkujate konkurentsianalüüs avalikult kättesaadava teabe ja huvikoolide vajaduste põhjal
- Konkurentsianalüüsi alusel huvikooli vajadusi rahuldava ja konkurentsieelist pakkuva tarkvaralahenduse sõnastamine
- Analüüsi põhjal kirjapandud nõuetele vastava teenuse loomine rakendades uuritud praktikaid
- Loodud veebiteenuse funktsionaalsuse testimine ja dokumenteerimine ning soovitud tulemuse saavutamise hindamine
- Lõpptulemuste formuleerimine

### **1.4 Ülevaade tööst**

Veebiteenustest ja teenuse loomisel kasutatavatest tehnoloogiatest annab ülevaate teine peatükk, milles tutvustatud tehnoloogiaid kasutades realiseeritakse antud töös analüüsitud veebiteenus. Töö kolmas peatükk kirjeldab huvihariduse pakkujate pävikupidamise infovajadused tuginedes avalikule kättesaadavale infole ja autori isiklikule kogemusele, mille põhjal veebiteenuse analüüsis sõnastatakse veebiteenuse funktsionaalsed nõuded. Lisaks viiakse komandas peatükis läbi konkurentsianalüüs, mis hindab esitatud vajaduste täitmist olemasolevate konkureerivate tarkvaralahenduste poolt ning analüüsib, miks üks või teine lahendus ei ole kinnistunud ainsaks eelistatud lahenduseks, produtseerides tarkvaralahenduse idee, mis rahuldaks huvikoolide infovajadusi ning kõrvaldaks

konkretentsipuudused. Neljandas peatükis tehakse huvikoolide vajaduste põhjal antud töö käigus valmiva veebiteenuse analüüs ning püstitatakse valmivale veebiteenusele funktsionaalsed nõuded. Viies peatükk kajastab veebiteenuse realiseerimist koos teenuse dokumentatsiooni loomisega ja testimist. Töö kuendas peatükis viiakse läbi tulemuste analüüs ning soovitud eesmärkide täitmise hindamine.

## 2 Veebiteenused ja tehnoloogiad

Jaroslav Tulah kirjutab raamatus „Practical API Design“ teadmatuse kontseptist (inglise keeles *concept of cluelessness*), mis programmeerimise seisukohalt kujutab endast uue tarkvara loomisel võimalikult palju olemasolevate tarkvaralahenduste ja raamistike kasutamist, et programmeerimistiimid saaksid keskenduda eelkõige äri loogika realiseerimisele. Sellist tarkvaraarendust võrreldakse ehitamisega, kus olemasolevat kasutades luuakse terviklahendus. Seejuures on tarkvaraarenduse eeliseks võimalus kasutada kõiki ehitusklotse nii palju kordi kui soovi on, ilma et materjal otsa lõppeks [2].

Veebiteenused on üks teadmatu programmeerimise alusväärtustest. Veebiteenuse näol on programmeerimismeeskondadel võimalik toota väiksema vaevaga keerukama funktsionaalsusega süsteeme, kasutades selleks juba arendatud ja testitud teenuseid. Käesolev peatükk tutvustab veebiteenuste arendamise põhimõtteid ja annab ülevaate veebiteenuste üldistest põhimõtetest ja tehnoloogiast.

### 2.1 Veebiteenused

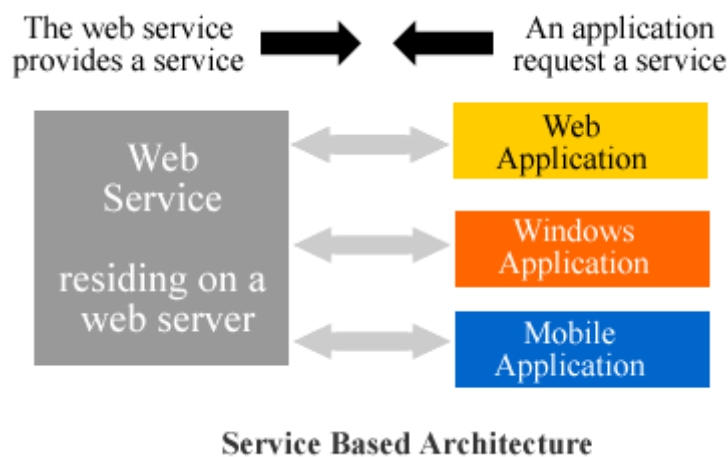
Veebiteenused on termin, mis iseloomustab standardiseeritud võimalust panna rakendused omavahel suhtlema. Teenused võimaldavad tarkvarasüsteemidel omavahel suhelda ja andmeid vahetada omamata vastastikuste süsteemide töötamise kohta mingisuguseid teadmisi. Erinevalt tavalistest klient-server süsteemidest ei paku veebiteenus graafilist kasutajaliidest, vaid jagavad äri loogikat, andmeid ja andmeprotsesse läbi programmeeriliste liidest. Tarkvaraarendajad saavad lisada veebiteenused arendatavatesse süsteemidesse ja kasutajaliidestesse, et pakkuda teenuse funktsionaalsust kasutajatele [3].

Veebiteenuste definitsioon ja peamised omadused võimaldab kirjeldada ilmekalt põhjused, miks ja millal tuleks tavalise programmi asemel luua veebiteenus. Siinkohal ei ole piir üheselt selge ning sõltub paljuski ka arendusettevõtete ja -tiimide võimekusest ja soovist. Kuna iga äri line probleem on lahendatav, tihti seda ka tehakse, veebiteenust kasutamata, ei ole tänasel päeval veebiteenuse kasutamine ilmtingimata möödapääsematu

lahendus. Küll aga on paljudes olukordades veebiteenused võimalikest meetoditest parim ja kõige otstarbekam.

Üks peamisi argumente veebiteenuse kasuks otsustamisel on veebiteenuste kasutajate vajadused ja vahendid. Kui ärilise probleemi puhul on enne programmi loomist teada, et probleemi lahendamiseks on vaja võimaldada erinevate programmeerimiskeelte tugega süsteemide vahel andmevahetusprotsesse, on veebiteenuse tugi kirjeldatavale protsessile potentsiaalselt parim lahendus. Sellisel lahendamisel saab võimaldada andmevahetusprotsesse nii võrgu-, töölaua- kui ka mobiilirakenduste vahel sõltumata toetatud programmeerimiskeeltest senikaua kuni kõik nimetatud rakendused omavad võrguühendust ja seeläbi ühendust konkreetse veebiteenusega [4].

Veebiteenuste puhul on oluline omadus, et teenus on täielikult eraldi teenust kasutava rakenduse ärioloogikast. Ainsaks teenuse kasutamise tingimuseks on teenusega ühendumise ja teenuse operatsioonide käivitamise võimaldamine API-de (inglise keeles *Application programming interface*) ehk rakendustarkvara liideste [1] kaudu. Sellest tulenevalt saab välja tuua ka teise peamise argumendina veebiteenuse kasuks eesmärgi võimaldada ja parandada erinevatel programmeerimiskeeltele baseeruvate rakenduste omavahelist infovahetust [4]. Erinevate rakenduste ja veebiteenuste vahelist seost iseloomustab Joonis 1.



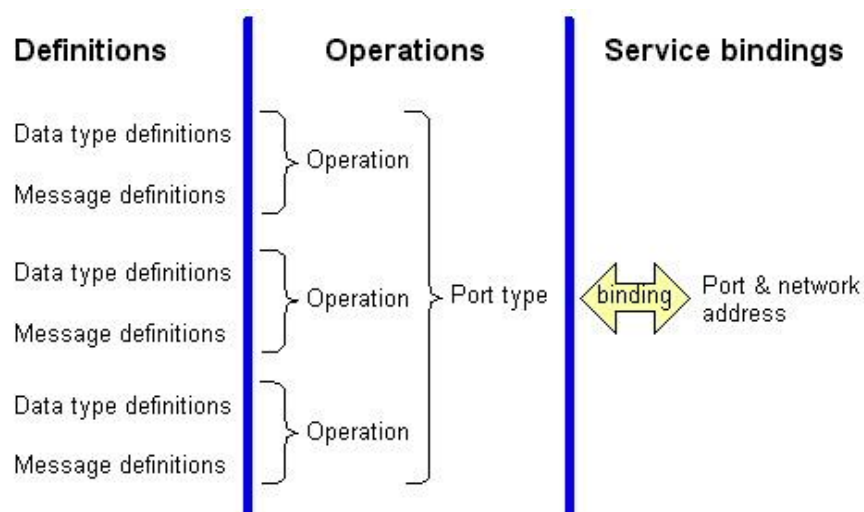
Joonis 1. Teenusepõhise arhitektuuri mudel [4]

Veebiteenus ei ole defineeritud ainult teenuse funktsionaalsuse poolt, vaid pakub tervikliku lahendusena lisaks funktsionaalsusele selget kirjeldust, kuidas teenust kasutada ja mida teenus päringute tulemusena pakub. Kasutajatest ja operatsioonidest sõltumatult

on oluline omadus ja sellest tulenevalt tugev argument teenuse kasuks otsustamisel teenuse kaudu vahetatavate andmete omandiõigus. Teenust kasutav klient või organisatsioon omab teenuse kaudu võimalust ja õigust oma andmetega ringi käia vastavalt oma soovile. Info omanikul on õigus veebiteenuse funktsionaalsuse kaudu omal soovil liidestada talletatavat infot avalikult mõnes klientkeskkonnas või töödelda ja hoida privaatset ning kasutada ainult enda jaoks toetatud ja usaldusväärsetes keskkondades või rakendustes [5].

## 2.2 WSDL

WSDL (inglise keeles *Web services description language*) on XML (inglise keeles *Extensible markup language*) ehk laiendatava märgistuskeele formaadis keel veebilehtede teenuste kirjeldamiseks [1]. WSDL kirjeldab veebiteenuse liidest ehk pakutavaid operatsioone, vahetatavate sõnumite struktuuri ja andmetüüpe. WSDL koosneb definitsioonidest, operatsioonidest ja teenuse sidumistest (inglise keeles *binding*). Definitsioonid on üldiselt esitatud XML-ina ja need sisaldavad andmetüüpide ja sõnumite kirjeldustest ning nendes kasutatavatest andmetüüpidest. Operatsioonid kirjeldavad vahetatavate sõnumite poolt toetatud tegevused. Operatsioonid on grupeeritud porditüüpideks (inglise keeles *port types*), mis koondavad mingi veebiteenuse poolt pakutavad operatsioonid. Sidumised ühendavad porditüübid ühte porti ja portide kollektsioon defineerib teenuse. WSDL põhiosade suhteid kirjeldab järgmine joonis [6].



Joonis 2. WSDL põhiosade suhted [6]

## 2.3 SOAP ja REST

SOAP (inglise keeles *Simple Object Access Protocol*) ehk lihtne objekti juurdepääsu protokoll [1] ja REST (inglise keeles *Representational State Transfer*) ehk kujutava olekuedastuse protokoll on kaks veebiteenuste suhtlusprotokolli käsitlemaks kuidas pääseda ligi ja suhelda veebiteenusega. Selleks, et veebiteenuste põhilised praktikad oleksid püsivad, peab olema sõnumivahetus kindlate protokollidega struktureeritud [7].

SOAP on XML formaadil baseeruv protokoll pakkumaks sõnumivahetusteenuseid. SOAP põhimõtte tugineb WSDL kirjelduskeeles defineeritud teenuse struktuurile ja sõnumites vahetatavatele andmetüüpidele. Sõltuvalt teenuse realiseerimiseks kasutatavatest programmeerimiskeeltest on SOAP tugevuseks võimalus WSDL dokumendi põhjal automatiseerida suur osa teenuse realiseerimisest. Seejuures on ka SOAP suureks plussiks sisseehitatud vigadehaldus, mis vigase päringu korral tagastab veainfo, et teenuse kasutajal ehk päringu autoril on võimalus oma päringut parandada [7].

SOAP aluseks olev XML sõnumistruktuur on üks peamisi põhjuseid REST protokolliga väljakujunemisele. Kuna ka kõige väiksemate päringute jaoks tuleb SOAP teenuse puhul iga kord luua XML struktuuriga sõnum, tekkis vajadus lihtsama teenuse poole pöördumise protokollile järele. REST on kergestruktuuriline alternatiiv, mis XML struktuuri kasutamise asemel võimaldab URL-i (inglise keeles *Uniform Resource Locator*) ehk universaalse ressursilokaatori [1] kaudu pöörduda veebiteenuse poole. Kergemate operatsioonide puhul on kogu päring võimalik teostada URL-iga, keerukamate operatsioonide korral on vaja lisada informatsiooni etteantud struktuuriga. Reeglina järgib REST siiski URL teel edastatavate päringute põhimõtet. Erinevalt SOAP protokollist, REST ei pea lisainformatsiooni edastamisel kasutama XML struktuuri, vaid võib ka võimaldada teistes andmeedastusformaatides andmeedastust, näiteks JSON (inglise keeles *Javascript Object Notation*) ehk Javascripti objektipõhise andmeedastusvormingu formaadis [7].

## 2.4 SaaS

Tarkvara teenusena ehk SaaS (inglise keeles *Software as a Service*) on tarkvara jaotusmudel, kus tarkvara autor või jagaja tagab tarkvara kasutajatele pakutavat rakendust üle interneti teenusena. SaaS on pilvetechnoloogiate kategooria, mis kaotab



organisatsioonidel ja tarkvarakasutajatel vajaduse paigaldada lokaalsesse seadmesse tarkvara pakkudes tarkvara funktsionaalsusele internetipõhist ligipääsu. Nii vähendab SaaS klientide jaoks riistvaranõudmisi ja -kulutusi [8]. Tarkvara teenusena ei erine lokaalsest tarkvaraprogrammist võimaldatava funktsionaalsuse poolest, kuid peamine erinevus tuleneb ligipääsust tarkvara funktsionaalsusele. Kui lokaalse tarkvara puhul peab kliendi füüsilises seadmes olema tarkvara ja ressursid tarkvara käivitamiseks, siis SaaS puhul on tarkvara kasutamiseks oluline vaid ühendus teenusega.

Paindlik maksustamispoliitika, ulatuslikum kasutusvõimalus, automaatsed uuendused, ligipääsetavus ja püsiv kättesaadavus on teenusena pakutava tarkvara alusväärtused. Selle asemel, et osta tarkvara ja tarkvaravärskendusi, võimaldab SaaS maksutada tarkvara kasutamist vastavalt kasutatavatele operatsioonidele või perioodipõhisele kasutamisele. SaaS pilveteenuse läbi on tarkvara kasutajal laiendatud valikud tarkvaraoperatsioonide kasutamisel. Kliendil on valikus kasutada rohkem või vähem teenuseid vastavalt vajadusele nii, et vastavalt maksustamispoliitika erisustele on kliendil võimalus maksta ainult kasutatud ressursside eest. Kuna kasutatav tarkvara paikneb teenusepakkuja poolsetes serverites, on teenusepakkujal uuenduste korral võimalus konfigureerida muutused kasutajale võimalikult vähe ebamugavusi põhjustades. Mitmekülgne ligipääsetavus ja püsivus ühendus tagavad lõppkasutajale pääsu oma andmetele ligi erinevatel viisidel ja igal ajal [8].

Tarkvara teenusepakkuja jaoks SaaS-i puhul on peamiseks kohustuseks klientide ees tagada suure klientide hulga korral püsiv ligipääs ja andmete korrektsus. Samuti on pakkuja poolelt eetilise kohustus tagada usaldusväärsus ja ausus andmete eesmärgipärasel kasutamisel ja kasutaja operatsioonide maksustamisel. Usaldusväärsed teenusepakkujad suudavad pakuvad andmete omanikule võimalust integreerida andmeid kliendi poolt valitud süsteemides [8].

Infoedastus- ja talletamisprotsesside puhul on peamiseks muredeks kasutaja jaoks andmete omandiõigus ning turvalisus. Kuigi töölauarakendustes hoitav informatsioon ei ole alati kergesti jaotatav ning on kasutaja puhul andmete salvestamiseks kasutatava füüsilise seadme omandikaotuse või turvariskide tõttu ohustatud, pakub kliendi isiklikus seadmes sisalduv informatsioon kindluse, et turvalekete puudumisel on kasutaja andmed täielikult kasutaja käsutuses. Selliselt on andmete omanikul kohustus hoolitseda isiklikult privaatsete andmete lekkimise ärahoidmise eest. Seevastu peab SaaS-i puhul klient

usaldama teenusepakkujat, et teenusepakkuja teenus paigutab privaatsed andmed turvaliselt ning piirab mitteautoriseeritud isikutele ligipääsu andmetele. Kuna teenuse puhul on kliendile ja teenusepakkujale sama teenuse külge ühendunud palju teisi kliente, on teenusepakkuja kohustus hoolitseda selle eest, et ühegi kliendi andmed ei läheks segamini. Üldiselt on SaaS teenusepakkujate alusdokumentides sätestatud andmete omandiõiguse ja turvalisuse sätted, deklareerides andmete omanikuks kliendi ja lubades tagada andmetele ligipääsu ainult kliendile või kliendi poolt määratud isikutele [9].

Kuna SaaS pakub tarkvara teenusena, on kliendil laiem vabadus oma andmete kasutamisel. Kui võrrelda teenust mõne veebirakendusega, pakub teenus kliendile palju laialdasemat võimalust andmeoperatsioonide teostamiseks, mis tähendab otseselt suuremat omandivabadust. Kuna andmete omanik on kasutaja, kes andmed süsteemi sisestab, siis on teenusena realiseeritud tarkvara andmete omandiõiguse põhimõtete järgimisel oluliselt omanikusõbralikum lahendus, sest kasutajal on võimalus oma andmed siduda ka teiste süsteemidega. Nii ei ole kasutaja privaatne info seotud ainult pakkuja rakendusega, vaid on seotud teenusega võimalusega pääseda ligi nii teenusepakkuja rakenduses kui ka pakutavate liideste läbi.

SaaS pakub kasutajasõbralikku, laialdase sidumisvõimalusega, omandipärast ja püsivat teenust klientidele, kelle jaoks on oluline andmetele ligipääs enda valitud vahenditega. SaaS tehnoloogia andmete turvalisuse ja andmete eraldatuse tagab teenusepakkuja ning olenevalt pakkuja usaldusväärsest võib SaaS olla klientidele isegi usaldusväärsem variant kui mõni lokaalne rakendus või veebitarkvara.

## **2.5 Swagger**

Veebiteenuse üheks oluliseks osaks on täpne teenuse dokumentatsioon. Teenuse kasutamise aluseks on kasutamisselgus ning korrektne andmete edastamine. On oluline tagada klientrakenduste loojatele täpne juhend teenuse sidumiseks ja vahetatava info kliendipoolsele struktureerimisele.

Swagger on raamistik API-de arendajatele, mis võimaldab ja mugavdab API arendamise elutsüklit disainist dokumentatsioonini. Swagger võimaldab loodava API struktuuri analüüsides genereerida interaktiivse dokumentatsiooni. Swagger võimaldab manuaalset

dokumentatsiooni kirjutamist või API lähtekoodi põhjal automaatset dokumentatsiooni genereerimist [10].

## **2.6 Postman**

Postman on API arendajate jaoks loodud interaktiivne ja mitmekülgne tööriist, mis võimaldab jagada, testida, dokumenteerida ja jälgida API-de arendamisprotsessi ja valmisprodukti töötamist. Postman võimaldab testida valmisteenuse REST liidest kontrollides tagastatavat infot ja tagastatavat andmete struktuuri. Postmanis on võimalik päringuid rikastada skriptidega, mis käivitavad etteantud testid ja seeläbi kontrollivad teenuse funktsionaalsust. Lisaks võimaldab Postman loodud testi- ja päringukollektsioonide ning muutujate keskkondade jagamist [11].

### **3 Info kogumine ja konkurentsianalüüs**

Huvihariduse pakkujatele ja tellijatele on päevikupidamine assisteeriv ja informatiivne protsess, mis üldiselt otsest mõju pakutava teenuse kvaliteedile ei oma. Küll aga on päevikupidamine oluline osa tagasiside protsessist ning aitab huvikoolide tegevuse koordineerimisel tegeleda mugavamalt ja kiiremini ilmnevate korralduslike probleemidega. Samuti on huvitegevuse info kättesaadavus aluseks kliendi rahulolu tekkimisel, mis vähendab teenuse tellimisest loobumise tõenäosust.

Käesolev peatükk viib avalikel veebilehtedel oleva info ja autori isikliku kogemuse põhjal läbi huvikoolide infovajaduste analüüsi ja esitab elektroonilise huvikooli päeviku süsteemi funktsionaalsed nõuded. Hetkeseisu elektroonilise päevikupidamise osas hinnatakse huvikoolide ja konkureerivate teenusepakkujate kodulehtedelt saadavate andmete põhjal. Konkurentsianalüüsis analüüsitakse kõiki enamlevinud vahendeid elektroonilisel päevikupidamisel ning hinnatakse kõikide olemasolevate meetodite konkurentsieeliseid ja -puuduseid.

Info kogumise ja analüüsi peatükis selgunud huvikoolide päevikupidamise vajaduste põhjal viiakse läbi loodava veebiteenuse analüüs, mida aluseks võttes luuakse bakalaureusetöö raames valmiv veebiteenus.

#### **3.1 Huvikoolide päevikupidamise hetkeseis**

Eestis on antud bakalaureusetöö kirjutamise hetkel Eesti Hariduse Infosüsteemi (edaspidi EHIS) avaliku otsingumootori abil leitava info põhjal registreeritud kokku 778 huvikooli. Nendest huvikoolidest 318 on spordikoolid, 152 muusika- ja kunstikoolid, 27 tehnika-, loodus-, loome- ja huvimajad või -keskused ning 281 muud huvikoolid [12].

Autori isikliku kogemuse ja avaliku teabe põhjal on tekkinud hinnang, et Eestis tegutsevatest huvikoolidest ja spordiklubidest ei ole ühtegi või on väga vähe selliseid, kus infoedastuseks ei kasutata ühtegi infosüsteemi või tarkvaralahendust. Paber kandjal päevikupidamine on reeglina vahetunud elektrooniliste vahendite vastu. Paljud huvikoolid ei kasuta ühtegi spetsialiseerunud infosüsteemi, vaid kasutavad reeglina

üldkasutatavaid tarkvaralahendusi, eelkõige Exceli arvutustabeleid info ja kontaktandmete talletamisel ja töötlemisel ning sellele vastavaid mugavaid kommunikatsioonivahendeid, näiteks üldiseid meilivahetusvahendeid. Teised huvi- ja spordikoolid kasutavad spetsiaalseid vahendeid, näiteks VEMI infosüsteemi spordi- ja huvikoolidele [13] või Sportlyzer spordiklubide haldustarkvara [14].

Hetkeolukorra uurimiseks saab kasutada erinevaid avaliku info põhjal töötavaid konkurents- ja turundustööriistaid. Sportlyzer kasutajaskonna uurimiseks pakub Sportlyzer ise otsingutööriista, mille põhjal saab otsida klientideks registreerunud spordiklubide infot. Antud otsingumootori info põhjal on Sportlyzeril Eestis 150 klienti, kelle hulgas on rohkem korvpalli- ja jalgpallikoole, vähem muude spordialadega tegelevaid spodikoole [15]. Teise suurema konkureeriva teenusepakkuja VEMI kohta infot otsides on võimalik Semrush tarkvara kasutades välja selgitada, et väljaspool VEMI süsteemi on VEMI-le viidatud linke 41, mis on kõik esitatud 20 erineval domeenil. Selle põhjal on võimalik leida vähemalt 20 suuremat huvihariduse pakkujat, kelle kodulehtedel on otselingid VEMI süsteemi [16]. SimilarWeb veebitarkvara otsinguga selgub, et VEMI veebiliiklusest on 100% Eestist, kusjuures konkureerival Sportlyzer süsteemil on Eestist tuleva liikluse suhtarvuks ligi 15% [17].

Teiste päevikupidamispraktikate kohta ei ole võimalik avaliku info põhjal põhjalikke järeldusi teha kuna iga huvikooli töökorraldus on täielikult huvikoolide põhine ning kujuneb välja vastavalt reaalsele huvikoolide vajadustele.

### **3.2 Vajadused**

Huvihariduse pakkujate kommunikatsiooniprotsessid on huvitegevuse läbiviimise koordineerimisel ja assisteerimisel vältimatud. Haldamaks huvikooli klientide nimekirjasid ja kontaktandmeid, peab huvikooli poolelt olema kättesaadav alati kõige värskem info, et vältida infokadusid ja rahulolematuid kliente puudulike infoedastusprotsesside tõttu. Huvikoolide klientide põhiline vajadus on omada alati kõige päevakohasemat korralduslikku infot. Huvikoolide ja nende klientide seisikohalt on oluline omada vahendit, mille abil saaks võimalikult mugavalt ja kiiresti edastada vastastikku muutuvat infot tõrgeteta, mis oleks põhjustatud kontaktandmete muutumisest.

Elektronilise päevikupidamise vajadused on huvikoolile võimaldada hallata ja kättesaadavaks teha vajalikele huvikooli töötajatele huvikooli klientide ja töötajate nimekirjad ja kontaktandmed. Huvikooli haldamisel on oluline hallata kõiki huvikooli huvigruppe, nende liikmete ja juhendajate kontaktinfot ja huvigruppide korralduslikku infot nii, et kõik huvigrupi tööga seotud osapooled saaksid endale vajaliku info kätte. Huvigruppide juhendajate seisukohalt on oluline omada ülevaadet juhendatavate gruppide liikmetest ning omada alati ligipääsu nende kontaktidele. Samuti on huvigrupi juhendajale oluline võimalus teha huvigruppide korraldusliku info, näiteks trenniaegade, muutumisel uus info kättesaadavaks kõikidele liikmetele. Huvigruppide liikmete ehk klientide seisukohalt on kõige olulisem ligipääs huvigruppide korralduslikule infole, huvikoolide üldisele avalikule ja huvigruppide infole ning huvigruppide juhendajate ja huvikoolide töötajate kontaktinfole. Huvikoolide ja huvigruppide korraldusliku ja töötajate info puhul on oluline võimaldada ka info kättesaadavus võimalikele tulevastele klientidele, et teha uute klientide saamine võimalikult mugavaks protsessiks.

Huvikoolide ja klientide vajadusi silmas pidades tuleb luua eeldus, et süsteemis sisalduva info omanik, antud töö kontekstis huvikool, omaks ligipääsu oma andmetele endale sobival viisil. Huvikooli infole ligipääsust tuleneb antud töö eesmärk luua päevikupidamise tarkvara teenusena, et andmete omanik omaks võimalus mitmel moel oma andmetele ligipääsuks ja saaks vastavalt oma soovidele andmetega ringi käia.

### **3.3 Sihtrühmad**

Antud bakalaureusetöö peamine fookus on luua veebiteenus, mis rahuldaks huvi- ja spordikoolide päevikupidamise funktsionaalsust. Seetõttu on käesoleva töö raames loodaval veebiteenusel järgmised sihtgrupid:

- Huvikoolid ja huvikoolide kliendid, kes kasutavad veebiteenuse funktsionaalsust huvitegevuse kommunikatsiooniprotsesside mugavdamiseks ja kontaktandmete haldamiseks ning ligipääsuks
- Huvikoolide tarkvaralahenduste pakkujad, kes kasutavad veebiteenuse funktsionaalsust huvikoolide veebikeskkondade arendamisel

Lisaks eelpool nimetatud kahele peamisele sihtrühmale loob käesoleva töö tulemusel valminud teenus edasiarendamise korral võimaluse teenindada ka teistsuguse

suunitlusega sihtrühmasid. Mõned näited, milliste sihtrühmade tööd võiks antud veebiteenuse baasil loodav süsteem tulevikus teenindada on välja toodud järgnevalt:

- Üldhariduskoolid, kellele võimaldada kasutada elektroonilise päeviku funktsionaalsust
- Ettevõtted, kellele pakkuda töögruppide loomise, haldamise ja graafkualase töötamise korraldamise võimalusi
- Ülikoolid, kellele pakkuda aineoengute ja praktikumitundide päevikupidamise ning õppegruppide koostamise funktsionaalsust
- Ühingud ja ühistud, kellele pakkuda koosolekute korraldamise võimalust pakutava teenuse näol

Sellisel on võimalik loodava veebiteenuse funktsionaalsusele luua lisafunktsionaalsust laiendades teenust kasutatavat sihtrühma ning olenevalt kasumiteenimismvormist suurendada teenust haldava ettevõtte kasumit.

### **3.4 Konkurentsianalüüs**

Konkurentsianalüüs peab andma turuletuleva toote või lahenduse jaoks selged hinnangud, millised on olemasolevad võimalused ja kas pakutav uus lahendus suudaks võistelda juba turul läbi löönud lahendustega. Konkurentsianalüüs vastab küsimustele, millised on üldised konkurentsijõud tegevusharus, kes on peamised konkurendid, kui suur on nende turuosa, millised on eelised pakutava lahenduse ees ning millised on pakutava lahenduse eelised konkurentide ees. Konkurentsianalüüsi tulemuseks on teadmine konkurentsipositsioonist ja vastuse küsimusele, mille unikaalsega on võimalik konkurentidest eristuda [18].

Käesolev konkurentsianalüüs analüüsib Eestis huvi- ja spordikoolidele suunatud elektroonilise asjaajamise valdkonnas peamisi tegutsevaid pakkujaid ja alternatiivseid lahendusi tutvustades ja hinnates iga pakkuja eeliseid ja olemasolevaid kliente. Eesti turul on selles valdkonnas peamisteks tegijateks Vemi OÜ [13] poolt pakutav huvi- ja spordikoolidele suunatud infosüsteem ja Sportlyzer spordimeeskondade haldamise tarkvara [14]. Alternatiivsete lahendustena hinnatakse Excel tabelarvutustarkvara info

töötlemisel ja talletamisel ning Studium [19] ja eKool [20] võimalike tulevaste konkurentidena.

### 3.4.1 VEMI

VEMI on osäühing Vemi poolt loodud infosüsteem, mis pakub huvi- ja spordikoolidele elektroonilise asjaaajamise funktsionaalsust. VEMI kodulehekülje andmetel võimaldab VEMI treeninggruppide moodustamist ja laste lisamist treeningrühmadesse, tasude määramist nii personaalse, tunni- kui ka kuupõhise arvestusega. Samuti võimaldab VEMI treeningtundide lisamist, kohalkäimise märkimist ja treeningute kirjeldamist. Võimaldatud on arvete genereerimine, laekumiste ja saldode jälgimine ning maksevõlgnike haldamine. Infosüsteem pakub statistika edastamist Eesti Hariduse Infosüsteemi (edaspidi EHIS), andmete edastamist Pearaha Arvestamise Infosüsteemi (edaspidi PAI) ja deklaratsioonide genereerimist Maksu- ja Tolliametile (edaspidi EMTA) jaoks. Lisaks võimaldab VEMI saata teateid lapsevanematele ja treeneritele [13].

Lähtudes huvikoolide päevikuprotsesside antud töös püstitatud vajadustest selgub, et VEMI infosüsteem võimaldab kogu soovitud funktsionaalsust. Seejuures pakub VEMI süsteem püstitatud probleemidele lisaks palju lisafunktsionaalsust huvikoolide tegemisi jälgivate ametite infosüsteemidega liidestumise näol.

Rahuldatus eesmärkidele ja ka väga laialdasele lisafunktsionaalsusele vaatamata on VEMI süsteemil mitmeid põhjuseid, miks ei ole kogu sihtrühmas saavutatud täielikku turueelist. Üks põhjus VEMI vähesel kasutamiskäitumise taga on maksekord. Olenemata huvigrupi vajadusest, tuleb maksta kuupõhist tasu aktiivsete liikmete arvu põhjal. Nii peavad sama liikmete arvuga huvikoolid maksma sama palju isegi ka juhul kui üks huvikool kasutab vaid huvigruppide kohalkäimise märkimise funktsionaalsust ja teine huvikool kasutab süsteemi kõiki funktsionaalsuseid. Nii on valmis süsteemi kasutama lisatasu eest vaid need huvikoolid, kes näevad endale kogu funktsionaalsusest kasulikku abivahendit.

Teine põhjus, miks VEMI süsteem ei ole ainus eelistatud päevikupidamise vahend seisneb selles, et süsteemil puudub liidestamise võimalus. Huviharidusettevõtete jaoks on oma huvikooli identiteedi rõhutamine kasutajale ning selleks, et kuvada näiteks huvikooli kodulehel huvigruppide liikmenimekirja, tuleb siiski eraldi VEMI süsteemist antud info hankida ning staatiliselt oma veebikeskkonnas kuvada. Liidese abil, mida pakuks



veebiteenus, oleks klient ise oma ressursside omanik ning omaks seejuures ka lihtsat võimalust enda andmete kättesaamiseks.

Liidestusvõimaluse puudumise tõttu on VEMI puhul oluliseks puuduseks selge andmete omandi poliitika. Sisestades andmed VEMI süsteemi, ei ole üheselt arusaadav, mida pakkuja andmetega teeb, kas pakkuja juurest lahkudes on võimalik andmed kaasa võtta ning kas teenusepakkuja kliendisuhete lõppemisel seotud andmed kustutab või mitte. Kliendi seisukohalt on oluline privaatsete andmete ligipääs ning omandiõigus.

Võttes arvesse VEMI süsteemi kohta kogutud infot võib pidada antud süsteemi tugevaks konkurendiks loodavale süsteemile.

### **3.4.2 Sportlyzer**

Sportlyzer on selgelt spordiklubide tegevusele fokuseeritud. Sportlyzer pakub omapoolselt kolme rakendust, milleks on treeneri, sportlase ja lapsevanema rakendused. Treenerirakendus pakub treeneritele võimalust planeerida treeninguid, täita treeningkalendrit, teavitada tegutsevaid sportlasi või sportlaseid esindavaid lapsevanemaid nii treeningutega seotud muudatuste osas kui ka trennis toimuva kohta kohalkäimiste märkimise või märkuste ja arenguteadete saatmise näol. Huvikoolidele pakub Sportlyzer võimalust koordineerida arvete saatmist ning laekumiste kontrolli. Lapsevanema ja sportlase rakendused võimaldavad kontrollida trennidest osavõtmist, anda treenerile teada trennis mitteosalemisest, kontakteeruda trennikaaslaste või treeneritega ja uuendada enda kontaktinfot [14].

Lisaks eelpoolnimetatud funktsionaalsusele, mis kattub üsna suurel määral ka VEMI poolt pakutavate funktsionaalsustega, pakub Sportlyzer spordikoolidele ka võimalust luua läbi süsteemi tasuta spordiklubi koduleht või liidestada mõningaid spordiklubi funktsionaalsuseid oma kodulehele. Sportlyzer pakub liidestust spordigruppide info liidestamiseks kodulehele gruppide reklaamimise eesmärgil, treenerite nimekirja liidestust näitamaks treenerite infot, treeningrühmade trennigraafikute liidestamist, spordiklubidesse registreerimise liidestamist ning Sportlyzer süsteemi sisse logimise liidestust [14].

Kuna Sportlyzer on suunitlusega spordiklubidele, ei pakuta teistele sarnastele teenusepakkujatele konkurentsi muudes huvivaldkondades tegutsevate huvikoolide osas.

Nii pakub Sportlyzer praktiliselt terviklikku lahendust spordiga tegelevatele huvikoolidele.

Teisalt on Eesti turul Sportlyzer võrreldes VEMI süsteemiga veidi nõrgem konkurent, kuna ei pakuta võimalust EHIS-e, PAI ega EMTA-ga liidestumist. Seejuures omab Sportlyzer liidestatavate komponentide pakkumise näol tugevat eelist, mis laiemat pilti vaadates võib olla isegi olulisem kui eelpoolnimetatud infosüsteemidega liidestumine.

Sportlyzeri puhul on võimalik probleemina välja tuua maksustamispoliitika. Olenemata spordikooli suurusest, aktiivsete kasutajate arvust ning soovitud funktsionaalsusest kehtib täispaketi kasutamiseks kindel kuutasu. Samuti on andmete omanikel Sportlyzeri süsteemi puhul oma andmetele ligipääsuks vaja kasutada teenusepakkuja süsteeme. Seejuures piiratud hulgal andmete liidestamise võimalus teeb antud süsteemi puhul andmete omanikele ligipääsetavuse osas tugevaid vastutulemisi.

Arvestades Sportlyzeri võimalusi ning probleeme saab väita, et spordiklubide vajadusi arvestades on Sportlyzer üks põhjalikumaid ja täiuslikumaid lahendusi spordikorralduse elektrooniliseks päevikupidamiseks nii spordikooli kui ka spordikooli kliendi vaatepunktist, kuid terviklahenduse pakkumisel kõikidele huvihariduse pakkujatele on nõrgem konkurent kui on eelmises punktis tutvustatud VEMI.

### **3.4.3 Excel**

Need huvikoolid, kelle liikmete arv ei ole eriti suur või kelle vajadused piirduvad vaid kontaktinimekirjade salvestamisega, mille põhjal saata arveid ning infot, kasutavad reeglina liikmeinfo hoidmiseks Exceli tabeleid. Kuna Excel on kasutatav nii töölauarakendusena kui ka veebis, pakub see võimalust vajalikule infole ligipääsuks ka internetis muutes vajaliku info ka volitatud inimestele jagatavaks. Samuti ei nõua Excel uute keskkondade selgeks õppimist eeldusel, et varasem kogemus Exceliga on olemas, ega ka huvikoolile lisaväljaminekuid. Kuigi töölauarakendus on osa tasulisest kontoritarkvarast, ei pea Exceli veebiversiooni eest tasuma [21].

Exceli üheks tugevamaks pooltargumendiks on täielik andmete omandiõigus. Andmed on andmesisestaja omandis ja privaatsed seni kuni andmete omanik ise otsustab andmeid jagada või avalikustada. Nii pakub Excel täielikku kontrolli andmete üle.

Seejuures ei paku Excel lahendust enamusele püstitatud probleemidele. Kuigi on võimalus luua huvigruppidepõhised tabelid ja säilitada seal vajadusel ka trenniinfo säilitamist või kohalkäimiste ülesmärkimist, ei oleks ühelgi huvikoolil privaatsuse huvides võimalik kogu infot korraga kõikidele liikmetele jagada. Seega tuleks päevikuinfo salvestada kas huvikooli liikme põhiselt eraldi tabelitesse või manuaalselt vajalik info kogu infost eraldada.

Võrreldes eelnevate konkurentidega ei paku Excel eriti tugevat konkurentsi. Excel on universaalne tööriist info töötlemiseks ning kindlat laadi info säilitamiseks, kuid koondades kogu huvikooli tegevuse info Exceli tabelisse ei saavutataks elektroonilise päeviku põhimõtete täitmist. Kasutajad hindavad Exceli puhul harjumuspärasust ja mugavust, sest suur osa inimestest suudab antud tööriista abil infot lugeda ning ka vajalikul määral töödelda.

Excel on sobilik vahend omamaks huvikooli tasemel listina ülevaadet huvikooli liikmetest, nende arvust ja vajalikest kontaktandmetest, mida sisseehitatud Exceli funktsionaalsusega ja sobivate filtreerimistööriistadega on võimalik mugavalt eraldada. Samas on Excel pävikupidamisprogrammina ebasobiv, kui pävikus säilitatud infole peaks ligi pääsema paljud inimesed. Info avalik kättesaadavus põhjustaks aga kontaktandmete privaatsuse rikkumise. Niisiis on Excel pävikupidamist silmas pidades huvikoolide vaatenurgast pigem assisteeriv tööriist, mille abil säilitatud info puhul peab selgelt piirama, kellele õigus info lugemiseks anda ja kellele mitte.

#### **3.4.4 eKool ja Stuudium**

eKool ja Stuudium on üldhariduskoolide elektroonilise pävikupidamise funktsionaalsust pakkuvad süsteemid. Antud töö kontekstis ei hinnata kumbagi süsteemi konkreetse funktsionaalsuse sobivust püstitaud probleemidele ja eesmärkidele, vaid käsitletakse mõlemat kui potentsiaalset konkurenti turu hõivamiseks.

Mõlema süsteemi aluseks on üldhariduskoolide korralduslik formaat ning põhifookus on hinnetepõhise õppetöö korraldamine ning info kättesaadavus lapsevanematele. Seejuures on võimalik mõlemale süsteemile luua lisafunktsionaalsus, mis võimaldab rahuldada ka huvikoolide vajadused. Kuna eKool ja Stuudium on reeglina koolides koolitöö kohustuslik osa, siis on nii vanematel kui ka lastel nende kasutamiskogemus olemas. Selliselt on võimalus mõlemal pakkujal kasutada ära klientide harjumuspärase eelistuse

tekkimist ning võimaldada ka huvi- ja spordikoolidel oma süsteemide kasutamist. Kuna baasfunktsionaalsus on mõlema süsteemi puhul loodud, siis on huvikoolide formaadile sobilikuks ümberkohandamine lihtsam ja odavam kui uue süsteemi valmis ehitamine.

Kuigi kumbki vahend ei paku otsest funktsionaalsust huvikoolide vajadusi silmas pidades, omavad mõlemad tugevat kasutajabaasi, mille põhjal oleks eelis teiste konkurentide ees saavutada väga lihtne. Seejuures on Studium tänasel päeval ka mõnede muusika- ja kunstikoolide poolt kasutusele võetud, sest sealne huvikooli töökorraldus sarnaneb kõige enam üldhariduskoolide korraldusele, kuna kasutatakse samuti hindamist. Küll aga puudub võimekus liikmete arvete haldamiseks, mis VEMI ja Sportlyzer puhul on olemas.

### **3.5 Konkurentsieelised**

Lähtudes konkurentsianalüüsist saab põhiliste konkurentidena arvestada siiski vaid VEMI ja Sportlyzer süsteeme. Kuna mõlemad süsteemid on selge suunitlusega täitmaks huvi- ja/või spordikoolide päevikupidamisest tingitud vajadusi, on nendele süsteemidele loodud ka kõige spetsiaalsem funktsionaalsus.

Konkureerivate pakkujate puuduseid ja huvikoolide vajadusi arvesse võttes saab loodava päevikutarkvara lahendamise aluseks võtta SaaS ehk tarkvara teenusena printsiibid. Pakkudes andmete omanikule võimalust oma andmete liidestamiseks ja kasutamiseks soovitud süsteemides, pakub loodav teenus selget konkurentsieelist. Kuna huvikoolide andmed kuuluvad huvikoolile, peab huvikoolil olema andmete omanikuna täielik vabadus andmete töötlemisel, hoiustamisel ja eemaldamisel. Teenus võimaldab andmete omanikul funktsionaalsust andmete täielikuks haldamiseks.

Mõlema konkurendi analüüsimise käigus selgus, et ühiseks murekohaks, miks huvikoolide seas tõenäoliselt täielikku edu ei saavutata, on maksustamiskord. Antud töö käigus loodava veebiteenuse üheks suurimaks võimalikuks konkurentsieeliseks võib kujuneda paindlik maksustamine. Kuigi käesolevas töös loodavale teenusele maksustamispoliitikat ei kehtestata, on veebiteenusena lahendades võimalik pakkuda paindlikke maksureegleid vastavalt sellele, millist funktsionaalsust klient realselt vajab ja kasutab. Nii ei pea ükski spordi- ja huvikool maksma funktsionaalsuse eest, mida tegelikkuses ei kasutata.

Teise eelise loob liidestamise võimaldamine. Kuigi Sportlyzer pakub ka mingite funktsionaalsuste liidestamisvõimalust, ei pakuta ressursside omanikule ehk kliendile võimalust liidestada vabal valikul kogu oma informatsiooni omamata seetõttu ka info üle täielikku omandiõigust ja kontrolli.

Kolmanda konkurentsieelisenä saab loodava süsteemi puhul välja tuua suurema sihtrühma. Kui veebiteenus luuä selliselt, et iga moodul on eraldiseisev ja eraldi kasutatav, saab pakkuda teenust ka väljaspool sihtvaldkonda tegutsevatele potentsiaalsetele klientidele. Selliselt on hästi loodud veebiteenusega võimalik teenindada kõiki sihtrühmade analüüsis väljatoodud võimalikke kliente samas kui VEMI ja Sportlyzer on kindla eesmärgi põhised.

## **4 Analüüs**

Käesolevas peatükis analüüsitakse, millistele nõudmistele peab vastama loodav veebiteenus. Teenuse nõuete analüüsi aluseks võetakse info kogumise ja konkurentsianalüüsi käigus ilmenenud huvikoolide pävikupidamise vajadused.

### **4.1 Veebiteenuse analüüs**

Veebiteenuse analüüs esitab töös realiseeritavat funktsionaalsust pakkuvale veebiteenusele seatud funktsionaalsed ja mittefunktsionaalsed nõuded ning kirjeldab veebiteenuse andmeressursid.

#### **4.1.1 Funktsionaalsed nõuded**

Teenuse funktsionaalsed nõuded on järgmised:

- Teenus peab võimaldama registreerida kasutajat, huvikooli ja huvigruppi
- Teenus peab võimaldama pärida kasutajat, huvikooli ja huvigruppi ning nende nimekirjasid
- Teenus peab võimaldama uuendada kasutaja, huvikooli ja huvigrupi infot
- Teenus peab võimaldama kustutada kasutajat, huvikooli ja huvigruppi
- Huvikooli lisamisel peab tekkima administraatori seos lisatud huvikooli ja lisaja vahel
- Huvigrupi lisamisel peab lisatama seos huvikooliga
- Teenus peab võimaldama pärida iga kasutaja, huvikooli ja huvigrupi andmeid
- Teenus peab võimaldama lisada ja eemaldada administraatori seost kasutaja ja huvikooli vahel

- Teenus peab võimaldama lisada ja eemaldada töötaja seost kasutaja ja huvikooli vahel
- Teenus peab võimaldama lisada ja eemaldada juhendaja seost kasutaja ja huvigrupi vahel
- Teenus peab võimaldama lisada ja eemaldada liikme seost kasutaja ja huvigrupi vahel
- Teenus peab võimaldama pärida huvikooli administraatoreid ja töötajaid
- Teenus peab võimaldama pärida huvigrupi juhendajaid ja liikmeid
- Kasutaja kustutamisel kustutatakse huvikooli/huvigrupi ja kasutaja vahelised seosed
- Administraatori kasutaja kustutamisel kustutamine ebaõnnestub kui huvikoolile pole määratud ühtegi teist administraatorit
- Huvigrupi ja huvikooli kustutamisel kustutatakse ka seosed kustutatud ressursi ja kasutajate vahel
- Veebiteenus peab vigade korral teavitama teenust kasutavat rakendust võimalikult informatiivselt

#### **4.1.2 Mittefunktsionaalsed nõuded**

Loodava veebiteenuse mittefunktsionaalsed nõuded on järgmised:

- Teenuse ressursside nimetamine peab olema võimalikult üldine ning looma eelduse teenuse sihtrühmade laiendamiseks
- Teenus peab olema kasutatav REST API kaudu ning omama kasutamist assisteerivat dokumentatsiooni
- Teenus peab realiseerima minimaalsel tasandil turvalisuse ligipääsuvõtme ehk tokeni näol

### 4.1.3 Andmeressursside kirjeldus

Andmeressursside kirjeldus kirjeldab kolme peamist ressurssi ja esitab iga objekti JSON-kujul näitepäringu vastuse. Kolm peamist ressurssi, mille omavahelisel ühendamisel ja kombineerimisel võimaldab veebiteenus luua päevikurakenduse on kasutaja ehk administraator, juhendaja või klient, institutsioon ehk huvikool ja grupp ehk huvigrupp.

Kasutaja ressurss sisaldab järgmisi andmeväljasid koos andmetüüpidega:

- id – integer: kasutaja süsteemi poolt määratud identifikaator
- username – string: süsteemis kasutaja poolt valitud unikaalne kasutajatunnus
- name – string: kasutaja nimi
- birthDate – date: sünnikuupäev
- idCode – string: kasutaja isikukood
- phoneNumber – string: kasutaja telefoninumber
- email – string: kasutaja meiliaadress
- address – string: kasutaja aadress

Kasutaja pärimisel süsteemist tagastatav näidis JSON formaat:

```
{
  "id": 100,
  "username": "testuser",
  "name": "Test Kasutaja Üks",
  "birthDate": "1995-04-04",
  "idcode": "11111111111",
  "phoneNumber": "53892333",
  "email": "i@ttu.ee",
  "address": "Raja 4, Tallinn"
}
```



Institutsiooni ressurss sisaldab järgmisi andmeväljasid koos andmetüüpidega:

- name – string: institutsiooni/huvikooli nimi
- registryCode – string: institutsiooni unikaalne registrikood
- address – string: institutsiooni registrijärgne aadress
- type – string: institutsiooni tüüp
- phoneNumber – string: institutsiooni üldtelefoni number
- email – string: institutsiooni meiliaadress
- webAddress – string: institutsiooni veebilehe aadress

Institutsiooni pärimisel süsteemist tagastatav näidis JSON formaadis objekt:

```
{  
  "name": "Koeru Muusikakool",  
  "registryCode": "12345678",  
  "address": "Kooli 1, Koeru",  
  "type": "Muusikakool",  
  "email": "muusikakool@koeru.ee",  
  "phoneNumber": "900292",  
  "webAddress": "koeru.edu.ee"  
}
```

Grupi ressurss sisaldab järgmisi andmeväljasid koos andmetüüpidega:

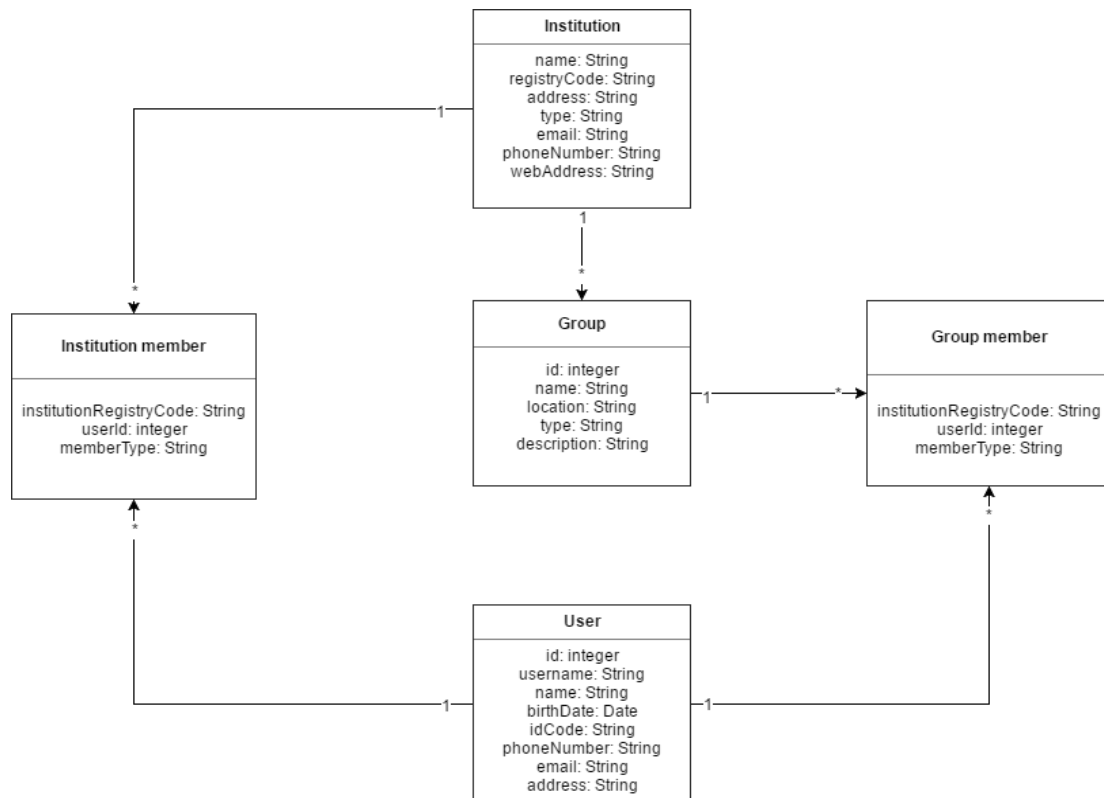
- id – integer: süsteemi poolt genereeritav unikaalne grupi id
- name – string: huvigrupi nimi
- location – string: huvigrupi tegutsemispaik
- type – string: huvigrupi tüüp

- description – string: huvigrupi kirjeldus

Grupi pärimisel süsteemist tagastatav näidis JSON formaadis objekt:

```
{
  "id": 100,
  "name": "Mudilaste klaveriklass",
  "location": "Koeru kooli klaveriklass",
  "type": "klaveriring",
  "description": "Klaveritund E, K ja R kell 17.00-19.00"
}
```

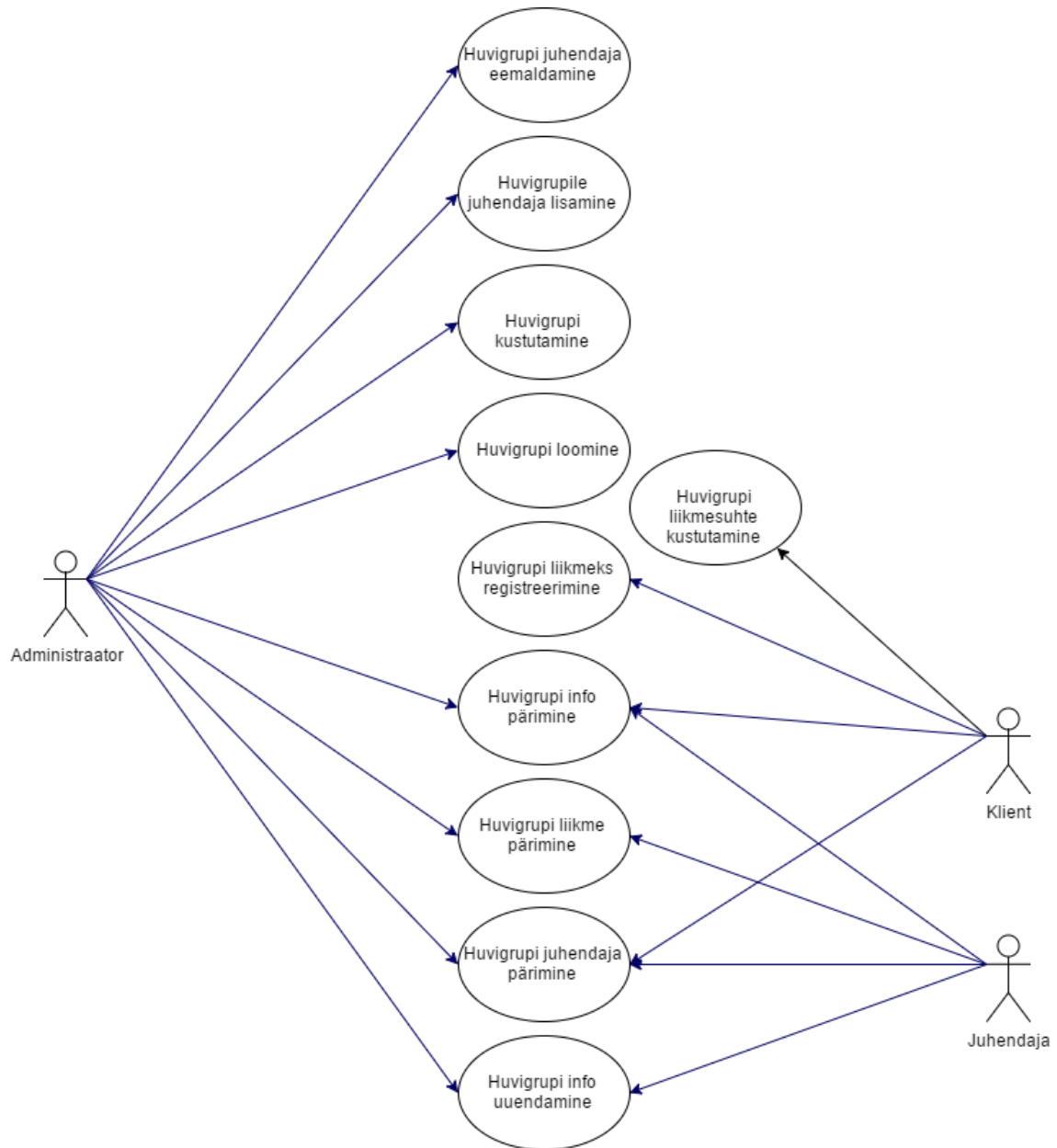
Kasutaja saab olla huvikooliga otseses seoses huvikooli administraatorina või huvikooli töötajana, samamoodi saab olla kasutaja seotud huvigrupiga otseses suhtes grupi juhendajana, kes täidab grupi tasemel administratiivset rolli, ja huvigrupi liikmena. Huvigrupi ja huvikooli vahel on otsene kuuluvussuhe, kus iga huvigrupp peab kuuluma ühte huvikooli. Kirjeldatud ressursside omavahelised suhted on esitatud järgmise klassidiagrammiga.



Joonis 3. Päevikuteenuse klassidiagramm

#### 4.1.4 Päevikuteenuse huvigrupiga seotud kasutusjuhud

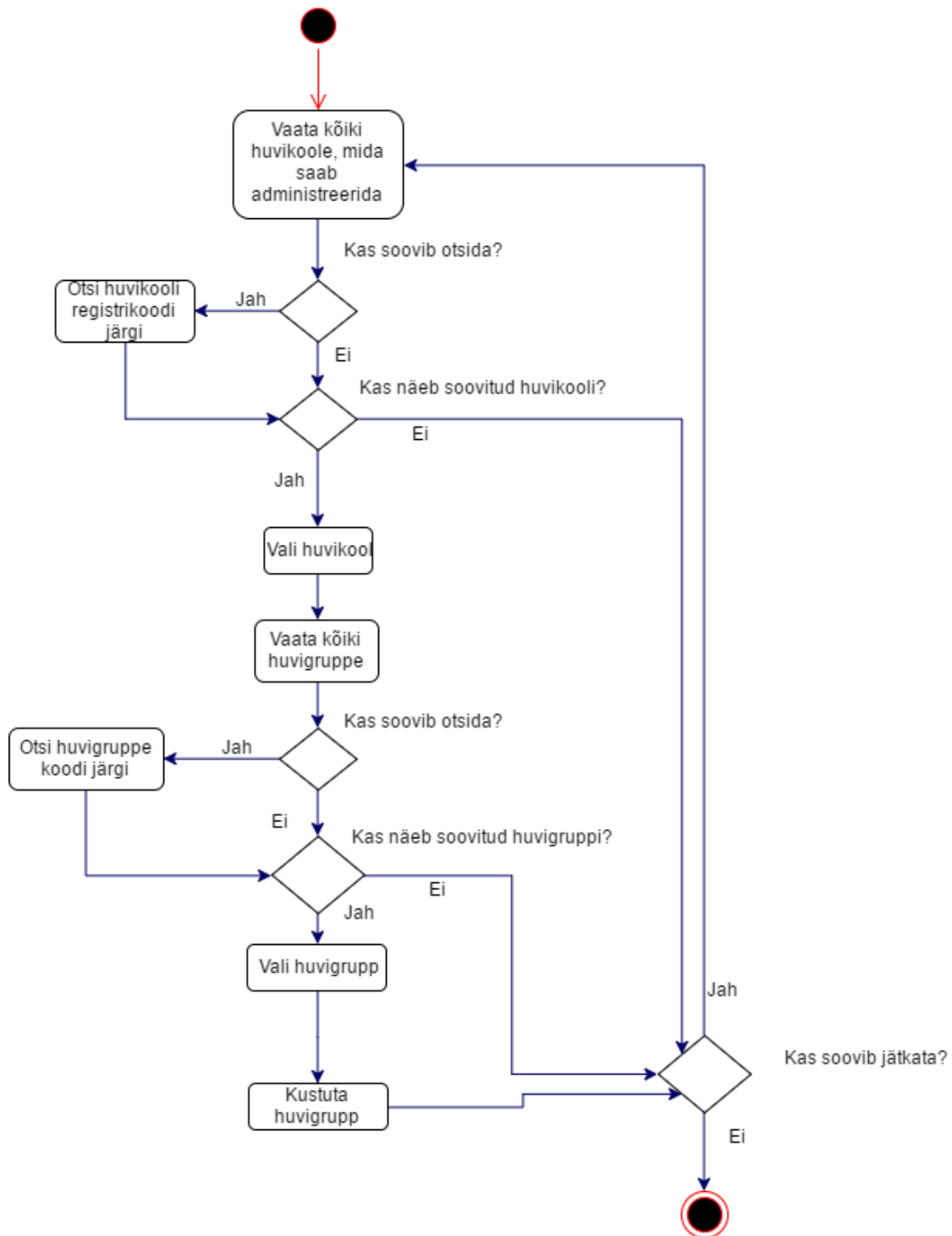
Käesolevas süsteemis on kõik kolm osapoolt otseselt seotud läbi huvigruppide. Joonis 4 kujutab süsteemi huvigruppide ressursiga seotud kasutusjuhud koos nende seotusega vastavate kasutajatega. Esitatud ei ole huvikooli ja kasutajatega seotud kasutusjuhud.



Joonis 4. Päevikuteenuse huvigrupi kasutusjuhtude diagramm

#### 4.1.5 Päevikuteenuse ühe põhiprotsessi tegevusdiagramm

Joonis 5 esitab huvigrupi kustutamise protsessi kirjelduse tegevusdiagrammina näitena teenuse realiseeritavatest operatsioonidest ja huvigrupi kustutamiseni jõudmise operatsioonide jadast.



Joonis 5. Huvigrupi kustutamise tegevusdiagramm

#### **4.1.6 Veebiteenuse turvalisus**

Antud bakalaureusetöö prototüüpteenuse raames realiseerib turvalisuse ligipääsuvõtmete süsteem. Igal kasutajakontol registreeritakse konto loomisel kasutajatunnus ja parool ning taustal genereeritakse igale kasutajale isiklik ligipääsuvõti. Sisselogimisinfo päringu saatmisel tagastatakse korrektse info korral kasutajale tema ligipääsuvõti, mille esitamisel saab kasutaja edaspidi päringuid teostada. Ligipääsuvõtme põhjal kontrollitakse teenuse tasemel, kas päringut tegeval kasutajal on õigus antud operatsiooni teostada.

## 5 Realisatsioon

Käesolev bakalaureusetöö peatükk kirjeldab veebiteenuse ja klientrakenduse arendamist eelmises peatükis kirjeldatud analüüsi põhjal.

Antud töö tulemusena realiseeritakse veebiteenuse analüüsis kirjeldatud nõudeid realiseeriv prototüüpteenus, mis on huvikoolide päevikutarkvara teenuse osalist funktsionaalsust realiseeriv allsüsteem. Teenuse realiseerimisel on loodud kõik eeldused arendada loodava prototüüpteenuse põhjal terviklahendus.

Päevikutarkvara lahendatakse teenusena SaaS põhimõtteid järgides. Sellisel looakse kõikide funktsionaalsete operatsioonide jaoks teenuseoperatsioon koos realisatsiooniga. Töö lahendatakse REST protokolliga järgiva teenusena, mis serveri poolel kasutab SOAP protokolliga järgivaid implementeeritud operatsioone. Veebiteenuse aluseks on teenust kirjeldav WSDL, mille põhjal genereeritakse operatsioonide ja objektidega programmikood. Genereeritud koodi operatsioonide loogika realiseeritakse ja implementeeritud operatsioone kasutades realiseeritakse REST veebiteenus lokaalses seadmes kasutamiseks. Lokaalse teenuse andmed säilitatakse teenusesisestes andmetalletamisstruktuurides ning testid ja dokumentatsioon looakse vastavalt Postman ja Swagger töövahendeid kasutades lokaalse REST teenuse alusel.

Veebiteenuse prototüüp on realiseeritud SOAP ja REST API-dena eesmärgiga pakkuda tervikteenuse arendamiseks laiemat toetuspinna ning võimalust otsustada, millise API tuge tervikteenuses pakkuda. Käesolevas töös loodud API-dest on dokumenteeritud ja testitud REST API.

### 5.1 Veebiteenuse realisatsioon

Veebiteenus realiseeritakse kolmele ressursile põhinevad operatsioonid. Operatsioonide aluseks on ressursid kasutaja (klient/töötaja), institutsioon (huvikool) ja grupp (huvigrupp). Iga ressursile realiseeritakse lisamise, pärimise, muutmise ja kustutamise ning lisaks ka ressursside omavaheliste seoste lisamise ja kustutamise

operatsioonid. Iga operatsiooni ebaõnnestunud päringu tulemusena tagastatakse veakood ja informatiivne veateade. Veebiteenuse operatsioonid ja operatsioonide sisend- ja väljundelemendid on defineeritud veebiteenuse WSDL dokumendis.

### 5.1.1 InstitutionService WSDL

Loodava veebiteenuse aluseks on WSDL ja kirjeldatud operatsioonid. Veebiteenuse kolm peamist ressursi - kasutaja, huvigrupp ja huvikool - on nimetatud WSDL-is vastavalt *user*, *group* ja *institution*. Teenuse WSDL kirjeldab kõikide elementide andmetüübid, kõikide operatsioonide sisend ja väljundsõnumite elemendid ja kõik veebiteenuse operatsioonid koos nende sisend- ja väljundsõnumitega. Kogu realiseeritava prototüüpteenuse WSDL kirjeldus on saadaval töö autori avalikus repositooriumis [22]. Järgnevalt on kirjeldatud WSDL põhjal grupi lisamise operatsioon.

- WSDL tüüpide (inglise keeles *types*) elemendis olevas skeemi (inglise keeles *schema*) elemendis on globaalsed tüübid *GroupType* ja *GetGroupType*, mis kirjeldavad grupi lisamisel grupi sisendandmete struktuuri ja grupi lisamisel edukalt loodud grupi väljundandmete struktuuri, kus on lisandunud *groupId*, mis on süsteemi poolt genereeritud unikaalne grupi identifikaator. Järgnevad joonised kujutavad WSDL kirjeldusfaili skeemi globaalseid grupitüüpe.

```
<xs:complexType name="GroupType">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="name" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="location" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="type" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="description" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Joonis 6. WSDL kirjeldusfaili grupi globaalne tüüp

```
<xs:complexType name="GetGroupType">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="id" type="xs:integer"/>
    <xs:element minOccurs="1" maxOccurs="1" name="name" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="location" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="type" type="xs:string"/>
    <xs:element minOccurs="1" maxOccurs="1" name="description" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Joonis 7. WSDL kirjeldusfaili grupi globaalne tagastustüüp

- Skeemi elemendid *AddGroupRequest* ja *AddGroupResponse* kirjeldavad päringute struktuuri. Grupi lisamise päringu element võtab sisendina

ligipääsuvõtme ehk tokeni, huvikooli registrikoodi, millesse grupp lisatakse ja globaalset *GroupType* tüüpi oleva grupp elemendi, mis vastavalt oma tüübile sisaldab ülalnimetatud *GroupType* tüübi elemente. Tagastatavas elemendis on edukuse element *success*, mis annab tõene ja väär väärtustena teada, kas operatsioon oli edukas. Vastavalt eduka operatsiooni puhul tagastatakse valikulisest elemendist grupi element, mille tüüp on eelmises punktis tutvustatud *GetGroupType*. Operatsiooni ebaedu korral on tagastatavaks elemendiks *error*, mis *ErrorType* globaalse tüübi struktuuri järgi sisaldab vea koodi ja sõnumit. Järgnev joonis kujutab WSDL kirjeldusfaili grupi lisamise päringu ja vastuse elemente koos elementide andmetüüpidega.

```
<xs:element name="AddGroupRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="token" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="registryCode" type="xs:string"/>
      <xs:element minOccurs="1" maxOccurs="1" name="group" type="tns:GroupType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="AddGroupResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="success" type="tns:SuccessType"/>
      <xs:choice>
        <xs:element name="group" type="tns:GetGroupType"/>
        <xs:element name="error" type="tns:ErrorType"/>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Joonis 8. WSDL kirjeldusfaili grupi lisamise päringu ja vastuse elemendid

- Sõnumi (inglise keeles *message*) elemendid *addGroupMessageRequest* ja *addGroupMessageResponse* defineerivad, mis andmeid vahetatakse veebiteenuse operatsiooni käigus. Grupi lisamise operatsiooni sõnumid defineerivad, et vahetatavad andmed on eelmises punktis kirjeldatud *AddGroupRequest* grupi lisamise päringu jaoks ja *AddGroupResponse* päringu vastusena. Järgnev joonis kujutab WSDL kirjeldusfaili grupi lisamise operatsiooni sisend- ja vastussõnumeid.



```

<wsdl:message name="addGroupMessageRequest">
  <wsdl:part name="parameter" element="tns:AddGroupRequest"/>
</wsdl:message>
<wsdl:message name="addGroupMessageResponse">
  <wsdl:part name="parameter" element="tns:AddGroupResponse"/>
</wsdl:message>

```

Joonis 9. WSDL kirjeldusfaili grupi lisamise sõnumielemendid

- WSDL liidese (inglise keeles *portType*) elemendi sees on kirjeldatud kõik vastavale veebiteenusele kuuluvad operatsioonid. Seal kirjeldatud *addGroup* operatsioon kirjeldab grupi lisamise operatsiooni ja defineerib, mis sõnumid päringu ja vastuse puhul saadetakse. Grupi lisamise sõnumid on kirjeldatud eelmises punktis ja grupi lisamise operatsioon koos sõnumitega järgmisel joonisel.

```

<wsdl:operation name="addGroup">
  <wsdl:input message="tns:addGroupMessageRequest"/>
  <wsdl:output message="tns:addGroupMessageResponse"/>
</wsdl:operation>

```

Joonis 10. WSDL kirjeldusfaili grupi lisamise operatsioon porditüübis

- WSDL sidumise (inglise keeles *binding*) elemendi sees on kirjeldatud WSDL operatsioonid, sealhulgas ka grupi lisamise operatsioon. Operatsioonide täpsed sisend- ja väljundsõnumid koos nõutud elementidega on defineeritud liidese elemendis kirjeldatud operatsiooni sisend- ja väljundelementidena.

```

<wsdl:operation name="addGroup">
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>

```

Joonis 11. WSDL kirjeldusfaili grupi lisamise operatsioon sidumise elemendis

Antud bakalaureusetöös valmiv veebiteenus koosneb 25 operatsioonist. Kogu WSDL koos kõikide kirjeldatud operatsioonidega ja sisend- ning väljundelementide kirjeldustega on saadaval viidatud repositooriumist [22].

### 5.1.2 Operatsioonide realisatsioon

Operatsioonid on realiseeritud SOAP teenusena Java programmeerimiskeeles. Käesolevas töös on kirjeldatud eelmises peatükis tutvustatud WSDL kirjelduse alusel realiseeritud grupi lisamise operatsiooni SOAP ja REST implementatsiooni, kirjeldatud

implementatsioonidele realiseeritud teste kasutades Postman rakendust ja dokumentatsioon Swagger rakendusega. Kõikide operatsioonide SOAP ja REST realisatsioonid on lokaalselt teenuse käivitamise juhend on saadaval autori avalikus repositooriumis [22]. Postman testikollektsioonide failid koos testide käivitamise juhendiga ja Swagger dokumentatsiooni fail koos käivitamise juhendiga on esitatud samas repositooriumis.

### 5.1.2.1 Operatsioon addGroup

Grupi lisamise SOAP operatsiooni implemetatsioon võtab sisendina grupi lisamise tüüpi objekti *AddGroupRequestType*. Sisendobjektis sisalduva ligipääsuvõtme token ja institutsiooni registrikoodi alusel kontrollitakse, kas ligipääsuvõtmele on lubatud sisestatud institutsioonile gruppide lisamine. Seejärel kontrollitakse, kas institutsioonis on juba olemas sisestatava nimega grupp. Kui operatsiooni kontrollid on teostatud, siis eduka grupi lisamise korral lisatakse institutsioonile grupp ning tagastatakse lisamisel lisatud grupi info. Ebaõnnestunud sisestamise või kasutajale lubamatu operatsiooni korral tagastatakse kasutajale veateade koos veakoodiga.

REST API võtab operatsiooni jaoks vajalikud sisendid URL parameetritena ning sisuparameetrina lisatava grupi info ja serveri tasemel edastab saadud info SOAP teenusele, mis vastavalt sisendite väärtustele tagastab eduka operatsiooni tulemusel grupi info või ebaõnnestunud operatsiooni korral veateate. Grupi lisamiseks vajalikust infost on institutsiooni registrikood ja ligipääsuvõti esitatavad URL-iga.

REST URL:

- `http://localhost:8080/InstitutionService/webresources/institution/{registryCode}/group?token={token}`

URL parameetritele lisaks tuleb grupi lisamise operatsioonile kaasa anda REST API korral grupi JSON formaadis sisu nagu näidatud järgmiselt:

```
{  
  
  "name": "Mudilaste klaveriklass",  
  
  "location": "Koeru kooli klaveriklass",  
  
  "type": "klaveriring",
```

```
"description": "Klaveritund E, K ja R kell 17.00-19.00"  
}
```

## 5.2 Veebiteenuse dokumentatsioon ja testimine

Veebiteenuse dokumentatsiooni loomiseks on kasutatud Swagger brauseripõhist töövahendit. Loodud dokumentatsioon on võimalik Swagger brauseripõhisesse redigeerijasse [23] importida ning keskkond genereerib faili põhjal interaktiivse dokumentatsiooni. Swagger dokumentatsiooni käivitamise fail ja juhend on koos projektiga autori avalikus repositooriumis kättesaadav [22]. Teenuse funktsionaalsuse käivitumist ja näiteandmete korrektset salvestamist kontrollitakse Postman testikollektsioonidega. Postmani kollektsioonifailid testidega ja Postmani failide käivitamise juhendid on samuti kättesaadavad loodud teenuse projekti repositooriumis [22].

Swagger dokumentatsioon kirjeldab kolme ressursi kõiki meetodeid koos REST URL-idega, sisend- ja väljundparameetritega. Iga ressursiga seotud operatsioonid ja vastavad URL-id on kategoriseeritud vastavalt ressursile ning on genereeritud dokumentatsiooni interaktiivses keskkonnas ka käivitatavad juhul kui lokaalses seadmes on teenus käivitatud. Dokumentatsiooni kolm põhilist andmeressurssi on esitatud järgmisel joonisel.

<b>User</b>	Resources describing user based actions	>
<b>Institution</b>	Resources describing institution related actions	>
<b>Group</b>	Resources describing group related actions	>

Joonis 12. Swagger dokumentatsiooni kolm andmeressurssi

Joonis 13 kujutab grupi alamressursi päringute dokumentatsiooni Swagger interaktiivses redigeerijas, kus iga meetodi puhul on käivitatav näidispäring ning kirjeldatud sisend- ja väljundandmete struktuur ja andmetüübid koos näiteandmetega.

Group		Resources describing group related actions
GET	/institution/{registryCode}/group	Gets all groups in an institution
POST	/institution/{registryCode}/group	Add a new group
GET	/institution/{registryCode}/group/{id}	Gets a specific group in an institution
GET	/institution/group/{groupId}/manager	Gets all managers of a specific group
POST	/institution/group/{groupId}/manager/{id}	Add a new manager to a group
DELETE	/institution/group/{groupId}/manager/{id}	Deletes an group's manager
GET	/institution/group/{groupId}/manager/{id}	Gets specific manager of a specific group
GET	/institution/group/{groupId}/member	Gets all members of a specific group
GET	/institution/group/{groupId}/member/{id}	Gets a specific member of a specific group
POST	/institution/group/{groupId}/member/{id}	Add a new member to a group
DELETE	/institution/group/{groupId}/member/{id}	Deletes an group's member
PUT	/institution/group/{id}	Update an existing group
DELETE	/institution/group/{id}	Deletes an institution's group

Joonis 13. Swagger grupi alamressursside dokumentatsioon

Testimistarkvarana kasutatavas Postman rakenduses on loodud kolme ressursi põhjal kolm testkolleksiooni. Kasutajatepõhine testkolleksioon testib kasutajate lisamise, pärimise, andmete muutmise ja kasutajate kustutamise funktsionaalsust. Institutsioonipõhine testkolleksioon kontrollib institutsioonide registreerimise funktsionaalsust koos pärimise, andmete muutmise ja kustutamise funktsionaalsusega ning testib kasutajate ja institutsioonide vaheliste seoste loomise ja eemaldamise operatsioonide töötamist. Grupipõhine testkolleksioon testib gruppide ja gruppide seoste lisamise ja eemaldamise funktsionaalsust koos kõikide grupipõhiste operatsioonide funktsionaalsuste testimisega.

Iga testkolleksioon loob kõik testitavad ressursid kolleksiooni käivitamise järel ning testide läbimisel kustutab kõik loodud ressursid. Testide korrektse läbimise korral testide käigus loodud ressursid ja seosed on teenuse süsteemist eemaldatud. Testide automaatseks läbimiseks on vaja testkolleksioonidele määrata Postmanis kasutajapõhine testide muutujate keskkond, et testid saaks URL parameetritena kasutatavat infot

ressursse luues salvestada ning päringute tegemisel kasutada. Järgnev joonis kujutab testide käigus keskkonnamuutujate salvestamist.

```
1 var jsonData = JSON.parse(responseBody);
2 postman.setEnvironmentVariable("userid", jsonData.user.id);
```

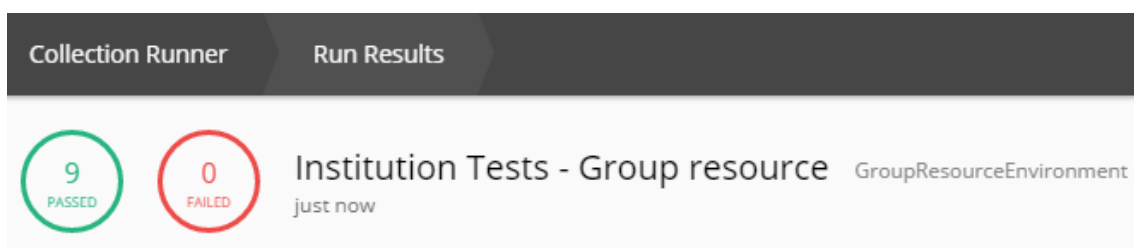
Joonis 14. Postman keskkonnamuutujate salvestamine

Postmani testkeskkonnas käivitatavad testkollektiioonid on implemeteeritud testima teenuse funktsionaalsust teenuseserveri käivitamise järel olukorras, kus teenusesse pole sisestatud ühegi ressursi andmeid. Testkollektiioonide töötamise protsessi lõppedes on säilitatud andmete seisukohalt olukord serveri käivitamise järgse olukorraga sama. Testkollektiioonide valitud operatsioonide tulemusel käivitatakse testid, mis kontrollivad vastuses saadud info korrektsust. Joonis 15 kujutab esimese grupi lisamise järel gruppide listi päringu tulemuse kontrolli.

```
1 var data = JSON.parse(responseBody);
2 tests["Get group 1"] = data.groupList.group.length === 1;
```

Joonis 15. Postman päringu tagastusväärtuse kontrolltest

Postmani rakenduses on kõik kollektiiooni testpäringud käivitatavad üksahaaval ja kogu kollektiioonina. Automaatsete testide kollektiiooni käivitamise tulemusena genereerib Postmani keskkond raporti, milles on andmed päringute õnnestumise ja testimise läbimise kohta. Näidet kollektiiooni käivitamise tulemusel läbitud testiteratsiooni tulemustest kujutab järgmine joonis.



Joonis 16. Postman kollektiioonitestide tulemuste raport

## **6 Tulemuste analüüs**

Käesoleva töö raames valminud veebiteenuse prototüübi eesmärk on pakkuda huvikoolidele tarkvara teenusena rahuldades põhilisi huvikoolide infovajadusi. Töö raames valminud veebiteenus realiseerib osalist funktsionaalsust terviklahendusest, mis loodud töö põhjal on edasiarendatav.

Käesolev peatükk hindab konkurentsianalüüsi ja huvikoolide vajaduste põhjal loodud teenuste eesmärkide täitmist ja konkurentsieeliste realiseerimist. Lisaks analüüsitakse loodud rakenduse alusel edasise arendamisvõimalusi tervikteenuse väljatöötamise eesmärgiga.

### **6.1 Realiseeritud eesmärgid**

Töö teoreetilise taustana tutvustatud veebiteenuste tehnoloogiad, protokollid ja toetavad rakendused annavad ülevaate teenuste realiseerimise põhisammudest analüüsist realisatsioonini. Tutvustatud tehnoloogiate põhjal on võimalik realiseerida terviklik veebiteenus ning veebiteenuse dokumentatsioon ja funktsionaalsuse testid.

Kahe peamise konkurenti süsteemide ja huvikoolide infovajaduste analüüsi tulemusel selgus, et konkureerivad pakkujad täidavad valdava osa huvikoolide infovajadustest pakkudes lisaks klientidele meelepärast lisafunktsionaalsust. Analüüsi tulemusel ka selgus, et vaatamata süsteemide terviklikkusele on konkurentide rakendustes puuduseid. Peamiseks puuduseks, mis konkurentsianalüüsis selgus, oli andmete omanikele andmete mitmekülgse kasutamise võimaldamine pakutavate liidest näol.

Lahendamaks konkurentide puuduseid on realiseeritud antud töö raames veebiteenus, mille operatsioonid võimaldavad huvikoolide kommunikatsiooniprotsesside tuumfunktsionaalsust. Töö tulemusena valminud teenus realiseeris kasutajate, huvikoolide ja huvigruppide lisamise, pärimise, uuendamise ja kustutamise funktsionaalsust. Lisaks eesmärgina seatud sihtrühmale on loodud veebiteenuse funktsionaalsus kasutatav ka alternatiivsete klientide äriprobleemide lahendamisel.

Antud töö eesmärgina seatud huvikoolide vajaduste rahuldamine ning eksisteerivate konkurentide võrdluses konkurentsieeliseid omava prototüüpteenuse analüüsi ja väljatöötamise käigus valminud tarkvarateenuse projekt on eesmärgipärane ning tugeva konkurentsipotentsiaaliga tarkvaralahendus, mis edasisel arendamisel võib kujuneda tugevaks konkureerivaks teenusepakkujaks.

## **6.2 Elektroonilise päevikuteenuse tulevik**

Loodud teenuse edasisel arendamisel on võimalik luua huvihariduse pakkujatele terviklik päevikuteenus koos teenuse operatsioone realiseeriva klientrakendusega. Samuti on loodud teenuse komponente võimalik realiseerida teistsuguse taustaga äriprobleemide lahendamisel pakkudes konkurentsi erinevates valdkondades. Jätkates teenuse arendamist on võimalik teenuse mooduleid realiseerida näiteks üldhariduskoolide, ülikoolide, ettevõtete või ühingute kommunikatsioonieesmärkide lahendamisel.

## Kokkuvõte

Käesoleva bakalaureusetöö põhieesmärk oli analüüsida huvihariduse kommunikatsiooni- ja päevikuprotsesside hetkeseisu ja vajadusi ning analüüsida ja realiseerida kogutud info põhjal elektroonilise päeviku veebiteenuse prototüüp. Nagu töös selgus, on töö kirjutamise hetkeks huvikoolidel sarnaselt üldhariduskoolidele tekkinud vajadus elektrooniliste kommunikatsioonitööriista järgi, mis võimaldab huvikooli korraldusliku info vahetamist, liikmete ja kontaktandmete haldamist.

Hetkeolukorra analüüsi tulemusena selgus, et huvikoolide infovajaduste hulka kuulub huvikoolide info talletamine, selle edastamine ning liikmenimekirjade haldamine. Avaliku info põhjal tehtud analüüsi tulemusel selgus, et kuigi suur hulk Eestis tegutsevaid huvikoole on oma kommunikatsiooniprotsessid automatiseerinud mõne spetsiaalse infosüsteemi või rakenduse abil, on veel selliseid huvikoole, kus kommunikatsioon toimub meili teel ning liikmeinfot talletatakse manuaalselt hallatavates nimekirjades.

Käesoleva töö raames valmis prototüüpteenus, mille funktsionaalsus realiseerib töö analüüsis püstitatud nõudeid ning täidab sõnastatud huvikoolide infovajadused. Loodud teenuse struktuur võimaldab laiendada olemasolevat funktsionaalsust, et arendada välja päevikupidamise tervikteenus või vajadusel lisafunktsionaalsus teistsuguse suunitlusega äriprotsesside võimaldamiseks. Loodud teenuse funktsionaalsust dokumenteerib Swagger rakendusega loodud interaktiivne dokumentatsioon ning teenuse funktsionaalsuse kontrollivad Postman rakenduse testkollektsioonid.

Lõputöö andis autorile arvestataval määral uusi teadmisi veebiteenuste tehnoloogiate ning teenustele orienteeritud tarkvaraarenduse põhimõtetest. Kuna töös lahendatud probleem on reaalne eksisteeriv probleem, on käesolev lõputöö potentsiaalne alusdokument tervikliku ärilisel eesmärgil realiseeritava teenuse arendamiseks ja turustamiseks.



## Kasutatud kirjandus

- [1] Liiklane, L., Kesa, M. Arvutisõnastik, 2006. [WWW]  
<http://www.keeleveeb.ee/dict/speciality/computer/> (22.05.2017)
- [2] Tulach, J. Practical API Design. New York: Apress, 2008, 15-16.
- [3] Beal, V. Web services. [WWW] [http://www.webopedia.com/TERM/W/Web\\_Services.html](http://www.webopedia.com/TERM/W/Web_Services.html)  
(22.05.2017)
- [4] Why use Web services. [WWW]  
<http://www.vkinfotek.com/webservice/whycreatewebservice.html> (22.05.2017)
- [5] DotGNU's Vision for Webservices. [WWW] <http://www.gnu.org/software/dotgnu/webservices.html> (22.05.2017)
- [6] Barry, D. K. Service Architecture. [WWW] [http://www.service-architecture.com/articles/web-services/web\\_services\\_description\\_language\\_wsdl.html](http://www.service-architecture.com/articles/web-services/web_services_description_language_wsdl.html)  
(22.05.2017)
- [7] Mueller, J. Understanding SOAP and REST Basics And Differences, 2013. [WWW]  
<http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/> (22.05.2017)
- [8] Rouse, M. Software as a Service (SaaS), 2016. [WWW]  
<http://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service> (22.05.2017)
- [9] Snider, E. Six data privacy questions to ask SaaS financial software vendors, 2012.  
[WWW] <http://searchfinancialapplications.techtarget.com/feature/Six-data-privacy-questions-to-ask-SaaS-financial-software-vendors> (22.05.2017)
- [10] What Is Swagger? [WWW] <http://swagger.io/docs/specification/what-is-swagger/>  
(22.05.2017)
- [11] Postman documentation. [WWW] <https://www.getpostman.com/docs/> (22.05.2017)
- [12] Eesti Hariduse Infosüsteem. [WWW] <http://www.ehis.ee> (22.05.2017)
- [13] VEMI. [WWW] <https://vemi.ee/wp/> (22.05.2017)
- [14] Sportlyzer. [WWW] <https://www.sportlyzer.com/et/> (22.05.2017)
- [15] Sportlyzer [WWW] <http://www.tuletrenni.ee/search> (22.05.2017)
- [16] Semrush. [WWW] <https://www.semrush.com/info/> (22.05.2017)
- [17] Similarweb. [WWW] <https://www.similarweb.com/> (22.05.2017)
- [18] Konkurentsianalüüsi olemus. [WWW]  
<http://konkurentsianaluus.weebly.com/konkurentsianaluumlumlsi-olemus.html>  
(22.05.2017)
- [19] Stuumium. [WWW] <https://stuumium.com> (22.05.2017)
- [20] eKool. [WWW] [https://ee.ekool.eu/index\\_et.html](https://ee.ekool.eu/index_et.html) (22.05.2017)
- [21] Excel. [WWW] <https://products.office.com/et-ee/excel> (22.05.2017)

- [22] Puntso, S. Github repositoarium, 2017. [WWW]  
<https://github.com/sanderpuntso/institutionserviceproject> (22.05.2017)
- [23] Swagger Editor. [WWW] <http://editor.swagger.io/#/> (22.05.2017)