

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

IT40LT
Kenn Jaakson 142685

**PROGRAMMEERIMISÜLESANNETE
KOGUMIKE TULEMUSTE
INTEGREERIMISE PISTIKPROGRAMM
MOODLE'I ÕPIKESKKONNALE**

Bakalaureusetöö

Juhendaja: Ago Luberg
MSc

Tallinn 2017

Autorideklaratsioon

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

Autor: Kenn Jaakson

22.05.2017

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on luua Moodle'i pistikprogramm, mis lihtsustab kolmandate osapoolte poolt loodud automatiseeritud hindamissüsteemiga ülesannete kogumike (näiteks UVa Online Judge) kasutamist õppetöös. Töö tulemus võiks pakkuda huvi kõigile õppejõududele, kes on soovinud kasutada mõnda sellist veebiteenust kursuse raames.

Töö kirjeldab pistikprogrammi loomisel kasutatud tehnoloogiaid ning nende omavahel sidumist, arhitektuuri, ettetulnud probleeme, kasutajaliidest ning integreerimist teiste sarnaste ülesannete kogumikega.

Töö tulemuseks on töötav ja paindlik Moodle'i pistikprogramm, mida saab kasutada kursuse õppetööle ülesannete lisamiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 30 leheküljel, 4 peatükki, 13 joonist.

Abstract

Moodle plugin for integrating programming problem collections results

The main objective of this thesis is to develop a Moodle plugin. The plugin simplifies the usage of third party automated programming problem judges (such as UVa Online Judge) in coursework. The plugin could be of interest to any lecturer interested in using such webservices in their coursework.

The thesis describes the technologies used and how they fit together, the architecture, the user interface and explains how to integrate with other similar third party services.

The result of this thesis is a working Moodle plugin with many customisable features that can be used for adding problems from problem services to a course.

The thesis is in Estonian and contains 30 pages of text, 4 chapters, 13 figures.

Lühendite ja mõistete sõnastik

Dependency injection, DI	<i>Dependency injection</i> ehk sõltuvuse sisestamine on metoodika, kus üks objekt täidab teise objekti sõltuvused.
Laravel	Avatud lähtekoodiga PHP-s kirjutatud veebiraamistik.
Moodle	Moodle ¹ (<i>Modular Object-Oriented Dynamic Learning Environment</i>) on avatud lähtekoodiga kursustehaldamisesüsteem.
MVC	<i>Model-View-Controller</i> on tarkvara arhitektuuri muster, mis jagab tarkvararakenduse kolmeks osaks.
Object-relational mapper, Object-relational mapping, ORM	<i>Object-relational mapping</i> on metoodika, kus luuakse virtuaalne objektide andmebaas eesmärgiga lihtsustada andmebaasiga töötamist.
PHP	<i>PHP: Hypertext Preprocessor</i> ² on programmeerimiskeel.
Plugin	<i>Plugin</i> ehk pistikprogramm on tarkvaramoodul, mis lisab mõnele teisele tarkvarasüsteemile funktsionaalsust.

¹ <https://moodle.org/>

² <http://php.net/>

Sisukord

1 Sissejuhatus	8
1.1 Eesmärk ja oodatav tulemus	8
1.2 Metoodika	9
2 Analüüs	10
2.1 Kasutusjuhud	10
2.2 Nõuded	10
2.2.1 Funktsionaalsed nõuded	10
2.2.2 Mittefunktsionaalsed nõuded	11
2.3 Olemi-suhte diagramm	12
2.4 Jadadiagramm	13
2.4.1 Kasutaja esituste saamise ning hinde lisamise jadadiagramm	13
3 Arendus	15
3.1 Tehniline ülevaade	15
3.1.1 Moodle	15
3.1.2 Moodle'i pistikprogrammi loomine	15
3.1.3 Laravel	18
3.1.4 Moodle'i sidumine Laraveliga	18
3.2 Pistikprogramm Castor	21
3.2.1 Vorm	21
3.2.2 Probleemi kogumike liidestamine	24
3.2.3 Grupid	27
3.2.4 Tähtajad	28
3.2.5 Ülesande vaade	29
4 Kokkuvõte	35
4.1 Edasised arendused	35
Kasutatud kirjandus	37

Jooniste loetelu

Joonis 1. Olemi-suhte diagramm.	13
Joonis 2. Kasutaja esituste saamise ning hinde lisamise jadadiagramm.	14
Joonis 3. Moodle'i activity tüüpi pistikprogrammi kohustuslike failide struktuur.	16
Joonis 4. Üldised vormi seaded.	22
Joonis 5. Probleemi seaded vormil.	22
Joonis 6. Hinnete seaded vormil.	23
Joonis 7. Gruppide seaded vormil.	24
Joonis 8. UVa Online Judge liidestamine pseudokoodina.	26
Joonis 9. UVa Online Judge liidese konfiguratsioon.	27
Joonis 10. Tähtaegade kuvamine kursusel osalejale.	28
Joonis 11. Ülesande informatsiooni vaade.	31
Joonis 12. Castori seadete vaade.	32
Joonis 13. Castori gruppide vaade kursuse haldamise paneelis.	34

1 Sissejuhatus

Kolmandate osapoolte poolt loodud automatiseeritud hindamissüsteemiga ülesannete kogumike (näiteks UVa Online Judge¹) kasutamine õppetöös on võimalik, kuid suure ajakuluga õppejõu jaoks. Õppejõud peaks kõigepealt üles märkima kõigi kursusel osalejate poolt kasutatavad kasutajanimed vastavas süsteemis ning seejärel neid perioodiliselt kontrollima juhuks kui kursusel osaleja on vahepeal lahendanud mõne ülesande. Parem lahendus oleks süsteem, mis suudaks seda teha ilma õppejõu sekkumiseta.

1.1 Eesmärk ja oodatav tulemus

Töö eesmärgiks on arendada süsteem, mis lihtsustab ja mugavdab eelmises lõigus kirjeldatud ülesannete kogumike kasutamist õppetöös. Kuna Tallinna Tehnikaülikooli Infotehnoloogia teaduskonnas on populaarseks õpikeskkonnaks Moodle², siis oleks mõistlik antud süsteem luua Moodle'i pistikprogrammina. Mugavdamise ja lihtsustamise all mõeldakse süsteemi, mis suudab anda kursusel osalejale ülesande esitamise eest hinde Moodle'i süsteemis.

Töö teiseks eesmärgiks on lisada süsteemile võimalus kasutada gruppe, kusjuures gruppide kasutamine peaks olema piisavalt paindlik, et moodustatud gruppe saaks kasutada, kas ühe ülesande, kogu kursuse või ainult harjutustunni vältel.

Töö kolmandaks eesmärgiks on kirjutada kood selliselt, et loodud pistikprogrammi saaks hiljem kergesti edasi arendada ning Moodle'i ametliku laiendusena välja anda.

¹ <https://uva.onlinejudge.org/>

² <https://moodle.org/>

1.2 Metoodika

Töö eesmärkide täitmiseks uuritakse kõigepealt, kuidas arendada Moodle'ile pistikprogrammi ning kas ja kuidas on võimalik Moodle'i pistikprogrammi kirjutamisel kasutada veebiraamistikku Laravel. Pistikprogramm arendamisel kasutatakse PHP programmeerimiskeelt serveri ning SCSS-i ja JavaScripti kliendi poolel.

2 Analüüs

Selles peatükis antakse ülevaade loodava Moodle'i pistikprogrammi kasutusjuhtudest, nõuetest ning tuuakse välja olemi-suhte diagramm ja kasutaja esituste saamise ning hinde lisamise jadadiagramm.

2.1 Kasutusjuhud

Loodava Moodle'i pistikprogrammi kasutusjuhtudeks on:

1. Kursuse haldaja saab luua uue ülesande.
2. Kursuse haldaja saab muuta ülesannet.
3. Kursuse haldaja saab kustutada ülesandeid.
4. Kursusel osaleja näeb ülesannet ning ülesandega seotud informatsiooni (tähtajad, grupi liikmed, ülesandele viitav link, esitused, hinded).
5. Kursuse haldaja näeb ülevaadet kursusel osalejatest (tulemused, probleemi kogumikega seotud kasutajad).
6. Kursusel osaleja saab luua ning liituda grupiga.
7. Kursuse haldaja näeb loodud gruppe ning saab neid muuta.
8. Kursusel osaleja saab uuendada oma esitusi.
9. Kursuse haldaja saab uuendada kursusel osalejate esitusi.
10. Kursuse haldaja saab dubleerida ülesannet.
11. Kursuse haldaja saab määrata vaikimisi seaded ülesande loomiseks.

2.2 Nõuded

Selles peatükis kirjeldatakse loodava Moodle'i pistikprogrammi funktsionaalseid ja mittefunktsionaalseid nõudeid.

2.2.1 Funktsionaalsed nõuded

1. Laiendus peab toetama Moodle'i sisseehitatud funktsionaalsust (näiteks dubleerimine, taastamine, kustutamine, peitmine jne).

2. Kursuse haldaja peab saama määrata ülesande kogumiku ja ülesande (kasutades ülesande kogumiku veebilehel olevat ülesande unikaalset identifikaatorit).
3. Kursuse haldaja peab saama valida ning muuta vaikimisi hinnet ning tähtaegu, mille möödumisel saadav hinne väheneb.
4. Kursuse haldaja peab saama lisada ning muuta ülesande (ka hindekategooria) nime ja kirjeldust.
5. Kursuse haldaja peab saama valida, kas kasutada gruppe või mitte.
6. Süsteem peab võimaldama määrata grupiliikmete maksimaalse arvu.
7. Süsteem peab võimaldama varem loodud gruppide taaskasutamist.
8. Süsteem peab võimaldama määrata grupis töötamise ajavahemikku (millal läheb esitus kirja grupi esitusena, millal mitte).
9. Süsteem peab oskama suhelda ülesande kogumike süsteemidega (saada kasutaja ja ülesannete kohta informatsiooni).
10. Kursusel osaleja peab saama siduda oma Moodle'i kasutajat ülesande kogumiku kasutajaga.
11. Kursusel osaleja peab saama luua grupi, kellega koos ülesandeid lahendatakse.
12. Grupi looja peab saama eemaldada grupi liikmeid.
13. Kursusel osaleja peab saama grupiga liituda ning grupist lahkuda.
14. Kursuse haldaja peab nägema ülevaadet gruppidest ning saama neid muuta (näiteks eemaldada grupi liikmeid).
15. Kursuse haldaja peab saama määrata vaikimisi seadeid uue ülesande loomiseks (näiteks vaikimisi hinne, tähtajad, ülesande kogumik).
16. Kursuse haldaja peab nägema ülevaadet kursusel osalejate esitustest (millal esitati, saadud hinne, millist programmeerimiskeelt kasutati).
17. Kursuse haldaja peab nägema ülevaadet kursusel osalejate poolt seotud kontodest ülesande kogumikega ning saama neid muuta.

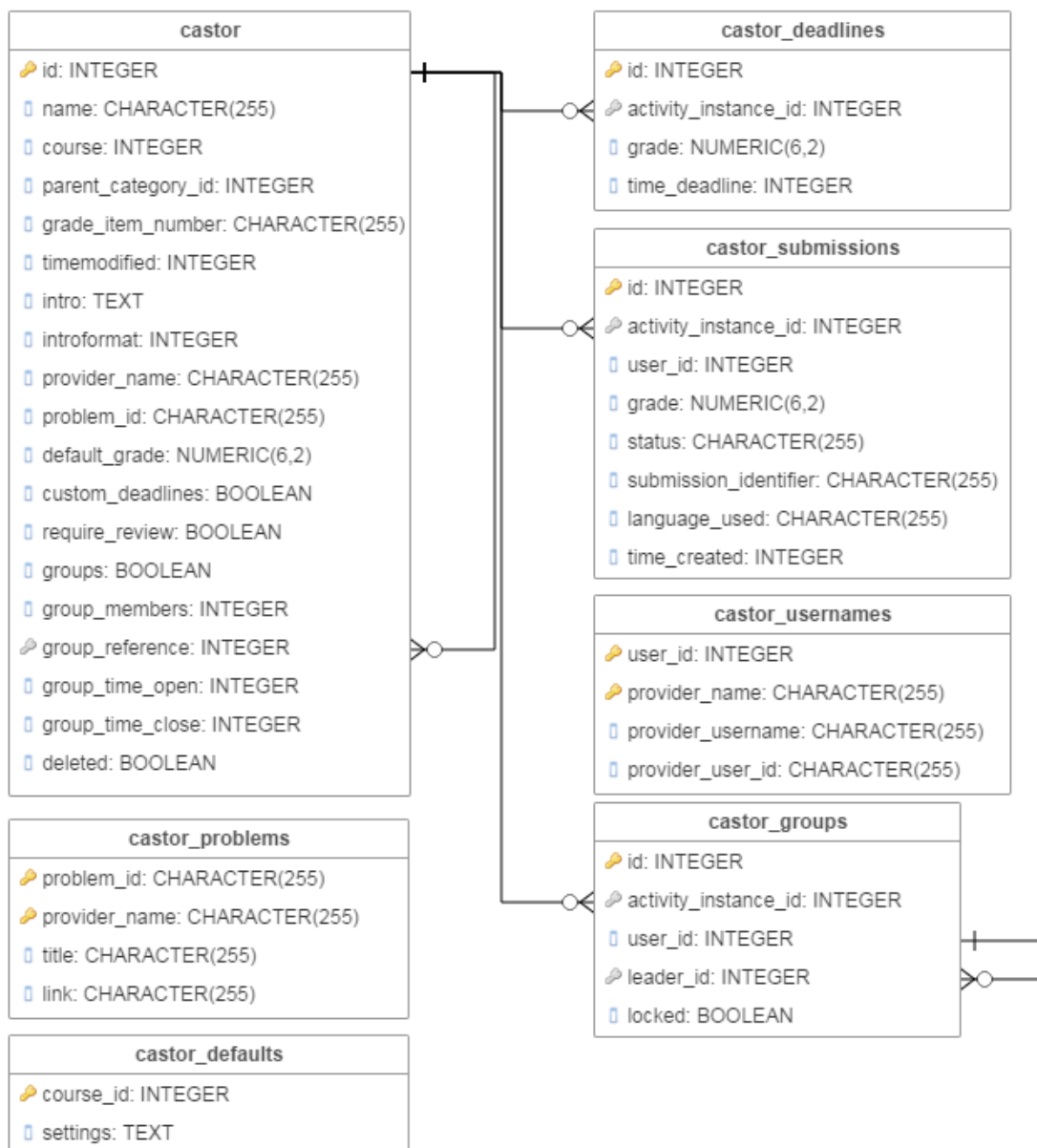
2.2.2 Mittefunktsionaalsed nõuded

1. Loodavat laiendust peaks olema võimalik välja anda Moodle'i ametliku laiendusena.
2. Laiendus luuakse *activity* moodulina.
3. Loodavat laiendust peaks olema võimalik liidestada erinevate ülesannete kogumikega.

4. Laiendus peaks töötama erinevate relatsiooniliste andmebaasi haldamise süsteemidega (näiteks MySQL ja PostgreSQL).
5. Laiendus peab olema kasutatav ka tahvelarvutil ning mobiilil.
6. Laienduse kood peab olema inglisekeelne.
7. Laienduse kasutajaliides peab olema saadaval nii eesti kui ka inglise keeles.
8. Laiendus peab kasutama MVC (*Model-View-Controller*) tarkvara arhitektuuri mustrit.
9. Laiendus peab olema dokumenteeritud.

2.3 Olemi-suhte diagramm

Järgnevalt on Joonis 1 ära toodud laienduse olemi-suhte diagramm. Süsteemi põhiliseks objektis on castor.

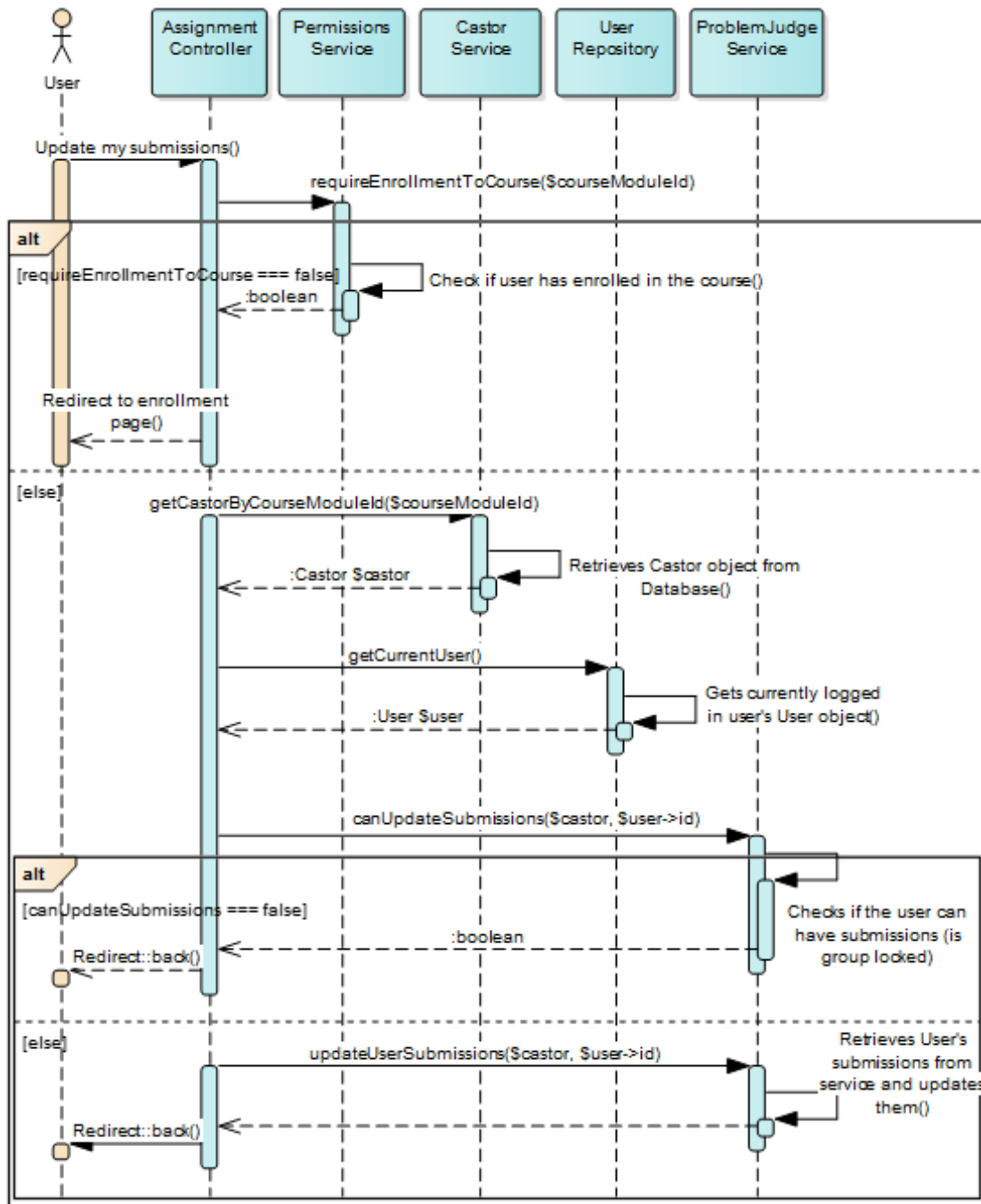


Joonis 1. Olemi-suhte diagramm.

2.4 Jadadiagramm

2.4.1 Kasutaja esituste saamise ning hinde lisamise jadadiagramm

Kasutaja esituse saamise ning hinde lisamise jadadiagramm asub Joonis 2.



Joonis 2. Kasutaja esituste saamise ning hinde lisamise jadadiagramm.

3 Arendus

Selles peatükis antakse ülevaade pistikprogrammi loomisel kasutatud tehnoloogiatest, nende sidumisest ning pistikprogrammi funktsionaalsusest.

3.1 Tehniline ülevaade

Selle peatüki eesmärk on anda ülevaade töös kasutatud tehnoloogiatest ning nende sidumisest tervikuks.

3.1.1 Moodle

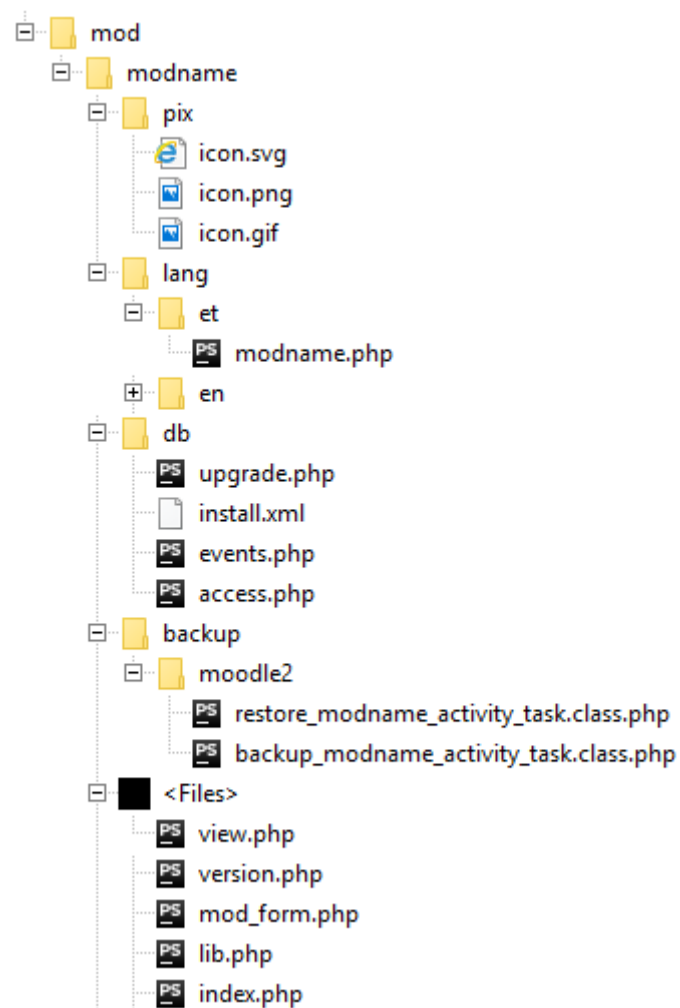
Moodle (*Modular Object-Oriented Dynamic Learning Environment*) on avatud lähtekoodiga õpikeskkond, mille eesmärgiks on pakkuda õpetajatele, administraatoritele ja õppijatele robustset, turvalist ja integreeritavat süsteemi isikupärastatud õpikeskkonna loomiseks [1]. Moodle'i esimene versioon avalikustati aastal 2001 Martin Dougiamase poolt [2]. Moodle'i keskkonda on maailmas installeeritud enam kui 80 000 serverile, millega hallatakse üle 100 miljoni kasutaja ning üle 12 miljoni kursuse 233 riigis [3].

Moodle'il on palju erinevaid funktsioone alates erinevatest ülesandetüüpidest ning lõpetades erinevate kursuste struktuuridega. Selleks, et Moodle'i funktsionaalsust veelgi laiendada, vastavalt kasutaja soovidele, on Moodle kasutusele võtnud *plugin*'id. Tänu Moodle'i suurele kasutajate arvule on Moodle'i ametlikus laienduste kataloogis saadaval üle 1300 erineva *plugin*'i [4]. Tänu Moodle'i ametlikule laienduste kataloogile on võimalik laiendusi lisada kiirelt ning turvaliselt. Lisaks funktsionaalsuse laiendamisele on võimalik kergesti muuta Moodle'i välimust ning tõlkida kogu süsteem sobivasse keelde, hetkel on Moodle tõlgitud üle 100 erinevasse keelde [5].

3.1.2 Moodle'i pistikprogrammi loomine

Kõige lihtsam ja hooldatavam viis Moodle'ile lisafunktsionaalsuse lisamiseks on luues *plugin* ehk pistikprogramm. Pistikprogrammide tegemise lihtsustamiseks on Moodle'i kogukond loonud üle 50 erineva standardiseeritud pistikuprogrammitüübi ning nende loomise töökäigu dokumenteerinud, kui aga ükski standardiseeritud tüüpidest ei sobi, saab luua ka täiesti uue tüübi luues *local* tüüpi pistikuprogramm [6]. Selle peatüki eesmärgiks on anda ülevaade *activity* tüüpi pistikprogrammi loomisest.

Kõige enam kasutataksegi *activity* tüüpi pistikprogramme, millega saab lisada tegevusi kursuste all (näiteks *Forum*, *Quiz*, *Assignment*). *Activity* tüüpi pistikprogrammid asuvad kaustas `/mod`. Uue *activity* tüüpi pistikprogrammi loomiseks on vaja luua kaust `/mod/<modname>`, kus `<modname>` on pistikprogrammile antud nimi. Kausta `/mod/<modname>` sisu koosneb mitmest kohustuslikust failist ning muudest failidest, mida arendaja kasutab, et juhtida pistikprogrammi tegevust. Kohustuslikud failid kirjeldavad seda, kuidas peaks pistikprogrammi installeerima, deinstalleerima ning siduma Moodle'i süsteemiga. Järgnevalt Joonis 3 on välja toodud *activity* tüüpi pistikprogrammi kohustuslike failide struktuur [7].



Joonis 3. Moodle'i *activity* tüüpi pistikprogrammi kohustuslike failide struktuur.

Kohustuslike failide sisu kirjeldus:

- `db` – installeerimisel ja deinstalleerimisel kasutatavad failid.

- `install.xml` – defineeritakse pistikprogrammi poolt kasutatava andmebaasi tabelite ja väljade struktuur XML (*Extensible Markup Language*) formaadis.
- `events.php` – defineeritakse sündmuste kuulajad, näiteks kui luuakse või muudetakse ülesannet.
- `upgrade.php` – defineeritakse instruktsioonid pistikprogrammi ühelt versioonilt teisele uuendamiseks.
- `access.php` – defineeritakse pistikprogrammi kasutamise õigused ning kontekst (näiteks kursus).
- `pix` – pistikprogrammi ikoonid erinevates formaatides.
- `backup` – defineeritakse kuidas dubleerida ning hiljem taastada moodulid.
 - `moodle2` – dubleerimise ja taastamise skeemi mudeli nimetus.
 - `restore_<modname>_activity_task.class.php` – defineerib moodulite taastamise protsessi.
 - `backup_<modname>_activity_task.class.php` – defineerib moodulite dubleerimise protsessi.
- `lang` – pistikprogrammi tõlked.
 - `et` – keelenime lühend.
 - `<modname>.php` – sisaldab antud keele tõlkeid.
- `view.php` – mooduli vaade, näiteks ülesande sisu.
- `version.php` – defineerib pistikprogrammi versiooni ning installeerimiseks vajaliku Moodle'i versiooni.
- `mod_form.php` – defineerib mooduli lisamise, muutmise vormi ning valideerimise loogika.
- `lib.php` – defineerib mooduli salvestamise, uuendamise ja kustutamise loogika.
- `index.php` – kuvab listi pistikprogrammi kõikidest moodulitest, mis on seotud avatud kursusega.

3.1.3 Laravel

Laravel¹ on avatud lähtekoodiga, tasuta, PHP (*PHP: Hypertext Preprocessor*) programmeerimiskeeles kirjutatud veebiraamistik. Pistikprogrammi loomisel kasutatakse Laraveli versiooni 5.3, mis nõuab PHP versiooni 5.6.4 või uuemat ning PHP-l peavad olema aktiveeritud järgnevad laiendused: OpenSSL, PDO, Mbstring, Tokenizer ja XML. Laravel on loodud selleks, et lihtsustada ning kiirendada hea ülesehituse ja struktuuriga nii suurte kui ka väikeste veebirakenduste loomist. Veebirakenduse arendamist lihtsustavad ja kiirendavad Laraveli valmiskomponendid, mida kasutatakse tihti veebirakenduste loomisel.

Nendest komponentidest olulisemateks on marsruutimine (*routing engine*), sõltuvuse sisestamise konteiner (*dependency injection container*), andmebaasi ORM (*Object-relational mapper*) [8] . Marsruudi mootor viib päringu vastavalt defineeritud marsruutidele õigesse kontrollerrisse. Sõltuvuse sisestamise konteiner võimaldab sõltuva objekti käitumist eraldada tema loomise protsessist, mis annab võimaluse kirjutada madala sõltuvusega (*loosely coupled*) tarkvara [9] . *Object-relational mapping* tähendab objektide talletamist andmebaasis mingi kindla eeskirja järgi ning võimaldab arendajal mitte muretseda selle pärast, kuidas on andmed salvestatud – programmeerija töötleb andmeid läbi objektide.

Laravel kasutab MVC (*model-view-controller*) arhitektuuri mustrit. *Model* ehk mudel iseloomustab andmeid andmebaasis objektide näol. *View* ehk vaade näitab kasutajale mudeli informatsiooni, tagastab kasutaja toimingud kontrolleriile. *Controller* ehk kontrolleri seob mudeli ja vaate, edastab vaatele vajaliku informatsiooni vaate loomiseks ning tõlgendab kasutaja tegevusi (näiteks vajutusi nupule) [10] .

3.1.4 Moodle'i sidumine Laraveliga

Moodle'i sidumise Laraveliga võib jaotada kahte osasse: Moodle'i päringute edastamine Laravelile ning Moodle'i funktsioonide kasutamine Laravelis.

¹ <https://laravel.com/>

Moodle'i päringute edastamine Laravelile

Moodle'i päringud on võimalik Laravelile edastada kasutades Apache¹ `.htaccess` konfiguratsiooni faili, mis võimaldab muuta konfiguratsiooni kausta põhiselt [11]. Eelnevalt mainitud konfiguratsiooni failis tuleb luua reeglid, millega suunatakse kõik `/mod/<modname>/` kausta tulnud päringud edasi Laravelile. Selliseks ümbersuunamiseks on vajalik, et Apache oleks aktiveeritud `mod_rewrite` moodul [12]. Sellist ümbersuunamist saab teha ka kasutades teisi veebiservereid, näiteks `nginx`'i² puhul tuleks selline suunamine kirjutada virtuaalse *hosti* konfiguratsiooni faili.

Eelnevas lõigus pakutud lahendus aga ei toimi mooduli loomisel, muutmisel ega kustutamisel. Kuna lisamine, muutmine ja kustutamine toimub Moodle'i enda koodis (kutsutakse välja `/mod/<modname>/lib.php` failis olevaid funktsioone), siis tuleb `/mod/<modname>/lib.php` failis kõigis funktsioonides kopeerida päringu sisu ning luua päringu objekt, mida Laravel tunnistab, ning anda päringu vastuse loomine Laraveli HTTP (*Hypertext Transfer Protocol Kernel*)'ile üle.

Moodle'i funktsioonide kasutamine Laravelis

Moodle'i funktsioonide kasutamiseks Laravelis kasutati Joosep Franz Moorits Alviste poolt loodud Moodle'i mähist [13]. Mähis kutsub välja Moodle'i funktsioone „Laraveli stiilis“, mis tähendab, et programmikood on organiseeritud klassideks ning kasutatakse objektorienteeritud programmeerimise võtteid (kapseldamine, modulaarsus, pärimine jne) [13]. Mähis on jaotatud järgmiselt: teenused (*Services*), mudelid (*Models*), globaalsed muutujad (*Globals*). Mudelid on loodud tabelitele *course*, *course_modules*, *grade_categories*, *grade_grades*, *grade_items*, *groups*, *modules*, *config_plugins* ja *user*. Mudelid on vaja selleks, et saaks pistikprogrammis luua seoseid pistikprogrammi enda mudelite ja Moodle'i mudelite vahel.

¹ <https://www.apache.org/>

² <http://nginx.org/>

Teenused:

- *GradebookService* – mähis Moodle'i *Gradebook* funktsioonidele. Võimaldab lisada hindeid, hinnete kategooriaid ning neid muuta.
- *GradingService* – võimaldab kursusel osalejale hindeid lisada või neid muuta.
- *LocalizationService* – tõlgib sõne vastavalt kasutaja eelistusele.
- *ModuleService* – tagastab pistikprogrammi unikaalse identifikaatori numברי Moodle'i moodulite tabelist.
- *PermissionsService* – võimaldab kontrollida, kas kasutaja on sisselogitud või mitte ning kas kasutajal on piisavad õigused, et näha või muuta kursuse lehte.
- *UserService* – Moodle'i kasutajaga seotud funktsioonid.

Globaalsed muutujad:

- *Course* – sisaldab Moodle'i globaalset muutujat \$COURSE.
- *Output* – sisaldab Moodle'i globaalset muutujat \$OUTPUT.
- *Page* – sisaldab Moodle'i globaalset muutujat \$PAGE.
- *User* – sisaldab Moodle'i globaalset muutujat \$USER.

Laraveli teekide installeerimine

Moodle'i pistikprogramme on võimalik installeerida läbi Moodle'i pistikprogrammide kataloogi (*Moodle plugins directory*), kasutades zip faili pakitud versiooni või tõstes pistikprogrammi sisu vastavasse kausta, kust Moodle ta üles leiab. Zip failist pistikprogrammi installeerimisel käivitab Moodle kontrolli, mis vaatab üle pistikprogrammi failid. Kuna kõik failid käiakse läbi mitu korda ja Laraveli teekidel on kokku üle 3000 faili, siis võtab see protsess liialt kaua aega ja installeerimine ei õnnestu.

Eelnevalt kirjeldatud kontrolli vältimiseks tuleb Laraveli teegid laadida hiljem pistikprogrammi installeerimise käigus. Üheks võimaluseks on luua `/mod/<modname>/install.php` fail, kus laetakse alla ning käivitatakse Composer¹, mis on laeb vajalikud teegid `/mod/<modname>/composer.json` faili põhjal.

¹ <https://getcomposer.org/>

Composer on vahend teekide haldamiseks PHP-s [13] . Composer installeerib teegid projekti põhiselt, vaikimisi kausta `vendor`. Pakutud lahenduse heaks küljeks on ka see, et alati kui pistikprogrammi installeeritakse või uuendatakse, laetakse ka teekide viimased versioonid.

3.2 Pistikprogramm Castor

Selles peatükis antakse ülevaade Moodle'ile loodud pistikprogrammist Castor. Castor on pistikprogrammile antud nimi ning ei oma mingit erilist tähendust.

3.2.1 Vorm

Selles peatükis antakse ülevaade pistikprogrammi Castor poolt kasutatavast mooduli lisamise ja muutmise vormist ehk üleüldiselt Castori seadete vormist.

Pistikprogrammi Castor vormi tegemisel kasutati Moodle'i sisseehitatud funktsioone, sest vormi elementide HTML (*Hypertext Markup Language*) struktuuri leidmine Moodle'i lähtekoodist osutus liiga ajakulukaks (Moodle ei kasuta malle, vaid HTML-i struktuur genereeritakse programmikoodis). Selline lähenemine tagab ka selle, et pistikprogramm töötab enamiku Moodle'i teemadega (disainidega).

Vormi saab jaotada järgnevalt: üldised seaded, probleemi seaded, hinde seaded, gruppide seaded ning Moodle'i poolt pakutud seaded.

Üldiste seadete kirjeldus (Joonis 4):

- *Name* – hinde kategooria ning ühtlasi ka mooduli nimi.
- *Description* – kursuse haldajal on võimalik lisada ülesande juurde kirjeldus, näiteks mõni vihje ülesande kohta või täpsustavad detailid. Välja tühjaks jätmisel komponenti ülesande vaates ei kuvata.

▼ General

Name ?

Description

Rich text editor toolbar with icons for text formatting (bold, italic, underline, link, unlink, list, link), image insertion, and other editing functions.

Joonis 4. Üldised vormi seaded.

Probleemi seadete kirjeldus (Joonis 5):

- *Problem provider name* – valik erinevate probleemi kogumike vahel (täpsemalt sellest peatükis 3.2.2).
- *Problem ID* – probleemi unikaalne identifikaator.

▼ Problem settings

Problem provider name *

Problem ID *

Joonis 5. Probleemi seaded vormil.

Hinnete seadete kirjeldus (Joonis 6):

- *Choose parent grade category* – valik kategooriatest, kuhu alla saaks Castor oma hindekategooria luua.
- *Default grade* – määrab vaikimisi hinde, kui ühtegi tähtaega ei ole möödunud või tähtajad puuduvad.
- *Require review* – lisab uue hinde Castori hindekategooria alla, millega korrutatakse ülesande eest saadud hinne läbi. Eesmärgiks on anda kursuse haldajale võimalus lasta kursusel osalejatel oma ülesande lahendust selgitada.
- *Custom deadlines* – võimaldab aktiveerida tähtaegade kasutamise.
- *Custom deadline grade x* – hinne ülesande lahendamise eest tähtaja möödudes.

- *Custom deadline x* – tähtaeg, mille möödumisel saab kursusel osaleja ülesande esitamise eest eelmisel väljal määratud hinde.
- *Add 1 field(s) to form* – lisab uue *custom deadline grade*'i ja *custom deadline*'i ehk võimaldab lisada mitu tähtaega ülesandele.

▼ Grade

Choose parent grade category ?

Default grade * ?

Require review ?

Custom deadlines ?

Custom Deadline Grade 1 ?

Custom Deadline 1








Add 1 field(s) to form

Joonis 6. Hinnete seaded vormil.

Gruppide seadete kirjeldus (Joonis 7):

- *Groups* – võimaldab aktiveerida grupis ülesande lahendamise.
- *Group reference* – lubab valida uute gruppide loomise ning mõne teise mooduli gruppide kasutamise vahel.
- *Maximum members* – määrab grupiliikmete maksimaalse arvu.
- *Count submissions as a group from* – saab määrata, mis ajast alates antakse ülesande lahenduse eest hinded kogu grupile.
- *Count submissions as a group to* – saab määrata, mis ajani antakse ülesande lahenduse eest hinded kogu grupile.

▼ Groups

Groups		Enabled						
Group reference		Group for this instance						
Maximum members		2						
Count submissions as a group from		14	May	2017	14	00		<input type="checkbox"/> Enable
Count submissions as a group to		14	May	2017	14	00		<input type="checkbox"/> Enable

Joonis 7. Gruppide seaded vormil.

Moodle'i poolt pakutud seaded:

- *Visible* – määrab mooduli nähtavuse.
- *Access restrictions* – keelab tingimuste põhjal osale kasutajatest ligipääsu.
- *Tags* – saab lisada silte moodulile.

3.2.2 Probleemi kogumike liidestamine

Selles peatükis antakse ülevaade probleemi kogumike liidestamisest pistikprogrammiga Castor. Liidestamine võimaldab kasutada erinevate veebiteenuste poolt pakutavaid probleemi kogumikke. Liidestamise võib jaotada kahte osasse: liidese programmeerimine ning konfigureerimine.

Liidese programmeerimine

Liidestamiseks on loodud abstraktne klass `ProblemService`. Liidese loomiseks oleks vaja luua uus klass, mis laiendab abstraktset klassi `ProblemService`. Castori faili struktuur näeb ette, et probleemi kogumike klassid asuvad `/mod/castor/plugin/app/Services/ProblemServices` kaustas. Joonis 8 on välja toodud UVa Online Judge liidestamine pseudokoodina. Abstraktne klass `ProblemService` näeb ette, et implementeeritud oleksid järgmised meetodid:

- `usernameToUserId($username)`, kus `$username` on probleemi kogumiku kasutajanimi sõnena. Meetod peaks konverteerima probleemi kogumiku kasutajanime vastava probleemi kogumiku kasutaja unikaalseks identifikaatoriks,

mille järgi on võimalik leida kasutaja poolt tehtud probleemide lahenduste esitused (selleks võib olla ka kasutajanimi ise). Juhul kui kasutajanimi ei eksisteeri peaks meetod tagastama null'i.

- `getAllSubmissionsByUserId($providerUserId)`, kus `$providerUserId` on `usernameToUserId` meetodis tagastatud sõne. Meetod peaks tagastama massiividest koosneva listi, mille massiivide võtmeteks on `submission_id` (sõne, mille järgi saab erinevaid esitusi eristada, näiteks esituse unikaalne identifikaator), `problem_id` (probleemi unikaalne identifikaator sõnena), `status` (probleemi kogumiku poolt antud staatus esitusele, näiteks „Accepted“, „Time limit“), `time_created` (esituse loomise aeg Unix'i ajana), `language_used` (programmeerimiskeel, mida kasutati esituse tegemiseks).
- `getProblemSubmissionsByUserid($problemId, $providerUserId)`, kus `$problemId` on probleemi unikaalne identifikaator ning `$providerUserId` on `usernameToUserId` meetodist tagastatud sõne. Meetod peaks tagastama ainult ühe probleemi kohta tehtud esitused listina, mis koosneb massiividest, mille võtmed on samad, mis `getAllSubmissionsByUserId` meetodis kirjeldatud massiivil.
- `getProblemInformation($problemId)`, kus `$problemId` on probleemi unikaalne identifikaator. Meetod peaks tagastama massiivi, millel on kaks võtit: `title` (probleemi nimi) ja `link` (veebiaadress, mis viitab probleemi kirjeldust hoidvale veebilehele). Meetod peaks tagastama null'i kui antud identifikaatoriga ülesannet ei leitud.
- `filterSubmissions($submissions, $currentSubmissions)`, kus `$submissions` on list massiividest, mis on tagastatud kas `getAllSubmissionsByUserId` või `getProblemSubmissionsByUserid` meetodi poolt ning `$currentSubmissions` on hetkel andmebaasis olemasolevate esituste massiiv. Meetod peaks tagastama listi, milles on alles ainult uute esituste massiivid. Meetodi ülekirjutamine on valikuline, vaikimisi filtreeritakse esitused `submission_id` põhjal.

```

class UvaOnlineJudgeService extends ProblemService {

    func usernameToUserId($username) {
        $userId = makeRequest(API_URL, $username);
        if ($userId) return $userId;
        return null;
    }

    func getAllSubmissionsByUserId($providerUserId) {
        $response = makeRequest(API_URL, $providerUserId);
        if ($response) return formatSubmissions($response);
        return [];
    }

    func getProblemSubmissionsByUserid($probId, $provUserId){
        $response = makeRequest(API_URL, [
            $probId,
            $provUserId
        ]);
        if ($response) return formatSubmissions($response);
        return [];
    }

    private func formatSubmissions($submissions) {
        $formatted = [];
        foreach($submissions as $submission) {
            $formatted[] = [
                'submission_id' => $submission['submission_id'],
                'problem_id' => $submission['problem_id'],
                'status' => $submission['status'],
                'time_created' => $submission['time_created'],
                'language_used' => $submission['language_used'],
            ];
        }
        return $formatted;
    }

    func getProblemInformation($problemId) {
        $response = makeRequest(API_URL, $problemId);
        if ($response)
            return [
                'title' => $response['name'],
                'link' => $response['url'],
            ];
        return null;
    }
}

```

Joonis 8. UVa Online Judge liidestamine pseudokoodina.

Liidese konfigureerimine

Pärast probleemi kogumiku klassi loomist ning selle programmeerimist oleks vaja lisada loodud klass ka konfiguratsiooni, et pistikprogramm Castor selle ära tunneks. Probleemi kogumike konfiguratsioon asub `/mod/castor/plugin/config/problems.php` failis. Antud fail koosneb massiivist, mille võtmeteks on probleemi kogumike lühendatud nimed ning väärtusteks massiivid, mis hoiavad endas probleemi kogumike kohta järgmist informatsiooni: *name* (probleemi kogumiku täisnimi), *short_name* (probleemi kogumiku nime lühend), *class* (eelmises punktis loodud klassi nimi), *status_required_for_grade* (staatused, mille puhul antakse kursusel osalejale hinne). Joonis 9 on välja toodud konfiguratsiooni faili sisu, kus on konfigureeritud UVa Online Judge probleemi kogumik.

```
use TTU\Castor\Services\ProblemServices\UVaOnlineJudgeService;
return [
    'uva' => [
        'name' => 'UVa Online Judge',
        'short_name' => 'UVa',
        'class' => UVaOnlineJudgeService::class,
        'status_required_for_grade' => ["Accepted"]
    ],
];
```

Joonis 9. UVa Online Judge liidese konfiguratsioon.

3.2.3 Grupid

Selles peatükis antakse ülevaade pistikprogrammi Castor poolt kasutatavatest gruppidest.

Gruppide struktuur

Castori grupid toimivad „lobby“ stiilis – üks kasutaja loob grupi ning teised saavad sellega liituda. Grupi loojal on õigus teisi liikmeid grupist välja visata ning gruppi lukustada. „Lobby“ stiilis grupp on vajalik, sest vastasel juhul tekiks küsimus, kellel on õigus otsustada, milline grupi liige võib grupis olla ja milline mitte ning millal on grupi koosseis lõplik. Lisaks vähendab selline lahendus kursuse haldaja tööd, sest kursusel osalejad saavad omavahel otsustada grupi koosseisu üle ning ei pea selleks kaasama kursuse haldajat. Grupiga liitumist ning grupi loomist on kirjeldatud peatükis 3.2.5.

Gruppide kasutusjuhud

Castori gruppide süsteem on väga paindlik, sest võimalik on kasutada mõne teise eelnevalt loodud mooduli gruppe, määrata maksimaalne liikmete arv ning valida millal loetakse esitus grupi esituseks. Selline lähenemine annab võimaluse kasutada samu gruppe kogu kursuse raames või ainult harjutustunni ajal määrates grupi esituste ajavahemiku.

3.2.4 Tähtajad

Selles peatükis antakse ülevaade pistikprogrammi Castor poolt kasutavatest tähtaegadest. Tähtaeg on määratud ajavahemik, millal saab ülesande sooritaja teatud hinde. Tähtaegu on vaja selleks, et kursuse haldajal oleks võimalik anda varem ülesande esitanud kursusel osalejale parem hinne kui hiljem esitanud kursusel osalejale.

Tähtaegu on võimalik lisada mooduli lisamise või hiljem muutmise vormis (Joonis 6). Selleks, et tähtaeg lisada tuleb kõigepealt märkida *Custom deadline*'ide kasutamine aktiivseks, seejärel on võimalik valida hinne, mille ülesande sooritaja saab pärast tähtaja möödumist ning valida tähtaja kuupäev ning kellaaeg. Vajadusel on võimalik lisada ka mitu tähtaega vajutades nupule „Add 1 field(s) to form“. Pistikprogrammis Castor on tähtajad lahendatud järgnevalt: enne tähtaja möödumist saab ülesande esitaja omale hindeks *Default grade* väärtuse, pärast tähtaja möödumist saab ta viimati möödunud tähtajale määratud hinde, kusjuures aja möödudes peavad hinded minema väiksemaks. Hinde välja tühjaks jätmisel tähtaega ei lisata. Tähtaegade kuvamine kursusel osalejale on välja toodud Joonis 10.

Deadlines	
Grade	After
3.00	Before first deadline
1.50	00:00:00 14-05-2017
0.00	15:00:00 31-05-2017

Joonis 10. Tähtaegade kuvamine kursusel osalejale.

3.2.5 Ülesande vaade

Ülesande informatsiooni vaateni jõudmine koosneb kahest etapist olenevalt Castori mooduli loomisel valitud sätetest ning kursusel osaleja varasemast kokkupuutest pistikprogrammiga Castor. Nendeks etappideks on probleemi kogumiku kasutaja sidumine Moodle'i kasutajaga ja grupiga liitumine.

Probleemi kogumiku kasutaja sidumine Moodle'i kasutajaga

Juhul, kui kursusel osaleja pole varasemalt sidunud kasutatava probleemi kogumiku kasutajat oma Moodle'i kasutajaga, siis kuvatakse vaadet, kus on võimalus seda teha. Vaade koosneb ühest sisendi väljast, kuhu kursusel osaleja peaks kirjutama oma kasutajanime probleemi kogumiku lehel. Pärast kasutajanime sisestamist kontrollitakse, kas kasutajanimi eksisteerib ning ega mõni teine Moodle'i kasutaja seda ei kasuta. Kui kontrollid on läbitud suunatakse kursusel osaleja järgmisesse etappi.

Grupiga liitumise vaade

Grupiga liitumise vaadet kuvatakse kasutajale vaid juhul kui kursuse haldaja on aktiveerinud gruppide kasutamise ning kasutaja pole kas liitunud mõne grupiga või tema grupp ei ole veel lukustatud.

Grupiga liitumise vaates on kasutajal kaks võimalust, kas liituda mõne grupiga (kasutades grupi liidri numbrit, mida kuvatakse grupi „ootamise“ vaatel) või luua uus grupp. Uue grupi loomisel muutub kasutaja uue grupi liidriks ning ühtlasi ka selle grupi liikmeks. Kui kasutaja on mõne grupi liige siis kuvatakse talle „ootamise“ vaadet. „Ootamise“ vaates tuleb kasutajal kas grupp lukustada (kui ta on antud grupi liider), kui kõik grupi liikmed on liitunud, või oodata, kuni grupi lukustab grupi liider.

Ülesande informatsiooni vaade

Pistikprogrammi Castor ülesande informatsiooni vaate (Joonis 11) nägemine eeldab, et ühendatud on probleemi kogumiku kasutaja Moodle'i kasutajaga ning kasutaja on kas lukustatud grupi liige või antud ülesannet ei lahendata grupina.

Ülesande informatsiooni vaade on jaotatud järgnevalt:

- *Description* – sisaldab kursuse haldaja poolt sisestatud informatsiooni antud ülesande kohta, komponent kuvatakse juhul kui väli ei ole tühi.
- *My submissions* – sisaldab kursusel osaleja poolt tehtud esituste informatsiooni tabelina. Tabeli veergude päisteks on *submission time* (millal ülesanne esitati), *verdict* (ülesannete kogumiku poolt antud staatus esitusele), *language* (programmeerimiskeel, mida kasutati), *grade* (hinne, mis esituse eest saadi).
- *Group Submissions* – tabeli struktuur on sama, mis *My submissions*'il, kuid kuvatakse kasutaja enda esituste asemel grupi poolt tehtud esitusi, komponent kuvatakse juhul kui antud ülesanne kasutab gruppe.
- *Group members* – grupi liikmete nimekiri, komponent kuvatakse juhul kui antud ülesanne kasutab gruppe.
- *Group submission dates* – näitab kuupäevi, mille vahemikus loetakse esitused grupi esitusteks, komponent kuvatakse juhul kui ülesanne kasutab gruppe ning on määratud grupi esituste kuupäevad.
- *Deadlines* – näitab tähtaegu ning nende möödumisel saadavat hinnet ülesande esitamise eest.
- *Problem link* – sisaldab linki ülesande probleemi kirjeldusele.
- *Update my submissions* – nupu vajutusel uuendatakse kasutaja esitusi.

Ülesanne 100

Description				
WYSIWYG editor ülesande kirjelduse lisamiseks.				

My submissions				
#	Submission time	Verdict	Language	Grade
1	19:13:56 03-05-2017	Accepted	C++11	5.00
2	16:54:11 05-12-2014	Accepted	Java	5.00

Group submissions				
Member name	Submission time	Verdict	Language	Grade
No submissions yet				

Problem link	
The 3n + 1 problem (uva 100)	

Update my submissions

Deadlines	
Grade	After
5.00	Before first deadline
3.00	00:00:00 14-05-2017
1.00	00:00:00 31-05-2017
0.00	00:00:00 01-07-2017

Group submission dates	
Date	
From	00:00:00 01-04-2017
To	00:00:00 01-05-2017

Group members	
Member name	
Admin User	
John Doe	

Joonis 11. Ülesande informatsiooni vaade.

Kursuse haldaja administratsiooni vaated

Kursuse haldajal on pistikprogrammi Castor haldamiseks neli võimalust: *Castor Settings*, *Castor Submissions*, *Castor Groups*, *Castor Users*. Eelnevalt mainitud lingid leiab kui navigeerida kursuse haldamise menüüsse. Järgnevalt kirjeldatakse kõiki nelja administratsiooni vaadet.

Castor Settings ehk Castori seadete vaade

Castori seadete vaates on võimalik kursuse haldajal määrata kursuse põhiselt Castori mooduli loomiseks vaikimisi väärtused. Castori seadete eesmärk on vähendada uue Castori mooduli loomise ajakulu.

Väljad Castori seadete vaates (Joonis 12):

- *Problem provider name* – valik erinevate probleemi kogumike vahel.
- *Default grade* – vaikimise hinne lahendatud ülesande eest.

- *Require review* – lisab uue hinde Castori hindekategooria alla, millega korrutatakse ülesande eest saadud hinne läbi.
- *Custom deadlines* – aktiveerib tähtaegade kasutamise.

Aktiveerides tähtaegade kasutamise tekib nupp, kust on võimalik lisada tähtaegu. Tähtaegade sisestamine on lahendatud järgnevalt: igal tähtajal on hinne, mille saab ülesande lahendaja pärast tähtaja möödumist ning iga tähtaeg määratakse kahe välja abil. Esimene väli määrab, mitme päeva pärast alates Castori mooduli lisamisest peaks tähtaeg olema, ning teine väli määrab kellaaja sellel päeval (formaadis hh:mm, kus hh tähistab tunde ning mm minuteid).

Näidis on olemas ka Joonis 12, kus määrame, et tähtaeg on 10 päeva pärast kell 12:00 (joonise tegemise hetkel oli kuupäev 14.05.2017). Kui selliste seadetega alustada mooduli lisamist 15.05.2017 siis oleks olemas tähtaeg ajaga 25.05.2017 12:00 ning hindega 2.

Problem provider name

UVa Online Judge

Default grade

3.00

Require review

Custom deadlines

Grade Delete

2

10 days from now at 12:00 24-May-2017 12:00 (Wednesday)

Add another custom deadline

Joonis 12. Castori seadete vaade.

***Castor Submissions* ehk Castori esituste vaade**

Castori esituste vaade on mõeldud kursuse haldajale ülevaate andmiseks ning kursusel osalejate kohta informatsiooni nägemiseks. Castori esituste vaade koosneb tabelist, kus on välja toodud kõigi kursusel osalejate poolt tehtud ülesannete esitused ning

rippmenüüst, millega saab uuendada kursusel osalejate ülesannete esitusi. Tabelist on võimalik otsida kõigi väljade järgi, filtreerida välju ning sorteerida tulemusi.

Tabeli veergude päised on:

- *Castor's Name* - Castori mooduli nimi.
- *Person's name* - esituse teinud inimese nimi.
- *Grade* - esituse eest saadud hinne.
- *Problem Provider* - ülesannete kogumiku nimi.
- *Problem ID* - probleemi unikaalne identifikaator.
- *Status* - ülesande kogumiku poolt antud esituse staatus.
- *Language used* - programmeerimiskeel, mida kasutati ülesande lahendamiseks.
- *Time* - esituse toimumise aeg.

Castor Groups ehk Castori gruppide vaade

Castori gruppide vaade annab kursuse haldajale võimaluse näha kursusel osalejate poolt loodud grupe ning neid muuta. Täpsemalt on gruppide süsteem selgitatud peatükis 3.2.3. Castori gruppide vaates tuleb kõigepealt valida Castori moodul lehel olevast rippmenüüst, mille peale laetakse kõik antud moodulis kursusel osalejate poolt loodud grupid. Castori gruppide vaate näidis on Joonis 13.

Iga grupi vaates on näha järgnev:

- *Group ID* - grupi unikaalne identifikaator.
- *Locked* - näitab, kas grupp on lukustatud või mitte.
- *Members* - näitab liikmete arvu ning ka grupi maksimaalset võimalikku liikmete arvu.
- *Leader* - grupi liidri nimi.
- List grupi liikmetest.

Kõigi grupi liikmete, kes ei ole grupi liidrid, juures on olemas nupp „*Kick member*“, millega saab liikme grupist eemaldada. Lisaks on veel iga grupi all, mis on lukustatud, nupp „*Unlock group*“, millega saab grupi lukust lahti teha, juhul kui grupi liikmed soovivad lisada uue liikme või mõne eemaldada.

Ülesanne 123 (2)	
Group ID: 1	
Locked	Yes
Members	2/2
Leader	Jane Doe
Member name	Actions
Jane Doe	N/A
John Doe	Kick member
Unlock Group	

Joonis 13. Castori gruppide vaade kursuse haldamise paneelis.

Castor Users ehk Castori kasutajate vaade

Castori kasutajate vaade annab kursuse haldajale võimaluse muuta Moodle'i kasutajatega seotud probleemi kogumike kasutajanimed. Castori kasutajate vaade koosneb tabelist, kus on välja toodud kõik probleemi kogumike kasutajanimed, mida on kursusele registreerunud Moodle'i kasutajad sidunud. Kogu tabelit saab veergude põhjal filtreerida ning sorteerida.

Tabel koosneb järgnevatest veergudest:

- *User ID* – Moodle'i kasutaja unikaalne identifikaator.
- *Name* – Moodle'i kasutaja pärisnimi.
- *Problem provider* - probleemide kogumiku nimi (näiteks UVa Online Judge).
- *Provider User ID* – Moodle'i kasutajaga seotud probleemi kogumiku kasutaja unikaalne identifikaator.
- *Provider username* – Moodle'i kasutajaga seotud probleemi kogumiku kasutajanimi.
- *Actions* - sisaldab nuppu, millega saab seotud kasutajanime muuta.

4 Kokkuvõte

Töö põhieesmärgiks oli arendada Moodle'i pistikprogramm, mis lihtsustaks ja mugavdaks kolmandate osapoolte poolt loodud automatiseeritud hindamissüsteemiga ülesannete kogumike kasutamist õppetöös. Töö teiseks eesmärgiks oli lisada pistikprogrammile paindlik gruppide süsteem, mis võimaldaks kasutada gruppe nii harjutustunni, ülesande kui ka kogu kursuse raames.

Töö lugeja peaks saama ülevaate pistikprogrammi loomisel kasutatud tehnoloogiatest ning nende omavahel sidumisest, arhitektuurist, ettetulnud probleemidest, kasutajaliidese funktsionaalsusest ning integreerimisest teiste ülesannete kogumikega.

Töö tulemusena valmis töötav Moodle'i pistikprogramm nimega Castor, mida saab kasutada põhieesmärgina püstitatud probleemi lahendamiseks ning millel on paindlikud sätted: grupid (teine eesmärk), tähtjad ning võimalus kergelt integreerida teiste ülesannete kogumikega. Pistikprogrammi Castor lähtekood on kättesaadav: <https://gitlab.cs.ttu.ee/Kenn.Jaakson/plugin-castor-moodle>

4.1 Edasised arendused

Selles peatükis toon välja potentsiaalsed pistikprogrammi Castor edasised arendused.

Ülesande esitamine läbi Moodle'i õpikeskkonna

Ülesande esitamine läbi Moodle'i õpikeskkonna võimaldaks kursusel osalejatel säästa aega ning lahendada ülesandeid läbi tuttava keskkonna. Kursuse haldaja jaoks tähendaks see automatiseeritud plagiaadi kontrollimise võimalust.

Võimalikuks probleemiks eelnevalt kirjeldatud arenduse juures on esituse saatmise API puudumine ülesannete kogumikes, mis tähendaks mitte nii töökindla lahenduse loomist.

Programmeerimiskeele valiku seade

Programmeerimiskeele valiku seade võimaldaks kursuse haldajal määrata programmeerimiskeel, millega tuleks ülesanne lahendada. Selline seade võimaldaks kursuse jooksul kokku puutuda mitme erineva programmeerimiskeelega.

Automaatne esituste uuendamine

Kui ülesannete kogumik võimaldab kasutada *callback*'e ülesande esitamisel, siis oleks seda võimalik esituste uuendamise automatiseerimiseks kasutada. Selline edasiarendus muudaks kursusel osaleja kogemuse paremaks.

Kasutatud kirjandus

- [1] Moodle, „About Moodle,“ [Võrgumaterjal]. Available: https://docs.moodle.org/32/en/About_Moodle. [Kasutatud 12 mai 2017].
- [2] Moodle, „Moodle HQ,“ [Võrgumaterjal]. Available: <https://moodle.com/hq/>. [Kasutatud 12 mai 2017].
- [3] Moodle, „Moodle statistics,“ Moodle, [Võrgumaterjal]. Available: <https://moodle.net/stats/>. [Kasutatud 12 mai 2017].
- [4] Moodle, „Moodle plugins directory,“ [Võrgumaterjal]. Available: <https://moodle.org/plugins/>. [Kasutatud 12 mai 2017].
- [5] Moodle, „Language packs,“ [Võrgumaterjal]. Available: https://docs.moodle.org/32/en/Language_packs. [Kasutatud 12 mai 2017].
- [6] Moodle, „Plugin types,“ [Võrgumaterjal]. Available: https://docs.moodle.org/dev/Plugin_types. [Kasutatud 13 mai 2017].
- [7] Moodle, „Activity modules,“ [Võrgumaterjal]. Available: https://docs.moodle.org/dev/Activity_modules. [Kasutatud 13 mai 2017].
- [8] Laravel, „Github - laravel/laravel,“ [Võrgumaterjal]. Available: <https://github.com/laravel/laravel>. [Kasutatud 13 mai 2017].
- [9] I. Pletnjov, „Sõltuvuste sisestamise raamistik Javas.,“ [Võrgumaterjal]. Available: <https://digi.lib.ttu.ee/i/?7083>. [Kasutatud 13 mai 2017].
- [10] T. Dalling, „Model View Controller Explained,“ [Võrgumaterjal]. Available: <http://www.tomdalling.com/blog/software-design/model-view-controller-explained/>. [Kasutatud 13 mai 2017].
- [11] T. A. S. Foundation, „Apache HTTP Server Tutorial: .htaccess files,“ [Võrgumaterjal]. Available: <https://httpd.apache.org/docs/current/howto/htaccess.html>. [Kasutatud 13 mai 2017].
- [12] T. A. S. Foundation, „Apache Module mod_rewrite,“ [Võrgumaterjal]. Available: http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html. [Kasutatud 13 mai 2017].
- [13] J. F. M. Alviste, „Joosep Alviste / moodle-wrapper · GitLab,“ [Võrgumaterjal]. Available: <https://gitlab.com/zeizig/moodle-wrapper>. [Kasutatud 20 mai 2017].
- [14] D. Biler, „Laravel: Best Practices,“ [Võrgumaterjal]. Available: <http://www.laravelbestpractices.com>. [Kasutatud 20 mai 2017].
- [15] Composer, „Getting Started,“ [Võrgumaterjal]. Available: <https://getcomposer.org/doc/00-intro.md>. [Kasutatud 13 mai 2017].