

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Arvutiteaduse instituut

ITV40LT  
Arsenti Morozov 134920

# **MOODLE ÕPIKESKKONNA SINGLE-PAGE RAKENDUS HINNETERAPORTI JAOKS**

Bakalaureusetöö

Juhendaja: Ago Luberg  
MSc  
Assistent

Tallinn 2016

## **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: Arseni Morozov

23.05.16

## Annotatsioon

Käesoleva bakalaureuse töö eesmärgiks on luua *single-page* rakendus Moodle'i õpikeskkona jaoks. Arendatav rakendus peaks lihtsustama hinnete haldamist ning parandama kasutajamugavust Moodle'i õpikeskkonna kasutamisel. Peamiseks probleemiks Moodle'i versioonis 3.0.3. olemasoleva funktsionaalsuse juures on navigeerimise keerukus ja kasutajamugavuse langus haldava andmete koguse suurenemisel. Töö tulemusena valmiv rakendus kasutab arhitektuuri, mis suhtleb andmebaasiga läbi Moodle'i poolt pakutava API-t. Loodud arhitektuur võimaldab kasutada veebiteenuse funktsioone *single-page* rakenduse laiendamiseks. Loodud rakendus vastab käesoleva töö raames püstitatud eesmärgile. Käesolev bakalaureuse töö sisaldab ka Moodle versiooni 3.0.3. hinneteraportite funktsionaalsuse uurimust.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 34 leheküljel, 7 peatükki, 11 joonist.

## **Abstract**

### **Moodle single-page application for grader report**

The main aim of present thesis is to create a single page application for Moodle. Moodle is a learning platform designed to provide educators, administrators and learners system to create personalised learning environments. The application should change the way of grade organisation and make it more easy to use. The main problems in grade report Moodle 3.0.3 version is its difficult navigation and weak user comfort, when result data is rapidly increasing.

The thesis is divided in 3 main parts. The first part describes Moodle already realized tools to organise and administrate user results. Main focus of the first part is to find out weak aspects of Grader report and use them for developing single-page application.

The second describes the first iteration of the application with its own architecture. All functions are written in JavaScript and uses AJAX queries to GET and POST information from Moodle database. In the first iteration the application was made as a moodle plugin and it wasn't single-page.

The third part describes second iteration application. In the second iteration application becomes single-page and totally external from moodle. During the second iteration there was created a new architecture which changed the way of communicating with Moodle. Relying on the architecture Moodle API was taken as a way to communicate with Moodle. As a development framework was chosen Angular JS. Also created single-page application uses MVC pattern and can be expanded in the future. Developed application corresponds to the main aim of the thesis.

The thesis is in Estonian language and contains 34 pages of text, 7 chapters, 11 figures.

## Lühendite ja mõistete sõnastik

Moodle	<i>Modular Object-Oriented Dynamic Learning Environment</i> on tasuta lähtekoodiga õpikeskonna platvorm
Plugin	Tarkavaramoodul, mis lisab suuremale süsteemile teatud omaduse või teenuse
Report	Raport ehk hinnete haldamiseks Moodle poolt pakutav lahend.
WAMP	Lühend sõnadest <i>Windows, Apache, MySQL, PHP</i> . WAMP on veebirakendusele mõeldud vabavaraline keskkond.
PHP	<i>PHP: Hypertext Preprocessor</i> on skriptimiskeel
SQL	<i>Structured Query Language</i> struktureeritud päringukeel
PHPmyAdmin	PHP-s kirjutatud töövahend, mis võimaldab hallata MySQL andmebaasi
Apache	Vabavaraline ja maailmas levinuim veebiserveri tarkvara
HTML	Hüpertexti märgistuskeel
AJAX	Lühend <i>Asynchronous JavaScript And XML</i> . Ajax on kogum omavahel seotud veebiarenduse tehnikaid interaktiivse veebirakenduse arendamiseks
Javascript	On objektorienteeritud programmeerimiskeel, mida kasutatakse peamiselt veebilehtede skriptimiseks
JSON	<b>JavaScript Object Notation</b> on lihtsustatud andmevahetusvorming, mis põhineb JavaScript programmeerimiskeele alamhulgal
API	<i>Application Program Interface</i> on tarkvara rakendamise protokollide, reeglite ja vahendite kogum
AMF	<i>Action Message Format</i> on binaarse andmete vahetamise protokoll
SOAP	<i>Simple Object Access Protocol</i> ehk lihtne objektipöördusprotokoll
REST	<i>Representational State Transfer</i> on tarkvaraarhitektuuri stiil, mida kasutatakse hüpermeedia haiussüsteemide valdkonnas.
XTML-RPC	<i>Extensible Markup Language Remote Procedure Call</i> on kaugprotseduurikutse protokoll
Angular JS	JavaScript raamistik

SPA	<i>Single-page application</i> veebirakendus mis kasutab üht HTML dokumenti kogu rakenduse käigus.
MVC	<i>Model-view-controller</i> on arhitektuuriline muster, mis lihtsustab rakenduse laiendamist.
CSS	<i>Cascading Style Sheets</i> on keel, milles märgitakse üles XML ja SGML- keelsete failide kujundust
XML	<i>Extensible Markup Language</i>
MVVM	<i>Model-View-ViewModel</i>

## Sisukord

Autorideklaratsioon .....	2
Annotatsioon.....	3
Abstract Moodle single-page application for grader report.....	4
Lühendite ja mõistete sõnastik .....	5
Sisukord.....	7
Jooniste loetelu .....	8
1 Sissejuhatus .....	9
2 Moodle'i installeerimine ja paigaldamine. ....	11
2.1 WAMP.....	11
2.2 Moodle'i installeerimine.....	12
3 Moodle'i hinneteraporti loomine.....	13
4 Moodle'i hinneteraportite funktsionaalsuse uurimine.....	16
4.1 Grader report'i funktsionaalsuse uurimine .....	18
5 Esimene iteratsioon .....	22
6 Teine iteratsioon. ....	27
7 Kokkuvõte .....	32
Kasutatud kirjandus .....	33

## Jooniste loetelu

Joonis 1. <i>Wampserver</i> 'i menüü. ....	12
Joonis 2. Faili <code>access.php</code> sisu näide.....	13
Joonis 3. Faili <code>version.php</code> sisu näide. ....	14
Joonis 4. Grader report'i keele faili demonstratsioon.....	15
Joonis 5. Grader report'i näide. ....	20
Joonis 6. Demonstratsioon andmete kuvamise horisontaal kerimise vältimiseks. ....	22
Joonis 7. AJAX päringu peal käivitatav kood, tabelite kuvamiseks. ....	24
Joonis 8. Esimese iteratsiooni <i>plugin</i> 'i arhitektuur. ....	25
Joonis 9. AJAX päring hinnete muutmisel.....	25
Joonis 10. MVC skeem.....	28
Joonis 11. Teise iteratsiooni arhitektuur.....	29



## 1 Sissejuhatus

Paljud ülikoolid kasutavad alates 20. sajandi lõpust peamiseks õpinguprotsesside automatiseerimise platvormiks Moodle'it<sup>1</sup>. Tõnis Tartes [1] kirjeldab Moodle'it (*Modular Object-Oriented Dynamic Learning Environment*), kui avatud lähtekoodiga tasuta e-õppe rakendust, mis on üldtuntud e-kursuste haldamissüsteemina. Moodle'i keskkonda on maailmas installeeritud rohkem kui 75 000 serverile, mis hõlmavad rohkem kui 86 miljonit kasutajat ja rohkem kui 9 miljonit kursust [2].

Moodle'is on väga palju võimalusi ja ka palju innovaatilisi lahendusi. Võimaluste realiseerimiseks on kasutusele võetud laiendused ehk *plugin*'id, mis sarnanevad Lego plokkidega. *Plugin*'id iseenesest on tarkvaramoodulid, mis võimaldavad süsteemi laiendada ehk teisiti öeldes nende kasutades on võimalik suuremale süsteemile lisada teatud omaduse või teenuse [3]. Moodle on mahuka dokumentatsiooniga rahvusvaheline süsteem. Tänu suure Moodle'i kasutajate kogukonnale on loodud palju erinevaid laiendusi, mis parandavad Moodle'i funktsionaalsust vastavalt vajalikele nõudmistele, mille tõttu Moodle omab suurepäraseid võimalusi püstitatud eesmärkide saavutamiseks.

Käesoleva töö eesmärgiks on Moodle õpikeskkonna *single-page* rakenduse loomine hinneteraporti jaoks, kuna üheks peamiseks Moodle'iga seotud probleemiks on hinnete haldamine tulemuste tabelites. Antud probleem esineb hinnete haldamisel ainetes, kus on palju tudengeid ja palju hindeid, mille tulemusena muutub hinnetetabelis navigeerimine keeruliseks ja ebamugavaks ehk teisiti öeldes, andmete suurendamisega kasutajamugavus langeb. Probleemi aktuaalsus seisneb selles, et tänapäeval nii Tallinna Tehnikaülikooli kui ka Tartu Ülikooli õppejõud kasutavad aktiivselt õppetöö organiseerimisel Moodle'it. Selle õppekeskkonna üheks eeliseks on suure osalejate arvuga kursuste administreerimise võimalus, kuid sellel korral hinnete haldamine on väga keerukas ning ebamugav, mille põhjusel õppejõud tihti keeldub selle funktsionaalsuse kasutamisest.

---

<sup>1</sup><https://moodle.com/>

Vajalikku *single-page* rakenduse loomine vajab suurt Moodle'is olemasoleva funktsionaalsuse uurimist, kuna Moodle'is on juba realiseeritud kuus raportit hinnete organiseerimiseks, kuid kahjuks ühtegi neist ei ole mugav kasutada suurte andmete hulga korral. Lähtudes püstitatud eesmärgist käesoleva bakalaureusetöö käigus on teostatud ülevaade olemasolevatest hinnete raportitest ning Grader report'i funktsionaalsuse mahukas uurimus. Uurimuse käigus tehtud järelduste alusel oli loodud iseseisev raport, mille ülesandeks on hinnete mugav organiseerimine. Raportist on antud hetkel valminud teine iteratsioon, kuid käesolevas töös on kirjeldatud ka esimene iteratsioon.

## 2 Moodle'i installeerimine ja paigaldamine.

Moodle on kergesti installeeritav ja madalate nõuetega tarkvara. Hetkel olemasolev Moodle'i stabiilne versioon 3.0.3 vajab, et serveris installeeritud tarkvara vastaks vähemalt järgmistele versioonidele: PHP 5.4.4, Microsoft SQL Server 2008, PostgreSQL 9.1, MySQL 5.5.31 või Oracle 10.2 [9] . Sellest tuleneb, et iga inimene, kellel on olemas sellistele nõudetele vastav tarkvara, võib omas serveris Moodle'it installeerida.

### 2.1 WAMP

Moodle'i raportite uurimuse läbiviimiseks ja enda poolt loodud raporti realiseerimiseks on vajalik arenduskeskkond. Lähtudes minimaalsetest nõuetest ja rahalistest võimalustest otsustasin kasutada WAMP platvormi. Seega esimeseks sammuks Moodle'i installeerimisel oli WAMP platvormi osade installeerimine ja paigaldamine. WAMP on lühend, mis tähendab komplekti, mis koosneb operatsioonisüsteemist ja kolmest vabavaralistest tarkvara komponentidest, mis võimaldavad luua dünaamilisi veebilehti või servereid [6] .

- Windows - operatsiooni süsteem
- Apache – veebiserver
- Mysql – andmebaas
- PHP – programmeerimiskeel (kuna Moodle baseerub selle keele peale)

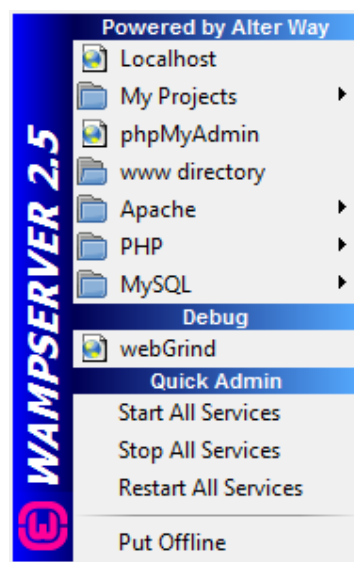
WAMP-i installeerimiseks on olemas mitu erinevat võimalust. Üks nendest on lihtsalt eraldi tarkvara osade installeerimine ning nende konfigureerimine vastavalt vajadustele. Teine võimalus on *wampserver*'i installeerimine [6] . *Wampserver*<sup>2</sup> on WAMP-i tarkvarade komplekt, mis lisaks sisaldab *Phpmyadmin*'i [7] . *Phpmyadmin* on PHP-s kirjutatud töövahend, mis võimaldab hallata *MySQL*'i kasutades veebi brauseri [8] .

---

<sup>2</sup> [https://en.wikipedia.org/wiki/LAMP\\_\(software\\_bundle\)#Variants\\_and\\_equivalents\\_on\\_other\\_platforms](https://en.wikipedia.org/wiki/LAMP_(software_bundle)#Variants_and_equivalents_on_other_platforms)

*Phpmyadmin*'i abil saab luua ja eemaldada andmebaase, tabeleid ning nende väljudes sisalduvaid andmeid. Kasutades *Phpmyadmin*'i samuti saab käivitada ükskõik, millist SQL'i käsku. SQL on faktikogumite ja nende omavaheliste seostega töötamiseks mõeldud arvutikeel [17]. *Phpmyadmin*'i ei ole otseselt vaja selleks, et Moodle korrektselt töötaks, kuid see lihtsustab tegevusi, mis on seotud Moodle andmebaasiga. Kuna WAMP platvorm kuulub vabatarkvarasse, siis *wampserver* on ka vabatarkvara.

*Wampserver*'i installeerimine on kirjeldatud ametlikul *wampserver*'i veebilehel, kust võib seda ka alla laadida [7]. Käesolevas töös oli kasutatud PHP versioon 5.5.12, MySQL versioon 5.6.17 ning Apache versioon 2.4.9, mida pakkus *wampserver 2.5*. Installeeritud versioonid olid edukalt aktsepteeritud Moodle'i poolt. Nagu näidatud Joonis 1 pärast installeerimist ja käivitamist tekib vasakul allpool *wampserver*'i ikoon ja sellele vajutamisel tekib menüü, mille abil on võimalus kogu AMP-i koos *Phpmyadmin*'iga konfigureerida ning tarkvara sisse ja välja lülitada.



Joonis 1. *Wampserver*'i menüü.

## 2.2 Moodle'i installeerimine

Eelmised sammud olid ettevalmistavad Moodle tarkvara installeerimiseks. Ametlikul Moodle'i veebilehel olevas dokumentatsioonis on olemas inglisekeelne installeerimisjuhend [9]. Moodle'i installeerimise käigus tekkis probleem sellega, et minu poolt loodud õigused Moodle'i andmebaasi instantsi kasutajale olid mittetäielikud, kuid Moodle iseenesest ei andnud mingid teadet, kus oleks see mainitud.

### 3 Moodle'i hinneteraporti loomine.

Moodle'it on võimalik laiendada kasutades *plugin*'aid ehk laiendusi. Nagu eelnevalt oli juba mainitud, laiendus on programmi osa või tarkvaramoodul, mis lisab teisele tarkvarasüsteemile funktsionaalsust [3] . Uue hinneteraporti loomisel tekib arusaam Moodle'i poolt pakutava raportite loomise algoritmist ning laienduste arhitektuurist. Moodle pakub ametliku samm-sammulist juhendit uue hinneteraporti loomiseks [10] , kuid kahjuks juhend on kirjutatud Moodle 1.9 versiooni jaoks ning praegusel ajal juhendis olev informatsioon on vananenud. Käesolevas töös on vormistatud uus juhend, mille järgimine võimaldab luua uut hinneteraportit.

Kõik olemasolevad hinneteraportid asuvad kaustas `moodle/grade/report` ning uue raporti kaust peab olema loodud samuti sinna. Esimesena on vaja luua fail nimega `access.php`. Faili sisuks peab olema õiguste nimekiri, mida kehtestatakse uues raportis. Fail `access.php` peab asuma kaustas `db`, mis omakorda peab asetsema `moodle/grade/report/{uusreport}` kaustas. Joonis 2 on toodud näidis `access.php` faili sisust.

```
defined('MOODLE_INTERNAL') || die();

$capabilities = array(
    'gradereport/{uusreport}:view' => array(
        'riskbitmask' => RISK_PERSONAL,
        'captype' => 'read',
        'contextlevel' => CONTEXT_COURSE,
        'archetypes' => array(
            'teacher' => CAP_ALLOW,
            'editingteacher' => CAP_ALLOW,
            'manager' => CAP_ALLOW
        )
    )
);
```

Joonis 2. Faili `access.php` sisu näide.

Teine fail, mis on nõutud Moodle poolt korrektse laienduse loomiseks on `version.php`. Faili `version.php` on vaja luua uue raporti kausta ehk `moodle/grade/report/{uusreport}`. `version.php` faili sisuks on versiooniga seotud informatsioon ning laiendamise nime sätestamine. Ametlikus Moodle'i poolt pakutavas juhendis versiooni faili `$plugin->version` tulemuseks on 2007081000 [10] . Need

numbrid tähendavad versiooni uuendamise kuupäeva formaadis YYYYMMDD, kus D - tähendab päeva (day), M – tähendab kuud (month), Y – tähendab aastat (year) ning kaks viimast numbri X on versiooni täpsustamiseks. Versioon 2007081000 on vananenud ja Moodle 3.0.3 loobub selle kasutamisest. Joonis 3 on kujutatud näide version.php failist.

```
defined('MOODLE_INTERNAL') || die();
$plugin->version = 2015111600;
$plugin->requires = 2015111000;
$plugin->component = 'gradereport_{uusreport}';
```

Joonis 3. Faili version.php sisu näide.

Kolmandaks failiks luuakse hinneteraporti keele faili. Keele fail peab asuma kaustas /grade/report/{uusreport}/lang/en\_utf8 ning faili nimeks peab olema gradereport\_{uusreport}.php. Ametlikus juhendis pakutakse kaks parameetrit keele faili sisuks: \$string['modulename'] ja \$string['user:view']. Esimene defineerib raporti nime ning teine on raportite listis oleva nime defineerimiseks. Tegelikult Moodle'i poolt realiseeritud raportite keele failid sisaldavad palju rohkem informatsiooni, kui pakutavas ametlikus juhendis. Joonis 4 on toodud näide Grader report'i keele failist.

```
$string['ajaxclicktoclose'] = 'Click this box to remove it';
$string['ajaxerror'] = 'Error';
$string['ajaxfailedupdate'] = 'Unable to update [1] for [2]';
$string['ajaxfieldchanged'] = 'The field you are currently editing has
changed, would you like to use the updated value?';
$string['ajaxchoosescale'] = 'Choose';
$string['eventgradereportviewed'] = 'Morozov report viewed';
$string['grader:manage'] = 'Manage the grader report';
$string['grader:view'] = 'View the grader report';
$string['pluginname'] = 'Morozov report';
$string['preferences'] = 'Morozov report preferences';
$string['summarygrader'] = 'A table with the names of students in the first
column, with assessable activities grouped by course and category across the
top.';
$string['useractivitygrade'] = '{$a} grade';
$string['useractivityfeedback'] = '{$a} feedback';
```

Joonis 4. Grader report'i keele faili demonstratsioon.

Neljandaks failiks on vaja luua index.php faili. Fail index.php prindib välja HTML-i sisu, mida kuvatakse veebilehel. Juhend pakub näidist index.php failist ning sellega lõpeb uue hinneteraporti loomine.

Raporti loomise käigus oli otsustatud kasutada olemasolevat Grader report'i, sest peamine uurimus ja rakenduse arendamine on sellega tihedalt seotud. Uue Grader report'i kopeerimise realiseerimiseks oli muudetud failide sisu vastavalt juhendile ning samuti oli muudetud Grader report'is juba olemasolevate failide sisu. Lõpuks oli valmistatud Grader report'iga identne raport, mida kasutati funktsionaalsuse uurimise jaoks ning esimese iteratsiooni käigus.

## 4 Moodle'i hinneteraportite funktsionaalsuse uurimine

Antud töö raames püstitatud eesmärgi saavutamiseks oli vaja uurida, missugune olemasolev Grader report'i funktsionaalsus on otstarbekas säilitada, milline tuleb välja võtta või parandada ning, missugust funktsionaalsust oleks vaja lisada maksimaalse võimaliku kasutamise mugavuse saavutamiseks. Püstitatud eesmärgist lähtudes oli uuritud ja hinnatud Grader report'is juba realiseeritud funktsionaalsus, millest suurem osa oli edaspidi muudetud. Samuti uurimus näitas, et Grader report'is puudub üldse hinnete statistika kogumise võimalus, mis lihtsustaks kasutajate jaoks hinnete ülevaadet ja analüüsi.

Moodle on hästi läbimõeldud keskkond stabiilse süsteemiga. Moodle omab suurt funktsionaalsuse hulka, mis võimaldab korrektselt tulemusi üle vaadata, sisestada ja muuta. Moodle versioonis 3.0.3 on hetkel realiseeritud 6 raportite laiendust:

- Grader report – võimaldab jälgida kursuse osalejaid, vaadata üle nende põhiandmeid ja osalejate poolt kursuse käigus saadud hindeid. Kuna Grader report'i hindamine ja uurimine on antud töö eesmärgi saavutamise aluseks, siis raporti funktsionaalsust kirjeldatakse laiemalt natuke hiljemalt antud töö samas peatükis.
- Grade history<sup>3</sup> – võimaldab jälgida kõike toiminguid, mis on tehtud hinnetega. Teisiti öeldes Grade history raport on hinnete muutmise, lisamise ning kustutamise logide vaatamise vahend.
- Outcomes report<sup>4</sup> – aitab õppejõududel kontrollida ja jälgida kursuse osalejate tulemusi kasutades *outcome* vahendid.
- Overview report<sup>5</sup> – võimaldab vaadata, missugustele kursustele on kasutaja ennast registreerinud. Samuti sealt saab vaadata erinevate kursuste koguhinnet ühe tudengi kohta võib vaadata kasutajate tulemusi ühe kaupa.

---

<sup>3</sup> Grade history dokumentatsioon: [https://docs.moodle.org/30/en/Grade\\_history](https://docs.moodle.org/30/en/Grade_history)

<sup>4</sup> Outcomes report dokumentatsioon: [https://docs.moodle.org/20/en/Outcomes\\_report](https://docs.moodle.org/20/en/Outcomes_report)

<sup>5</sup> Overview report dokumentatsioon: [https://docs.moodle.org/29/en/Overview\\_repo](https://docs.moodle.org/29/en/Overview_repo)



- Single view<sup>6</sup> –Single review näitab valitud kursuse, kas ühe tudengi kõiki hindeid või kõikide tudengite ühte hinnet ning lisaks annab võimalust hindeid muuta. Võrreldes Grader reportiga hinnete muutmine ei ole dunaamiline ning iga hinde muutmine nõuab salvestamist.
- User report<sup>7</sup> – võimaldab vaadata kasutajaga seotud informatsiooni.

---

<sup>6</sup> Single view dokumentatsioon: [https://docs.moodle.org/29/en/Single\\_view](https://docs.moodle.org/29/en/Single_view)

<sup>7</sup> User report dokumentatsioon: [https://docs.moodle.org/30/en/User\\_report](https://docs.moodle.org/30/en/User_report)

## 4.1 Grader report'i funktsionaalsuse uurimine

Nagu eelnevalt oli mainitud kõige rohkem uuriti ja hinnati antud töö raames ja Grader report'i, kuna antud raport on peamine vahend, mida Moodle pakub õppejõududele tulemuste vaatamiseks. Selle raporti kaustas README failis asub järgnev tekst: 'The default gradebook interface, that teachers will use'. Huvi Grader report'i uurimise vastu on põhjendatud sellega, et raporti kasutamine muutub väga keeruliseks andmete hulga suurenemisel ning sellega, et hetkel realiseeritud funktsionaalsus ei võimalda kasutajaliidese ebamugavuse tõttu raportit efektiivselt kasutada.

Esimeseks sammuks oli uurida ja hinnata Grader report'is juba realiseeritud funktsionaalsust. Esimesest pilgust on arusaadav, et Grader report baseerub tavalise tabeli põhjal ning suure tabeli puhul kerimine (nii horisontaalselt kui vertikaalselt) on üks nendest aspektidest, mida antud juhul ei saa vältida. Kerimise kasutamine on samuti kirjeldatud esimestes punktides ametliku Grader reporti dokumentatsioonis [4]. Kerimine iseenesest ei põhjusta suurt probleemi, kui tegemist on väikese andmete hulgaga. Kuid suurema andmete koguse juures teeb kerimine hinnete vaatamise ja sisestamise keeruliseks ja ajamahukaks ning Moodle'i kasutamisel nutiseadmetes võimatuks. Uurimus näitas, et 150 kasutaja 15 sooritatud testi tulemuste vaatamine muutub Grader report'i kasutamist keeruliseks.

Kerimise lihtsustamiseks Grader report'is on realiseeritud sorteerimise funktsionaalsus. Sorteerimine on võimalik vaid veergude järgi. Vajutades veeru nimele toimub valitud veeru mittedünaamiline sorteerimine. Mittedünaamilisus käesoleva töö raames tähendab veebilehe uuendamist iga teostatud tegevuse järel. Sorteerimise käigus aga andmete kogus ei vähene vaid andmeid lihtsalt sorteeritakse valitud parameetri järgi, mis omakorda vajab kerimist vajalikke andmete leidmiseks.

Grader report'is on realiseeritud otsimise funktsionaalsus, mis peaks olukorda parandama. Tegelikult otsimise funktsionaalsus on samamoodi mittedünaamiline. Otsingut on võimalik teostada nime ja perekonnanime esimese tähe järgi. Iga uue otsinguparameetri lisamine vajab lehe värskendamist. Otsingu tulemusel ning pärast värskendamist ilmub tabel, kuid selleks et leida antud tabelist vajalikku testi ning vastavat hinnet on samuti vaja kerida lehte horisontaalses suunas.

Moodle võimaldab kasutada kategooriaid, selleks et hindeid struktureerida. Kategoorias sisalduvate andmete läbi vaatamiseks tuleb kategooria avada, mis omakorda kutsub esile lehe uuendamise. Antud funktsionaalsuse kasutamine vähendab horisontaalses suunas kerimise vajadust, kuid hinnete struktureerimine ja arusaam testide organiseerimise põhimõttest nõuab kogemust ja aegsasti teadmist kursuse jooksul tulenevate testide arvust.

Ametliku Grader report'i dokumentatsioonis on pakutud kolm võimalust kategooriate ja nendes sisalduvate andmete kuvamiseks [4]. Esimese võimaluse juures näidatakse kõike hindeid ja peidetakse ära kategooriaid. Teise võimaluse korral kuvatakse vaid kategooria keskmist tulemust. Kolmas võimalus kujutab endast nii kõikide kategoorias olevate tulemuste kuvamist, kui ka keskmist kategooria tulemuse näitamist. Kõik kategooriatega teostatavad tehinguid on mittedünaamilised ja toovad kaasa veebilehte värskendamise.

Grader report võimaldab samuti vaadata kasutaja põhiandmeid vajutades kasutajanime peal. Vajutamisega laetakse uus veebileht samas aknas, kus võib vaadata kasutaja andmeid. Eelneva veebilehe sätteid nt kategooria, sorteerimine ning otsinguparameetrid aga säilitatakse sellisel kujul, nagu nad olid enne kasutaja andmete veebilehele üleminekut.

Samuti üheks Grader report'i funktsiooniks on hinnete muutmine. Hinnete muutmine on ainuke dünaamiliselt töötav osa Grader report'is. Tuleb aga silmas pidada, et funktsionaalsuse aktiveerimiseks on vaja lubades AJAX-i kasutamine. Ajax on kogum omavahel seotud veebiarenduse tehnikaid interaktiivse veebirakenduse arendamiseks [11]. Topelt klõpsamisel tulemuste peale ilmub võimalus neid muuta ja „Enter“ vajutades toimub sisestatud hinnete dünaamiline ehk ilma veebilehe uuendamiseta salvestamine. Nimetatud funktsionaalsuse realiseerimiseks kasutatakse AJAX päringut, mis omakorda kutsub `update_final_grade` funktsiooni.

Joonis 5 on toodud Grader report'ist tehtud pilt, kus punase ristkülikuga on ära märgitud mittedünaamiline funktsionaalsus ja rohelise ristkülikuga olemasolev dünaamiline funktsionaalsus. Jooniselt on samuti näha horisontaalse kerimise ribasid.

First name :  
All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z  
Surname :  
All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

		Test1	Test2	Test3	Test4	Test5	Test6	Test7	Test8
Surname	First name	Email address							
asd2	asd2	asd2@asd.ee	2.00	-	-	-	-	-	-
asd3	asd3	asd3@asd.ee	3.00	-	-	-	-	-	-
asd	asd	asd1@asd.ee	2.00	-	-	-	-	-	-
Overall average			3.00	-	-	-	-	-	-

Joonis 5. Grader report'i näide.

Grader report'i administreerimiseks ja konfigureerimiseks on võimalik kasutada *Preferences: Grader report*. Sõna *preference* inglise keeles tähendab "eelistus". Grader report'i juhtimiseks kasutatavaid eelistused jagunevad kolmeks põhigrupiks. Esimene põhigrupp on „*Show/hide toggles*“, mis juhib informatsiooni kuvamise ja peitmisega seotud seadistusi. Kasutades antud eelistustes olemasolevaid häälestusi võib näiteks kasutajaprofiilide piltide või ikoonide kuvamist juhtida. Teiseks grupiks on eelistused nimega „*Special rows*“, mis juhib kuvatavate tulemuste vormiga seotud spetsiaalseid sätteid. „*Special rows*“ sätete abil võib määrata tulemuste ühikud, kohtade arvu pärast koma jne. Kolmas grupp nimetusega „*General*“ juhib üldiste sätete seadistamist. Üheks väga tähtsaks sätteks, mida võimaldab „*General preference*“ kasutamist seadistada on AJAX'i lubamine. Nagu eelnevalt oli mainitud, AJAX'i lubamine Grader report'is on vajalik selleks, et lubada hinnete muutmine dünaamiliselt.

Kõik loetletud funktsionaalsus on hetkel realiseeritud Moodle versiooni 3.0.3 Grader report'is ning võetud aktiivse kasutusele hinnetega tehingute teostamisel ja tulemuste kuvamisel. Uurimuse käigus autor veendus, et Moodle'is puudub üldse funktsionaalsus tulemustega seotud statistika kogumiseks.

Statistika võimaluse puuduse tõttu kaob ära visuaalne ülevaade kasutajate tulemustest ja kursuse seisundist. Statistika saamise jaoks on Moodle poolt pakutud andmete

eksportimise võimalus Excel vormi dokumendina. Allalaetavas Excel dokumendis on andmed esitatud Grader report'i tabeli kujul.

Teostatud uurimus näitas, et iseenesest Grader report on vahend, mis ideaalselt sobib väikeste andmete hulga vaatamiseks ja töötlemiseks. Suurema andmete koguse ilmumisel muutub Grader report'i kasutamine ebamugavaks ja ajamahukaks ning sellel juhul kursuse korralik administreerimine nõuab kogemust, kindlat planeerimist ja lisa teadmisi. Statistika kogumise võimaluse puudumine takistab kursuse seisundi visuaalse ülevaade saamist.

Olemasoleva informatsiooni põhjal võib välja tuua, et raporti funktsionaalsuse uurimise käigus sai välja selgitatud kaks peamist Grader report'i puudust. Esimeseks puuduseks võiks nimetada Grader report'is hetkel realiseeritud funktsionaalsuse mittedünaamilisust. Teiseks puuduseks on vähese statistika näitamise võimalus.

## 5 Esimene iteratsioon

Kõigepealt on vaja aru saada, mis oleks parem viis kasutajamugavuse suurendamiseks. Kuna funktsionaalsuse uurimuse käigus oli välja toodud, et suurema andmete koguse korral muutub lehe kerimine ebamugavaks, siis esiteks oleks vaja vabaneda horisontaalsest kerimisest. Üks võimalus, kuidas võiks horisontaalset kerimist vältida on näidatud Joonis 6.

Nimi Perekonnanimi

Kategooria	Tulemus
Test1	5.00
Test2	4.00
Test3	3.00
Test4	2.00
Test5	2.00
Test6	5.00
Test7	3.00
Test8	2.00
Test9	3.00

Joonis 6. Demonstratsioon andmete kuvamise horisontaal kerimise vältimiseks.

Jooniselt on hästi näha, et andmete haldamiseks ja hinnete sisestamiseks ei ole vaja omada rohkem informatsiooni, kui kasutaja nimi ja perekonnanimi ning testide tulemusi.

Teiseks probleemiks võiks nimetada kasutajate andmete saamist. Hetkel andmete vahetuseks Moodle'is on olemas PHP's kirjutatud funktsioonid, mis suhtlevad andmebaasiga ja tagastavad nõutud andmed juba printimiseks valmis kujul, kuid nad ei sobi kokku Joonis 6 kujutatud andmete kuvamise põhimõttega.

Esimese iteratsiooni käigus oli otsustatud kasutada Moodle'i andmebaasi otseselt selleks, et teostada päringuid vajalikku informatsiooni saamiseks ja uuendamiseks. *Phpmyadmin*'ist on näha, et Moodle'i andmebaas koosneb 84 tabelitest, mis on omavahel tihedalt seotud. Tabelite kasutamine informatsiooni kuvamiseks nõuab arusaamist nende struktuuridest ja omavahelistest seostest. Käesolevas töös struktuuri visuaalseks

kuvamiseks ja seoste välja selgitamiseks on kasutatud Mysql Workbench'i 6.3 [15] ja ametlikku Moodle'i andmebaasi dokumentatsioone [16] . Joonis 6 kujutatud mudeli realiseerimiseks on vaja teostada päring, mille tulemusel saadakse kursusel osalejate nimikiri ja nende tulemused. Vastuseks saadud andmete kuvamine peab olema Joonis 6 kujutatud näidismudeliga kooskõlas. Eelmainitud andmete saamiseks on vaja kasutada nelja tabelit: `mdl_grade_grades`, `mdl_user`, `mdl_user_enrolments`, `mdl_grade_items`.

Esimene „`mdl_grade_grades`“ tabel sisaldab andmeid kõikide kasutajate tulemustest. Teist tabelit nimega `mdl_user` kasutatakse kasutajate nimikirja saamiseks. Kolmas tabel „`mdl_user_enrolments`“ haldab informatsiooni kasutajatest, kes on antud kursusele lisatud. Viimasest ehk neljandast tabelist „`mdl_grade_items`“ võib vaadata, millisesse kategooriasse kuulub mingi test. Nendest tabelitest saadud andmetest piisab kasutajate ees- ja perekonnanimedele ning kasutajate tulemuste kuvamiseks vastavalt Joonis 6 demonstreeritud mudelile.

Järgmiseks sammuks oli antud töö peatükis 3 *Moodle hinneteraporti loomine* loodud Grader report'i koopia konfigureerimine joonisel 6 toodud demonstratsiooni realiseerimiseks. Grader report'i muutmiste realiseerimiseks oli otsustatud valida keeleks JavaScript ja dünaamilisuse saavutamiseks AJAX päringut. Päringu suhtlemine andmebaasiga on kirjutatud PHP keeles. Päritud andmed tagastatakse mahukas JSON formaadis vastusena. JSON on lihtsustatud andmevahetusvorming, mis põhineb JavaScripti programmeerimiskeele alamhulgal [12] . Vastuseks saadud andmed konfigureeritakse kasutades JavaScript, ning vastavalt joonisel 6 demonstreeritud mudeli eeskujule.

Esimese iteratsiooni loomise käigus tekkis idee realiseerida dünaamiline otsing nime järgi, selleks et vabaneda ka vertikaalsest kerimisest. Idee realiseerimiseks oli loodud tekstilahter ehk andmete sisestuse võimalus, mille järgi teostatakse AJAX päringut. Põhimõtteliselt iga uue tähe sisestus kutsub välja uue AJAX päringu.

```

<?php
header('Content-Type: text/html; charset=utf-8');
$tmp = $_GET["firstname"];
$link = mysqli_connect('localhost', 'root', '') or die('ei saa connect: ' .
mysqli_error());
mysqli_select_db($link,'moodle') or die('ei saa valida db');
$myArray = array();
$query ="SELECT
firstname,lastname,finalgrade,itemname,mdl_grade_grades.userid,mdl_user.id,mdl
L_grade_grades.id as gradeid FROM
mdl_grade_grades,mdl_user,mdl_user_enrolments,mdl_grade_items WHERE firstname
Like '$tmp%' AND mdl_user.id=mdl_user_enrolments.userid AND
mdl_grade_grades.userid=mdl_user.id AND
mdl_grade_items.id=mdl_grade_grades.itemid";
$result = mysqli_query($link,$query) or die('päring ei õnnestunud: ' .
mysqli_error());
    while($row = $result->fetch_array(MYSQL_ASSOC)) {$myArray[] = $row;}
echo json_encode($myArray);
mysqli_free_result($result);
mysqli_close($link);
?>

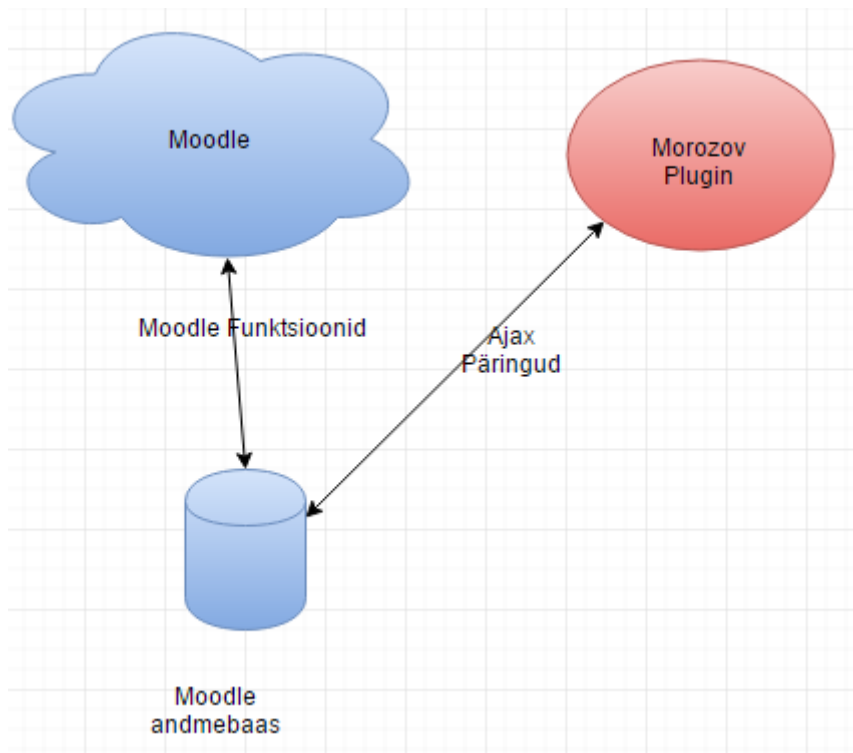
```

Joonis 7. AJAX päringu peal käivitatav kood, tabelite kuvamiseks.

Joonis 7 on kujutatud AJAX päringu peal käivitatav kood, mis oli kasutatud andmebaasiga suhtlemisel. Esiteks võetakse sisendparameeter kasutades `$_GET`'i. Seejärel vormistatakse andmebaasi sisselogimine ja ühendatakse andmebaasiga. Teiseks saadetakse mahukas SQL SELECT päring ning päringu vastuseks saadud andmeid konverteeritakse JSON formaati [17]. Viimaste sammudena vabastatakse mälu hoitavatest päringu tulemustest ning pannakse kinni ühendust andmebaasiga.

Grader report'i koopia failis index.php kustutatakse kõik, mida kasutab ametlik Grade report, selleks et kokku panna uut raporti. Tabelite ja sisendite kuvamine ning päringu loogika on realiseeritud JavaScript keeles kasutades JQuery. JQuery on JavaScript raamistik, mis lihtsustab koodi kirjutamist ja muudab veebilehe kasutajaliidese mõttes ilusamaks ja dünaamilisemaks [13]. JQuery on mõeldud veebiarendajatele, kes soovivad luua rohkemat kasutades vähemat. JavaScript oli otsustatud kasutada sellepärast, et iga järgmise Moodle'i versiooni väljatulemisega on oht, et praegused PHP-s kirjutatud funktsioonid võivad muutuda ja nende kasutamine võib põhjustada tulevikus loodava laienduse mittekorrektset tööd ning lähtudes autori JavaScript'i kasutamise oskusest ja kogemusest Joonis 8 on välja toodud esimese iteratsiooni arhitektuur.





Joonis 8. Esimese iteratsiooni *plugin*'i arhitektuur.

Raporti nimeks oli valitud minu perekonnanimi. Raporti arendamise käigus tekkis mõtte viia ellu ka dünaamiline hinnete muutmise funktsionaalsus, selleks et säilitada ametliku Grader report'i funktsionaalsust. Selle eesmärgi saavutamiseks oli kasutusele võetud POST AJAX päring, mis hinde peale klõpsamisega võimaldab seda dünaamilist muuta. Joonis 9 illustreerib koodi, mis käivitatakse serveris POST AJAX päringu peale.

```

<?php
  $connect = mysqli_connect("localhost", "root", "", "moodle");
  $id = $_POST["id"];
  $finalgrade = intval($_POST["finalgrade"]);
  $column_name = $_POST["column_name"];
  $sql = "UPDATE mdl_grade_grades SET ".$column_name."='".$finalgrade.'" WHERE
  id='".$id.'" ";
  if(mysqli_query($connect, $sql)){echo 'Data Updated';}
?>
  
```

Joonis 9. AJAX päring hinnete muutmisel.

Ametliku Grader report'i hinnete muutmise funktsionaalsuse säilitamisel tekkis probleem sellega, et lihtne dünaamiline hinde muutmine muudab ainult `final_result` veergu, kuid tegelikkuses Moodle'i poolt pakutud `update_grade` funktsioon teeb palju rohkem, kui lihtsalt ühe välja muutmine andmebaasis. See fakt tõi ülesse arhitektuurilised probleemid, kuna vastavalt arhitektuurile pole võimalik kasutada Moodle'i poolt PHP-s kirjutatud funktsioone.

Esimese iteratsioon kokkuvõtteks võib öelda, et Moodle'i andmete muutmine nõuab kasutada Moodle'isse sisse ehitatud funktsionaalsuse kasutamist, seega “jalgratta” välja mõtlemine dubleerides Moodle'is juba hetkel olemasolevat andmebaasiga suhtlemise funktsionaalsust ei ole ratsionaalne.

Esimese iteratsiooni käigus loodud laienduse (plugin) võib alla laadida siit:  
[https://github.com/arsenti/grader\\_report\\_plugin\\_v0.1](https://github.com/arsenti/grader_report_plugin_v0.1)

## 6 Teine iteratsioon.

Esimese iteratsiooni käigus avastatud probleemide kõrvaldamiseks oli vaja arhitektuuri täielikult muuta. Kõige tähtsam põhimõtte oli võtta kasutusele Moodle'i sees juba olemasolevat funktsionaalsust ja andmebaasiga suhtlemise printsiipi, kuid realiseerida seda ilma otseselt PHP-s kirjutatud funktsioonide kasutamist. Eesmärgi saavutamiseks on kirjutatud API, mis suhtleks Moodle'iga ja edastaks andmeid kasutades GET ning POST päringuid .

Moodle pakub andmete vahetamiseks veebiteenus võimalust. Veebiteenus toetab nelja järgmist protokoll: AMF<sup>8</sup>, REST<sup>9</sup>, SOAP<sup>10</sup> ja XML-RPC<sup>11</sup>. Antud töö raames püstitatud eesmärgi saavutamiseks oli valitud REST protokoll. REST on tarkvaraarhitektuuri stiil, mida kasutatakse hajutatud rakenduse koostisosade koostööks. REST kujutab endast kooskõlastatud piirangute komplekti, mida on vaja arvesse võtta hajutatud infosüsteemi projekteerimisel ja realiseerimisel. REST protokoll kasutamine antud töö raames oli kõige loogilisem valik, kuna REST võimaldab edastada päringuid JSON formaadis. JSON formaadis olevaid andmeid on väga lihtne kasutada JavaScript'is.

JavaScript'ile on kirjutatud mitu raamistikku, mis sobiksid antud ülesande lahendamiseks. Rakenduse raamistikuks sai valitud Angular JS. Angular JS on struktuurne raamistik, mida kasutatakse dünaamiliste veebirakenduste jaoks [14] . Antud raamistik võimaldab kasutada HTML-i baasmalli ning seda laiendada, mille tulemusena väljendatakse kõiki rakenduse komponente väga selgelt.

Rakenduses oli otsustatud kasutada SPA (*Single Page Application*) põhimõtteid. SPA on veebirakendus või veebileht, mis kasutab üht HTML faili kõikide veebivaadete kuvamiseks, rakendades dünaamiliselt laetavaid HTML-i, ja CSS-i ning JavaScripti jaoks tavaliselt rakendades AJAX-i päringuid [19] . SPA meenutab pärismaist (*native*) rakendust, kuid erinevusega, et kõik toimub brauseris, mitte eraldiseisva protsessina.

---

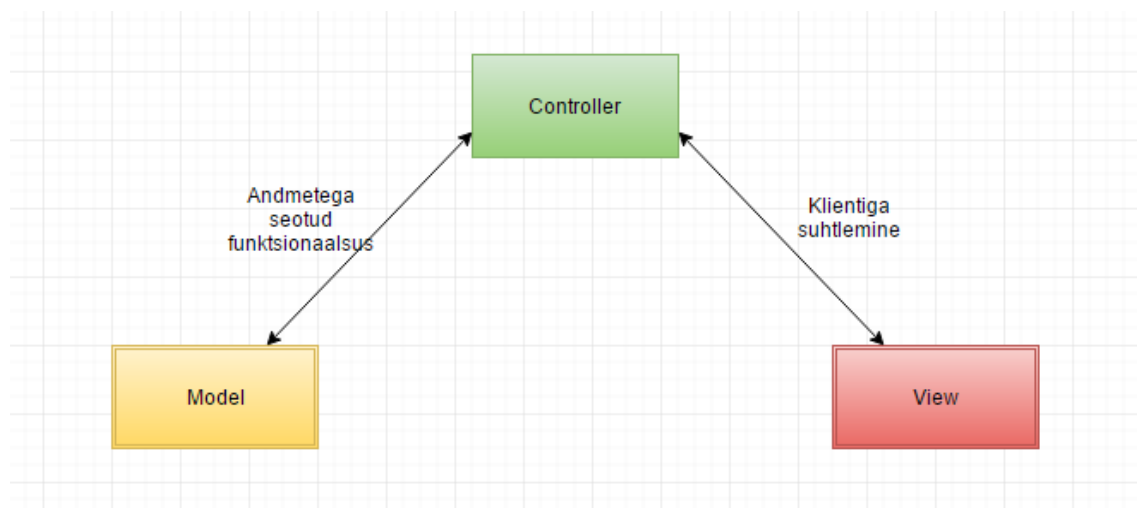
<sup>8</sup> AMF: [https://en.wikipedia.org/wiki/Action\\_Message\\_Format](https://en.wikipedia.org/wiki/Action_Message_Format)

<sup>9</sup> REST: [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)

<sup>10</sup> SOAP: <https://et.wikipedia.org/wiki/SOAP>

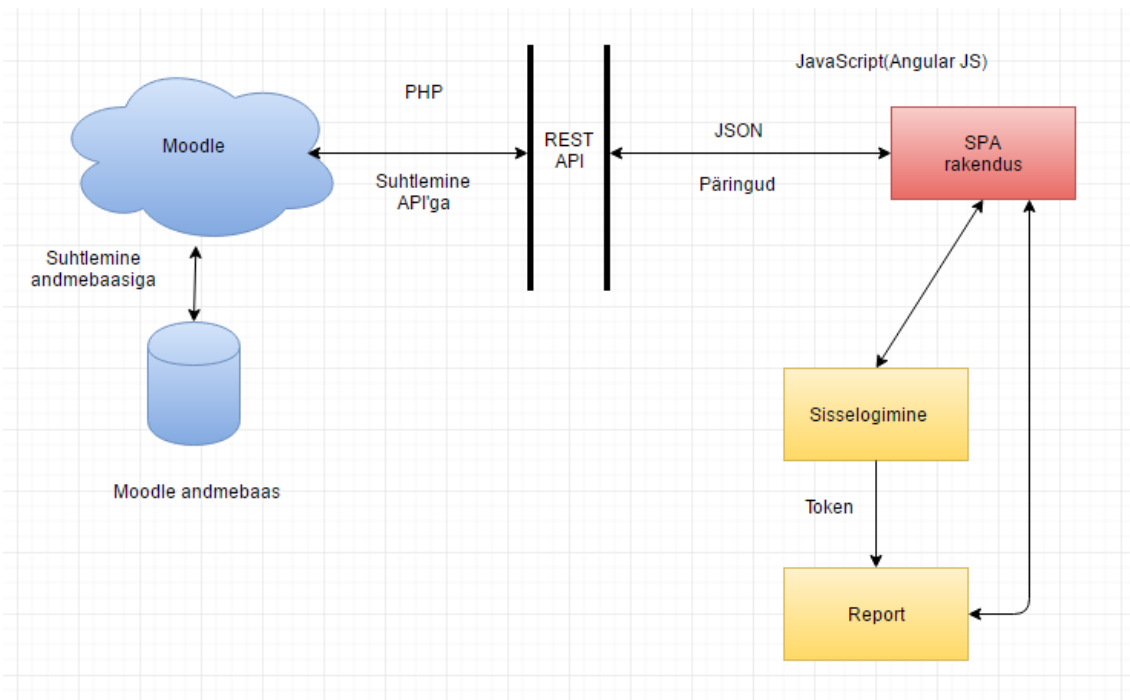
<sup>11</sup> XML-RPC: <https://et.wikipedia.org/wiki/XML-RPC>

SPA rakenduste põhimõtete saavutamiseks tuleb viis põhipunkti täita. Esiteks on vaja valida JavaScript raamistik, mis vastaks MVC või MVVM arhitektuurimustritele. MVC on muster, mille põhimõtte seisneb selles, et jagada kogu rakendust kolmeks osaks. Esimene osa on *model*'i kiht, mis vastutab andmetega seotud funktsionaalsuse eest. *Model*'i kiht reageerib andmete muutmise või kuvamise päringutele ning sellel kihil puudub võimalus andmete visualiseerimiseks. Teine osa on *view* kiht, mis vastutab informatsiooni visualiseerimise eest. Teisiti öeldes *view* on mustri kiht, mida kasutab klient rakendusega suhtlemisel. Viimaseks osaks on *controller*, mis vastutab *modeli* ja *view* sidumise eest. *Controller*'i põhimõtteks on kasutaja poolt tehtud otsuste üleandmine *model* kihisse. MVC arhitektuuri põhieelisteks on paindlikkus, kuna ühe kihi muutmine ei nõua alati muudatusi teistes kihtides ning rakenduste laiendamine muutub mugavamaks.



Joonis 10. MVC skeem.

Teiseks SPA rakenduste realiseerimise põhipunktis on vaja formuleerida *routing* ehk marsruutimine. *Routing* on rakenduse *view* kihis realiseeritud navigatsioon, teisisõnu, kuna kõik on realiseeritud ühes HTML failis, siis selle dünaamilise muutmise realiseerimiseks on vaja HTML sisu uuendamist. HTML sisu uuendamise navigatsiooniga tegeleb marsruutimine. Kolmandaks põhipunktiks on vaja luua malli, selleks, et *model* kihist saadud andmetest kasutades *controller*'i kihis olevaid funktsioone, kuvada *view* kihi osi. Neljanda põhipunkti täitmiseks on vaja API-it. Nagu varem oli mainitud selleks antud rakenduse raames kasutatakse Moodle REST API-it. Viiendaks põhipunktiks on päringute saamine ja dünaamilise veebilehe põhimõtte säilitamine, mille jaoks kasutatakse AJAX päringuid. Täites kõik varem loetletud SPA nõudeid oli loodud joonisel 11 kujutatud arhitektuur:



Joonis 11. Teise iteratsiooni arhitektuur.

Peamine aspekt, mis on andmebaasiga suhtlemisel paranenud võrreldes esimese iteratsiooniga, on andmete ringkäik. Joonise järgi on näha, et suhtlemist andmebaasiga teostatakse ainult läbi Moodle'i ning uus SPA Grader report'i rakendus suhtleb Moodle'iga läbi REST API. Antud arhitektuur aitab vältida Moodle'i uue versiooni väljatulemise korral PHP-s olevate funktsioonide võimalike muutumiste probleemi. Ainuke võimalus rakenduse suuremaks ümbertegemiseks versiooni uuendamisel on veebiteenuste funktsioonide muutmise Moodle'i poolt.

Rakenduse realiseerimiseks on vaja lubada veebiteenuse kasutamine Moodle'is, mis vaikimisi pole lubatud. Antud võimalus on olemas ainult Moodle administraatoril. Administraator peab *site administration* menüüs valima *plugin* menüü ja sealt *web service*. Veebiteenuse administreerimiseks on olemas kuus välja. Järgneva rakenduse realiseerimiseks on vaja kasutada valikut *external service*, mille seadistamiseks Moodle pakub samm-sammulist juhendit, mis asub *overview* väljas. Juhendis loetletud tingimuste täitmisel saavutatakse töötav *external service* ning genereeritakse autentimise *token*.

Autentimise *token*'i genereerimine Moodle'i poolt tähendab seda, et kõiki autenditud kasutajate õigused päritakse Moodle'i poolt. Autentimise *token*'i väljavõtmine kasutajatunnust ja parooli kasutades on realiseeritud moodle/login/token.php failis. Selle veebirakenduse REST päringuks on:

<http://localhost/moodle/login/token.php?username=username&password=password&service=service>.

*Username* on autentimise kasutajatunnuse nimi ning *password* on parool ja *service* on veebiteenuse lühinimetuse. Päringu korrektse *token*'i vastamiseks peavad olema kolm eeldust täidetud. Esiteks, kasutaja, kes tahab kasutada päringu *token*'i saamiseks peab olema veebirakenduse kasutajate nimekirja lisatud. Teiseks, kasutajal peab olema administraatori poolt genereeritud autentimise *token*. Kolmandaks, kasutajal peab olema õigus kasutada veebiteenuseid ja sellele vastavat protokollit. *Token* annab võimaluse kasutada põhimõtteliselt kõiki veebiteenuse poolt pakutavaid funktsioone *token*'i poolt kehtestatud õigustega.

Pärast kasutaja autentimise edukat läbimist antud rakendusse saadud *token*'it hakatakse kasutama veebiteenuste funktsioonide väljakutsemisel. Funktsiooni, mis tagastab kõikide kursuste kohta informatsiooni, nimeks on `core_course_get_courses`. REST päringu näide:

```
http://localhost/moodle/webservice/rest/server.php?wstoken=token&wsfunction=core_course_get_courses&moodlewsrestformat=json
```

*wstoken* on autentimise *token*, *wsfunction* on väljakutsetud funktsiooni nimetus ning *moodlewsrestformat*'iks on valitud JSON. `core_course_get_courses` funktsiooni vastuseks on võimalus saada kursuse id ning kasutada seda järgmistes päringutes.

Funktsiooni, mis tagastab valitud kursuse tulemust informatsiooni nimeks on `gradereport_user_get_grades_table`. REST päringu näide:

```
http://localhost/moodle/webservice/rest/server.php?wstoken=token&wsfunction=gradereport_user_get_grades_table&moodlewsrestformat=json&courseid=course_id
```

*Wstoken*, *wsfunction* ja *moodlewsrestformat* tähendavad sama, mis `core_course_get_courses` funktsiooni korral. *Courseid* tähendab kursuse id, mida saadakse eelmise funktsiooni väljakutsemisel.

Need kolm veebiteenuse funktsiooni on piisavad selleks, et autentida kasutaja ja printida välja dünaamiline hinneteraaport.

Tulemuste dünaamilise muutmise realiseerimiseks pakub Moodle'i veebiteenus funktsiooni `core_grades_update_grades`. See funktsioon tagastab ühe või nulli vastavalt funktsiooni tulemusele. Null tähendab, et hinne on edukalt muudetud, üks aga et hinne muutmine ebaõnnestus. Antud funktsioon nõuab 8 parameetrit: *wfunction*, *wstoken* ja *moodlewsrestformat* tähendused on selgitatud varem samas peatükis, *source* on hinde uuendamise allikas, *activityid* on rakendatud hinnete muutmise tegevuse id. Funktsiooni parameeter *courseid* on konkreetse kursuse id. Seda parameetri on võimalik saada kasutades eelnevalt kirjeldatud funktsiooni `Core_course_get_courses`. Funktsiooni parameeter *component* on komponent, mis on kirjeldatud `mdl_component` tabelis ja näitab, mille Moodle'i aspektiga tegemist on. *Component* parameetri näiteks on `mod_forum` ja `mod_quiz`, Funktsiooni parameeter *itemnumber* on id moodulite kategooriatele, mille väärtuseks pannakse null, kui vajalik id on iseseisev ning kuulu hinnete kategooriatesse

Funktsiooni välja kutsumine ning veebilehe struktuur on realiseeritud Angular JS raamistiku kasutades. Turvalisuse saavutamiseks autentimisel kasutatakse standardset sisselogimise mustrit, mis on pakutud Jason Watmore poolt [18]. Mustris on realiseeritud standardne autentimine kasutades Angular JS raamistiku. Mustrit on modifitseeritud nii, et rakendatakse *token*'i väljakutsemise funktsiooni ning eduka autentimisega saadakse *token*.

Rakenduse loomise käigus oli otsustatud kasutada bootstrap'i lehe viimistluse paremaks muutmiseks. Bootstrap on tasuta veebilehe esipoole raamistik, mis sisaldab HTML ja CSS šabloone tüpograafia, vormide, tabelite, nuppude, navigatsiooni, piltide ja paljude muude asjade disainimiseks.

Teine iteratsiooni ei ole veel lõpuni viidud ning rakendus on pidevalt arenduses. Teise iteratsiooni tänase seisuga oleva versiooni võib alla laadida siit: <https://github.com/arsenti/Teine-Iteratsioon>

## 7 Kokkuvõte

Käesoleva bakalaureusetöö põhietappideks oli Moodle'i keskkonda installeerimine, Moodle'i poolt pakutavate hinneteraportite funktsionaalsuse uurimise läbiviimine ning uurimuse käigus saadud andmeid kasutades hinnete organiseerimiseks realiseerimine iseseiseva rakenduse Moodle's. Rakenduse arendamisprotsess oli jagatud kaheks iteratsiooniks. Teise iteratsiooni tulemusena sai loodud arhitektuur andmebaasiga suhtlemiseks, kasutades Moodle'i poolt pakutavat veebiteenust. Autor väidab, et loodud arhitektuuri võib kasutada ka teiste veebiteenuste poolt pakutavate võimaluste realiseerimiseks.

Eesmärgiks oli luua rakendus Moodle'i õpikeskkonna hinneteraporti jaoks, mis parandaks kasutajamugavust Moodle'i tulemuste organiseerimisel. Vastava rakenduse realiseerimisel oli kasutatud kaasaegseid tehnoloogiat. Hinneteraportite uurimuse käigu ülesehitus ja ühe raporti peale rõhu panemine on põhjendatud sellega, et Moodle pakub peamise vahendina tulemuste sisestamise, muutmise ja jälgimise jaoks uuritava Grader report'i. Lähtudes püstitatud eesmärgist oli uurimuse põhiülesandeks välja selgitada Grader report'i puuduseid ja neid aspekte, mis vajavad parandamist.

Esimese iteratsiooni tulemusena sai loodud arhitektuur, mis parandas kasutajamugavust, kuid antud rakendus suhtles otseselt Moodle'i andmebaasiga, mis ei sobinud andmete sisestamiseks.

Teise iteratsiooni käigus oli otsustatud kasutele võtta SPA arhitektuurilisi põhimõtteid ja andmebaasiga suhtlemiseks Moodle API-t. Sellel viisil oli loodud iseseisev *single-page* rakendus hinnete haldamiseks. Rakenduse eeliseks on dünaamilisus ja veebilehel mugav marsruutimine. Tulevikus plaanitakse rakenduse laiendamine kasutades teisi veebiteenuse funktsioone.

Antud töö tulemus on tähtis, sest Moodle on üldtuntud e-kursuste haldamissüsteem ning populaarseim keskkond tasuta e-õppe läbiviimiseks. Moodle'i uute laienduste arendamine toob kaasa uue funktsionaalsuse lisandumise, mis omakorda täiendab keskkonda ja laiendab tema poolt pakutavate võimaluste hulka. Hinneteraportite kasutajamugavuse parendamine ja funktsionaalsuse lisamine muudab keskkonda täiuslikumaks.



## Kasutatud kirjandus

- [1] T. Tartes, „eTÜ | Tartu Ülikooli e-õppe ajakiri,“ 21 märts 2010. [Võrgumaterjal]. Available: <http://etu.ut.ee/kevad-2010/mis-on-moodle/>. [Kasutatud 8 mai 2016].
- [2] Moodle™, „Moodle Statistics,“ Moodle™, 25 jaanuar 2016. [Võrgumaterjal]. Available: <https://moodle.net/stats/>. [Kasutatud 19 mai 2016].
- [3] K. Varrik, „EESTIKEELSE ARVUTIALASE SÕNAVARA ARENDAMINE JA,“ TARTU ÜLIKOOLI VILJANDI KULTUURAKADEEMIA , Viljandi, 2009.
- [4] Moodle™, „Grader report,“ 20 jaanuar 2016. [Võrgumaterjal]. Available: [https://docs.moodle.org/29/en/Grader\\_report](https://docs.moodle.org/29/en/Grader_report). [Kasutatud 3 mai 2016].
- [5] Oracle Corporation, „MySQL,“ 2016. [Võrgumaterjal]. Available: <https://dev.mysql.com/doc/workbench/en/>. [Kasutatud 4 aprill 2016].
- [6] metshein.com, „WAMP server paigaldus,“ 2012. [Võrgumaterjal]. Available: <http://metshein.com/index.php/veeb/ueksikud-artiklid/707-wamp-server-paigaldus>. [Kasutatud 4 märts 2016].
- [7] R. Bourdon, „WampServer,“ [Võrgumaterjal]. Available: <http://www.wampserver.com/en/#>. [Kasutatud 8 märts 2016].
- [8] Wikimedia Foundation Inc., „Vikipeedia: phpMyAdmin,“ 10 märts 2013. [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/PhpMyAdmin>. [Kasutatud 2 märts 2016].
- [9] Moodle™, „Installing Moodle,“ 18 jaanuar 2013. [Võrgumaterjal]. Available: [https://docs.moodle.org/23/en/Installing\\_Moodle\(11\)](https://docs.moodle.org/23/en/Installing_Moodle(11)). [Kasutatud 6 märts 2016].
- [10] Moodle™, „Gradebook reports,“ 30 jaanuar 2012. [Võrgumaterjal]. Available: [https://docs.moodle.org/dev/Gradebook\\_reports](https://docs.moodle.org/dev/Gradebook_reports). [Kasutatud 6 märts 2016].
- [11] Wikimedia Foundation Inc., „Vikipeedia: Ajax,“ Wikimedia Foundation Inc., 16 november 2015. [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/Ajax>. [Kasutatud 15 märts 2016].
- [12] Wikimedia Foundation Inc., „Vikipeedia: JSON,“ Wikimedia Foundation Inc., 24 august 2015. [Võrgumaterjal]. Available: <https://et.wikipedia.org/wiki/JSON>. [Kasutatud 5 aprill 2016].

- [13] metshein.com, „01 - Mis on jQuery,“ metshein.com, 2012. [Võrgumaterjal]. Available: <http://metshein.com/index.php/veeb/jquery/589-01-mis-on-jquery>. [Kasutatud 15 aprill 2016].
- [14] S. Leetus, „AngularJS raamistiku õppematerjal,“ Tallinna Ülikool, Tallinn, 2015.
- [15] Oracle Corporation, „MySQL Workbench,“ 2016. [Võrgumaterjal]. Available: <http://dev.mysql.com/doc/workbench/en/>. [Kasutatud 16 aprill 2016].
- [16] „Moodle docs 3.0,“ 25 jaanuar 2016. [Võrgumaterjal]. Available: [https://docs.moodle.org/30/en/Main\\_page](https://docs.moodle.org/30/en/Main_page). [Kasutatud 2 märts 2016]
- [17] Microsoft, „Sissejuhatus Accessi SQL-keelde,“ 2016. [Võrgumaterjal]. Available: <https://support.office.com/et-ee/article/Sissejuhatus-Accessi-SQL-keelde-d5f21d10-cd73-4507-925e-bb26e377fe7e#bm1>. [Kasutatud 5 märts 2016]
- [18] J. Waltmore, „Jason Waltmore blog,“ 10 märts 2015. [Võrgumaterjal]. Available: <http://jasonwatmore.com/post/2015/03/10/AngularJS-User-Registration-and-Login-Example.aspx>. [Kasutatud 20 aprill 2016]
- [19] Wikimedia Foundation, Inc., „Wikipedia : Single-page application,“ Wikimedia Foundation, Inc., 16 mai 2016. [Võrgumaterjal]. Available: [https://en.wikipedia.org/wiki/Single-page\\_application](https://en.wikipedia.org/wiki/Single-page_application). [Kasutatud 10 aprill 2016]
- [20] E. Puuman, „JavaScript,“ 2014. [Võrgumaterjal]. Available: <http://kodu.ut.ee/~epuman/javascript.html>. [Kasutatud 18 aprill 2016].