

# Ylläpitodokumentti

Sheeple

Helsinki 11.05.2009

Ohjelmistotuotantoprojekti

Helsingin Yliopisto

Tietojenkäsittelytieteen laitos

## Versiohistoria

Versio	Päivitys
0.5	Lisätty Uusien tehtävätyyppien lisääminen
0.6	Lisätty asennusohje
1.0	Dokumentti viimeistelty

## Table of Contents

1. Johdanto.....	3
2. Sanasto.....	3
3. Koodin ulkoasu.....	5
4. Arkkitehtuuri.....	6
4. Monikielisyyden toteutus.....	8
5. Tarkennuksia suunnitelmiin.....	8
6. Uusien tehtävätyyppien lisääminen.....	9
7. Moduulin asennus.....	10
8. Huomioita.....	11
9. Kehitysideoita.....	13

# 1. Johdanto

Tässä dokumentissa esitellään Helsingin Yliopiston ohjelmistotuotantoprojektissa kehitetyn ryhmäpalautustyökalun keskeisimmät ratkaisut. Ryhmäpalautustyökalu toimii Moodle-ympäristössä ja se on Moodleen erillisenä liitettävä moduuli. Dokumentissa käydään pääpiirteittäin läpi kehitetyn moduulin arkkitehtuuri sekä ohjelmointiratkaisut, ohjelmointiympäristö ja yleinen ohjelmointityyli.

Lisäksi dokumentissa käsitellään ohjelmiston ylläpitoa helpottavia näkökulmia, laajennettavuusratkaisuja ja kehitysnäkymiä. Dokumentti esittelee myös laajennus- ja kehitysideoita, joita ryhmäpalautustyökalun suunnittelun ja toteutuksen aikana kartoitettiin.

Ohjelmisto kehitettiin ohjelmistotuotantoprojektina ryhmä *Sheeple*:n toimesta keväällä 2009. Ohjelmiston tilasi Anni Rytönen.

# 2. Sanasto

Sana	Selitys
Moodle	Oppimisjärjestelmä johon ryhmäpalautustyökalu toteutetaan.
Ryhmäpalautustyökalu	Laajennus Moodle -oppimisjärjestelmään, joka mahdollistaa tehtävän arvioinnin kohdistumisen useaan opiskelijaan.
Ylläpitäjä	Henkilö, jolla on Moodle asennusoikeus.
Opiskelija	Henkilö, joka käyttää Moodlea opiskelijaoikeuksilla.
Opettaja	Henkilö, joka käyttää Moodlea opettajan oikeuksilla ja jolla on oikeus luoda ja arvostella tehtäviä ja ryhmätehtäviä kurssien alle.
Tehtävä	Moodle -oppimisjärjestelmän sisäinen toiminto jolla opettaja voi arvioida jonkin opiskelijan tuotoksen. Tehtävällä viitataan Moodlen assignment -aktiviteettiin, joka toteuttaa edellä mainitun toiminallisuuden.
Palautus	Jostakin ryhmätehtävästä annettu ryhmäkohtainen vastaus.
Aktiviteetti	Moodle -oppimisjärjestelmään liitettävän moduulin ilmentymän kutsumanimi.

Ryhmätehtävä	Toteutettavan aktiviteetin tyyppi, joka mahdollistaa ryhmäpohjaisten tehtävien palauttamisen ja ryhmäytymisen Moodlessa.
Avoin ryhmätehtävä	Ryhmätehtävä on avoin, kun opettaja on määritellyt sen näkyväksi opiskelijoille.
Tehtävätyyppi	Moduulin tarjoamat vaihtoehtoiset tyypit tehtäville, esimerkkeinä tiedostojen lähetys, verkkoteksti, lähetä yksi tiedosto tai offline -tehtävä (otettu Moodlen perustehtävätyypeistä).
Ryhmä	Opettajan tai opiskelijoiden ryhmätehtävän alle perustama instanssi, jolla on muille näkyvä nimi ja kuvaus ja johon opiskelijat voivat liittyä.
Aihe	Ryhmän nimen synonyymi. Ryhmiä voidaan kutsua myös aiheiksi, jos opettaja esimerkiksi haluaa määritellä ennalta tietyt aiheet mistä ryhmät kirjoittavat.
Kurssi	Kurssi jolla käytetään Moodle -oppimisjärjestelmää.
Tietokanta	Tietokanta jota Moodle ja ryhmäpalautustyökalu käyttävät.
Palvelin	Web-palvelin jolla ohjelmisto suoritetaan.
Moduuli	Projektin tuotoksena toteutettava lisäosa Moodle -oppimisjärjestelmään.
Käyttöliittymä	Moduulin käyttäjille näkyvä osuus jonka avulla opettaja ja opiskelijat kommunikoiivat sen kanssa.
Vertaisarviointi	Opiskelijoiden mahdollisuus arvioida muiden opiskelijoiden palautuksia.
Arvostelu	Opiskelijalle annettava tehtäväkohtainen arvosana.
Palaute	Opettajan ryhmälle antama tehtäväkohtainen kirjallinen kommentti.
Uudelleenpalautus MVC	Tehtävän palautus uudelleen sen arvioinnin jälkeen. Malli-Näkymä-Ohjain -arkkitehtuurityyli, jota käytetään ryhmäpalautustyökalun suunnittelun perustana.
Ryhmän profiili	Ryhmän nimi, kuvaus ja mahdollisesti muuta tietoa.

### 3. Koodin ulkoasu

Ryhmäpalautustyökalu on toteutettu PHP (väh. vrs. 5.2.9), HTML ja Javascript -ohjelmointikielillä ja se käyttää mySQL tietokantaa. Moodlessa käytetty merkistökoodaus on UTF8. Moodle on OpenSource-ohjelmisto, jota on kehitetty jo useiden vuosien ajan eri tahojen toimesta. Ryhmäpalautustyökalun ohjelmointityyli mukailee Moodlen jo olemassaolevia ohjelmointiratkaisuja. Ohjeet näihin ratkaisuihin löytyvät osoitteesta <http://docs.moodle.org/en/Development>.

Moodle tarjoaa valmiina paljon funktiota tietokantojen, lomakeparametrien, virheiden ja näkymien tulostamiseen. Ryhmäpalautustyökalu pyrkii tarvittaessa käyttämään näitä funktioita hyödyksi parhaaksi katsomallaan tavalla. Moodle määrittelee tarkasti mitä siihen kehitettävien lisämoduulien tulisi sisältää ollakseen Moodle yhteensopivia. Ryhmäpalautustyökalu on toteutettu näitä ohjeita noudattaen.

Tietokanta on määritelty Moodlen omalla XMLdb-schemalla, joka mahdollistaa tietokannan yhteensopivuuden tarvittaessa myös PostgreSQL, Oracle tai jonkin muun SQL-tietokannan kanssa. PHP koodi on pyritty ohjelmoimaan Moodlen ohjeiden mukaan ja kommentoimaan PHP-Doc kommentointityylin mukaisesti. Ohjelmiston API:na toimivat kommentoidut lähdekooditiedostot. Ratkaisu kirjoittaa API suoraan lähdekooditiedostoihin helpottaa ohjelmiston jatkokehitystä, koska suurin osa Moodlen kehittäjistä lukee ja kirjoittaa kommentteja pelkästään lähdekooditiedostoihin.

Ryhmäpalautustyökalu käyttää hyväkseen Ajax-tekniikkaa *ryhmien hallinta* -sivulla. Lisäksi ohjelmiston toimintalogiikasta vastaava controller.php käyttää hyödykseen Javascriptiä ryhmien hallinnassa käytettävien nappien toiminnallisuuden kontrollointiin. Painikkeiden toiminnallisuuteen ja käyttöoikeuksiin vaikuttavat *access.php* -tiedostossa määritellyt käyttäjäryhmäkohtaiset oikeudet käyttää ryhmäpalautustyökalun palveluja. Myös tietokantafunktioiden joukko seuraa käyttäjien oikeuksia ja tarjoaa sen mukaan käyttäjälle mahdollisuuden muuttaa tietokannan tilaa: tallentaa ryhmiä, liittyä ryhmään tai poistaa ryhmiä.

## 4. Arkkitehtuuri

Ryhmäpalautustyökalu on toteutettu MVC-arkkitehtuurin (malli-ohjain-näkymä) mukaisesti. Ohjaimena toimii controller.php, joka liitetään sivuun aina ryhmäpalautustyökalua käytettäessä. Poikkeuksena on ryhmäpalautustyökalun instanssin luonti- ja muokkaussivu mod\_form.php sekä tehtävätyyppien yksilöllisiä palautussivuja. Moodle tarjoaa valmiina kurssin alle sijoittuvien instanssien luontiin ja hallintaan sivun Mod\_form.php sekä valmiit tehtävätyypit verkkoteksti (online) ja tiedoston lähetyksen (file).

Ryhmäpalautustyökalun toimintalogiikka ei ulotu koskemaan mod\_form.php sivua. Verkkoteksti ja tiedostonlähetyksen tehtävätyyppien oikeuksia ja niiden tarvitsemia tietoja kontrolloidaan ryhmäpalautustyökalun ohjaimesta, mutta niiden varsinainen toiminnallisuus on Moodlen valmiina tarjoama palvelu ja se toteutetaan tehtävätyyppien toimesta.

### Ohjain

Ohjain controller.php käsittelee kaikki ryhmien selailuun, hallintaan, palautuksiin ja arvostelemiseen liittyvät toimintalogiikan ratkaisut. Controller tarkistaa palvelupyynnön tyyppin, annetut parametrit, käyttäjän oikeudet, hakee tarvittavat tiedot, tallentaa ne ja huolehtii poikkeustenhallinnasta. Lisäksi controller.php ilmoittaa virheistä niiden sattuessa.

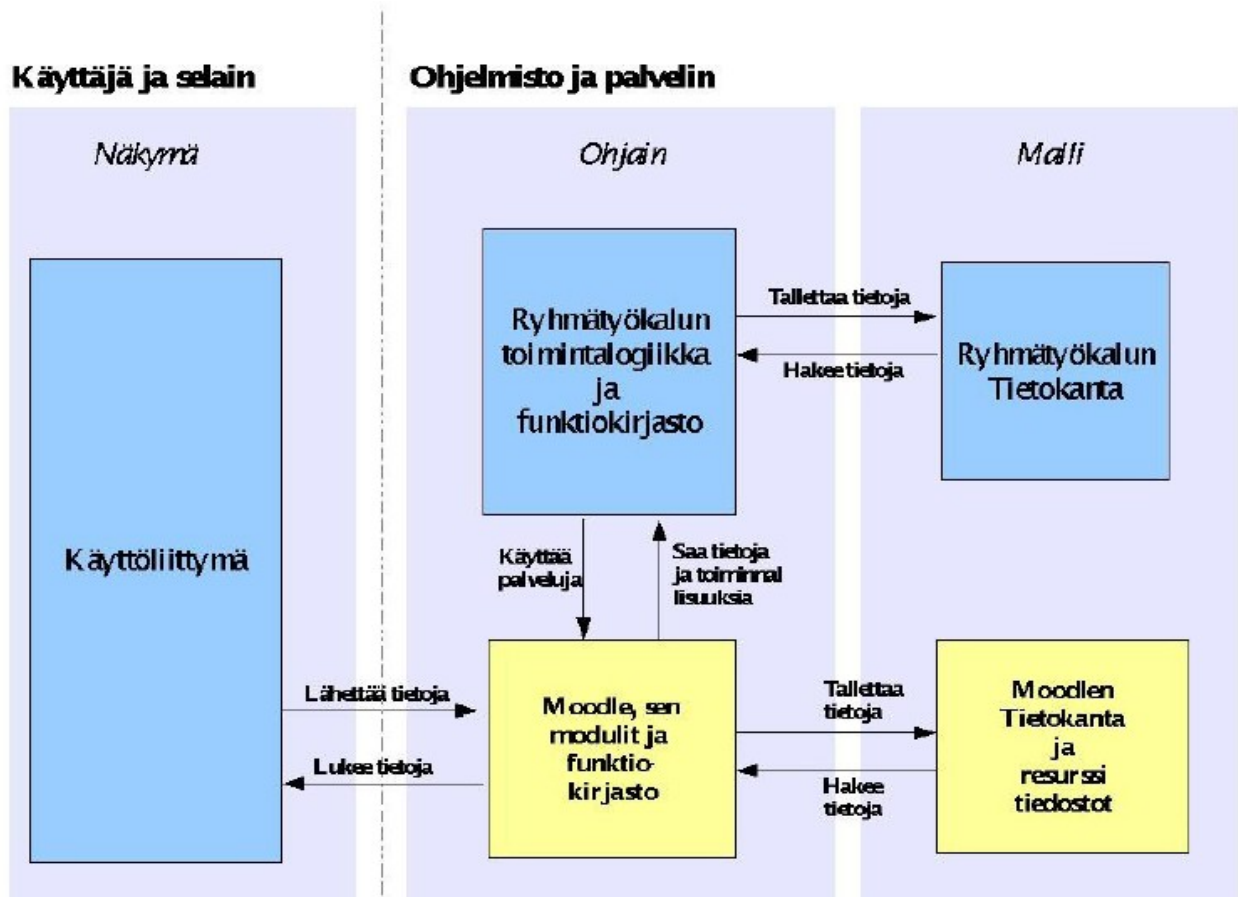
Ryhmien hallintaikkunassa ryhmiä valittaessa controller.php suorittaa ajax-palvelupyynnön getgroup.php -sivulle. Vastauksena saadaan profiili, joka on käyttöoikeuksia ja pyyntöä vastaava. Lisäksi Ajax-pyyntöllä toteutetaan ensimmäistä kertaa sivulle tultaessa haettava käyttäjän oma profiili. Ajax-pyyntömahdollisuutta käytetään tämän lisäksi tarvittaessa ryhmien lisäysten, profiilipäivitysten ja poistojen yhteydessä.

### Näkymä

Ohjelmiston näkymäsivut (index.php, group.php, emptyprofile.php, editableprofile.php ja noneditableprofile.php) ovat toimintalogiikan ulkopuolella ja ne keskittyvät tiedon esittämiseen ja sen ulkoasuun. Näitä sivuja pyydetessä controller.php hakee tarvittavat tiedot tietokannasta ja käsittelee ne näkymäsivua varten ja käynnistää näkymäsivun. Lib.php:ssa sijaitsee näkymän tarvitsemat funktiot esim. header-, footer- ja navikko-osien tulostamiseen.

## Malli

Mallista vastaa tietokantafunktiojoukko, joka sijaitsee lib.php:ssa. Ohjain käyttää näitä funktioita hyväkseen aina tietojen haku-, päivitys-, poisto- ja lisäyspalveluja tarvittaessa.



## 4. Monikielisyyden toteutus

Moduuli on toteutettu Suomeksi. Ohjelmisto mahdollistaa täydellisen monikielisyyden tuen ts. Kaikki käytetyt ilmaukset haetaan erikseen kielitiedostosta. Ryhmäpalautustyökalun käyttämät kielitiedostot sijaitsevat moduulin alikansiossa: /lang/fi\_utf8/sheeple.php.

Ryhmäpalautustyökalun ohje on toteutettu Moodlen valmiilla help\_button funktiolla, joka hakee ohje -painiketta painettaessa pyydetyn .html päätteisen ohjetiedoston omaan pop-up ikkunaan. Ohje-tiedostojen sijainti: /lang/fi\_utf8/help/sheeple/X.html.

### Uuden kielen lisääminen

Uutta kieltä lisättäessä /lang/ kansioon määritellään jokaista \$string muuttujaa vastaava arvo halutulla kielellä. Käyttäjä voi tämän jälkeen Moodlesta määritellä itselleen tarjolla olevat kielet käyttäjäkohtaisesti.

## 5. Tarkennuksia suunnitelmiin

### Arvostelu

Arvostelu toteutettiin siten, että opettaja voi alutessaan arvioida ryhmän ja jokaisen opiskelijan eri arvosanoilla tai vaihtoehtoisesti kaikki samalla arvosanalla. Opettajan on myös mahdollista arvioida pelkästään opiskelijat tai ryhmä. Ryhmän arvostelu koostuu arvosanasta ja palautteesta, mutta opiskelijan voi arvioida ainoastaan arvosanalla.

Ryhmän arvostelu tallentuu moduulin omaan tietokanta tauluun mdl\_sheeple\_group, jossa on kaikki muukin ryhmään liittyvä tieto. Opiskelijoiden arvostelussa käytetään moodlen arviointi -funktioita, joten opiskelijoiden arviointi tallentuu moodleen omaan arvostelu tauluun mdl\_grade\_grades.

Opiskelija näkee oman sekä ryhmän arvioinnin palautuksen yhteydessä, kun opettaja on tehnyt arvioinnin sekä moodlen oman "Arvioinnit" -toiminnon kautta.



## 6. Uusien tehtävätyyppien lisääminen

Moduulin suunnitteluvaiheessa pyrimme toteuttamaan modulaarisen arkkitehtuurin, jossa osia voi vaihtaa tai lisätä vaikuttamatta muiden osien toimintaan. MVC -arkkitehtuuriin pyrkiminen ja uusien tehtävätyyppien lisääminen tai poistaminen ovat esimerkkejä modulaarisesta arkkitehtuurisuunnittelusta.

Tehtävätyypit ja niiden sisältämät palautus -sivut ovat irrallisia itse ryhmytyimisestä, joten ne voidaan vaihtaa ilman, että ryhmytymistoiminnallisuus kärsii. Jokainen tehtävätyyppi on oma alihakemistonsa moduulin type -alikansiossa. MVC-arkkitehtuurin ohjain sisällyttää tehtävätyypin hakemistosta löytyvän luokkatiedoston ja luo uuden tehtävätyyppiolion, jonka jälkeen se kutsuu olion view -funktiota.

Uuden tehtävätyypin lisääminen tapahtuu luomalla uusi alihakemisto type -hakemiston alle ja kirjoittamalla sheeple.class.php -tiedosto, joka sisältää "sheeple\_TTNIMI" nimisen luokan joka perii sheeple\_base -luokan. sheeple\_base -luokka sisältää jokaiselle tehtävätyypille ominaisia piirteitä ja funktioita. Tietokantataulu johon uusi tehtävätyyppi tallentaa palautukset on tällä hetkellä luotava käsin, koska automaattisen taulunluontijärjestelmän toteuttaminen olisi ollut liian aikaa vievää (kts. Kehitysideoita)

## 7. Moduulin asennus

Moduuli asennetaan kuten mikä tahansa Moodlen moduuli. Paketti `sheeple.tar.gz` -paketti puretaan komennolla `tar -pxvzf sheeple.tar.gz` Moodlen `mod` -hakemistossa jonka jälkeen `admin` -oikeuksilla olevan Moodlen käyttäjän täytyy käydä hallintapaneelistä painamassa “ilmoitukset” -painiketta, jolloin moduuli asentuu. Tämän jälkeen moduuli on täysin käyttövalmis. `Tar` -komennon `-p` -parametri (säilytä oikeudet) on tarpeellinen niissä ympäristöissä, joissa oikeudet kumoutuvat tai muuttuvat (esim. `db.cs.helsinki.fi` -palvelimella paketin purkaminen ilman `-p` parametria antaisi kaikille tiedostoille oikeudet pelkästään käyttäjälle jättäen ryhmän ja maailman oikeudet asettamatta, mikä johti ongelmiin asennusta testattaessa).

## 8. Huomioita

Ryhmäpalautustyökalua kehitettäessä ryhmämme tutustui Moodlen toteutukseen, ohjelmointityyliin, arkkitehtuuriin, teknisiin ratkaisuihin sekä sudenkuoppiin perinpohjaisesti. Tässä luvussa listataan huomioita ja ongelmakohtia, joita kohtasimme kehitystyön ohessa.

### Moodlen dokumentointi ja tuki

Moodlen dokumentaatio on minimaalista. Moodle tarjoaa kehittäjille selkeän ohjeistuksen, mitä kaikkea tulee ottaa huomioon ja mukaan kehitettäessä moduuleja siihen liitettäväksi. Kuitenkaan dokumentaatiota jo tehdyistä toiminnallisuuksista on hyvin vaikea löytää Moodlesta. Usein kuitenkin koodi on kommentoitu suhteellisen kattavasti ja siellä löysimmekin suurimman avun ryhmätyökalua kehitettäessä.

### Tietokannan indeksointi

Moodle vaatii jokaiselle taululle oman yksilöivän ideksin. Tämä ei välttämättä ole paras ja joustavin mahdollinen ratkaisu. Toisaalta se varmistaa sen, että tietokantakyselyt pysyvät suhteellisen yksinkertaisina kautta linjan. Tämän lisäksi numeroituun pääavainindeksiin perustuvat tietokantataulut nopeuttavat kyselyitä, koska erillistä indeksiä ei tarvitse luoda. Tämä tyyli on käytössä useissa sovelluksissa siitä huolimatta, että se ei toteuta Boyce-Codd -normaalimuotoa.

**XMLdb-Schema** ei tarjoa tietokannan viiteavain ja -eheys määrittäjiä. Yrittäessämme sisällyttää tietokantamäärittelyihimme *on update cascade* ja *on delete cascade* rivejä, Moodlen XMLdb-schema ei tarjonnut siihen mahdollisuutta. Asiaan tutustumisen jälkeen, kävi ilmi että ehysmäärittelyt on mahdollista syöttää Moodleen *create table* lauseita käyttäen. Tämä ei kuitenkaan ole Moodlen pyytämä taulujen määrittelyn tekniikka. Koska XMLdb-scheman idea on säilyttää geneerisyys siirryttäessä eri tietokantamoottorien piiriin. Päädyimme poistamaan toisistaan riippuvat relaatiot omilla delete käskyillään käsin koodin puolelta. Tämä tietenkin lisäsi työtä toteutusvaiheeseen, mutta koska Moodle pyrkii mahdollistamaan tuen eri alustoille suosien LAMP -tuotaratkaisupinoa (Linux käyttöjärjestelmänä, Apache www-palvelimena, MySQL tietokantamoottorina ja PHP ohjelmointikielenä), on tärkeää noudattaa sen vaatimia ohjelmointiparadigmoja.

### Moodlen ohjelmointityyli

Moodlen ohjelmointityylejä on monenlaisia. Vaikka Moodle määrittelee selkeästi toivotun ohjelmointityylin, ei siitä löydy yhtenäistä linjaa. Päädyimme valitsemaan mahdollisimman selkeän ja havainnollisen hybridi ohjelmointityylin näiden kaikkien eri variaatioiden joukosta. Kuitenkin pitäytyen Moodlen virallisen asennuspaketin ohjelmointityylissä. Lisäksi pyrimme kommentoimaan koodimme huolellisesti ja havainnollisesti.

### **Moodlen arkkitehtuuri**

Moodlen arkkitehtuuriratkaisuista ei löytynyt käytännössä minkäänlaisia kuvauksia. Internetistä löytyi muutamia isoja UML-kuvia, joiden abstraktiotaso oli kuitenkin liian alhainen kokonaisarkkitehtuurin määrittämiseksi. Myöskään Moodlen aktiviteettien huonosta dokumentaatiosta johtuen, saattoi käytetyn arkkitehtuurityylin vain parhaimmillaan arvata.

### **Ohje -tietostojen sijainti**

Moodlen aktiviteettien ohjetiedostot sijaitsevat pääsääntöisesti yhden kansiotason ylempanä, kuin itse Moodlen juuri. Tämä rikkoo ajatusta, siitä että uusien Aktiviteettien asentaminen olisi nopeaa ja näppärää, koska sellaisessa tapauksessa uuden ohjetiedostot tulisi käsin siirtää haluttuun paikkaan. Löysimme kuitenkin myös toisen mahdollisuuden sijoittaa ohje-tiedostot suoraan moduulin alle kielitiedoston sisään. Herää kysymys miksi ohje-tiedostojen sijoittelulle ei tarjota yhtä selkeää toteutuspaikkaa.

### **Parametrien keruu funktio**

Moodle tarjoaa lomakeparametrien talteenottamiselle oman funktionsa `OPTIONAL_PARAMETER(name, default, type)`. Funktio osaa poimia painikkeen painalluksesta syntyvän Boolean arvon, sekä GET ja POST -parametrit erikseen. Funktiokutsu on kuitenkin jo itsessään raskaampi kuin PHP:n jo valmis yleinen komento lomakeparametrien tarkasteluun. Lisäksi funktiossa joudutaan ehtolauseella tarkastelemaan syötteen tyyppiä. Herää kysymys onko tämänkaltaisen funktioiden yhdisteleminen tarpeellista pidemmällä juoksulla.

## 9. Kehitysideoita

Vaatimusanalyysissä ryhmämme kartoitti paljon vaatimuksia, jonka jälkeen ne priorisoitiin toteutusta varten. Ryhmän pienen koon ja resurssien mukaan valittiin ensisijaisesti toteutettavaksi vain korkean priorisointitason käyttäjävaatimukset, alla listassa vaatimuksia jotka ovat toteutuneet ja lisäksi vaatimukset, joita ei ole toteutettu tähän versioon, mutta saattavat olla ohjelman ylläpidon ja jatkokehityksen kannalta tutustumisen arvoisia. Tarkemmat tiedot vaatimusanalyysistä löytyvät vaatimusdokumentista.

Vaatimuksille on jaettu prioriteetti seuraavasti:

**Tärkeä:** Olennainen osa käyttökelpoista ohjelmistoa. Se toteutetaan projektin puitteissa.

**Keskinkertainen:** Parantaa ohjelmiston käyttökelpoisuutta merkittävästi. Ohjelmisto kuitenkin toimii ilman sitä. Ominaisuus toteutetaan projektin puitteissa. Pääpaino on kuitenkin tärkeiksi luokitelluissa ominaisuuksissa.

**Matala:** Tuo lisäarvoa jo käyttökelpoiseen ohjelmistoon. Se voidaan toteuttaa, jos on ylimääräistä aikaa tai toteutus osoittautuu triviaaliksi.

Kaikki opettajalle kartoitetut käyttövaatimukset vaatimusanalyysistä:

*Tähdellä \* merkityt toteutettu tässä projektissa:*

### **Prioriteetti: korkea**

- \* **T1.** Opettaja voi lisätä ryhmätehtävän Moodlessa olevalle kurssille.
- \* **T2.** Opettaja voi poistaa ryhmätehtävän.
- \* **T3.** Opettaja voi muokata olemassa olevaa ryhmätehtävää.
- \* **T4.** Opettaja voi määritellä ryhmätehtävälle sen alkamis- ja päättymisajan.
- \* **T5.** Opettaja voi määritellä ryhmätehtävässä käytettävän tehtävätyypin, jonka Moodle tarjoaa valmiina.
- \* **T6.** Opettaja voi antaa kaikille ryhmätehtävään osallistuville ryhmille mahdollisuuden ryhmätehtävän uudelleenpalautukseen.
- \* **T7.** Opettaja voi lukea ryhmien tekemiä palautuksia.
- \* **T8.** Opettaja voi kohdistaa palautetta ryhmälle.
- \* **T9.** Opettaja voi arvostella ryhmän palautuksen.
- \* **T10.** Opettaja voi estää opiskelijoita lisäämästä uusia ryhmiä ryhmätehtävään.
- \* **T11.** Opettaja voi lisätä ryhmän.

\* **T12.** Opettaja voi selailla ryhmiä.

**Prioriteetti: keskinkertainen**

T13. Opettaja voi lisätä opiskelijoita ryhmiin.

T14. Opettaja voi poistaa opiskelijan ryhmästä.

\* **T15.** Opettaja voi poistaa ryhmän.

T16. Opettaja voi rajoittaa ryhmiin kuuluvien opiskelijoiden määrää.

T17. Opettaja voi kohdistaa palautetta yksittäiselle ryhmän jäsenelle.

\* **T18.** Opettaja voi arvostella yksittäisen ryhmän jäsenen.

T19. Opettaja voi kopioida ryhmät Moodlen käyttämistä ryhmistä uutta ryhmätehtävää luodessaan.

**Prioriteetti: matala**

T20. Opettaja voi rajoittaa ryhmien lukumäärää ryhmätehtäväkohtaisesti.

T21. Opettaja voi antaa palautetta ryhmälle, niin että jo olemassa oleva vanha palaute säilyy tallessa.

T22. Opettajalla on mahdollisuus täyttää ryhmät opiskelijoilla automaattisesti.

T23. Opettaja voi kopioida ryhmät toisesta ryhmätehtävästä uutta ryhmätehtävää luodessaan.

**Kaikki opiskelijalle kartoitetut käyttövaatimukset vaatimusanalyysistä:**

*Tähdellä \* merkityt toteutettu tässä projektissa:*

**Prioriteetti: tärkeä**

\* **S1.** Opiskelija voi lisätä ja nimetä ryhmän ryhmätehtävään.

\* **S2.** Opiskelija voi liittyä ryhmään.

\* **S3.** Opiskelija voi kuulua vain yhteen ryhmään ryhmätehtävän sisällä ja voi vaihtaa ryhmää, vain ensin eroamalla ryhmästä, johon kuuluu.

\* **S4.** Opiskelija voi uudelleen nimetä ryhmänsä, johon itse kuuluu.

\* **S5.** Opiskelija voi määritellä omalle ryhmälleen kuvauksen.

\* **S6.** Opiskelija voi muokata oman ryhmänsä kuvausta.

\* **S7.** Kuka tahansa ryhmän jäsen voi tehdä palautuksen ryhmätehtävään.

\* **S8.** Opiskelija voi tehdä uudelleenpalautuksen.

\* **S9.** Opiskelija voi erota ryhmästä.

\* **S10.** Opiskelija voi muokata palautusta.

\* **S11.** Opiskelija voi selailla ryhmiä.

**Prioriteetti: keskinkertainen**

S12. Opiskelija voi lisätä ryhmälleen kuvan.

**Prioriteetti: matala**

S13. Opiskelija voi lukita palautuksen, niin ettei sitä voi enää uudelleen palauttaa.

S14. Opiskelija voi avata lukitsemansa palautuksen.

S15. Opiskelija voi katsella toisten opiskelijoiden tekemiä palautuksia.

S16. Opiskelija voi antaa toisille opiskelijoille palautetta.

S17. Opiskelija ei voi lisätä useita ryhmiä.

S18. Opiskelija voi hyväksyä ja hylätä ryhmään pyrkiviä opiskelijoita.

S19. Opiskelija voi vertaisarvioida toisia opiskelijoita Moodlen tarjoamin välinein.

\* **S20.** Opiskelija voi poistaa tyhjän ryhmän.

S21. Jos monta opiskelijaa yrittää muokata samaa palautusta, ohjelmisto huolehtii samanaikaisuuden hallinnasta, niin että opiskelija, joka ensimmäisenä on aloittanut palautuksen saa muokata palautusta ensin ja vasta sitten seuraava opiskelija.

Näiden lisäksi esille tuli ainakin moduulin tehtävätyyppien automaattinen haku hakemistorakenteen perusteella. Sen toteuttaminen vaatisi oman "ohjauspaneelin" (aktiviteettimoduulin asetukset) toteuttamista moodlen pääkäyttäjän hallintapaneeliin. Kyseisen ominaisuuden voisi toteuttaa esimerkiksi sijoittamalla aktiviteettimoduulin asetukset -sivulle painikkeen "Etsi uusia tehtävätyyppejä", jonka jälkeen ohjelma etsii ei-tietokannassa olevat alihakemistot type -hakemistosta ja lisäisi tarvittavat tiedot tietokantaan.

Yllä mainitun ominaisuuden toteuttaminen vaatisi kuitenkin huomattavan suuren työmäärän verrattuna saatavilla oleviin resursseihin, joten se jää ainoastaan kehitysideaksi.