

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Aleksei Kondratjev 155150IAPB

MOODLE'I KURSUSE LÕPPTULEMUSTE EDASTAMINE ÕIS-I

Bakalaureusetöö

Juhendaja: Ago Luberg
MSc

Tallinn 2019

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: Aleksei Kondratjev

13.01.2019

Annotatsioon

Töö eesmärk on õppekeskkonnast Moodle'i hinnete eksportimine õppeinfosüsteemi (ÕIS), et säästa õppejõu aega ja lihtsustada hinnete ülekandmist.

Antud töö raames luuakse pistikprogramm Moodle'i õppekeskkonna jaoks (edaspidi nimetatakse "plugin"), mis lihtsustab õppetulemuste ülekandmist Moodle'ist ÕIS-i. Kasutades imporditud hinnete CSV faili ÕIS-ist ja spetsiaalset vormi, saab õppejõud Moodle'is siduda tudengi matrikli ja UNI-ID ning eksportida Moodle'i hindeid ÕIS-i.

Töö kirjeldab plugina loomisel kasutatud tehnoloogiaid ja nende vahelisi seoseid, arhitektuuri, kasutajaliidest ning realiseerimisel ettetulnud probleeme. Arendamisel on kasutatud võimalikult palju Moodle'isse sisse ehitatud funktsionaalsust, mis teeb plugina mahu väikeseks (*lightweight*).

Töö tulemuseks on töötav ja lihtsalt modifitseeritav Moodle'i plugin, mida õppejõud saab kasutada hinnete ülekandmiseks. Pluginat on võimalik kasutada teistes Moodle'i õppekeskkondades.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, 5 peatükki, 22 joonist, 5 tabelit.

Abstract

Exporting the final grades of Moodle into SIS

The thesis main goal is to create a Moodle plugin which can be used in any Moodle workspace. This plugin should be used in order to give a teacher opportunity to export Moodle students' grades to SIS with less work than in case of manual grading. The plugin uses several Moodle functions and API: Template, String API, External API, AMD, RequireJS, XMLDB editor, AJAX module. In addition, third-party library has been added to this plugin: jQuery-csv.

In this thesis a problem has been stated which occurs to every teacher who works in Moodle and SIS. These 2 workspaces are learning management system and study information system. The teachers must export students' grades from Moodle to SIS. It might lead to the problem where a teacher has a several hundred students in one subject. The teacher must find connection between SIS student and Moodle student. It takes a lot of time for the teacher to complete it for a subject.

This plugin resolves a problem with connecting Moodle and SIS students. Plugin knows only Moodle student UNI-ID and SIS student number. To resolve this problem two solutions have been proposed in this thesis. First, ained.ttu.ee supports searching student UNI-ID using student number. This works only in ained.ttu.ee Moodle workspace. Secondly, the system tries to find a connection between SIS and Moodle students using their names. If these two ways could not help in solving a problem, then user should connect them manually.

The result of this thesis is a workable and simply modifiable Moodle plugin that can be used by teachers to transfer the grades. It must be possible to use the plugin in other Moodle learning environments.

The thesis is in Estonian and contains 30 pages of text, 5 chapters, 22 figures, 5 tables.

Lühendite ja mõistete sõnastik

AJAX	<i>Asynchronous JavaScript and XML</i>
AMD	<i>Asynchronous Module Definition</i>
API	<i>Application programming interface</i> , Rakendusliides
CSV	<i>Comma-separated values</i> , faili tüüp
Moodle	Avalik e-õppekeskkond
Plugin	Pistikprogramm
UNI-ID	Unikaalne identifikaator
ÕIS	Õppeinfosüsteem

Sisukord

2.1 Kasutuslugu	11
2.2 Tulemuse spetsifikatsioon	13
3.1 Moodle.....	14
3.2 jQuery-csv	14
4.1 Funktsionaalsus	16
4.1.1 Kasutajaliides	16
4.1.2 Kliendi ja serveri suhtlemine	19
4.1.2.1 Esimene päring: PHP päring	19
4.1.2.2 Teine päring: String API	20
4.1.2.3 Kolmas päring: Templates	21
4.1.2.4 Neljas päring: AJAX	22
4.1.3 Tudengite leidmine	27
4.1.4 ÕIS-i hinnete lugemine Moodle'isse	28
4.1.5 Failide genereerimine	29
4.1.5.1 ÕIS-i importimise fail	30
4.1.5.2 ÕIS-i konfliktsete tudengitega fail	30
4.1.5.3 Mitte imporditud ÕIS-i tudengite fail	30
4.1.5.4 Mitte imporditud Moodle'i tudengite fail	31
4.2 Moodle'i plugina struktuur	31
4.3 Andmebaas	33
4.4 JavaScript moodulid	36
Kasutatud kirjandus	39

Jooniste loetelu

Joonis 1. Kursuse hinnete vorm.....	10
Joonis 2. Moodle vorm.	11
Joonis 3. Esimene samm.....	17
Joonis 4. Teine samm.	18
Joonis 5. Kolmas samm.....	19
Joonis 6. JavaScripti käivitamine PHP koodist.	19
Joonis 7. Tõlgete kaardistamine	20
Joonis 8. Keeletõlgete kaart.....	21
Joonis 9. Mustache koodi kuvamine JavaScripti kaudu	22
Joonis 10. AJAX päring.	23
Joonis 11. AJAX mooduli kasutamine.	24
Joonis 12. Veebiteenuse funktsiooni deklareerimine.	25
Joonis 13. Veebiteenuse päringu argumentide valideerimine.	26
Joonis 14. ained.ttu.ee teenuse kasutamine curl kaudu.	27
Joonis 15. Tudengite valik sidumiseks.....	28
Joonis 16. Faili allalaadimine JavaScriptis.....	29
Joonis 17. Failide struktuur	32
Joonis 18. Olemi-suhte diagramm.	34
Joonis 19. Plugina andmebaasi uuendamine.	36
Joonis 20. Kolmanda osapoole raamastiku konfigureerimine.....	37
Joonis 21. AMD mooduli defineerimine.	37
Joonis 22. AMD mooduli kasutamine.	37

Tabelite loetelu

Tabel 1. ÕIS-ist eksporditud faili struktuur.....	29
Tabel 2. Moodle'ist eksporditud faili struktuur.	30
Tabel 3. Moodle'ist eksporditud faili struktuur koos hinnete konfliktidega.....	30
Tabel 4. Faili struktuur, kus on ÕIS-i tudengid ilma vasteta.....	31
Tabel 5. Faili struktuur, kus on Moodle'i tudengid ilma vasteta.	31

1 Sissejuhatus

Antud töö eesmärk on luua Moodle'i plugin, millega õppejõud saab kanda hinded üle Moodle'ist ÕIS-i. Moodle'i plugin peab olema lihtne ja struktureeritud. Õppejõule luuakse mugav kasutajaliides, mille kaudu ta saab eksportida Moodle'i hindeid ÕIS-i.

Need õppejõud, kes kasutavad Moodle'it, peavad üle kandma hinded, mis on pandud Moodle'i keskkonnas. Kursusel osaleb suur hulk tudengeid ja hinnete ülekandmine on ajamahukas töö. Peamiseks probleemiks on see, et Moodle'is ei ole seost tudengi matrikli ja UNI-ID vahel. See tähendab, et õppejõud peab tegema seosed käsitsi tudengi nime järgi. Selleks on vaja palju aega ja protsess on tülikas, kuna kursusel võib olla kaks tudengit sama nimega.

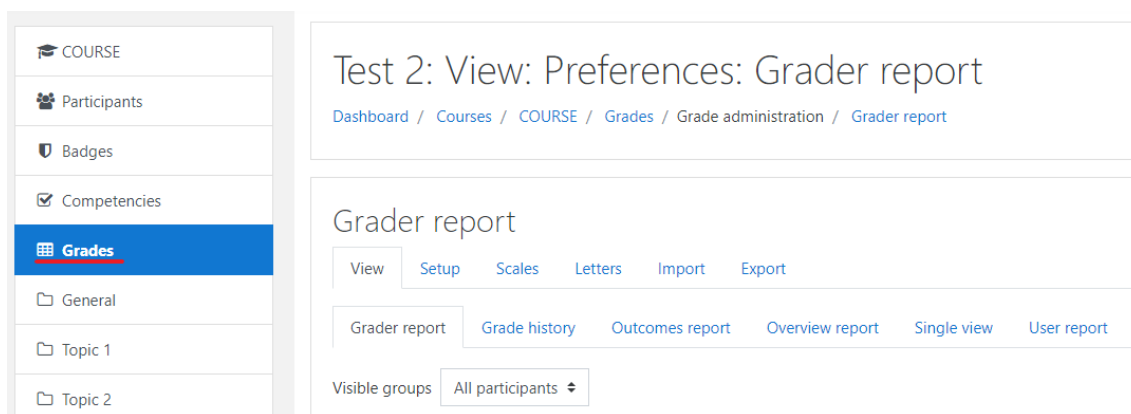
Eesmärk on, et seda Moodle'i pluginat saaks kasutada ka teistes keskkondades (tingimusel, et nendel kasutatakse sama ÕIS-i, mis on kasutamisel TTÜ-s ja TLÜ-s töö kirjutamise jooksul), mitte ainult ained.ttu.ee serveris. Plugina installeerimine peab olema lihtne, et iga kasutaja saaks seda kasutada oma Moodle'i õppekeskkonnas ilma probleemideta.

2 Analüüs

Selles peatükis antakse ülevaade loodava Moodle'i plugina tulemuse spetsifikatsioonist ja kirjeldatakse kasutuslugu.

ÕIS toetab võimalust eksportida ja importida tudengite andmeid/tulemusi kasutades CSV formaadis faili. ÕIS-ist saab eksportida: hindamisleht, tudengi matrikli number, tudengi nimi. Kui tudengil on olemas hinne ÕIS-is, siis eksporditakse ka hinne ja kuupäev. Moodle'i poolel on teada tudengi UNI-ID, tudengi Moodle'i hinne ja tudengi nimi.

Õppejõud saab eksportida oma ÕIS-i õppeainega seotud tudengeid koos hindega ja ilma hindeta. Moodle'is õppejõud avab kursuse lehte. Menüüs ta leiab punkti "Grades", mille peale vajutades kasutajale avatakse hinnete vaade (Joonis 1).



Joonis 1. Kursuse hinnete vorm

Selles vaates (Joonis 1) saab kasutaja valida kursuse hinnete ekspordi lehe, kust leiab valiku "ÕIS" ("SIS" plugina inglisekeelses versioonis) (Joonis 2).

Test 2: Export: OpenDocument spreadsheet

[Dashboard](#) / [Courses](#) / [COURSE](#) / [Grades](#) / [Grade administration](#) / [Export](#) / [OpenDocument spreadsheet](#)

Export to OpenDocument spreadsheet

[View](#) [Setup](#) [Scales](#) [Letters](#) [Import](#) [Export](#)

[OpenDocument spreadsheet](#) [SIS](#) [Plain text file](#) [Excel spreadsheet](#)

[XML file](#)

Joonis 2. Moodle vorm.

Enne Moodle'isse faili importimist peab õppejõud seadistama vajalikud parameetrid: miinimum punktid iga hinde jaoks ja võimalus valida ülesandeid, mis peavad tudengil tehtud olema.

2.1 Kasutuslugu

Tudengite tulemuste eksportimine Moodle'ist ÕIS-i toimub kahes keskkonnas, mis on Moodle ja ÕIS. Selline protsess toimub selles järjekorras:

- Kasutaja logib ÕIS-i sisse kui õppejõud.
- Kasutaja läheb ÕIS-is "Eksamid/Arvestused" lehele, kus saab eksportida kursuse iga tudengi andmeid ja hinnete olemasolul õppetulemusi.
- Kasutaja ekspordib tudengite andmed CSV-faili.
- Kasutaja logib Moodle'isse sisse kui õppejõud.
- Kasutaja läheb Moodle'is kursuse "Grades" lehele ja valib "Export" all "SIS" (inglisekeelne ÕIS versiooni lühend).
- Serverist küsitatakse ekspordi vorm ja keelestringe plugina jaoks.

- Kasutaja seadistab eksportimise parameetrid, mida kasutatakse hinnete konverteerimiseks ja tudengite filtreerimiseks.
- Kasutaja laeb üles ÕIS-ist eksporditud faili, parameetrid saadetakse serverisse.
- Parameetrid hinnete konverteerimiseks salvestatakse andmebaasi.
- Server tagastab Moodle'i tudengite andmeid, mis vastavad antud parameetritele.
- Moodle'i klientrakenduses toimub automaatne ÕIS-i ja Moodle'i tudengite vastavusse viimine.
- Kasutaja seob ÕIS-i ja Moodle'i tudengeid, kellele süsteem ei leidnud automaatselt vastavust.
- Kasutaja saab 4 linki, mille kaudu ta saab alla laadida CSV faile.
 - 1) Fail (põhifail), mille kasutaja laeb üles ÕIS-i
 - 2) Fail (õppetulemused käsitsi sisestamiseks), kus on tudengid, kellel on juba olemas hinne ÕIS-is. Nende jaoks peab kasutaja hinded panema käsitsi ÕIS-i.
 - 3) Fail (informatiivne), kus on ÕIS-i tudengid ilma vastavuseta (vastet Moodle'ist ei leitud).
 - 4) Fail (informatiivne), kus on Moodle'i tudengid ilma vastavuseta (vastet ÕIS-ist ei leitud).
- Kasutaja läheb ÕIS-i ja impordib esimese faili Moodle'ist.
- Juhul, kui tekivad hinnete konfliktid (tudengil on juba olemas hinne ÕIS-is), siis kasutaja parandab/paneab käsitsi hinded ÕIS-i kasutades teist faili.

2.2 Tulemuse spetsifikatsioon

Tulemuseks on väljatöötatud pluginaga genereeritud CSV fail, mis sisaldab tudengi Moodle'i hinnet ja hinde kuupäeva. Faili struktuur on sarnane ÕIS-ist eksporditud faili struktuuriga. Faili importimisel ÕIS-i on tudengi Moodle'i tulemused nähtavad ÕIS-is.

Faili sisu on täpsemalt kirjeldatud punktis 4.1.4.

3 Tehnoloogia ülevaade

Selles peatükis antakse ülevaade Moodle'ist ja kasutatud raamastikust.

3.1 Moodle

Moodle on e-õppe rakendus, mis on üldtuntud e-kursuste haldamissüsteemina. Moodle on tasuta ja avatud lähtekoodiga süsteem, mis on kirjutatud PHP programmeerimisekeeles. Moodle toetab paljusid funktsioone ja pakub võimalust pluginate kasutamiseks. [16]

Plugina arendamiseks annab Moodle võimaluse kasutada mitut raamastikku:

- tõlkimiseks on olemas *String API*, mille kohta on kirjutatud peatükis 4.1.2.2
- *External API* kasutatakse suhtlemiseks serveriga, mille kohta on kirjutatud peatükis 4.1.2.4
- *AMD moodulid* ja *RequireJS* kasutatakse JavaScripti moodulite loomiseks ja kasutamiseks, mille kohta on kirjutatud peatükis 4.4
- HTML lehe genereerimiseks on olemas *Template*, mille kohta on kirjutatud peatükis 4.1.2.3
- XMLDB redaktor lihtsustab tööd andmebaasiga, mille kohta on kirjutatud peatükis 4.3

Moodle on struktureeritud süsteem, kus on palju pluginaid, mis on kasutatud suure hulga funktsioonide toetamiseks. Pluginad on jagatud erinevateks tüüpideks. Süsteemis on rohkem kui 50 tüüpi. [8]

3.2 jQuery-csv

CSV failidega töötamiseks on valitud jQuery-csv raamastik. Raamastik on alla laaditav GitHubist (<https://github.com/typeiii/jquery-csv>). Raamastik toetab võimalust konverteerida JavaScripti objektid CSV faili ja vastupidi [15]:

- `$.csv.toArray(csv)` - konverteerib CSV faili sisu listi, mis koosneb stringidest.
- `$.csv.toArrays(csv)` - konverteerib CSV faili sisu listi, mis koosneb stringide listidest.
- `$.csv.toObjects(csv)` - konverteerib CSV faili sisu listi, mis koosneb objektidest, kus CSV faili esimene rida on võtmed väärtuste jaoks.
- `$.csv.fromArrays(arrays)` - konverteerib listi CSV faili.
- `$.csv.fromObjects(objects)` - konverteerib objekti CSV faili.

Iga konverteerimise funktsiooni käivitamisel on võimalik kasutada lisa-parameetreid ja funktsiooni, mis käivitatakse konverteerimise lõpus.

4 Lahendus

Selles peatükis antakse ülevaade probleemi lahendusest.

4.1 Funktsionaalsus

Moodle'i plugin peab olema iseseisev, et seda oleks võimalus kasutada teises keskkonnas, mis ei ole ained.ttu.ee. Plugin on tehtud kui *single-page application (SPA)*, mis vähendab lehelt lehele liikumisi.

ÕIS-is salvestatakse hinded kui väärtus on 0-5 või A (arvestatud) / MA (mitte arvestatud). Samal ajal Moodle'is on hinded tavaliselt salvestatud kui punktide arv, mis jääb vahemikku 0-100 (vaikimis seadistus). Tuleb anda õppejõule võimalus valida, kui palju punkte peab olema, et saada ÕIS-i hinnet. See on vaja salvestada andmebaasi, et õppejõud ei peaks neid väärtusi iga kord uuesti sisestama.

4.1.1 Kasutajaliides

Moodle'i hinnete eksportimine toimub ühel lehel. Kasutatakse ühte vormi, mis on jagatud 3 ossa. Kõik vormi elemendid on tehtud kasutades *bootstrap* raamastikku, mida tavaliselt kasutatakse Moodle'i keskkonnas.

Esimese sammuna (Joonis 3) sisestab kasutaja Moodle'is väärtused hinnete konverteerimise jaoks. Iga hindele ÕIS-is vastab punktide arv Moodle'is. Väärtus peab olema numbriline. Iga kord kui kasutaja vajutab nupu "Import grades", salvestatakse hinnete jaoks väärtused andmebaasi, et järgmine kord, kui kasutaja lehele tuleb, ei ole vaja neid väärtusi uuesti sisestada.

[OpenDocument spreadsheet](#)

[SIS](#)

[Plain text file](#)

[Excel spreadsheet](#)

[XML file](#)

Grade letters

Pass Exam

5	95
4	85
3	76
2	61
1	51
0	0

Student has to have a grade:

Exam Has to acquire a minimum

Choose file	Browse
Import grades	

Joonis 3. Esimene samm

Teise sammuna (Joonis 4) on vaja ühendada ÕIS-i ja Moodle'i tudengid. Vormi päises kuvatakse informatsioon importimise kohta: kui palju oli ühendatud tudengeid, paljudel on konflikt hindegaga ja paljusid ei saanud ühendada.

OpenDocument spreadsheet

SIS

Plain text file

Excel spreadsheet

XML file

Students who has already a grade in SIS:2

Students without a pair:3

Student uniid	Student name	Student number	Grade
user001	User001 Moodle	<input type="text"/>	4
user002	User002 Moodle	<input type="text"/>	3
user004	User004 Moodle	<input type="text"/>	0

Back

Next

Joonis 4. Teine samm.

Tudengid, kellel on hinne ÕIS-is olemas, kuvatakse tabelis kollase värvitooniga (nagu Joonis 15 kuvatud tudeng user004), mis on antud HTML-elementidele *table-warning* klassi kaudu.

Kolmandas sammus (Joonis 5) on antud 4 linki, mille kaudu kasutaja saab alla laadida failid ÕIS-i importimise jaoks. Nendest räägitakse täpsemalt peatükis 4.1.5.

Download

[Grades for SIS](#)[Pairs with conflicted grades](#)[SIS students without a pair](#)[Moodle students without a pair](#)[Back](#)

Joonis 5. Kolmas samm.

4.1.2 Kliendi ja serveri suhtlemine

Kasutaja suhtleb Moodle serveriga ainult 4 korda hinnete eksportimise jooksul.

4.1.2.1 Esimene päring: PHP päring

Esimene päring tehakse, kui kasutaja liigub Moodle's ÕIS-i ekspordi lehele. Päring läheb PHP lehele, kust tulevad vaikimised elemendid lehe jaoks. PHP kaudu käivitatakse JavaScripti mooduli funktsioon *init()*, mille kaudu algab eksportimise protsess. Selleks kasutatakse lehe lõpus PHP funktsiooni (Joonis 6):

```
$PAGE->requires->js_call_amd('gradeexport_sis/grade_export_sis', 'init',  
array($first, $last));
```

Joonis 6. JavaScripti käivitamine PHP koodist.

See käivitab funktsiooni *init* ja saadab tagasi kliendi parameetrid. Funktsiooni *js_call_amd* argumendid on:

- *'gradeexport_sis/grade_export_sis'* - esimene osa on komponendi nimi, teine osa on faili nimi ilma ".js" osata. [4]
- *'init'* - JavaScripti mooduli funktsioon, mis käivitatakse. [4]

- `array($first, $last)` - argumendid, mis antakse PHP poolt JavaScripti moodulisse. [4]

4.1.2.2 Teine päring: String API

Teises päringus küsitakse serverist tõlgitud keele tekstid. Selle jaoks on kasutatud Moodle'i moodulit `core/str` (*String API*). *String API* on moodul, mille kaudu on võimalik kasutada teisi keeli. See tegeleb nõ rahvusvahelisuse küsimusega ning püüab sobitada teksti iga kasutaja jaoks. Moodle'il on mehhanism, mis võimaldab mitmes kohas otsida keele tekste. Seetõttu on võimalik pakkida tõlked koos pluginaga. [1]

Mitme keele toetamiseks luuakse plugina kataloog `lang/en` (eesti keelse versiooni jaoks `lang/et`) ja pannakse sinna tekstifail. Tekstifaili nimi peab järgima plugina komponendi nime, mis selles töös on `gradeexport_sis.php`. [1]

PHP failis saab kasutada funktsiooni `get_string('sis:import', 'gradeexport_sis')`:

- `'sis:import'` - see on keele tõlke nimi, mis on tavaliselt antud järgmisel kujul: PHP koodis `$string['sis:import'] = 'Import grades';`.
- `'gradeexport_sis'` - plugina komponendi nimi.

JavaScripti failis on vaja kasutada moodulit `core/str`, kus on olemas funktsioon `get_string(array...)`. Funktsioon tagastab *Promise* objekti, mis tähistab asünkroonse toimingu õnnestumist (või ebaõnnestumist) ja toimingu lõpptulemust. Probleem on selles, et tagastatakse tõlgete järjend, kus antud keele tõlked on selles järjekorras, mis need olid päringus. Keele tõlgete vastavusse panekuks on kirjutatud funktsioon (Joonis 7):

```
function mapStrings(langStrings, mapper) {
  for (var i in mapper) {
    if (typeof mapper[i] === 'object' && mapper[i] !== null) {
      mapStrings(langStrings, mapper[i]);
    } else {
      mapper[i] = langStrings[mapper[i]];
    }
  }
}
```

Joonis 7. Tõlgete kaardistamine

Joonisel (Joonis 7) on `langStrings` tagastatud list ja `mapper` on objekt, mille kaudu küsitatakse keele tõlkeid, näiteks selline `mapper` (Joonis 8):

```

var stringMapping = {
  letters: 0,
  buttons: {
    back: 1,
    next: 2,
    importGrades: 6
  },
  gradeType: {
    pass: 3,
    exam: 4
  },
  <...>
  isAlreadyUsed: 27
};

```

Joonis 8. Keeletõlgete kaart

4.1.2.3 Kolmas päring: Templates

Kolmanda päringuga laetakse alla HTML leht. Kasutades Moodle'i moodulit *core/templates* (*Templates*) küsitakse serverist fail, kus on HTML leht ja kood. Kood on kirjutatud Mustache malli keeles.

Mustache süntaks:

- `{{raiden}}` - see on lihtne muutuv asendus. Otsitakse muutujat nimega "raiden" praeguses kontekstis (ja ükskõik millises vanemate kontekstis) ja kui väärtus leitakse, asendatakse muutujaga (HTML elemendid kaovad). [2]
- `{{{galaga}}}` - see on muutumatu asendus. Asendatakse muutuja nii, nagu see on antud. See on kasulik, kui muutuja sisaldab HTML-i plokki (näiteks). [2]
- `{{#lemmings}} jump off cliff {{/lemmings}}` - need on sektsioonisiltide avamine ja sulgemine. Kui *lemmings* muutuja eksisteerib ja väärtus ei võrdu *false*, antakse muutuja edasi, sektsiooni sisu analüüsitakse ja lisatakse tulemusse. Kui muutuja ei eksisteeri või võrdub *false* - jäetakse sektsioon vahele. Kui muutuja *lemmings* on massiiv, siis korratakse sektsiooni iga massiivi elemendi kohta tema kontekstiga. [2]
- `{{^lemmings}} enjoy view {{/lemmings}}` – vastupidine eelmisele, ekvivalentne *if-not* blokile. [2]
- `{{> franken_style/name_of_template }}` - see on osaline mallide importimine. [2]

- `{{ $blockvar }} ... {{ /blockvar }}` - see on *block* muutuja, mida kasutades on võimalik üle kirjutada sektsiooni (ploki sisu). Kui mallis on defineeritud plokid, siis mallide importimisel sellel kujul `{{ < template_name }} ... {{ /template_name }}` on võimalik plokis üle kirjutada sisu (sektsiooni). [2]

Mustache fail renderdatakse ainult üks kord kasutades Moodle'i moodulit *Templates*. Järgmised muutmised toimuvad otseselt kasutades JavaScripti. Renderdamiseks kasutatud funktsioon (Joonis 9):

```
templates.render('gradeexport_sis/main', context).then(successCallback)
```

Joonis 9. Mustache koodi kuvamine JavaScripti kaudu

Kus:

- `'gradeexport_sis/main'` - esimene osa on plugina komponendi nimi, teine osa on Mustache faili nimi (`main.mustache`) ilma laiendita.
- `context` - kontekst (JavaScripti objekt), mida kasutatakse faili renderdamise jooksul.
- `successCallback` - funktsioon, mis käivitatakse pärast vastuse saamist ja renderdamist.

Funktsioon `render(...)` tagastab *Promise* objekti. Vastusena tuleb faili sisu, mis on renderdatud kasutades Mustache malli keelt.

4.1.2.4 Neljas päring: AJAX

Serveriga suhtlemiseks kasutatakse AJAX päringut, mis on kasutab Moodle'i moodulit *core/ajax*. See AJAX-i moodul tagab kõikide päringute turvalisuse. [2]

Selle mooduli eelised on:

- Uued AJAX skriptid ei vaja turvaaukude kontrollimist. [3]
- Mitu päringut saab siduda ühte HTTP päringusse. [3]
- Kõigi parameetrite ja tagastamisviiside range kontrollimine. [3]

- Uued veebiteenuste funktsioonid on kasulikud AJAX liideste ja veebiteenuste klientidele. [3]

AJAX interaktsiooni loomiseks vajalikud sammud:

- Kirjutada või leida olemasolev veebiteenuse funktsioon *ajax* interaktsiooni käsitlemiseks. [3]
- Sisse lülitada AJAX veebiteenuses. Selleks on vaja defineerida '*ajax*' => *true* funktsioonis, mis asub failis *db/services.php*. Ainult need funktsioonid, kus AJAX on sisse lülitatud, on saadaval AJAX kaudu. [3]
- Kasutaja toimingule vastuseks kutsuda JavaScripti veebiteenust. [3]

▼ General

Request URL: https://ained.ttu.ee:9717/lib/ajax/service.php?sesskey=ZjIsVCpCyA&info=gradeexport_sis_get_grades
 Request Method: POST
 Status Code: ● 200 OK

Joonis 10. AJAX päring.

Pildi peal (Joonis 10) on näha, et päringus on vähemalt 2 parameetrit, mis on *sesskey* ja *info*. Turvalisuse mõttes on kõige olulisem *sesskey*, mis on seansi võti.

Kui kasutaja logib sisse, siis Moodle lisab seansile juhusliku sõne (võtme). Kui lehel on link või nupp, mida vajutades toimub mingi tegevus serveri poolt, siis lisatakse see *sesskey* andmetele lisaväärtusena. Enne tegevust võrreldakse päringus oleva *sesskey* väärtust sessioonis oleva võtme väärtusega ja tegevus toimub ainult siis, kui need kaks võrduvad. [5]

Seetõttu on näiteks kasutaja kustutamise päring tegelikult midagi sellist, nagu <http://example.com/moodle/user/delete.php?id=12&confirm=1&sesskey=E8i5BCxLJR>, ja teised kasutajad ei saa teada, milline on *sesskey*, nii et nad ei saa koostada URL-i, millega administraator eemaldab kasutajat. [5]

Plugina AJAX päringu kasutamise näide (Joonis 11):

```

ajax.call([
  {
    methodname: 'gradeexport_sis_get_grades',
    args: {
      data: {
        course_id: <...>,
        grade_items: <...>,
        students: <...>,
        grade_settings: <...>
      }
    },
    done: function(response) {
      <...>
    },
    fail: function(error) {
      <...>
    },
  }
]);

```

Joonis 11. AJAX mooduli kasutamine.

Kasutades funktsiooni *call([...])* moodulist *core/ajax* saadab kasutaja andmeid serverisse.

Funktsioon võtab argumendiks listi, kus saab olla mitu päringut. Antud argumendid on:

- *methodname* - see on funktsiooni nimi, mis on defineeritud failis *db/services.php*. Seal peab olema sisse lülitatud AJAX.
- *args* - see on saadetavad parameetrid.
- *done* - funktsioon, mis käivitatakse, kui päring on edukas.
- *fail* - funktsioon, mis käivitatakse, kui ilmneb viga.

AJAX päring jõuab plugina veebiteenusesse, kus toimub lisakontroll:

- kasutajal on õigused funktsiooni käivitamise jaoks. [6]
- AJAX päringu argumendid vastavad ootusele. [6]
- Veebiteenuse tagastatud tulemus vastab ootusele.

Veebiteenuse parameetrid on *db/services.php* failis. Parameetrite fail on järgmise kujuga (Joonis 12):


```

$functions = array(
    'gradeexport_sis_get_grades' => array(
        'classname' => 'grade_export_sis_external',
        'methodname' => 'get_grades',
        'classpath' => 'grade/export/sis/externallib.php',
        'description' => 'Gets a list of grades for course.',
        'type' => 'write',
        'ajax' => true
    )
);

```

Joonis 12. Veebiteenuse funktsiooni deklareerimine.

\$functions on list, kus on kirjutatud kõik veebiteenuse funktsioonid. Igas funktsioonis on parameetrid:

- *classname* - see on veebiteenuse API klass, kus on kirjas funktsioonide sisu.
- *methodname* - see on funktsiooni meetodi nimi, mis käivitatakse, kui kõik päringu argumendid on korras.
- *classpath* - see on tee failini, kus asub klass nimega *classname*.
- *description* - funktsiooni kirjeldus.
- *type* - andmebaasi õigused funktsiooni jaoks (*read* või *write*). [7]
- *ajax* - kas veebiteenuse funktsioon on avalik AJAX päringute jaoks. [7]

Veebiteenuse funktsiooni jaoks on tavaliselt kirjutatud 3 funktsiooni:

- *methodname_parameters()* - siin toimub päringu argumentide kontroll.
- *methodname(\$args)* - see on käivitav peafunktsioon.
- *methodname_returns()* - siin toimub tagastatud objekti kontroll.

Funktsiooni *get_grades* parameetrid peavad vastama järgmistele tingimustele (Joonis 13):

```

return new external_function_parameters(
    array(
        'data' => new external_single_structure(
            array(
                "course_id" => new external_value(PARAM_INT,
"Course id"),
                "grade_items" => <...>,
                "students" => new external_multiple_structure(
                    new external_value(PARAM_RAW, 'Student
number', VALUE_OPTIONAL)
                ),
                "grade_settings" => new external_single_structure(
                    array(<...>)
                )
            )
        )
    )
);

```

Joonis 13. Veebiteenuse päringu argumentide valideerimine.

PHP objektide seos JavaScriptiga on selline:

- *external_single_structure* - see on JavaScripti objekt.
- *external_multiple_structure* - see on JavaScripti *array* (list).
- *external_value* - see on primitiivne väärtus, mille kohta on vaja anda informatsioon esimese parameetriga (*PARAM_RAW*, *PARAM_INT*, ...).

Funktsiooni argument on objekt, kus on parameeter *data*, mis on ka objekt. Seal on 4 parameetrit:

- *course_id* - kursuse identifikaator, mis on number.
- *grade_items* - ülesanded, mis peavad tudengil olema tehtud.
- *students* - tudengite matrikli numbrid.
- *grade_settings* - seaded, mida kasutatakse hinnete konverteerimisel.

Funktsiooni käivitamise jooksul toimub *grade_settings* väärtuste salvestamine andmebaasis.

Protsessi lõpus genereerib veebiteenuse funktsioon vastuse ja saadab selle kasutajale AJAX päringu vastusena. Kui kasutaja saab vastuse, siis genereeritakse tabel, kus on

kuvatatud kõik tudengid koos hinde, nime, UNI-ID ja seotud tudengiga (või HTML *select* elemendiga).

4.1.3 Tudengite leidmine

Peaprobleem on kuidas leida seos Moodle'i tudengi ja ÕIS-i tudengi vahel. Töös on kasutatud 2 varianti selle probleemi lahendamiseks.

Esimene variant, mis toimub serveris, on suhtlemine LDAP-iga, kust võetakse tudengi matrikli numbriga järgi tudengi UNI-ID. See variant töötab ainult TTÜ Moodle'i keskkonnas. Kui meil on teatud tudengi UNI-ID Moodle'is ja tudengi matrikli number ÕIS-is, siis kasutades teenust on võimalik need siduda.

Teenuse päring tehakse GET päringuga serveris. Teenus tagastab tudengi UNI-ID matrikli numbriga järgi. Näiteks, 'http://localhost:port/155150IAPB' tagastab 'alkond'.

Teenuse parameetrid on konfigureeritavad. Kui päringute ebaõnnestunud vastuste arv on suurem kui `$MAX_CURL_ERRORS` (Joonis 14), siis rohkem päringuid ei tehta.

```
$error_counter = 0;
foreach ($params["data"]["students"] as $unused => $studentNumber) {
    $curl = curl_init();
    curl_setopt_array($curl, array(
        CURLOPT_RETURNTRANSFER => 1,
        CURLOPT_CONNECTTIMEOUT => self::$CURL_TIMEOUT,
        CURLOPT_URL =>
'http://localhost:'.self::$WEBSERVICE_PORT.'/'. $studentNumber,
    ));
    $resp = curl_exec($curl);
    if (curl_errno($curl)) {
        $error_counter++;
    } else {
        if (strpos($resp, 'ERROR:') === false) {
            $resp = substr(trim($resp), 1, -1);
            $grade_data["students"][$studentNumber] = $resp;
        }
    }
    curl_close($curl);
    if ($error_counter > self::$MAX_CURL_ERRORS) {
        break;
    }
}
```

Joonis 14. ained.ttu.ee teenuse kasutamine curl kaudu.

Tudengite sidumine toimub AJAX päringu jooksul, kus kasutaja failist saadetakse tudengite matrikli numbrid, mille järgi serveris toimub nende UNI-ID leidmine. Kui veebiteenusest tuleb vastus, siis toimub tudengite sidumine kasutaja töökohal. See sidumine on lõplik ja ei muutu.

Teine variant töötab nii, et klientrakenduses proovitakse leida seoseid nime järgi. See töötab igas Moodle'i keskkonnas. Juhul, kui leitakse 2 või rohkem sama nimega tudengit, siis need on vaja käsitsi siduda.

Student uniid	Student name	Student number	Grade
Students who has already a grade in SIS:2			
Students without a pair:3			
user001	User001 Moodle	Aleksei Kondratjev 1500001A	4
user002	User002 Moodle		3
user004	User004 Moodle	Aleksei Kondratjev 1500001APB Viktor Putronen 150001IAPB Viktoria Zahharova 150002IAPB	0

Back Next

Joonis 15. Tudengite valik sidumiseks.

Kasutajal on vaja iseseisvalt siduda need Moodle'i tudengid (Joonis 15), kellele ei ole leitud vastet automaatselt. Selle jaoks on vaja valida selline tudeng, kes kasutaja arvates sobib.

4.1.4 ÕIS-i hinnete lugemine Moodle'isse

ÕIS-i poolt genereeritakse CSV fail, kus asuvad andmed tudengite hinnete kohta. Faili struktuur on ühesugune iga kord ja selle tõttu on võimalik lugeda faili ilma probleemideta.

Fail on CSV formaadis. Eraldaja on sümbol ";" (*semicolon*). Faili struktuur on järgmine (Tabel 1):

IAY0520	Arvutiarhitektuurid	Ago Luberg			
jrk.	hindamisleht	tudengi kood	tudengi nimi	hinne	kuupäev
1.	XXX	155150IAPB	Aleksei Kondratjev	5	01.01.2019

Tabel 1. ÕIS-ist eksporditud faili struktuur.

Esimene ja teine rida on vajalikud faili importimiseks ÕIS-i. Alates kolmandast reast on tudengite andmed. Kõik tudengite andmed on salvestatud klientrakenduse poolt eksportimise protsessi jooksul.

Faili lugemine toimub kasutades jQuery-csv raamastikku.

4.1.5 Failide genereerimine

Lõpuks on kasutajal 4 linki, et alla laadida genereeritud failid. Kõik failid on genereeritud Moodle'i klientrakenduse poolt. Moodle'ist allalaadimise jaoks on tehtud funktsioon *exportToCsv(filename, content)* (Joonis 16):

```
function exportToCsv(filename, csvFile) {
    var blob = new Blob([csvFile], {type: 'text/csv;charset=utf-8;'});
    if (navigator.msSaveBlob) { // IE 10+
        navigator.msSaveBlob(blob, filename);
    } else {
        var link = document.createElement("a");
        if (link.download !== undefined) { // Feature detection
            // Browsers that support HTML5 download attribute
            var url = URL.createObjectURL(blob);
            link.setAttribute("href", url);
            link.setAttribute("download", filename);
            link.style.visibility = 'hidden';
            document.body.appendChild(link);
            link.click();
            document.body.removeChild(link);
        }
    }
}
```

Joonis 16. Faili allalaadimine JavaScriptis.

Juhul, kui brauser on Internet Explorer 10+ versioon, siis kasutatakse brauseri funktsiooni, mille kaudu saadetakse fail kasutajale. Kui see on muu brauser, kus

toetatakse *download* atribuuti, siis luuakse HTML *a* element, mille kaudu toimub allalaadimine.

4.1.5.1 ÕIS-i importimise fail

Esimene faili link, mis kuvatakse kasutajale, on fail koos uute hinnetega. See fail sobib importimiseks ÕIS-i. Faili struktuur sarnaneb selle faili struktuurile, mis imporditi Moodle'isse (Tabel 1).

Tudengite järjekord failis ei ole tähtis. Faili kirjutatakse ainult need tudengid, kellel on leitud paar ja juba ei ole hinnet ÕIS-is. Fail näeb välja nii (Tabel 2):

IAY0520	Arvutiarhitektuurid	Ago Luberg			
jrk.	hindamisleht	tudengi kood	tudengi nimi	hinne	kuupäev
1.		155150IAPB	Aleksei Kondratjev	5	12.05.2019

Tabel 2. Moodle'ist eksporditud faili struktuur.

4.1.5.2 ÕIS-i konfliktsete tudengitega fail

Teine faili link, mis kuvatakse kasutajale, on fail tudengitega, kellel on konflikt hindega. Kasutaja näeb tudengeid, keda ei saanud importida kasutades faili. Nende hinded on vaja panna ÕIS-i käsitsi, sest faili importimise jooksul süsteem ei tea, kas see on parandamine esimese hinde jaoks või see on teine hinne. Faili struktuur on selline (Tabel 3):

jrk.	uniid	hindamisleht	tudengi kood	tudengi nimi	Moodle hinne	Moodle kuupäev	ÕIS hinne	ÕIS kuupäev
1.	alkond	XXX123	155150IAPB	Aleksei Kondratjev	5	12.05.2019	4	10.05.2019

Tabel 3. Moodle'ist eksporditud faili struktuur koos hinnete konfliktidega.

Tudengi nimi on ÕIS-i tudengi nimi. Moodle kuupäev on viimane Moodle'i hinde muutmise kuupäev.

4.1.5.3 Mitte imporditud ÕIS-i tudengite fail

Kolmas faili link, mis kuvatakse kasutajale, on fail koos ÕIS-i tudengitega, keda ei ole esimeses või teises failis. Nendele ÕIS-i tudengitele ei ole leitud vastet Moodle'i tudengite hulgast. See fail on informatiivne. Struktuur on selline (Tabel 4):

IAY0520	Arvutiarhitektuurid	Ago Luberg			
jrk.	hindamisleht	tudengi kood	tudengi nimi	hinne	kuupäev
1.	XXX123	155150IAPB	Aleksei Kondratjev	5	10.05.2019

Tabel 4. Faili struktuur, kus on ÕIS-i tudengid ilma vasteta.

4.1.5.4 Mitte imporditud Moodle'i tudengite fail

Neljas faili link, mis kuvatakse kasutajale, on fail koos Moodle'i tudengitega, keda ei ole esimese või teises failis. Nendele Moodle'i tudengitele ei ole leitud vastet ÕIS-i tudengite hulgast. See fail on informatiivne. Faili struktuur on selline (Tabel 5):

jrk.	uniid	tudengi nimi	hinne	kuupäev
1.	alkond	Aleksei Kondratjev	5	12.05.2019

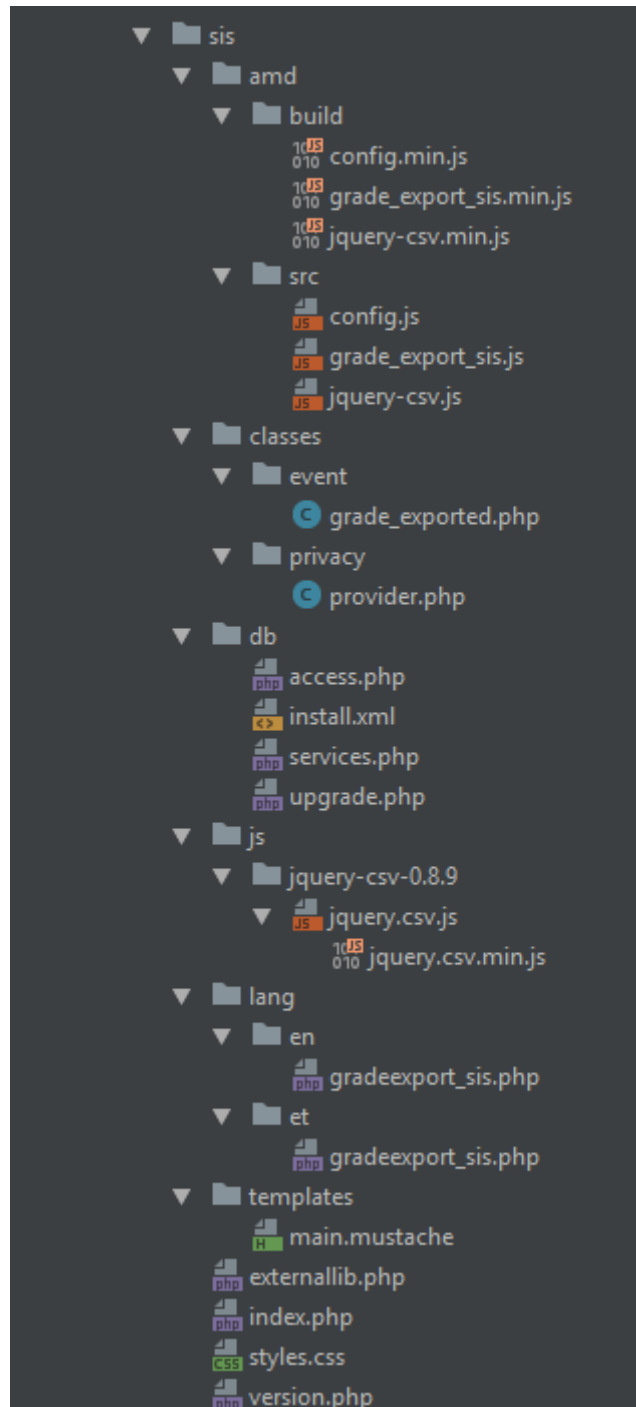
Tabel 5. Faili struktuur, kus on Moodle'i tudengid ilma vasteta.

4.2 Moodle'i plugina struktuur

Funktsionaalse programmi loomiseks kasutatakse Moodle'is pluginaid. Moodle'i plugin on struktureeritud plugin, mis põhineb kindlal plugina tüübil. Plugina tüübi järgi on defineeritud struktuur ja on lihtsam arendada programmi. Iga plugina tüübi jaoks on kohustuslikud failid, mis peavad olema loodud. [9]

Arendatava plugina jaoks on valitud plugina tüüp *Gradebook export*. Seda plugina tüüpi on kasutatud Moodle'i hinnete eksportimiseks ja on kasutuses alates Moodle'i versioonist 1.9. [8]

Seda tüüpi pluginad asuvad *grade/export* kaustas (Joonis 17).



Joonis 17. Failide struktuur

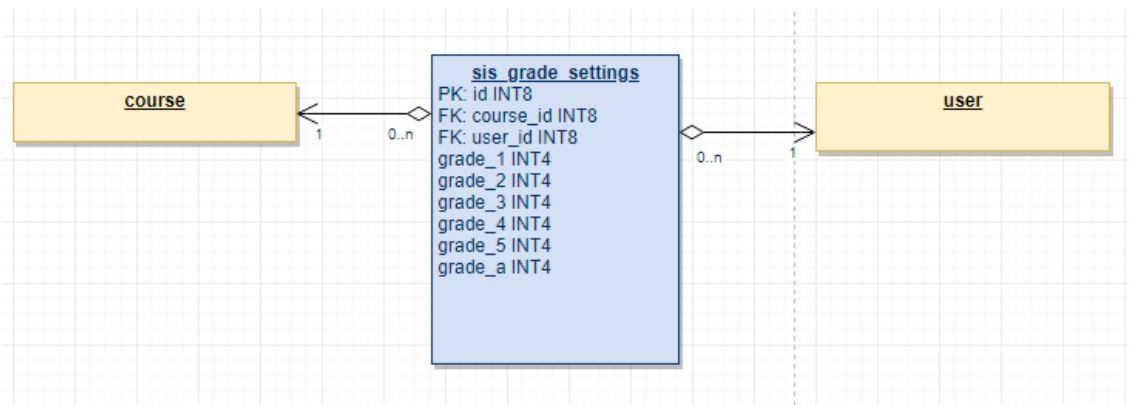
Plugina kausta sisu:

- *version.php* - see on oluline fail, mis peab olema plugina juurkataloogis. See fail sisaldab mitmeid parameetreid, mida kasutatakse plugina installimise ja uuendamise käigus. See fail võimaldab veenduda, et plugin on ühilduv antud Moodle'i saidiga, samuti määrab, kas uuendamine on vajalik. [10]

- *index.php* - see on plugina pealeht. Siin tavaliselt kuvatakse esimene vorm ja kontrollitakse kasutaja õiguseid.
- *db/install.xml* - see on fail, mis tavaliselt genereeritakse kasutades XMLDB redaktorit. Defineerib andmebaasi tabelite ja väljade struktuuri XML kaudu. Fail käivitatakse plugina installeerimisel ja lisab muudatused andmebaasi.
- *db/upgrade.php* - see on fail, mis sisaldab uuendamise funktsioone. Kasutatakse juhul, kui süsteem näeb, et käivitatakse vana versioon ja on võimalik kasutada uut versiooni.
- *db/access.php* - see on fail, kus on defineeritud kasutaja õigused.
- *db/services.php* - see on fail, kus on defineeritud veebitenuse funktsioonid, täpsemalt on nendest kirjutatud peatükis 4.1.2.4.
- *lang/** - see on kaust, kus iga keele jaoks on loodud alamkaust. Inglise keele jaoks peab kausta nimi olema *en*. Seal peab olema PHP fail plugina komponendi nimega. Fail peab sisaldama vähemalt ühte tõlget. Täpsemalt on sellest kirjutatud peatükis 4.1.2.2.
- *templates/** - see on kaust, kus asuvad mallid. Mallid on kirjutatud Mustache malli keeles. Täpsemalt on kirjutatud peatükis 4.1.2.3.
- *js/** - see on kaust, kus asuvad JavaScripti raamistikud. Need lisatakse rakendusse kasutades *requirejs*. Täpsemalt on kirjutatud peatükis 4.4.
- *amd/** - see on plugina JavaScripti moodulite kaust, mis kasutavad AMD moodulit. Seal asuvad 2 kausta, *src* ja *build*. Esimeses on tavaline JavaScripti kood, teises on minimeeritud kood. Täpsemalt on kirjutatud peatükis 4.4.

4.3 Andmebaas

Andmebaasi arhitektuur on koostatud kasutades Moodle API-t. Järgnevalt on toodud andmebaasi arhitektuur plugina piires.



Joonis 18. Olemi-suhte diagramm.

Joonis 18 näitab, et plugina jaoks on tehtud uus tabel, mille nimi on *sis_grade_settings*. Sellesse tabelisse salvestatakse parameetrid, mida kasutatakse Moodle'i hinnete konverteerimisel ÕIS-i hinneteks.

Moodle installeerib plugina vajalikud andmebaasi tabelid ja väljad kasutades *install.php* või *install.xml* faili. Selles pluginas kasutatakse andmebaasi tabelite loomiseks XML faili. Moodle soovib kasutada XMLDB redaktorit. XMLDB redaktor võimaldab luua korrektse install.xml faili ja genereerida PHP koodi, mida tavaliselt kasutatakse plugina uuendamiseks juhul, kui on tehtud muudatusi andmebaasis. Enne alustamist peab olema tehtud kaust nimega *db*.

XMLDB redaktor muudab tabelite / väljade / võtmete / indeksite lisamise praktiliselt triviaalseks ülesandeks, võimaldades arendajal kulutada aega kodeerimisele ja asjade parandamisele selle asemel, et võidelda XML-failide ja käsitsi redigeerimisest põhjustatud vigadega. [11]

Arendaja saab muuta välju ja määrata nende nime, tüübi, pikkuse, kümnendkoha, nullväärtuse, vaikeväärtused ja nii edasi. Täpselt sama nii võtmete kui ka indeksite puhul. [11]

On vaja kasutada XMLDB redaktorit vastavalt järgmistele reeglitele [11]:

- Nimed:
 - Nimedes peavad olema kõik väiketähed (tabelid, indeksid, võtmed ja väljad).

- Tabelite ja väljade nimed peavad kasutama ainult a-z, 0-9 ja _ märke. Tabelite nimed võivad olla kuni 28 tähemärki; veerus on maksimaalselt 30 tähemärki. [11]
- XMLDB failide all olevad võtme- ja indeksinimed peavad olema moodustatud võtmes/indeksis olevate väljade nimedest, mis on eraldatud "-" märgiga. [11]
- *Primary key* nimi peab alati olema "primary" (see on üks erand eelmisest konventsioonist). [11]
- Soovitav on vältida täielikult reserveeritud sõnu. [11]
- NULL väärtused
 - Vältida kõigi väljade loomist, kus on NOT NULL ja vaikimisi väärtus on "" (tühi string). Tabelite loomiseks kasutatav kood oskab seda korralikult käsitseda, kuid see võib viia probleemideni. [11]

Kui on vaja uuendada pluginat koos muudatustega andmebaasis, siis on vaja kirjutada PHP koodi *upgrade.php* failis. XMLDB redaktor saab genereerida selle jaoks osa koodist. Uuendamise funktsiooni nimi peab olema *xmlldb_{component_name}_upgrade*, funktsioonis kontrollitakse plugina versiooni (Joonis 19).

```

if ($oldversion < 2019050101) {
    // Define table sis_grade_settings to be created.
    $table = new xmldb_table('sis_grade_settings');
    // Adding fields to table sis_grade_settings.
    $table->add_field('id', XMLDB_TYPE_INTEGER, '10', null, XMLDB_NOTNULL,
XMLDB_SEQUENCE, null);
    $table->add_field('course_id', XMLDB_TYPE_INTEGER, '19', null,
XMLDB_NOTNULL, null, null);
    <...>
    // Adding keys to table sis_grade_settings.
    $table->add_key('primary', XMLDB_KEY_PRIMARY, array('id'));
    <...>
    // Conditionally launch create table for sis_grade_settings.
    if (!$dbman->table_exists($table)) {
        $dbman->create_table($table);
    }
    // Sis savepoint reached.
    upgrade_plugin_savepoint(true, 2019050101, 'gradeexport', 'sis');
}

```

Joonis 19. Plugina andmebaasi uuendamine.

Koodis (Joonis 19) on \$dbman andmebaasi halduri API.

4.4 JavaScript moodulid

JavaScripti moodul on JavaScripti koodi kogum, mida saab kasutada (usaldusväärselt) teistes JavaScripti moodulites. [4]

Pärast Moodle'i versiooni 2.9 on soovitatav kasutada JavaScripti AMD moodulit plugina jaoks. Selle kasutamise jaoks on vaja installeerida Grunt. Grunt on käsurea tööriist, mida kasutatakse rohkem JavaScripti ettevalmistamiseks ja vähem genereeritud CSS tootmiseks. Pärast seda, kui JavaScripti muudatus on tehtud, on vaja käivitada käsurealt (*grunt amd*), et korrigeerida ja pakendada JavaScripti nii, et seda saaks Moodle kasutada. [12]

JavaScripti moodulid töötavad kasutades RequireJS raamastikku. RequireJS on JavaScripti faili ja mooduli laadija. See on optimeeritud brauseris kasutamiseks, kuid seda saab kasutada ka teistes JavaScript-keskkondades, nagu Rhino ja Node. [13]

AMD mooduli jaoks on vaja luua 2 kausta: *amd/src* ja *amd/build*. Juhul, kui on lisatud kolmanda osapoole JavaScripti raamastik, siis see on vaja panna *js* kausta. Pluginas on kasutatud jQuery-csv raamastikku.

Kolmanda osapoole JavaScripti raamastik lisatakse kasutades RequireJS. Selle jaoks on vaja teha *amd/src/config.js* faili (Joonis 20).

```
define([], function() {
  window.requirejs.config({
    paths: {
      "jquery-csv": M.cfg.wwwroot + '/grade/export/sis/js/jquery-csv-0.8.9/jquery.csv'
    },
    shim: {
      "jquery-csv": {deps: ["jquery"], exports: "$.csv"}
    }
  });
});
```

Joonis 20. Kolmanda osapoole raamastiku konfigureerimine.

Koodis on *path* tee failini, kust süsteem saab leida JavaScripti faili, ja *shim* on konfiguratsioon faili jaoks. Konfiguratsiooni kaudu laaditakse vajalikud raamastikud ("jquery") ja pannakse vajalik nimi ("\$.csv") mooduli jaoks. Pärast seda on vaja defineerida kolmanda osapoole JavaScripti raamastik kui AMD moodul, mis toimub failis *amd/src/jquery-csv.js* (Joonis 21).

```
define(['gradeexport_sis/config', 'jquery-csv'], function(unused, csv) {
  return csv;
});
```

Joonis 21. AMD mooduli defineerimine.

AMD mooduli kasutamiseks on piisav defineerimisel kirjutada mooduli nimi (Joonis 22):

```
define([
  'jquery',
  'core/templates',
  'core/notification',
  'core/ajax',
  'core/str',
  'gradeexport_sis/jquery-csv'
], function($, templates, notification, ajax, str, csv) {<...>});
```

Joonis 22. AMD mooduli kasutamine.

Koodis (Joonis 22) on imporditud *gradeexport_sis/jquery-csv*, mis on kolmanda osapoole raamastiku AMD mooduli nimi.

5 Kokkuvõte

Töö eesmärgiks oli arendada Moodle'i plugin, mis lihtsustab ja teeb mugavaks õpetulemuste ülekandmise Moodle'st ÕIS-i. Plugin pidi olema lihtne ja struktureeritud, mis võimaldaks seda kasutada ka teistes Moodle'i õppekeskkondades.

Eesmärgi saavutamiseks on välja töötatud spetsiaalne Moodle'i plugin. Kasutatud on erinevaid Moodle'i funktsionaalsusi ja API-sid: *Templates*, *String API*, AMD moodulid, RequireJS, jQuery-csv, XMLDB redaktor, *External API* (veebiteenused), AJAX moodul.

Tudengite vastavuste leidmiseks on välja töötatud kaks varianti: kasutades teenust, mis on kättesaadav ainult ained.ttu.ee keskkonnas, ja nime järgi.

Kogu suhtlus kasutaja ja serveri vahel toimub turvaliselt. Kõiki andmeid on võimalik saada ainult juhul, kui on olemas seansi võti, mis on erinev iga kasutaja seansi jaoks.

Pluginal on mitmekeelsuse tugi, mis võimaldab tõlkimist teistesse keeltesse. Kasutamisel osa parameetritest salvestatakse andmebaasi, mis säästab aega, sest kasutajal ei ole vaja iga kord sisestada eksportimiseks vajalikke andmeid. Salvestamine toimub kasutaja id ja kursuse id põhjal.

Moodle'i plugina lähtekood on kättesaadav lingil: <https://gitlab.cs.ttu.ee/alkond/moodle-sis-export-plugin>.

Kasutatud kirjandus

- [1] String API - MoodleDocs. [WWW] https://docs.moodle.org/dev/String_API (04.05.2019)
- [2] Templates - MoodleDocs. [WWW] <https://docs.moodle.org/dev/Templates> (04.05.2019)
- [3] AJAX - MoodleDocs. [WWW] <https://docs.moodle.org/dev/AJAX> (05.05.2019)
- [4] Javascript Modules - MoodleDocs. [WWW] https://docs.moodle.org/dev/Javascript_Modules (05.05.2019)
- [5] Security:Cross-site request forgery - MoodleDocs. [WWW] https://docs.moodle.org/dev/Security:Cross-site_request_forgery (06.05.2019)
- [6] Web services - MoodleDocs. [WWW] https://docs.moodle.org/dev/Web_services (06.05.2019)
- [7] Adding a web service to a plugin - MoodleDocs. [WWW] https://docs.moodle.org/dev/Adding_a_web_service_to_a_plugin (07.05.2019)
- [8] Plugin types - MoodleDocs. [WWW] https://docs.moodle.org/dev/Plugin_types (08.05.2019)
- [9] Gradebook export - MoodleDocs. [WWW] https://docs.moodle.org/dev/Gradebook_export (08.05.2019)
- [10] version.php - MoodleDocs. [WWW] <https://docs.moodle.org/dev/version.php> (08.05.2019)
- [11] XMLDB defining an XML structure - MoodleDocs. [WWW] https://docs.moodle.org/dev/XMLDB_defining_an_XML_structure (11.05.2019)
- [12] Grunt - MoodleDocs. [WWW] <https://docs.moodle.org/dev/Grunt> (12.05.2019)
- [13] RequireJS. [WWW] <https://requirejs.org/> (12.05.2019)
- [14] Overview - MoodleDocs. [WWW] <https://docs.moodle.org/dev/Overview> (12.05.2019)
- [15] GitHub - typeiii/jquery-csv: A jQuery CSV parser plugin. Battle Tested | Optimized | 100% IETF RFC 4180 Complete. [WWW] <https://github.com/typeiii/jquery-csv> (12.05.2019)
- [16] Mis on Moodle? – eTÜ. [WWW] <https://etu.ut.ee/kevad-2010/mis-on-moodle/> (16.05.2019)