

Tallinna Ülikool  
Informaatika Instituut

# **AJAVEEBIDE AGREGEERIMINE E-KURSUSEL EDUFEEDRI NÄITEL**

Bakalaureusetöö

Autor: Maddis Reitav

Juhendaja: Hans Põldoja

Autor:..... 2014

Juhendaja ..... 2014

Instituudi direktor ..... 2014

Tallinn 2014

# AUTORIDEKLARATSIOON

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....

(kuupäev)

.....

(bakalaureusetöö kaitsja allkiri)

# SISUKORD

Sissejuhatus .....	4
1 Ajaveebid.....	6
1.1 Levinumad tarkvarad .....	6
1.2 Põhimõisted .....	8
1.3 Ajaveebide kasutamine õppetöös.....	9
1.4 TLÜ IFI kogemused ajaveebide kasutamisel õppetöös .....	10
2 Erinevad tarkvararakendused ajaveebide agregeerimiseks .....	13
2.1 Tavaotstarbelised tarkvara rakendused ajaveebide agregeerimiseks.....	13
2.2 Õppeotstarbelised tarkvara rakendused ajaveebide agregeerimiseks .....	14
3 Metoodika.....	20
3.1 Arendusmetoodika .....	20
3.2 Testimismetoodika.....	22
4 EduFeedr tarkvara lühitutvustus.....	24
4.1 Funktsionaalsus ja struktuur .....	24
4.2 Elgg .....	27
4.3 EduSuckr.....	28
5 Tracist piletite ümberkolimine GitHubi .....	29
6 Edufeedri agregeerimislahenduste täiendamine .....	32
6.1 Ajaveebipostituste ühendamine ülesannetega (pilet #154).....	32
6.2 Kommentaaride ühendamine autoritega (pilet #155) .....	36
6.3 Õppija profiil (pilet #156).....	41
7 Testimine .....	46
7.1 Testlood .....	46
7.1.1 Testlugu 1: Postituse ühendamine ülesandega .....	46
7.1.2 Testlugu 2: Kommentaari ühendamine autoriga .....	47
7.1.3 Testlugu 3: Üliõpilase profiili kuvamine.....	48
7.1.4 Testlugu 4: Üliõpilase progressi kuvamine .....	49
7.2 Ettepanekud EduFeedri edasiarenduseks.....	50
Kokkuvõte .....	52
Kasutatud kirjandus .....	53
Summary.....	56

## Sissejuhatus

Tänapäeval kasutatakse õppetöö huvitavamaks ja efektiivsemaks muutmiseks mitmeid erinevaid kaasaegseid meetodeid. Nendest üheks laiemalt kasutusel olevaks lahenduseks võib pidada erinevaid õpiahaldussüsteeme ning virtuaalseid õpikeskkondi. Õpiahaldussüsteemideks (ingl *Learning Management System*) nimetatakse veebipõhist serveritarkvara, mille abil on võimalik organiseerida võrgus olevaid õppeteenuseid õpilaste ning õppejõudude jaoks. Antud süsteemid koosnevad autentimise, õppetöö korraldamise, suhtluse ja administreerimise moodulitest (Paulsen, 2003). Virtuaalseteks õpikeskkondadeks (ingl *Virtual Learning Environment*) nimetatakse konkreetset vahendit õppetöö läbiviimiseks, mida saab iseloomustada järgneva seitsme tunnuse põhjal (Dillenbourg, Schneider & Paraskevi, 2002):

- 1) disainitud inforuum;
- 2) sotsiaalne ruum;
- 3) virtuaalne ruum on selgepiirilisel esitatud;
- 4) õppijatel on õpikeskkonnas aktiivne roll;
- 5) ei ole mõeldud ainult kaugõppe jaoks;
- 6) ühendavad erinevaid tehnoloogiaid ja pedagoogilisi lähenemisi;
- 7) täiendavad füüsilist keskkonda.

Üks üha enam populaarsust kogunud lahendus selleks on ajaveebide kasutamine õpikeskkonnana. Ajaveebide abil toimub erinevate õppeainete läbiviimine nii üli- kui ka üldhariduskoolides, kaasa arvatud Tallina Ülikoolis. Ajaveebipõhiste kursuste läbiviimise on keeruliseks teinud asjaolu, et õppejõul on ilma spetsiaalsete abivahendite olemasoluta keeruline pidada arvestust ajaveebipõhiste kursuste raames esitatud erinevate andmete kohta, nagu üliõpilaste poolt esitatud ülesanded või postitatud kommentaarid. Üheks ajaveebipõhiste kursuste läbiviimise abivahendiks on Tallinna Ülikooli Haridustehnoloogia keskus arendanud õpikeskkond EduFeedr. Hetke seisuga on antud õpikeskkonna funktsionaalsus poolik, kuna pole võimalik igas olukorras automaatselt ühendada üliõpilase postitust konkreetse ülesandega ning kommentaari selle autoriga.

Eelnevalt tulenevast on antud bakalaureusetöö eesmärgiks täiendada ajaveebipõhiste kursuste haldusvahendi EduFeedr funktsionaalsust selliselt, et see võimaldaks ühendada õppija postitusi ülesannetega ning kirjutatud kommentaare autoritega. Antud eesmärgi saavutamiseks on püstitatud järgnevad ülesanded:

1. analüüsida ajaveebide õppetöös kasutamise seonduvat kirjandust;
2. migreerida EduFeedr koodibaasi GitHub repositooriumisse;
3. täiendada EduFeedr haldusvahendi agregeerimislahendusi;
4. testida valminud funktsionaalsusi;

Antud bakalaureusetöö põhiosa koosneb seitsmest peatükist:

1. Esimeses peatükis keskendutakse erinevate kirjandusallikate analüüsile ning antakse lühiülevaade ajaveebidest - defineeritakse autori seisukohalt vajalikud mõisted, tuuakse välja enim kasutatavad tarkvarad ajaveebide loomiseks ja haldamiseks, kirjeldatakse kuidas tänaseks on rakendatud ajaveebe õppetöös, millised on nende eelised võrreldes õpihaldussüsteemidega ning millised on Tallinna Ülikooli Informaatika Instituudi kogemused ajaveebide kasutamisega õppetöös.
2. Teises peatükis kirjeldatakse tavaotstarbel ja õppeotstarbel kasutatavaid tarkvararakendusi ajaveebide agregeerimiseks.
3. Kolmandas peatükis kirjeldatakse arendus- ja testimismetoodikaid, mida antud bakalaureusetöö raames kasutati.
4. Neljandas peatükis tutvustakse lühidalt EduFeedr ajaveebide agregeerimise tarkvara, kirjeldades selle funktsionaalsust ning arhitektuuri, mis koosneb kolmest peakomponendist, milleks on EduFeedr, Elgg ja EduSuckr.
5. Viimasel peatükis selgitatakse, kuidas toimus EduFeedr koodibaasi migreerimine GitHub repositooriumisse.
6. Eelviimasel peatükis antakse ülevaade EduFeedri agregeerimislahenduste läbiviidud täiendustest.
7. Viimasel peatükis kirjeldatakse kuidas toimus valminud funktsionaalsuste testimine, millised olid antud testimise tulemused ning on autori poolsed soovitused edasiseks arendamiseks.

Autori jaoks mängisid antud lõputöö valikul rolli asjaolud, et lõputöö pakub arenguvõimalusi, väljakutset ja on autori jaoks uudne ning saab seostada Tallinna Ülikooli Informaatika erialaga. Lõputöö autoril eelnev kokkupuude õppetöökasutavate ajaveebidega puudus ning samuti oli minimaalne ka reaalne kogemus.

# 1 Ajaveebid

Ajaveebideks nimetatakse veebilehti, mis koosnevad perioodilistest tekstide, piltide või muudest meediaobjektide pävikulaadsetest postituste sissekannetest, mida regulaarselt täiendatakse. Enamasti esitatakse sisestatud postitusi tagurpidi ajalisel järjestuses – esimene postitus kuvatakse viimasena perioodiliselt ning seejuures vanemad postitused arhiveeritakse (Antriainen, 2012).

Nende sisu on väga erinev: autorid võivad kirjutada oma igapäeva tegemistest, kui ka keskenduda kindlale teemale ja nende postituste juures kasutatakse tihti linkimist ja tsiteerimist. Ajaveebid on üldjuhul on kättesaadavad kõigile Interneti kasutajatele ning samuti on igal ajaveebi külastajal enamasti võimalus autori poolt loodud postitusi kommenteerida.

Ajaveebe erisitatakse ka erinevate tüüpide järgi, milleks on järgnevad (Antriainen, 2012):

- 1) autori järgi — isiklik-, rühma-, asutuse- või „kummituse“ ajaveeb;
- 2) teemade järgi — poliitika, eluolu, reisimine, haridus, mood või muusika;
- 3) multimeedia järgi — teksti-, foto-, muusika-, audio- või videoajaveeb;
- 4) sisu eripära järgi — sisu-, seire- või tsitaatajaveeb;
- 5) tehnilise lahenduse järgi — blogimootoril, sisuhaldussüsteemil või staatilisel kodulehel põhinev ajaveeb.

## 1.1 Levinumad tarkvarad

WordPress<sup>1</sup> on avatud lähtekoodiga vabavaraline ajaveebi tarkvara, mille peal jookseb pea iga kuues veebileht, nende seas näiteks ühed maailma tuntuimad ajaveebid Wall Street Journal Blogs, People Magazine ning TechCrunch. Antud platvormi eelisteks on paindlikkus ja töökindlus.

Blogger<sup>2</sup> on Google'i poolt pakutav tasuta lihtne ajaveebiplatvorm, mis on WordPressi kõrval üks populaarsemaid ajaveebikeskkondi. Oma lihtsuse tõttu on see eriti sobiv inimestele, kes soovivad ajaveebide pidamisega algust teha. Bloggeri lähtekood ei ole avalik.

---

<sup>1</sup> WordPress — <http://wordpress.org/>

<sup>2</sup> Blogger — <https://www.blogger.com/home>

Tumblr<sup>3</sup> on tasuta kinnisel lähtekoodil põhinev ajaveebi tarkvara, mis kuulub Yahoole. Antud platvormi kasutajatel on võimalik postitada lühitekste, videosid ning pilte. Võimalus valida erinevate postituste tüüpide vahel võimaldab kasutajal pidada korraga mitut erinevat ajaveebi.

Typepad<sup>4</sup> on üks vanemaid kinnise lähtekoodiga ajaveebide platvorme, mis enne Bloggerit ja WordPressi omas suurt osa turust ning sisaldab eraldiseisvat rakendust statistika andmete loomiseks.

SquareSpace<sup>5</sup> on tasuline kinnise lähtekoodiga ajaveebi tarkvara, mis on mõeldud professionaalsetele ajaveebide kasutajatele ning ettevõtetele. Antud tarkvaraga on võimalik kujundust muuta vastavalt soovile, kasutades antud platvormi pukseerimise kasutajaliidest, samuti omab nimetatud tarkvara ajaveebide sisestamise võimalust teistest ajaveebide tarkvaradest.

Allpool olevas tabelis (vt tabel 1) on välja tood viie enimkasutava ajaveebi tarkvara võrdlus, võttes arvesse kättesaadavuse, kasutusmugavuse, kujundusvõimalused, funktsionaalsuse ning turvalisuse (Mikoluk, 2013).

**Tabel 1. Võrdlus viie enim levinud ajaveebide tarkvarade kohta**

	WordPress	Blogger	Tumblr	Typepad	SquareSpace
<b>Kättesaadavus</b>	On nii vaba tarkvara (avatud lähtekoodiga ja tasuta) kui ka tasuta teenus	Tasuta teenus	Tasuta teenus	Tasuline teenus	Tasuline teenus
<b>Kasutusmugavus</b>	Algajale keeruline	Algajale lihtne kasutada	Algajale lihtne kasutada	Algajale keeruline	Algajale keeruline
<b>Kujundusvõimalused</b>	Piiratud WordPressi tasuta konto kasutamisel, eraldiseisva serverina piiramatud	Piiratud	Piiratud	Piiratud	Piiratud
<b>Funktsionaalsus</b>	Piiramatu	Piiratud	Piiratud	Piiratud	Piiratud
<b>Turvalisus</b>	Ebaturvaline	Turvaline	Ebaturvaline	Turvaline	Turvaline
<b>Visutamine</b>	Piiratud WordPressi tasuta konto kasutamisel	Lubatud	Lubatud	Lubatud	Lubatud

Ülaltoodud tabel näitab, et avatud lähtekoodiga tasuta tarkvara kui ka tasuta teenus on WordPress ning Blogger ja Tumblr on tasuta kasutatavad teenused suletud lähtekoodiga ning

<sup>3</sup> Tumblr — <https://www.tumblr.com/>

<sup>4</sup> Typepad — <http://www.typepad.com/>

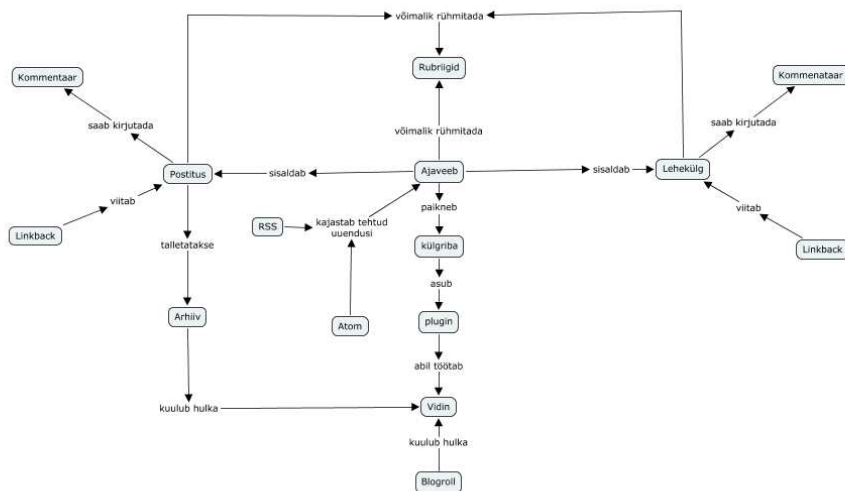
<sup>5</sup> SquareSpace — <http://www.squarespace.com/>

tasulised teenused on Typepad ja SquareSpace. Kasutusmugavuse poolest WordPress, Typepad ja SquareSpace algajatele ei sobi, kõikidest viiest kõige sobilikumad on Blogger ja Tumblr. WordPressi tasuta konto kasutamisel ei lase kujundust HTML/CSS'na muuta, kui antud platvorm on paigaldatud eraldiseisva serveri peale siis piirangud puuduvad, samas kui teistel tarkvaradel on ajaveebis etteantud kujunduspaketid ning juurdepääs kujunduse muutmist võimaldavale koodile puudub. WordPressi eeliseks on lisaks kujunduse loomisele võimalustele ka funktsionaalsus, teiste tarkvarade puhul saab kasutada ainult olemasolevaid lisavidinaid, kuid WordPress võimaldab neid ise juurde luua. Viiruste ja pahavarade eest on kõige enam kaitstud Blogger, Typepad ja SquareSpace. WordPress ning Tumblr on kergesti kahjustatavad pahavarade ja viiruste poolt. Visutamine on piiratud ainult WordPressi tasuta konto jaoks, see on võimalik ainult oEmbed abil (WordPress, kuupäev puudub), oma serverile paigaldatud WordPressil HTML koodi visutamise piirangud puuduvad.

## 1.2 Põhimõisted

Ajaveebid koosnevad erinevatest lehekülgedest ja postitusi, mida sisestab ajaveebi omanik. Postitusi on võimalik sildistada, antud siltide alusel tekib siltidepilv. Samuti on võimalik määrata kategooriaid, mis hiljem hõlbustavad konkreetsete postituste ja lehe otsimist. Ajaveebi lehekülje ja postitusi saavad kommenteerida kõik postitusel või leheküljel olevad lugejad, sisestades vastava teksti kommentaarlahtrisse. Ajaveeb võib omada mitmeid erinevaid vidinaid ja *blogrolli*. Ajaveebil toimuvate uuenduse kohta antakse teada RSS (Zhu, Sun & Choi, 2011) või Atom (Nottingham & Sayre, 2005) uudisvoogude abil isikutele, kes on märkinud antud ajaveebi endale jälgimiseks. *Linkbackid* tõstavad iga ajaveebi populaarust, kuna need on viited kellegi teise ajaveebilt, mis viitavad mingi konkreetse ajaveebi postitusele või lehele. Antud seoses on ka välja toodud joonisel 1.





Joonis 1. Mõistekaart WordPressi näitel

Igal komponendil on oma definitsioon, mida autor üritab enda sõnadega tabelis (vt tabel 2) lahti seletada.

Tabel 2. Autori definitsioonid ajaveebide komponentidele

Postitus ( <i>post</i> )	Kõigile ajaveebi lugejatele tehtud sissekanne ajaveebi autori poolt
Lehekülg ( <i>Page</i> )	Veebileht, mis koosneb pealkirjast, sisust milleks on postitused ning kommentaaridest
Kommentaari ( <i>Comment</i> )	Lugejate poolne tagasiside ajaveebi postitustele
RSS ( <i>RSS</i> )	Rakendus, mis võimaldab jälgida ajaveebi sisuga seonduvaid uuendusi ilma seda külastamata
Atom ( <i>Atom</i> )	XML-il baseeruv dokumendi formaat, mis edastab infot ajaveebides toimuvate uuenduste kohta
Rubriik ( <i>Category</i> )	Mingi tunnuse alusel rühmitatud ajaveebide postitused
Silt ( <i>Tag</i> )	Otsingusõnad, millega on hiljem arhiivist võimalik kindlaid postitusi otsida
Arhiiv ( <i>Archive</i> )	Arhiivis talletatakse ajaveebi vanemaid postitusi, värskemaid postitusi kuvatakse ajaveebi esilehel
Külgriba ( <i>Sidebar</i> )	Ajaveebi kujunduse üks osadest, kuhu saab paigutada erinevaid vidinaid ja Blogrolli
Blogroll ( <i>Blogroll</i> )	Külgribal asuv teistele ajaveebidele suunav linkide nimekiri., mida konkreetse ajaveebi autor lugejatele soovib
Vidin ( <i>Widget</i> )	Rakendus, mille abil on võimalik ajaveebi omanikul muuta enda ajaveebi kujundust ja sisu läbi „point-and-click“ ja „drag-and-drop“ funktsioonide (Soylu, Wild, Mödritscher, Desmet, Verlinde & De Causmaecker, 2011).
Linkback ( <i>Linkback</i> )	Ajaveebi postituses olev link, mis viitab seda postitust linkinud postitusele samas või teises ajaveebis. Linkback tehnoloogia eriliigid on Refback, Trackback, Pingback ja WebMention (Linkback, kuupäev puudub)
Pistikprogramm ( <i>plugin</i> )	Lisamoodul mille abil on võimalik täiustada üldist tarkvara funktsionaalsust

### 1.3 Ajaveebide kasutamine õppetöös

Õppetegevuse huvitavamaks muutmiseks on aina rohkem õpikeskkonnana kasutama hakatud ajaveebe. Kuigi ajaveebidel on õpiahaldussüsteemidega võrreldes teatavaid eeliseid, võib kahe

süsteemi omavahelisel võrdlusel välja tuua ka ajaveebide teatavaid miinuseid. Ajaveebide eelised ning puudused on välja toodud all olevas tabelis (vt tabel 3).

**Tabel 3. Ajaveebide kasutamise eelised ja puudused võrreldes õpiahaldussüsteemide kasutamisega õppetöös**

EELISED	PUUDUSED
Motiveerib õppijaid	Arutelud on fragmenteeritud erinevate ajaveebide vahel
Soodustavad osavõtu ja koostööd õpilaste seas	Puuduvad õpiahaldussüsteemides tavapärased vahendid
Pakuvad võimalusi grupitöödeks	On raskendatud õppijate haldus
Lubavad õppijatel arendada ja väljendada enda ideid	Keeruline reguleerida õppijate vahel ülesandeid
Kutsuvad esile tagasisidet teistelt	Hindamine keeruline
Suurendavad analüüsimise ja kritiseerimise võimet	
Pakuvad originaalseid ülesandeid	
Rikastavad õppekeskkonda	
Tõstavad esile erinevaid õpetamis metoodikaid	
Võimaldavad õppijatel end välja arendada eksperdi tasemele	
Aitavad kaas õppija-õppija, õppija-sisu, õppija-juhendaja vahelisele koostööle	
Annavad aluse kõrgemal tasemel õppimiseks ja mõtlemiseks	

Ajaveebide kasutamine motiveerib õpilasi iseseisvat mõtlema. Selline õppekorraldus arendab koostööd ja eneseväljendamise oskust. Grupitööd võimaldavad tagasisidet teistelt, mis tõstab analüüsimise ja kritiseerimise võimet, parandades õppija-õppija, õppija-sisu ja õppija-juhendaja omavahelist koostööd. Ajaveebide kasutamisel on võimalik luua originaalseid ülesandeid, mis rikastavad õppekeskkonda, tõstes esile erinevaid õppemetoodikaid.

Puudustena võib välja tuua selle, et ajaveebide fragmenteerituse tõttu on raskendatud arutelude jälgimine, õpilastele ülesannete andmine ja nende hindamine. See on põhjustatud sellest, et puuduvad õpiahaldussüsteemides esinevad tavapärased töövahendid ning on raskendatud õppijate haldamine.

Kõik eelnimetatud eelised võrreldes õpiahaldussüsteemidega on pedagoogilise tähtsusega ning puudused on pigem tehnilist laadi. Sellest järeldub, et ajaveebide kasutamine õppetöös pedagoogika seisukohalt on igati õigustatud (Goktas & Demirel, 2012)

## 1.4 TLÜ IFI kogemused ajaveebide kasutamisel õppetöös

Tallinna Ülikooli Informaatika Instituudi teaduritel ja õppejõududel on mitmeid avaldatud artikleid ajaveebide kasutamise kohta õppetöös. Üks selline artikkel on seotud üliõpilaste iseseiseva juhendamise eesmärkide täitmiseks erinevates õpiprojektides läbi sotsiaalmeedia (Väljataga & Fiedler, 2009), kus uuriti millised olid tudengite väljakutsed sellist tüüpi kursusel ning loodi kursuse läbiviimise eesmärgil õpilastele reaalseid situatsioone ja

väljakutseid arvestades kasutatavat tehnoloogiad, mis toetaks neid nende poolt kasutatavates õpikeskkondades võttes initsiatiivi ja vastutust nende tegemistes. Antud artikli raames läbiviidud uurimusest selgusid peamised väljakutsed, milleks olid:

- 1) raskused seoses terminoloogiaga;
- 2) raskused seoses lepingute õppimisega;
- 3) raskused seoses ülesannetega;
- 4) raskused seoses töövahendite ja teenustega;

Teine artikkel, mis on avalikustatud, analüüsis olukorda, kuidas saada üle kõrghariduses õppetööga kaasnevatest väljakutsega seonduvatest müütidest selleks sekkudes (Fiedler & Väljataga, 2013). Antud artiklist järeldub, et müütide likvideerimiseks tuleks selgitada ajaveebide tarkvarade võimalusi ning tagada vajalik tehniline tugi nii õpilaste kui ka õpetajate jaoks. Samuti peab julgustamiseks ja motivatsiooni tekitamiseks andma üliõpilastele tehtu kohta pidevat tagasisidet. Väga tähtis on õppejõudude poolne juhendamine teatud määral, mis aitab kaasa kogukonna põhise õppimise arengule ning stiimulid, et üliõpilased oleks motiveeritud andma tagasisidet kaastudengite kohta läbi kommenteerimise või ajaveebi postituste.

Kolmas artikkel rääkis Veeb 2.0 (O'Reilly, 2005) tehnoloogial põhinevatest teenustest ja vahenditest, mida kasutatakse üha rohkem õppetöö läbiviimisel, et luua personaalseid õpikeskkondi (Tomberg, Laanpere, Ley & Normak, 2013). Antud keskkondades on õppejõu roll üle kantud koolitajale, mis toob paratamatult endaga kaasa õiguste kadu. Seetõttu on keeruline hallata õppetööga seonduvaid protsesse, kui kursus peaks üle kanduma institutsioonilisest õpikeskkonnast personaalsesse ajaveebi põhisesse õpikeskkonda. Õppejõu seisukohalt on aga oluline, et säiliks teatud õigused õppetöö läbiviimiseks nagu: õpilaste kursusele registreerimine, ülesannete püstitamine ja õpilaste hindamise võimalus. Selle probleemi kõrvaldamiseks arendati WordPressi jaoks spetsiaalne kursuse lisamoodul nimega LePress. Selle kasutamine võimaldab õppejõul hallata ning koordineerida õppetegevust erinevatel kursustel ajaveebipõhistes õpikeskkondades, säilitades õpilaste vabaduse ja iseseisvuse õppimise võimaluse. LePressi kasutamise kohta tehti vastav uuring, millest võttis osa 37-liikmeline õppejõududest koosnev grupp. LePressi kasutati kuus kuud, mille vältel järeldus, et antud keskkonna kasutamine muutis efektiivsemaks õppejõudude võimalusi koordineerida erinevatel kursustel õppetegevust ning suurendas nende arusaama ajaveebipõhiste õppesüsteemide kasutuslihtsusest

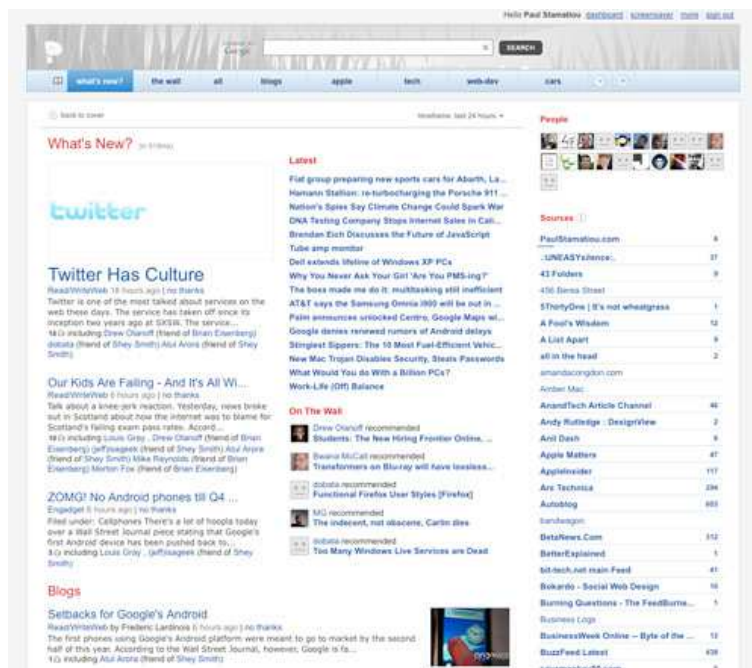
Sellised artiklid aitavad asjahuvilistel saada palju suuremat arusaama ajaveebide kasutamisest õppetöös, nende võimalustest ja probleemidest. Lisaks näitab suur artiklite arv seda, et inimestel on selliste teemade vastu huvi ning ajaveebide kasutamist õppetöös püütakse edasi arendada.

## 2 Erinevad tarkvararakendused ajaveebide agregeerimiseks

Ajaveebide andmete agregeerimiseks leidub mitmeid erinevaid tarkvara lahendusi, mille võib nende otstarbe alusel jaotada kaheks: tava ning õppeotstarbelised. Mõlemad grupid jagunevad veel omakorda alagruppideks, näiteks esimese grupi saab oma kasutuse poolest jagada personaalseteks ja jagatavateks agregeerimistarkvaradeks. Õppeotstarbelised jagunevad omakorda voogude agregeerimisel põhinevateks ning pistikprogrammil põhinevateks tarkvaradeks.

### 2.1 Tavaotstarbelised tarkvara rakendused ajaveebide agregeerimiseks

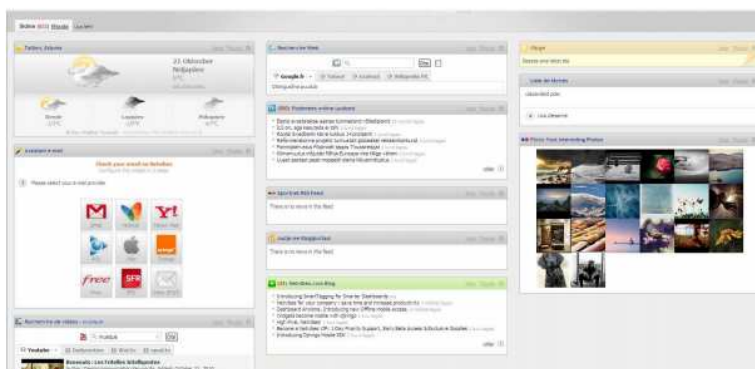
Feedly<sup>6</sup> – on uudisvoogude agregeerimiseks loodud tasuta veebileht, mida toetavad mitmed veebilehitsejad ning nutiseadmed, mis töötavad operatsioonisüsteemidel iOS või Android. Esmalt kandis see nime Feeddo, kui tegemist oli ainult veebilaiendiga. Peale nutiseadmete toe tulekut sai antud rakendus nimeks Feedly. Nimetatud rakenduse ametlik väljalaske kuupäev on 15. juuni 2008. aasta. See võimaldab pääseda ligi lemmikuteks märgitud ajaveebidele, uudiste lehtedele, raadioülekannetele ja YouTube'i kanalitele, sünkroniseerides antud vood erinevate seadmete vahel, kasutades olemasolevaid veebiallikaid (vt joonis 2).



Joonis 2. Feedly vaalet

<sup>6</sup> Feedly — <https://feedly.com/>

Netvibes<sup>7</sup> – on tasuta kättesaadav veebileht, mille kasutusvaldkond võib ulatuda isiklikust veebilehest õpikeskkonnani. Selle abil on võimalik personaalsesse keskkonda koondada erinevad veebilehed, ajalehed, videod pildid, ajaveebid ja erinevad vidinad nagu ilmateade, mängud jne (vt joonis 3). Netvibes'i abil on võimalik saada kiirelt kätte igast arvutist või mobiilsest seadmest vajalik info. Samuti võimaldab see korraga uuendada erinevaid sotsiaalse võrgustiku kontosid nagu Facebook ja Twitter. Peale selle võimaldab Netvibes luua privaatseid ja avalikke veebilehti ning jälgida veebis olevaid kasutaja poolt märgitud lemmik teemasid.



Joonis 3. Kasutaja vaade Netvibes'sse loginud kasutajale

Tavaotstarbeliste ajaveebide agregeerimiskanduse puhul puudub võimalus hallata erinevaid kursuseid ning teisi õppetöö läbiviimist toetavad funktsionaalsusi. Õppejõud ei saa üliõpilasi kursusele registreerida, luua iseseisvaks lahendamiseks vastavaid ülesandeid ning neid hinnata jne.

## 2.2 Õppeotstarbelised tarkvara rakendused ajaveebide agregeerimiseks

gRSSHopper<sup>8</sup> on Perl<sup>9</sup> keeles kirjutatud vabavaraline avatud lähtekoodiga personaalne veebikeskkond individuaalseks või väiksematele gruppidele kasutamiseks, kombineerides ressursside agregeerimist, isikliku andmebaasi ning võimaldades avaldada isiklike tekste (Downes, 2010). Antud tarkvara puhul on tegemist nii ajaveebi mootoriga, kogukonna veebilehega kui ka sisuhaldussüsteemiga personaalsete õpikeskkondade jaoks.

Õpilastel on võimalus hallata postituste sisu, lisada enda või teiste kasutajate poolt loodud postitusi teistelt ajaveebiplatvormidelt ning levitada neid kas uudisvoogudena või

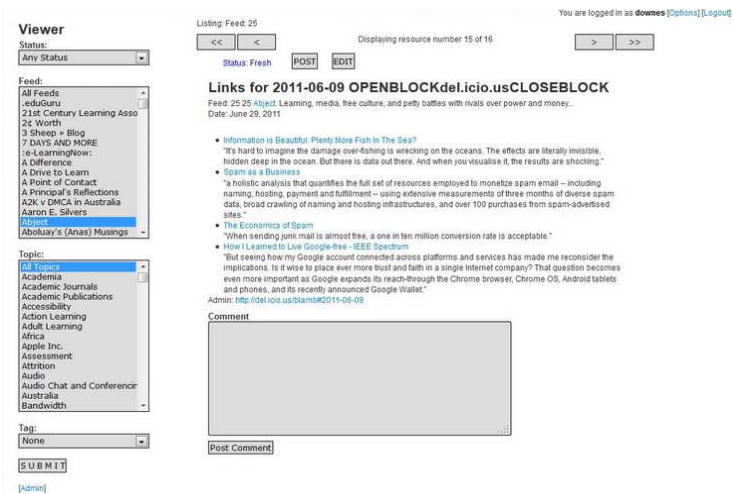
<sup>7</sup> Netvibes — <http://www.netvibes.com/en>

<sup>8</sup> gRSSHopper — <http://grsshopper.downes.ca/>

<sup>9</sup> Perl — interpreteeritav programmeerimiskeel <http://www.perl.org/>

veebilehtedena. Samuti on võimalik andmete levitamine JSON<sup>10</sup> andmevoona ja XMLi põhineva andmevorminguna (RSS).

Selline sisu loomine ja käsitlemine toimub läbi süsteemi koodi, mille abil eristatakse mitmeid andmetüüpe. Mõningad tüübid on laialdase kasutusega ning teised tüübid on spetsiifilised ja ainult gRSSHopperi jaoks mõeldud, kuigi andmetüübid on kõik omavahel seotud. Allpool toodud näide gRSSHopperi vaatest sisse loginud kasutajale (vt joonis 4). gRSSHopperi puuduseks on ülesannete haldamise võimaluse puudumine.



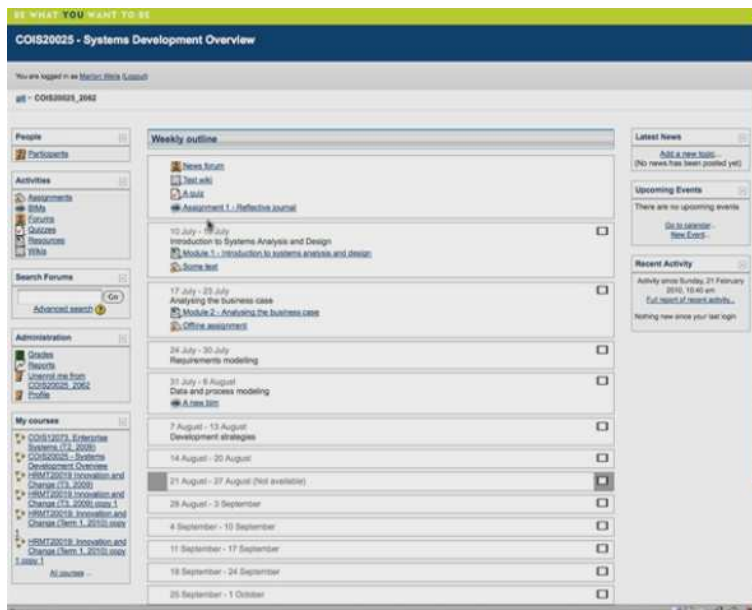
Joonis 4. gRSSHopperi vaade sisse loginud kasutajale

BIM on avatud lähtekoodiga spetsiaalselt Moodle õpikeskkonna jaoks loodud lisamoodul, mille abil toimub erinevate uudisvoogude agregeerimine üliõpilaste ajaveebidelt, registreerides need antud keskkonda (vt joonis 5). Sõna BIM tuleb väljendist BAM<sup>11</sup> lisamine Moodle'sse (ingl *BAM into Moodle*) (Jones, 2013).

Üliõpilased saavad läbi enda ajaveebide vastata õppejõu poolt püstitatud küsimustele, luues selleks vastava pealkirja ja sisuga ajaveebi postituse. See võimaldab õppejõududel paremini jälgida üliõpilaste progressi seoses õppetöoga, anda tagasisidet ning nende poolseid lahendusi hinnata. Nagu eelnevalt sai mainitud, siis BIM töötab ainult koos Moodle'ga, mis ajaveebipõhiste kursuste jaoks on suur ja keerukas vahend ning annab vähe lisaväärtust, kui kõik õpitegevused toimuvad ajaveebides.

<sup>10</sup> JSON — on kergekaaluline andmevahetusformaad, mis põhineb JavaScripti programmeerimiskeele alamhulgal

<sup>11</sup> BAM — Blog aggregation management



Joonis 5. Kursuse vaade BIMis

eMUSE on suletud lähtekoodiga Veeb 2.0 põhinev õpikeskkonna platvorm, mis võimaldab kasutada kõiki Veeb 2.0 poolt toetavaid tööriistu ühes kohas (vt joonis 6). Antud platvorm meenutab enda olemuselt MUPPLE personaalset õpikeskkonda, kuigi suureks erinevuseks võib pidada õiguste omamist seoses erinevate tööriistade kasutamisega (Wild, Mödritscher & Sigurdarson, 2008). Nimelt MUPPLE keskkonnas omavad õigusi üliõpilased, siis eMUSE keskkonnas omab antud õigusi õppejõud. Iga kursuse jaoks on need võimalik õppejõu poolt eraldi määrata.

Õppejõul on võimalus jälgida üliõpilaste poolt tehtut graafiliselt, võttes õpilaste poolt salvestatud andmed andmebaasist. See on kasulik vastava tagasiside andmiseks ning lõpphinde arvutamisel eMUSE poolt kasutatava hindamissüsteemi juures. Samuti pakub antud keskkond õppejõu jaoks kõiki õpihaldussüsteemidele omaseid teenuseid nagu autentimist, õpilaste kursusele lisamist, õpilaste profiilide redigeerimist. eMUSE platvormi kasutamisel on suureks eeliseks erinevatel kursustel kasutatavad veebirakendused nagu Blogger, MediaWiki<sup>12</sup>, Twitter<sup>13</sup>, Delicious<sup>14</sup> ja YouTube<sup>15</sup>, kuna nendega on kursustelt osavõtjad juba varasemalt kokku puutunud.

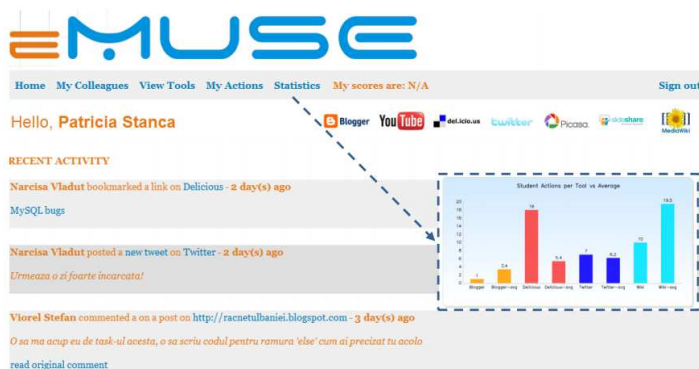
<sup>12</sup> MediaWiki — <http://www.mediawiki.org/wiki/MediaWiki>

<sup>13</sup> Twitter — <https://twitter.com/>

<sup>14</sup> Delicious — <https://delicious.com/>

<sup>15</sup> YouTube — <https://www.youtube.com/>





Joonis 6. vaade õpilasena sisse loginud kasutajale eMUSE keskkonnas

LePress<sup>16</sup> on avatud lähtekoodiga ajaveebi agregeerimisrakendus, mille abil on võimalik muuta WordPress kergesti jagatavaks personaliseeritud õpikeskkonnaks. Antud rakendus arendati välja Tallinna Ülikooli Informaatika Instituudi poolt.

Antud rakendus võimaldab õppejõududel luua erinevaid kursuseid kasutades endi ajaveebe, millele on lisatud LePressi lisamoodul kasutades WordPressi platvormi (vt joonis 7). Õppejõu poolt püstitatud ülesanded märgistatakse erinevate kategooriate järgi, mille abil on võimalik erinevate sisudega postituste kiire filtreerimine, mis on suunatud kursusele lisatud üliõpilastele. Filtreerimine lubab õppejõul anda kiiret tagasisidet esitatud tööde kohta ja vajadusel neid hinnata kasutades agregeeritud postituse koondlehte.

Kursusel osavõtmiseks peab igal üliõpilasel olema lisatud enda WordPressi platvormile samuti vastav pistikprogramm. Üliõpilased, kes on juba kursusele lisatud saavad jälgida õppejõu poolt jäetud teateid ning püstitatud ülesandeid, kasutades selleks külgribal olevat vidinat. Üliõpilased esitavad lahendusi postituste kujul, kust vajalik info automaatselt päritakse ning salvestatakse õppejõule kuuluvasse koondtabelisse, et õppejõud saaks anda tagasisidet tehtu kohta ning vajadusel seda ka hinnata. See ilmub üliõpilase poolt kasutatavasse WordPressi haldusliideses olevasse LePressi menüüsse, kuvades õppejõu poolt sisestatud tagasisidet ja hinnet. Nii õppejõud kui ka üliõpilased saavad kasutada üheaegselt sama ajaveebi erinevatelt kursustelt osavõtmiseks. LePressi puuduseks on kasutades WordPress.com tasuta majutusteenust ei ole võimalik kasutajatel endale vajalikku pistikprogrammi lisada. Selle jaoks peaks iga õppija enda WordPressi serveri püsti panema või ülikool peaks kõigile õppijatele vastavat teenust pakkuma hakkama. See tähendaks aga lisatööd hulga WordPresside haldamisega.

<sup>16</sup> LePress — <http://lepress.net>



### Welcome to LePress.net!

Hello, here is WordPress blog that we use for testing and evaluating of our LePress plugin.

LePress is used for organization of courses in WordPress. It support implementing courses on the base of WordPress blogs, connects learning activities of a teacher and students, and allows a quick assessment of students' homework.

LePress works as the plugin for WordPress. If you wish to install LePress into your own blog you can download LePress plugin from [WordPress repository](#).

If you are a teacher in University of Tallinn you can register your account in this blog on [Register](#) page. Please let us know about your new course by email [admin \[at\] lepress.net](mailto:admin[at]lepress.net) so we can place a link to your course on first page.

Regards,

Your LePress development team

#### WordPress Links

[Registration](#) / [Register](#)

#### Registered courses

[Dean Laanpere's profile](#)

#### Design for all

[Webaccessibility Workshop](#)

#### Active LePress Courses

## Joonis 7. LePress avalehe vaade

Dippler<sup>17</sup> on ajaveebide agregeerimise mõeldud pilverakendus, mille südameks on keskne vahevararakendus DBOS<sup>18</sup> koos tsentraalse andmebaasiga, võimaldades igal õppeasutusel tellida oma institutsionaalne keskkond (Laanpere, Pata, Normak, & Põldoja, 2014). Selle abil on võimalik lisada igale asutusele omane visuaalne stiil, struktuuriüksused ja ainekaardid (vt Joonis 8). Pedagoogilisest vaatenurgast toetab Dippler kolmele teoreetilisele nurgakivile: esimene on väljundipõhine õpe, teine enesejuhitav õpe ja kolmas refleksiivõpe e-portfoolio abil.

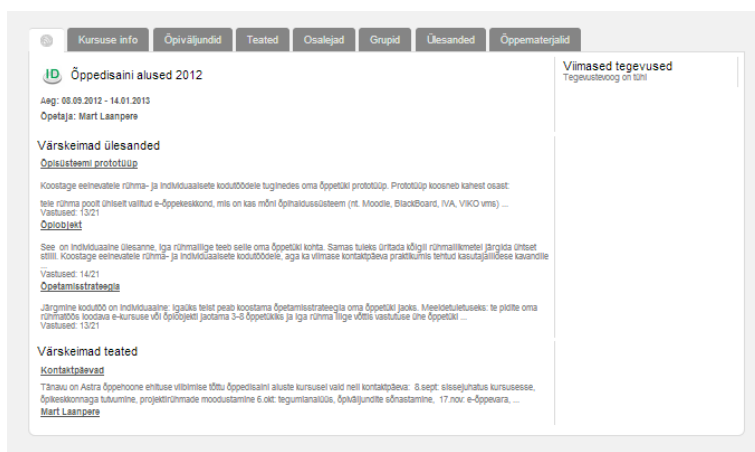
Dippler vahendusel on üliõpilasel võimalik kasutada enda personaalset õpikeskkonda, milleks tuntuim on ajaveebitarkvara WordPress. Selleks tuleb WordPressi platvormile paigaldada Dippleri lisamoodul, et enda vastavale kursusele registreerida. Dippleri kasutamine võimaldab vajadusel üliõpilastel end lahti siduda õppeasutuse küljest, säilitades oma portfoolios kõik õppetöö jooksul kogunenud materjalid, kodutööd, diskussioonid jne.

Õppejõudude jaoks on Dippleris õppematerjalide haldamine radikaalne — erinevalt eelmise põlvkonna õpihaldussüsteemidest ei saa õppejõud Dipplerisse üles laadida õppematerjale ja neid sinna lukustada. Selle asemel peab õppejõud need paigutama avatud keskkonda nagu Slideshare, Issuu, YouTube, visutades või linkides need kursuse õppematerjalide lehele.

<sup>17</sup> Dippler — <http://htk.tlu.ee/dippler/>

<sup>18</sup> DBOS (Dippler back office service) — Javal baseeruv rakendustarkvara, koos SOAP veebiteenustega ja MySQL andmebaasiga, mida kasutatakse andmete talletamiseks ja teiste tarkvaraliste komponentidega suhtlemiseks Dippleri süsteemis.

Sarnaselt LePressile vajab ka Dippler vastavat pistikprogrammi, et õppija saaks kasutada enda ajaveebi õppetöös. See välistab WordPress.com kasutamise.



### Joonis 8. Sektsioonide vaade Dippleri keskkonnas

EduFeedri suurimaks eeliseks võrreldes üleval kirjeldatud tarkvaradega võib pidada lisarakenduste vajalikkuse puudumise seoses kursuselt osavõtmisega. See tähendab seda, et õppijad saavad kasutada populaarseid tasuta ajaveebi majutusplatvorme WordPress.com ning Blogger. Samuti ei pea üliõpilane omama antud keskkonnas kasutajakontot, et näha toimuvat erinevatel kursustel ning kogu vajalik info on kättesaadav ühest keskkonnast. Ka sisaldab EduFeedr erinevaid visualiseeringuid, mis näitavad õppijate aktiivsust ja omavahelist suhtlemist.

## 3 Metoodika

Antud bakalaureusetöö juures kasutati erinevaid arendus- ja testimismetoodikaid. Arenduse juures kasutati elemente Scrum arendusmetoodikast ning testimise juures dünaamilise testimise funktsionaalset ehk musta kasti, halli kasti ja visuaalset testimismetoodikat. Antud metoodikaid kirjeldatakse all olevates alampeatükkides.

### 3.1 Arendusmetoodika

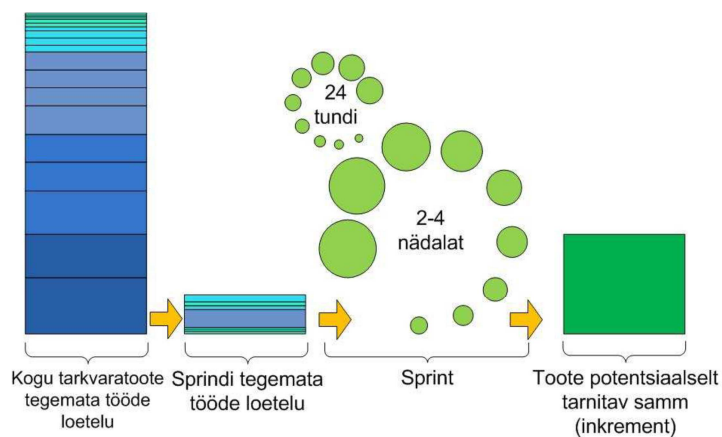
Scrum on projekti haldamisele suunatud iteratiivne agiilne arendusmetoodika kontseptsioon, mille järgi toimub tarkvaraarendus kiirete pistiktsükklitena, kus iga tsüklil iseenesest on traditsiooniline tarkvaraarenduse pisikordus, mis sisaldab analüüsi, koodi kirjutamist, testimist, veaparandust ning dokumenteerimist (Schwaber & Sutherland, 2009).

Antud metoodika jaotab arendusprojekti osalevate isikute rollid põhirollideks ja abistavateks rollideks. Antud metoodika jaoks on defineeritud kolm põhirolli, millele viidatakse tihti sünonüümiga porsad (ingl *pigs*) ja ülejäänud rollid on abistava tähendusega ning varieeruvad. Neid rolle kutsutakse sünonüümiga kanad (ingl *chickens*). Põhirollid on tihedalt seotud tarkvara arendusprojekti lõpule viimisega ning esindavad töövõtjat. Sellest tulenevalt defineeritakse Scrum arendusmetoodika juures kolme põhirolli järgnevalt:

- 1) Projektijuht (ingl *scrum master*) – isik, kes tagab metoodika korrektse rakendamise abistades toote omaniku tehniliste küsimustega erinevate Scrumi elementide kohta. Antud isik aitab kaasa arendusmeeskonna ning toote omaniku omavahelisele suhtlusele kui ka teiste huvigruppidega suhtlemisele. Projektijuht tagab, et arendusprotsess toimub plaanikohaselt, korraldades selleks vastava teemalisi koosolekuid. Samuti viib läbi antud isik Scrumiga seotud arenguetappe ehk sprinte<sup>19</sup> ja nende rakendavate tulemuste hindamist ning valmis tulemi juurutamist kliendi organisatsioonis (vt Joonis 9).

---

<sup>19</sup> Sprint — On pingutus mille sooritamiseks on ette antud piiratud aeg. Antud aeg määratakse iga sprinti jaoks eraldi ja on tavaliselt nädala kuni kuu pikkune, tüüpiline on 2 nädalat.



**Joonis 9. Scrumi arenguetapid ja nende ajaline kestvus (Scrum.ee, kuupäev puudub)**

- 2) Arendusmeeskond (ingl *development team*) on sama grupp inimesi, tavaliselt 3–9 inimest, kes on võrdsed, sõltumata oskuste erinevusest ja erinevate tegevustega seotud rollidest, kuna vastutatakse ühiselt välja arendatava projekti eest. Nemad analüüsivad, disainivad, arendavad ja testivad arenduses olevat tarkvara ning koostavad sellele vastava tehnilise dokumentatsiooni.
- 3) Toote omanik (ingl *product owner*) - huvirühma esindaja, kes omab detailset ülevaadet huvirühma soovidest ning ootustest konkreetse toote suhtes. Tema kirjutab kasutajalood (ingl *user stories*) ning määrab neile tähtsuse. Seejärel paneb ta kõik lood tähtsuse järjekorra ning lisab nõudmiste nimekirja. Tihti on toote omanik kombineeritud projektijuhi rolliga, et tagada parim nähtavus tehtud töö suhtes.

Projektijuht viib läbi projekti esialgse planeerimise ning arendusmeeskond koostab disaini käigust *backlogi*, mis kirjeldab tegemata töid antud tarkvara arenduse juures. See koosneb nii arendajate kui ka kasutajate soovidest, ideedest ja ettepanekutest. Scrumi juures jaguneb *backlog* kaheks, esimest võib nimetada välja arendatava tarkvara *backlogiks*, milles kirjeldatakse kliendi soove kasutajalugude (ingl *user stories*) abil ja vajalikke funktsionaalsusi. Antud *backlog* omakorda moodustab sprindi *backlogi*, kus on kirja pandud meeskondade ülesanded ning kirjeldused erinevate arendustööde kohta.

Tarkvaraarendusprojekti ajagraafiku jälgimiseks kasutatakse läbipõlemise graafikut (ingl *burndown chart*), mis on koostatud *backlogi* põhjal. Selleks hinnatakse *backlogi* koostamisel kui kaua kulub aega mingi funktsionaalsuse loomiseks. Liites antud ajahinnangud saadakse kogu projekti eeldatav kestvus. Kui mingi funktsionaalsuse arendus on lõpule viidud, lahutatakse sellele kulunud aeg kogu arendusprojekti ajalisest mahust ning saadakse teada allesjäänud töötundide maht.

Sprindid on Scrumi elutsüklit iseloomustavad projekti arendusetapid, mille vältel toimuvad arendustööd. Antud tööd on seotud ühte sprindi *backlogiga*, kuhu on klient välja valinud ühe kogumi kasutajalugusid (ingl *users stories*). Peale iga sprindi lõppemist vaadatakse üle tehtu ning lepatakse kokku tegevustest seoses järgmise sprindiga. Kui tekivad uued funktsionaalsuse täiendamise vajadused siis lisatakse need järgmisesse sprindi *backlogi*, kuna sprindi käigus tehtavaid tegevusi ei muudeta.

Kasutajalood on lõppkasutajale suunatud olulised tegevuste, valikute, tulemuste funktsionaalsuste kirjeldused, mida on vaja teostada kogu tarkvaraarendus protsessi vältel. Antud kirjeldus peab olema mõistetavalt sõnastatud disaineritele, arendajatele kui ka lõppkasutajatele, võttes aluseks testimisülesanded. Suuremad kasutajalood jaotakse omakorda väiksemateks ning iga loo juures on esindatud korraga ainult üks kasutaja.

Antud bakalaureusetöö raames kasutati mõningaid Scrumile iseloomulike elemente, milleks ühed olid kasutajalood. Need kirjeldasid EduFeedri rakenduse funktsionaalsuse täiendamise vajadust lõppkasutaja seisukohtalt, mis lisati ühe sprindi *backlogi*. Sprint ise oli konkreetse ajalise pikkusega, mis omakorda jaotati erinevateks lühemateks arenguetappideks. Selle tulemusena kujunes ühe sprindi kogupikkuseks 14 päeva ehk 2 nädalat ning lühemad arenguetapid olid ajaliselt erineva pikkusega. Scrumile iseloomuliku kolme põhirollide jaotust ei olnud, kuna toote omanik kui ka projektijuht oli üks ja see sama isik ning arendusmeeskond koosnes ühest isikust. Scrumi arendusmetoodika võimaldas töö autoril paremini mõista kuidas reaalse arendusprojekti juures antud metoodikat on võimalik rakendada.

### **3.2 Testimismetoodika**

Antud bakalaureusetöö juures kasutati toote omaniku (Scrumi *product owner*) poolt dünaamilise testimise funktsionaalset ehk must kasti (ingl *black box testing*) testimismetoodikat, mis tähendab testija ei olnud täpselt teadlik antud tarkvara programmi koodist ja struktuurist (Ostrand, 2002).

Antud bakalaureuse töö autor ise kasutas arenduse käigus läbi viidud täienduste funktsionaalsuse testimiseks dünaamilise testimise kahte erinevat meetodikat. Esmalt kasutati halli kasti (ingl *grey box testing*) testimismetoodikat, kuna autori teadmised antud tarkvara programmi koodi ja andmestruktuuri koha pealt olid piiratud (Dadeau & Peureux, 2011). Teise testimismetoodikana kasutas töö autor visuaalset testimist, et näha ebaõnnestumise korral, mis selle põhjustas (Lönnerberg, 2003).

## 4 EduFeedr tarkvara lühitutvustus

EduFeedr on Elgg<sup>20</sup> platvormile juurde ehitatud õppetöökäsitatav voogude lugemistarkvara, mida kasutatakse õpikeskkonna andmete agregeerimisel, kuvades konkreetse kursuse andmeid ning mille kohta tehakse päring lokaalsest andmebaasist (Põldoja, Savitski & Laanpere, 2010). Antud tarkvara sai loodud spetsiaalselt kursuste läbiviimiseks kasutades avatud õpikeskkondi, kus kursusel osavõtjad kasutavad enda personaalseid ajaveebe. EduFeedriga on hetkel võimalik kasutada ajaveebe, mis jooksevad Bloggeri või WordPressi platvormil, tulenevalt piirangust kommentaaride ja uudisvoogude asukoha tuvastamisega. EduFeedr võimaldab hetkel seada ülesse konkreetne kursus, registreerida end soovitud kursusele, luua kursusega seonduvaid ülesandeid, agregeerida postitusi ja kommentaare ja visualiseerida kursusest osavõtjate progressi seoses neile antud ülesannetega ning kuvada omavahelisi seoseid sotsiaalse võrgustikuna. Lähemalt tuleb antud funktsioonidest juttu alampeatükis 4.1.

### 4.1 Funktsionaalsus ja struktuur

Konkreetse kursuse ülesseadmiseks peab selle eestvedaja (õppejõud) omama EduFeedr keskkonnas kontot, kus peale autentimist tekib valik kursuse lisamiseks (ingl *Add Course*). Vajutades antud valikule avaneb kursuse loomise vorm, kus tuleb anda kursusele nimi (ingl *Title*), kirjeldada lühidalt millega täpsemalt tegemist on (ingl *Description*), määratleda silt (ingl *Course Tag*), sisestada ajaveebi link, kuhu kõik kursusega seonduvad ülesanded postitatakse (ingl *Course Blog*), määrata kursusele registreerimise tähtaeg (ingl *Enrollment deadline*) ning millal antud kursus algab ning lõpeb. Kui vastavad lahtrid on täidetud ning vajutatud nupul salvesta (ingl *Save*), siis luuakse uus kursus.

Minnes loodud kursusele jaguneb EduFeedr leht kuueks erinevaks sektsiooniks:

- 1) Kursuse uudisvoog (ingl *Course feed*) — Antud sektsioonis kuvatakse kümme viimast ajaveebi postitust ja kommentaari, mis viitavad konkreetsele läbiviidavale kursusele. Õppejõul on võimalus kursusega mitte seotud postitused ja kommentaarid eemaldada.
- 2) Kursuse info (ingl *Course info*) — Antud sektsioonis näeb teavet loodud kursuse kohta ning vajadusel saab antud infot muuta või kursus kustutada.

---

<sup>20</sup> Elgg - <http://elgg.org/>



- 3) Osavõtjad (ingl *Participants*) — Osavõtjate sektsoonis näeb isikuid, kes on antud kursusele end registreerinud. Registreerimiseks on vaja kursusest osavõtjatel täita lihtne vorm, kus on nõutud eesnimi (ingl *Firstname*), perekonnanimi (ingl *Lastname*), e-maili aadress (ingl *E-mail*) ning kasutatava ajaveebi täpne link (ingl *Blog*). Kursusele registreerimiseks ei pea omama kasutajakontot EduFeedr keskkonnas, kuid peale registreerimistähtaja ületamist, saab ainult kursuse eestvedaja lisada isikuid kursusele.

Osavõtjatel on võimalik jälgida kursusega seotuid postituste ja kommentaaride vooge kasutades endale sobivat tarkvara, kui laadida alla OPML<sup>21</sup> laiendiga failid. Õppejõul on veel võimalus alla laadida vCard<sup>22</sup> ning CSV<sup>23</sup> formaadis failid, millest esimene sisaldab kursusel osalejate kontaktandmeid aadressiraamatu kujul, mida on võimalik lisada teistesse rakendustesse, mis kasutavad samuti aadressiraamatut. CSV laiendiga fail sisaldab kursusel osalejate nimesid, nende e-maili ja ajaveebi aadressi ning hinde veergu iga ülesande jaoks.

- 4) Ülesanded (ingl *Assignments*) — On tegemist ajaveebi postitustega, mis sisaldavad osavõtjatele suunatud ülesandeid. Ülesande loomiseks on vaja märkida ära ajaveeb, kust antud ülesanne postituse kujul asub ning määrata lõpptähtaeg, mis ajaks ülesanne peab lahendatud saama.

Ülesannete lahenduste postitamiseks peavad kursuselt osavõtjad tegema postituse, mis sisaldab lingi kujul viidet ülesande postituse aadressist.

- 5) Progress (ingl *Progress*) — Vastavalt tudengite poolt esitatud ülesannete lahendustele joonistatakse antud sektsioonis diagramm, mis näitab mitu ülesannet on ühe üliõpilase poolt õigeaegselt esitatud sisaldades viiteid õppejõu poolt postitatud ülesannetele kasutades tumesinist värvi. Helesinisega kuvatakse üliõpilaste poolt tehtud postitusi, mis loodi mingi perioodi vahemikus, aga ei viita konkreetsele ülesandele. Vajutades mingil ülesandel navigeeritakse üliõpilase poolt tehtud postitule EduFeedr keskkonnas. Antud postitusi saab aga ainult kommenteerida minnes vastavale ajaveebile, mis antud postitust sisaldab (vt joonis 10).

- 6) Sotsiaalne võrgustik (ingl *Social network*) — Viimane sektsioon näitab seoseid üliõpilaste vahel. Seos tekib, kui keegi kursuselt osavõtjatest lingib või

---

<sup>21</sup> OPML — <http://dev.opml.org/>

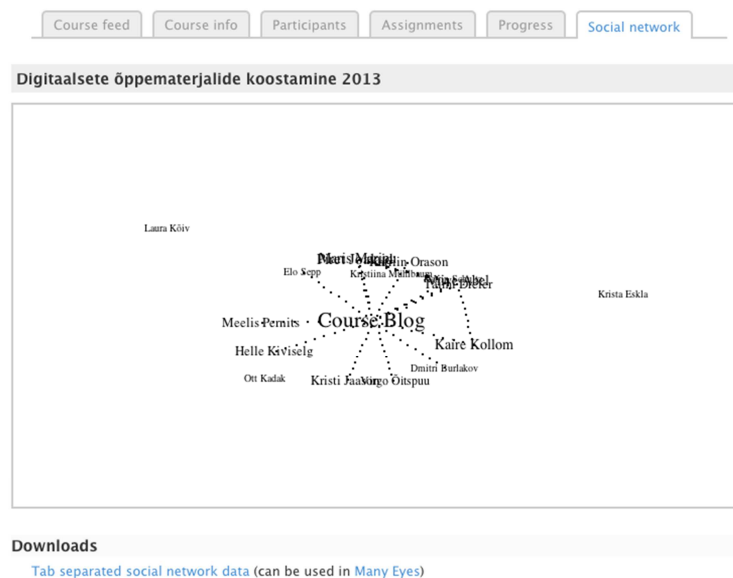
<sup>22</sup> vCard — <http://www.imc.org/pdi/vcardoverview.html>

<sup>23</sup> CSV — <http://creativyst.com/Doc/Articles/CSV/CSV01.htm>

kommenteerib kellegi teise ajaveebi postitust, see salvestatakse vastava aadressi lingina metaandmetesse. Antud võrgustik on võimalik alla laadida TSV<sup>24</sup> formaadis, mida saab sisestada IBMi poolt loodud Many Eyes<sup>25</sup> visualiseerimistarkvarasse (vt joonis 11).



Joonis 10. Progressi diagramm õpilaste poolt esitatud ülesannete kohta



Joonis 11. Kursusel osalejate sotsiaalne võrgustik

<sup>24</sup> TSV — <http://www.iana.org/assignments/media-types/text/tab-separated-values>

<sup>25</sup> Many Eyes — <http://manyeyes.alphaworks.ibm.com/manyeyes/> 5

Vastava võrgustiku loomiseks kasutatakse JSViz<sup>26</sup> Javascripti teeki, mille abil agregeeritakse antud sotsiaalse võrgustiku diagramm kohalikule kettale parandades EduFeedri reageerimisega.

Peale selle suudab EduFeedr genereerida HTML koodi, mis sisaldab *blogrolli* koos kõikide kursusest osavõtjatega, mis on võimalik lisada ajaveebi külgribal olevale vidinale.

## 4.2 Elgg

Nagu eelnevalt mainitud on EduFeedr Elggi veebirakendusele juurde ehitatud õppetöök kasutatav voogude lugemistarkvara, mille läbi toimub õpikeskkonna andmete agregeerimine. Elgg ise on avatud lähtekoodiga GPL v2<sup>27</sup> poolt litsentseeritud sotsiaalse võrgustiku mootor, mis omab jõulist raamistikku, võimaldades luua väga erinevaid sotsiaalseid keskkondi nii haridusasutuste kui ka erinevate ettevõtete jaoks (Sharma, 2008).

Antud veebirakenduse funktsionaalsus pakub rohkeid kujundamise ning kohendamise võimalusi, et tagada kiire prototüüpimine, kasutades LAMP<sup>28</sup> (Linux, Apache, MySQL ja PHP) või WAMP<sup>29</sup> (Windows, Apache, MySQL ja PHP) paketi. Elggi veebirakendus pakub võrguplatvormi, mis kombineerib omavahel ajaveebi elemente, e-portfooliot, uudiste voogude agregeerimist, failide jagamist ning sotsiaalset võrgustiku, kasutades enda raamistikul põhinevat lisavidinate arhitektuuri.

Elggil on mitmeid avatud standardeid nagu RSS ja LDAP<sup>30</sup>, millest viimast kasutatakse autentimiseks. Samuti on integreeritud FOAF<sup>31</sup> ja XML-RPC<sup>32</sup>, mida kasutatakse kolmandate osapoolte ajaveebide kliendipoolsete rakenduste integreerimiseks Elggi.

---

<sup>26</sup> JSViz — <http://code.google.com/p/jsviz/>

<sup>27</sup> GPL v2 — <http://www.gnu.org/licenses/gpl-2.0.html>

<sup>28</sup> LAMP — <http://lamphowto.com/>

<sup>29</sup> WAMP — <http://www.examiner.com/article/what-is-wamp>

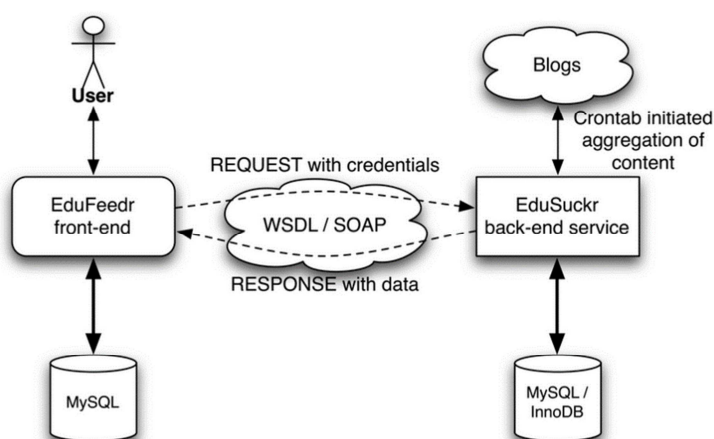
<sup>30</sup> LDAP — [http://msdn.microsoft.com/en-us/library/aa367008\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/aa367008(v=vs.85).aspx)

<sup>31</sup> FOAF — <http://xmlns.com/foaf/spec/>

<sup>32</sup> XML-RPC — <http://xmlrpc.scripting.com/default.html>

### 4.3 EduSuckr

EduSuckr on eraldi seisev veebiteenus, mille abil agregeeritakse ja talletatakse õppetöös läbiviivate kursustega seonduvaid andmevooge registreeritud ajaveebidelt. Andmevoogude hoidmine toimub nii EduFeedri kui ka EduSuckri poolses MySQL andmebaasis, kasutades InnoDB andmete säilitamis mootorit, mille abil on võimalik omavaheliste seoste loomiseks kasutada võõrvõtmeid (ingl *foreign keys*) (vt joonis 12). Selline omavaheliste seoste loomine võimaldab ajaveebi sisestada täpselt ühe korra, kuigi ajaveebi omanik võib kasutada sama ajaveebi ka mõnel teisel kursusel.



Joonis 12. EduFeedri ja EduSuckri süsteemiarhitektuur (Põldoja, Savitski, & Laanpere, 2010)

EduSuckr agregeerib ja talletab iga tunni tagant postitusi ja kommentaare süsteemi registreeritud ajaveebidelt, kasutades Crontabi, mis on UNIX operatsioonisüsteemil põhinev tööplaneerimistarkvara. Andmevooge hoitakse nii EduFeedri kui ka EduSuckri andmebaasis.

## 5 Tracist piletite ümberkolimine GitHubi

Trac<sup>33</sup> ja GitHub<sup>34</sup> on avatud lähtekoodiga vabavaralised tarkvara arendustööriistad, kus on võimalik hallata erinevaid arendusprojekte ning jagada neid erinevate kasutajate vahel. GitHub ja Trac on oma võimaluste poolest aga väga erinevad.

Traci puhul on tegemist keskse repositooriumiga, see tähendab – muudatusi ei ole võimalik viia lokaalselt, vaid muudatusi saab läbi viia ainult repositooriumisse kuuluvas projektis ning need kajastatakse kohe.

GitHubis seevastu on võimalus projekt pearepositooriumist ümber kopeerida igal kasutajal lokaalsele süsteemile ning seejärel vastavad muudatused teha. Kasutaja poolt lokaalis tehtud muudatused saab hiljem üle kanda repositooriumisse GitHubi lokaalse rakenduses oleva käskluse „registreeri (ingl *Check In*)“ abil.

GitHub võimaldab ka erinevaid repositooriume ühe kasutaja alt teise alla kopeerida. Kopeeritud repositooriumis saab teha kõrvaline kasutaja erinevaid muudatusi ning hiljem saab selle lähte repositooriumis need üle kanda andmete tõmbamise päringu (ingl *Pull request*) abil (vt joonis 13)



Joonis 13. Lähte repositooriumisse andmete ülekandmise käsklus GitHubi veebikeskkonnas

Tracist koliti GitHubi, kuna Trac asub Tallinna Ülikooli Haridustehnoloogia keskuse serveris, millele on piiratud juurdepääs (pääsevad ainult ligi ainult isikud, kes omavad Haridustehnoloogia keskuse serveris kasutajakontot). GitHubi puhul oli võimalik kaasata arendusse ka inimesi väljastpoolt.

Tracist oli vaja üle tuua järgnevad muutujad seoses erinevate piletitega EduFeedri kohta:

<sup>33</sup> Trac — <http://trac.edgewall.org>

<sup>34</sup> GitHub — <https://github.com>

**Tabel 4. Vajalikud muutujad iga pileti kohta, mis toodi Tracist GitHubi üle koos väärtustega**

Teadvustaja ( <i>Reporter</i> )	Isik, kes loob arenduspileteid
Tüüp ( <i>Type</i> )	Pileti tüüp ( <i>story, enhancement, defect</i> )
Osa ( <i>Component</i> )	Projekti osa mille juurde antud piletit kuulub
Versioon ( <i>Version</i> )	Projekti versioon mille, juurde antud piletit kuulub
Otsingusõnad ( <i>Keywords</i> )	Otsingusõnad, millega antud piletit otsida
Tähtsus ( <i>Priority</i> )	Pileti tähtsus, mille järgi neid ette võtta
Versapost ( <i>Milestone</i> )	Versaposti lahendamise lõpp tähtaeg
Suunatud ( <i>Assigned to/Owner</i> )	Isik, kes on vastutav antud pileti lahendamise eest
Koopia ( <i>Cc</i> )	Isik keda veel teavitada, seoses mingi pileti kohta tehtud muudatuse osas
Lahendus ( <i>Resolution</i> )	Lahenduse kirjeldus pileti sulgemise kohta ( <i>fixed, invalid, wontfix, duplicate, worksforme</i> )
Seis ( <i>Status</i> )	Pileti praegune olek ( <i>new, assigned, closed, reopened</i> )
Kokkuvõte ( <i>Summary</i> )	Lühikokkuvõte pileti kirjeldatud probleemist
Kirjeldus ( <i>Description</i> )	Pileti olev põhjalik probleemi kirjeldus

Tracist erinevate andmete üle toomiseks (vt tabel 4) GitHubi kasutati PHP-skripti nimega „trac2github.php”<sup>35</sup>.

Antud skriptis viid läbi mõningad muudatused, mis olid vajalikud GitHubi kasutaja autentimiseks ning ajutiste failide tekitamiseks lokaalkettale. All olevas tabelis (vt tabel 5) on täpselt välja toodud, mis väärtust muutujatele määrata tuli, et kõikide pileтите kohta olev informatsioon kanduks Tracist GitHubi.

**Tabel 5. Skriptis *Trac2Github.php* vajalike muutujate määramine, pileтите üle toomiseks GitHubi**

<i>\$username</i>	GitHubi kasutajanimi
<i>\$password</i>	GitHubi kasutaja parool
<i>\$project</i>	GitHubis oleva projekti nimi
<i>\$repo</i>	GitHubis oleva repositooriumi nimi
<i>\$trac_db</i>	Traci andmebaasi nimi ja asukoht, kust andmed võetakse, et GitHubi kanda
<i>\$save_milestones</i>	Versapostide ajutine salvestamise asukohta lokaalkettal
<i>\$save_tickets</i>	Pileтите ajutine salvestamise asukohta lokaalkettal
<i>\$save_labels</i>	Siltide ajutine salvestamise asukohta lokaalkettal

Traci puhul oli tegemist SQLite andmebaasiga ning seepärast tuli määrata *\$trac\_db* väärtuseks „*sqlite:trac:db*“ (vt koodinäide 1) ehk siis kogu koodirida oleks kujul:

```
$trac_db = new PDO('sqlite:trac.db');
```

**Koodinäide 1. Ühenduse loomine Trac andmebaasiga „*Trac.db*“**

<sup>35</sup> Trac2github.php — <https://github.com/trustmaster/trac2github>

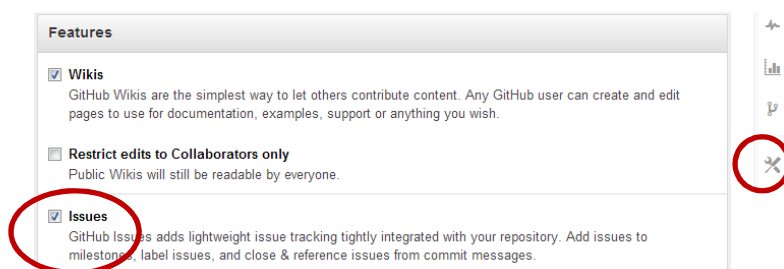
Traci andmebaasi fail Trac.db pidi PHP-skriptiga trac2github.php asuma samas kaustas, mille asukohaks tol hetkel oli C:\xampp\htdocs\.

Muutujad `$save_milestones`, `$save_tickets` ning `$save_labels` kasutati ajutiselt info salvestamiseks vastavate nimelistesse listideisse, milleks olid `save_milestones.list`, `save_tickets.list` ja `save_labels.list`. Antud listid olid enda failitüübilt tekstifailid ning skriptis endas tuli ära määrata, kus antud listid asuvad kuhu andmed salvestatakse (vt koodinäide 2)

```
$save_milestones = 'C:\xampp\ajutine\trac_milestones.list';  
$save_tickets = 'C:\xampp\ajutine\trac_tickets.list';  
$save_labels = 'C:\xampp\ajutine\trac_tickets.list';
```

#### Koodinäide 2. Ajutiste tähiste, piletite, siltide andmete loomine ajutisse faili

GitHubi veebikeskkonnast oli vaja suvandite alt lubada valik tõrked (ingl *Issues*), märkides linnukese antud valiku ette (vt joonis 14). Vastasel juhul sai tööautor tähised ülekantud, aga piletite teiste parameetrite kopeerimisel andis URL veateate „*Failed to convert a ticket: Array ( [message] => Not Found [documentation\_url] => http://developer.github.com/v3 )*“



Joonis 14. Tõrgete (ingl *Issues*) lubamine GitHubi keskkonnas, piletite ülekandmiseks

Tracist GitHubi kandusid kõik 171 piletit ja 10 versteposti (3 avatud ning 7 suletud) üle. Antud piletid ja verstepostid paigutati automaatselt vastavatesse olekute gruppi avatud/suletud (ingl *Open/Closed*) ning viimase mõlema puhul oli ka juurde märgitud põhjus, miks antud pilet või verstepost suletud sai.

## 6 Edufeedri agregeerimislahenduste täiendamine

EduFeedr agregeerimislahenduste täiendamine toimus antud töö raames kasutades eelnevalt loodud kasutajalugusid. Kasutajalugude hulgast valiti kolm Traci versioonihaldustarkvarast GitHubi üle kantud piletit #154<sup>36</sup>, #155<sup>37</sup> ja #156<sup>38</sup>, millest tehakse järgnevat peatükkides juttu. Arendusplatvormina kasutati Windows 8 operatsioonisüsteemi, kuhu oli paigaldatud XAMPP<sup>39</sup> tarkvarapakett, mis tugines WAMP arenduskeskkonnale. Arendusjärgus olevate funktsionaalsuste testimiseks paigaldati antud tarkvarapaketile Elgg, koos EduFeedr ja EduSuckr agregeerimistarkvaraga. Peamine arendus toimus PHP programmeerimiskeeles, mille kirjutamiseks kasutati koodiredaktorit Notepad++.

Arendusprotsessi sai jagada kaheks faasiks: esimeses faasis loodi vastav funktsionaalsus pileтите #154 ja #155 järgi, teises faasis täiendati EduFeedri funktsionaalsust pileti #156 järgi, mille eeliseks olid eelnevalt mainitud piletid. Autor tekitas arenduse jaoks eraldi forki GitHubis ja kood asub neil aadressidel ning tööle lisatud CD plaadil:

- <https://github.com/maddisr/EduFeedr>
- <https://github.com/maddisr/EduSuckr>

### 6.1 Ajaveebipostituste ühendamine ülesannetega (pilet #154)

Varasemalt ei olnud võimalik siduda üliõpilaste poolt tehtud postitust õppejõu poolt loodud ülesandega, kui üliõpilase postitus ei sisaldanud linki antud ülesande postitusele. Selleks loodi vastav lisafunktsionaalsus EduFeedri keskkonna jaoks, mille abil on võimalik käsitsi üliõpilase poolt loodud postituse kujul olev lahendus siduda õppejõu poolt loodud ülesandega (vt joonis 15).

---

<sup>36</sup> Pilet #154 — <http://www.edufedr.org/ticket/154>

<sup>37</sup> Pilet #155 — <http://www.edufedr.org/ticket/155>

<sup>38</sup> Pilet #156 — <http://www.edufedr.org/ticket/156>

<sup>39</sup> XAMPP — <https://www.apachefriends.org/index.html>



## Digitaalsete õppematerjalide koostamine 2014

### Digitaalsete õppematerjalide kvaliteet

28.04.2014 13:50 by Aivi Reimand

10 soovitud kvaliteetse digitaalse õppematerjali loomiseks: Õppematerjali loomisel tuleb arvestada sihtgrupi vajaduste ja lubavustega – mistahes õppematerjal peab olema olemasolevate õppijate jaoks arusaadav, motiveeriv ja ka kättesaadav. Õppematerjal peab olema läbimõeldud ja ühtne – õppetegevused ja sisu vastavad õppeeesmärkidele, struktuur ja õppetegevused on loogilise ülesehitusega ja toetavad õppimisprotsessi. Õppematerjal peab olema õppijakeskne ja õppijat toetav. [...]

28.04.2014 13:50 by Aivi Reimand

Originally posted at: <http://aivixyz.wordpress.com/2014/04/28/digitaalsete-oppematerjalide-kvaliteet/>

- ✓ Assignment 1: Esimene teema: Õpiobjekti mõiste ja õppematerjalide levitamise vahendid
- Assignment 2: Teine teema: õppematerjalide koostamise vahendid
- Assignment 3: Kolmas teema: arvutipõhine testimine
- Assignment 4: Neljas teema: uued tehnoloogiad
- Assignment 5: Viies teema: õppematerjalide autoriõigus
- Assignment 6: Viimane teema: õppematerjalide koostamise protsess ja kvaliteet

edufedr:connect\_post

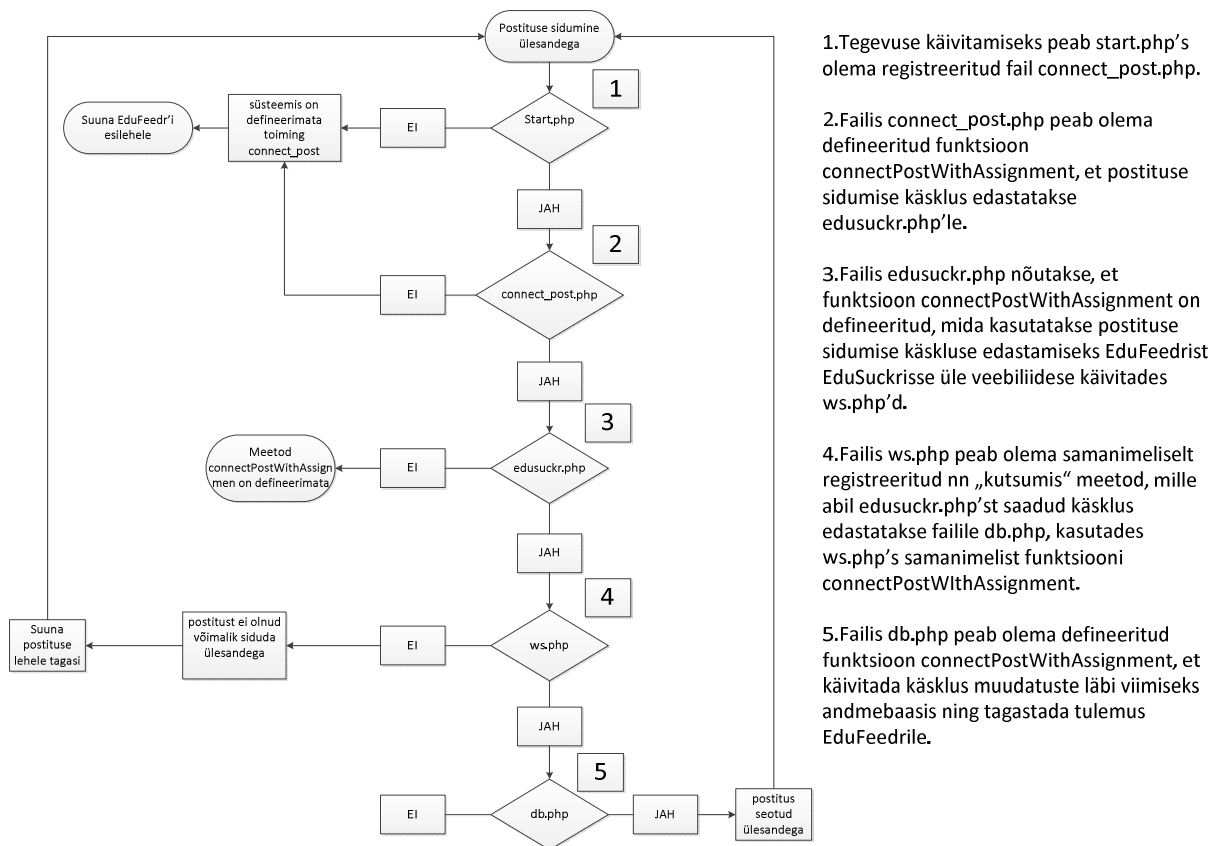
Joonis 15. Postituse sidumine ülesandega

Antud protsess näeks graafiliselt välja järgnev (vt joonis 16).

Piletiga #154 seotud tabelid ja failid:

- *prefix\_course\_rels\_posts* — tabel, kuhu salvestatakse postituse sidumisel ülesandega *assignment\_id*
- *prefix\_assignments* — kursuse ülesannete hoidmiseks mõeldud tabel
- *prefix\_posts* — tabel kursuse postituste hoidmiseks
- *educourse\_post.php* — fail, mille abil käivitatakse postituse sidumise ülesandega käsklus
- *start.php* — fail, kuhu on registreeritud käsklus, mis käivitatakse kasutades *connect\_post.php*'d
- *connect\_post.php* — fail, mille abil edastatakse funktsioon *connectPostWithAssignment* *edusuckr.php*'le
- *edusuckr.php* — fail, kuhu on registreeritud funktsioon *connectPostWithAssignment*, et postituse sidumise funktsioon edastada *EduSuckrisse*
- *ws.php* — fail, mille abil edastakse *EduFeedrist* käsklus *EduSuckrisse*, kasutades funktsiooni *connectPostWithAssignment*

- db.php — fail, mille abil kirjutatakse tabelisse *prefix\_course\_rels\_posts* *assignment\_id* väärtus ning näidatakse EduFeedrile, milline postitus ülesandega seoti



Joonis 16. Protsess diagramm postituse sidumiseks õppejõu poolt loodud ülesandega

Esmalt tuli selle jaoks EduSuckri poolset tabelit nimega *prefix\_course\_rels\_posts* laiendada tulbaga *assignment\_id*, mis võeti tabelist *prefix\_assignments*. Tabelis *prefix\_assignments* omasid *assignment\_id* tulba väärtused unikaalset väärtust. See oli vajalik, et korraga saaks siduda ainult ühte postitust õppejõu poolt loodud ülesandega. Antud tabeli laiendamiseks tuli lisada failile *edusuckr.sql* juurde vastav SQL käsklus (vt koodinäide 3)

```
CREATE TABLE IF NOT EXISTS prefix_course_rels_posts (
course_guid bigint(20) NOT NULL,
link char(255) NOT NULL,
hidden int(1) DEFAULT 0,
assignment_id bigint(20) DEFAULT NULL,
FOREIGN KEY (course_guid) REFERENCES prefix_educourses (course_guid) ON DELETE CASCADE,
FOREIGN KEY (link) REFERENCES prefix_posts (link) ON DELETE CASCADE,
FOREIGN KEY (assignment_id) REFERENCES prefix_assignments (assignment_id) ON DELETE SET NULL,
PRIMARY KEY (course_guid, link),
)ENGINE=InnoDB DEFAULT CHARSET=UTF8;
```

Koodinäide 3. *prefix\_course\_rels\_posts* laiendamine muutujaga *assignment\_id*'ga

Edasi tuli `start.php`'s registreerida vastav tegevus (ingl *action*) `connect_post` (vt koodinäide 4).

```
register_action('edufedr/connect_post', false, $CONFIG->pluginspath .  
'edufedr/actions/connect_post.php');
```

#### Koodinäide 4. Tegevuse defineerimine `start.php`

Fail `connect_post.php` mille abil postituste ühendamine õppejõu poolt püstitatud ülesandega võimaldati, asus ise asukohas `edufedr/actions`. Antud failis oli defineeritud, et sisse loginud kasutajal on ainult õigus siduda üliõpilaste postitusi ülesannetega. Kui postituste sidumine õnnestus, sai õppejõud teatekse „postitus seotud (ingl *post connected*)“, vastasel juhul „postitus ei ole seotud (ingl *post not connected*)“. Mõlemal juhul suunati sisse logitud kasutaja postituse lehele tagasi (vt koodinäide 5).

```
if ($educourse->getSubtype() == 'educourse' && $educourse->canEdit() &&  
edufedrCanEditEducourse($educourse) && $post_id && $assignment_id) {  
    $es = new EduSuckr();  
    $result = $es->connectPostWithAssignment($course_guid, $post_id, $assignment_id);  
    if ($result) {  
        /*translation:Post connected.*/  
        system_message(elgg_echo('edufedr:message:post_connected'));  
    } else {  
        /*translation:Post could not be connected.*/  
        register_error(elgg_echo('edufedr:error:post_not_connected'));  
    }  
    forward("pg/edufedr/view_post/$course_guid/$post_id");  
}
```

#### Koodinäide 5. Postituse sidumiseks ülesandega loodud fail `connect_post.php`

Failis `connect_post.php` nõudis muutuja `$result`, et `edusuckr.php`'s oleks defineeritud funktsioon `connectPostWithAssignment` koos muutujatega, mille väärtused on vaja tagastada. See oli vajalik, et päring edastakse `EduSuckri` poole edasi läbi veebiliidese kasutades `ws.php`'d. Selle jaoks pidi ka antud andmete pärisismeetod `ws.php`'s registreeritud olema ja vastav funktsioon koos muutujatega defineeritud (vt koodinäide 6).

```

$server->register('connectPostWithAssignment', array("course_guid"=>"xsd:int",
"post_id"=>"xsd:int", "assignment_id"=>"xsd:int"), array("result"=>"xsd:int"), WS_URL);
function connectPostWithAssignment($course_guid, $post_id, $assignment_id) {
global $db;
return !isAuthenticated() ? 0 : $db->connectPostWithAssignment($course_guid, $post_id,
$assignment_id);
}

```

#### Koodinäide 6. ws.php's meetodi registreerimine ja funktsiooni defineerimine

Edasi käivitati Edusuckri poolses failis db.php vastav käsklus, mille abil kirjutati tabelisse *prefix\_course\_rels\_posts* tabelist saadud *assignment\_id* väärtus tingimusel, et mõlema tabeli *course\_guid* ja tabelite *prefix\_posts* ja *prefix\_course\_rels\_post* *link* väärtused olid samad (vt koodinäide 7).

```

$post = $this->getPostById($post_id);
if ($post) {
query = "UPDATE ".DB_PREFIX."course_rels_posts SET assignment_id=$assignment_id WHERE
course_guid=$course_guid AND link='".$post['link']."'";
return $this->query($query);
}

```

#### Koodinäide 7. *assignment\_id* väärtuse kirjutamine *prefix\_course\_rels\_posts*

Edasi sai õppejõud teate, et postitus seotud (ingl *post connected*) ning suunati ta tagasi vastava üliõpilase postitusele ning oli näha, et postitus on ülesandega seotud (vt joonis 17).

Originally posted at: <http://martinkleinert90.wordpress.com/2014/03/31/my-home-town-attraction/>  
Task 3 (deadline 1.05). Evaluating destination. 🌸

#### Joonis 17. Postituse kuvamine seotud ülesandega

Sarnane protsess oli ka postituse lahtiühendamisel ülesandega, kus defineeriti *connectPostWithAssignment* funktsiooni asemel *disconnectPostWithAssignment* ning fail, mis vastava tegevuse käivitas oli nimega *disconnect\_post.php*, kus lahti sidumise õnnestumise korral sai kasutaja teateks „postitus lahti ühendatud (ingl *post disconnected*)“ ja ebaõnnestumise korral sai teateks „postitust ei õnnestunud lahti ühendada (ingl *post could not be disconnected*)“.

## 6.2 Kommentaaride ühendamine autoritega (pilet #155)

Õppejõul peab olema võimalus siduda, üliõpilaste poolt kirjutatud kommentaare ajaveebi postituse autoriga (vt joonis 18), nii et peale sidumist kuvatakse kommentaari autori eesnime

ja perekonna nime ning ajaveebi postituse autorit eesnime ja perekonnanime, kelle postitusega kommentaar seotud on. Antud protsess on graafiliselt kujutatud joonisel 19.

Piletiga #155 seotud tabelid ja failid:

- *prefix\_course\_rels\_comments* — tabel, kuhu salvestatakse kommentaari sidumisel autoriga *participant\_id*
- *prefix\_participants* — kursuselt osavõtjate kajastamiseks mõeldud tabel
- *prefix\_comments* - üliõpilaste kommentaaride tabel
- *educourse\_post.php* — fail, mille abil käivitatakse kommentaari sidumise autoriga käsklus
- *start.php* — fail, kuhu on registreeritud käsklus, mis käivitatakse kasutades *connect\_comment\_with\_participant.php*'d
- *connect\_comment\_with\_participant.php* — fail, mille abil edastatakse funktsioon *connectCommentWithParticipant* *edusuckr.php*'le
- *edusuckr.php* — fail kuhu on registreeritud funktsioon *connectCommentWithParticipant*, et kommentaari sidumise funktsioon edastada edasi *EduSuckrisse*
- *ws.php* — fail, mille abil edastatakse *EduFeedrist* käsklus *EduSuckrisse*, kasutades funktsiooni *connectCommentWithParticipant*
- *db.php* — fail, mille abil kirjutatakse tabelisse *prefix\_course\_rels\_comments* *participant\_id* ning näidatakse *EduFeedrile*, milline kommentaar autoriga seoti

Katrin Uurman to Tuuli Perolainen

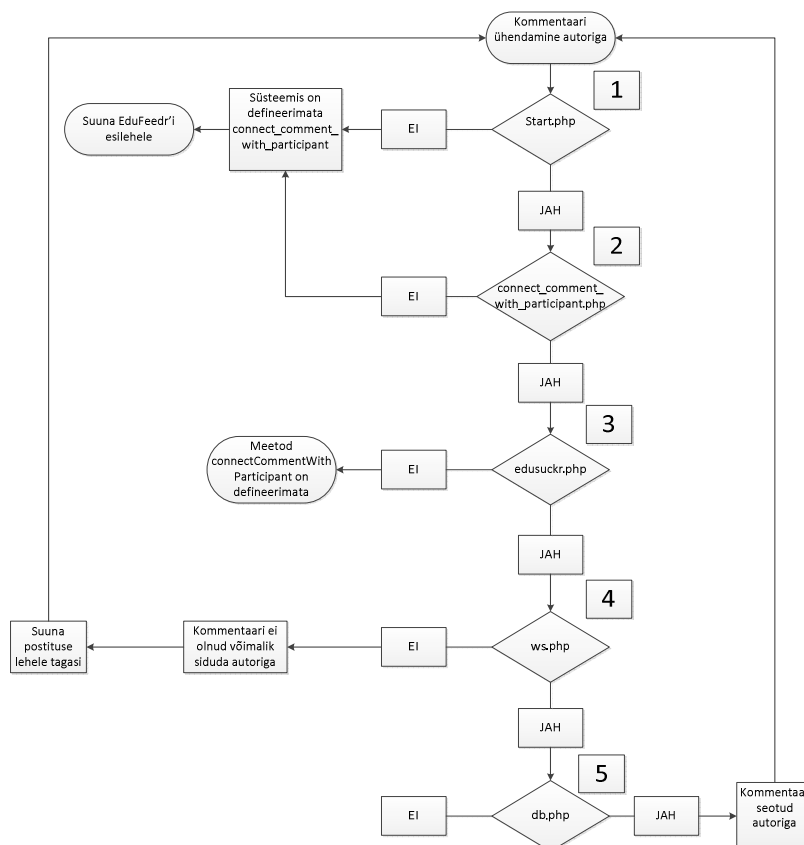
04.05.2014 11:00

Oleks tahtnud ka viitamise lisada, kuid kuna Eve märkis selle juba ära, siis minu poolt: kasutatavus erinevates keskkondades ja seadmetes.

Connect this comment with participant:

[Post a comment](#)

**Joonis 18. Kommentaari sidumine autoriga**



1. Tegevuse käivitamiseks peab start.php's olema registreeritud fail connect\_comment\_with\_participant.php.

2. Failis connect\_comment\_with\_participant.php peab olema defineeritud funktsioon connectCommentWithParticipant, et kommentaari sidumise käsklus edastataks edusuckr.php'le.

3. Failis edusuckr.php nõutakse, et funktsioon connectCommentWithParticipant on defineeritud, mida kasutatakse kommentaari sidumise käskluse edastamiseks EduFeedrist EduSuckrisse üle veebiliidese käivitades ws.php.

4. Failis ws.php peab olema samanimeliselt registreeritud nn „kutsumis“ meetod, mille abil edusuckr.php'st saadud käsklus edastatakse failile db.php, kasutades ws.php's samanimelist funktsiooni connectCommentWithParticipant.

5. Failis db.php peab olema defineeritud funktsioon connectPostWithAssignment, et käivitada käsklus muudatuste läbi viimiseks andmebaasis ning tagastada tulemus EduFeedrile.

**Joonis 19. Kommentaari sidumist autoriga kujutav protsess diagramm**

Nagu eelneva funktsionaalsuse täiustamise juures, tuli ka praeguse juures esmalt laiendada EduSuckri poolset tabelit. Tabel mida laiendati kandis nime *prefix\_course\_rels\_comments*. Antud tabelilt laiendada tulbaga *participant\_id*, mis võeti tabelist *prefix\_participants*. Tabelis *prefix\_participants* omasid *participant\_guid* tulba väärtused unikaalset väärtust. See oli vajalik, et korraga saaks siduda ainult ühte isikut ajaveebi postitusele kirjutatud kommentaariga. Antud tabeli laiendamiseks tuli lisada failile *edusuckr.sql* juurde vastav SQL käsklus (vt koodinäide 8).

```
CREATE TABLE IF NOT EXISTS prefix_course_rels_comments (
course_guid bigint(20) NOT NULL,
link char(255) NOT NULL,
hidden int(1) DEFAULT 0,
participant_id bigint(20) DEFAULT NULL,
FOREIGN KEY (course_guid) REFERENCES prefix_educourses (course_guid) ON DELETE CASCADE,
FOREIGN KEY (link) REFERENCES prefix_comments (link) ON DELETE CASCADE,
FOREIGN KEY (participant_id) REFERENCES prefix_participants (participant_guid) ON DELETE SET NULL,
PRIMARY KEY (course_guid, link)
) ENGINE=InnoDB DEFAULT CHARSET=UTF8;
```

**Koodinäide 8. *prefix\_course\_rels\_comments* tabeli laiendamine *participant\_id* tulbaga.**

Seejärel tuli samuti start.php's vastav tegevus nimega *connect\_comment\_with\_participant* registreerida, et faili *connect\_comment\_with\_participant.php* abil oleks võimalik kommentaari sidumine autoriga (vt koodinäide 9).

```
register_action('edufedr/connect_comment_with_participant', false, $CONFIG->pluginspath .  
'edufedr/actions/connect_comment_with_participant.php');
```

#### **Koodinäide 9. Tegevuse registreerimine, selle esile kutsumiseks**

Fail *connect\_comment\_with\_participant.php* asus samuti asukohas *edufedr/actions*. Kommentaari sidumise õigus autoriga oli ainult sisse loginud kasutajal. Kui kommentaari sidumine autoriga oli edukas, saadi teateks „kommentaari ühendatud (ingl *comment connected*)“, vastasel juhul „postitus ei ole seotud (ingl *comment could not be connected*)“. Peale seda suunati sisse logitud kasutaja postituse lehele tagasi, kuhu kommentaar kirjutatud oli (vt koodinäide 10).

```
if ($educourse->getSubtype() == 'educourse' && $educourse->canEdit() &&  
edufedrCanEditEducourse($educourse) && $comment_id && $participant_id) {  
    $es = new EduSuckr();  
    $result = $es->connectCommentWithParticipant($course_guid, $comment_id, $participant_id);  
    if ($result) {  
        /*translation:Comment connected.*/  
        system_message(elgg_echo('edufedr:message:comment_connected'));  
    } else {  
        /*translation:Comment could not be connected.*/  
        register_error(elgg_echo('edufedr:error:post_not_connected'));  
    }  
    forward("pg/edufedr/view_post/$course_guid/$post_id");  
}
```

#### **Koodinäide 10. Fail, mis defineeris tegevust kommentaari sidumiseks üliõpilasega**

*connect\_comment\_with\_participant.php* olev muutuja *\$result* nõudis, et *edusuckr.php*'s oleks defineeritud funktsioon *connectCommentWithParticipant* koos muutujatega, mille väärtused on vaja tagastada. Selle abil edastati päring *EduSuckrisse* kasutades *ws.php*'d. *ws.php*'s pidi andmete pärimis meetod koos väljastavate tulemustega registreeritud olemas ning samuti vastav funktsioon koos muutujatega (vt koodinäide 11).

```

$server->register('connectCommentWithParticipant', array("course_guid"=>"xsd:int",
"comment_id"=>"xsd:int", "participant_id"=>"xsd:int"), array("result"=>"xsd:int"), WS_URL);
function connectCommentWithParticipant($course_guid, $comment_id, $participant_id) {
global $db;
return !isAuthenticated() ? 0 : $db->connectCommentWithParticipant($course_guid, $comment_id,
$participant_id);
}

```

#### Koodinäide 11. *connectCommentWithParticipant* meetodi ja funktsioon

Seejärel käivitati `db.php`'s vastav käsklus, mille abil kirjutati tabelisse *prefix\_course\_rels\_comments* tabelist saadud *participant\_guid* väärtus tingimusel, et mõlema tabeli *course\_guid* ja tabelite *prefix\_comments* ja *prefix\_course\_rels\_comments* link väärtused olid samad (vt koodinäide 12).

```


$comment = $this->getCommentById($id);
if ($comment) {
$query = "UPDATE ".DB_PREFIX."course_rels_comments SET participant_id=$participant_id WHERE
course_guid=$course_guid AND link='". $comment['link']."'";
return $this->query($query);
}

```

#### Koodinäide 12. Andmebaasi tabelis *prefix\_course\_rels\_comments* olevasse välja *participant\_id* väärtuse kirjutamine

Õppejõud sai teate, et kommentaar ühendatud (ingl *comment connected*) ning suunati ta tagasi vastava üliõpilase postitusele kus oli näha, et kommentaar on autoriga seotud (vt joonis 20).

#### 1 comments:

Helis Oja to Martin Kleinert 

04.04.2014 13:23

Hey! Põltsamaa seems a lot more attractive place, after reading Your suggestions. I would absolutely like to visit wine cellar, it looks great, depending on Your comments on it. Thank You, now i can say i know more about Your home town.

[hide](#)

[Post a comment](#)

#### Joonis 20. Kommentaari autor on seotud kommentaariga

Sarnaselt kommentaari sidumisega autoriga, toimus ka kommentaari lahti ühendamine autorist. See toimus läbi funktsiooni *disconnectCommentWithParticipant* ning fail, mis vastava tegevuse käivitas oli nimega *disconnect\_comment\_with\_participant.php*, kus lahti sidumise õnnestumise korral sai kasutaja teateks „kommentaari lahti ühendatud (ingl *comment disconnected*)“ ja ebaõnnestumise korral sai teateks „kommentaari ei õnnestunud lahti ühendada (ingl *comment could not be disconnected*)“.



### 6.3 Õppija profiil (pilet #156)

Igal kursusel osavõtjal on enda profiili leht EduFeedris, kus kuvatakse järgnevat infot:

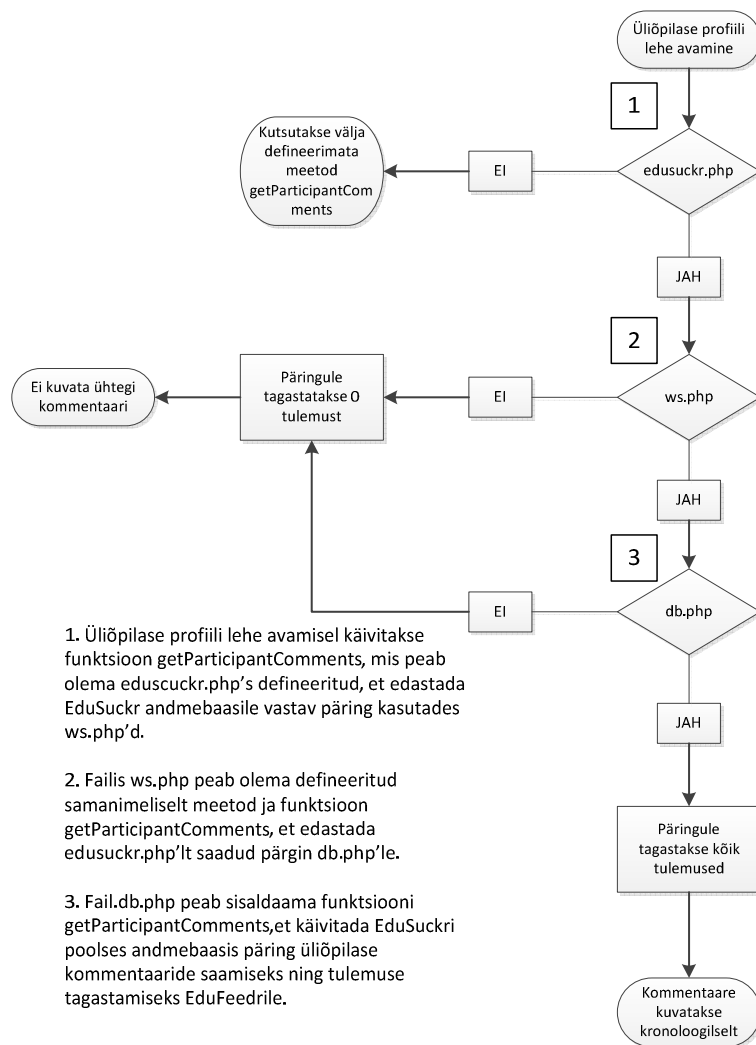
- 1) Profiilipildi kuvamine
- 2) Kursusel osavõtja eesnimi ja perekonnanimi
- 3) Üliõpilase e-maili aadress (Seda näevad ainult õppejõud, kui on EduFeedri keskkonda sisse loginud)
- 4) Üliõpilase poolt tehtud ajaveebi postitusi, koos kuupäevaga. Antud postitused peavad olema kronoloogilises järjekorras – värskemad postitused kuvatakse esimesena (ülevalt – alla)
- 5) Kuvatakse üliõpilase poolt kirjutatud kommentaare koos kuupäevadega, asetsedes kronoloogilises järjekorras — värskemad kommentaarid kuvatakse esimesena minnes ülevalt- alla

Kui vajutatakse üliõpilase profiili all olevale postitusele või kommentaarile siis suunatakse õigele postitusele või kommentaarile edasi.

Antud pileti all oli varasemalt valmist tehtud ülal olevast nimekirjast esimene, teine, kolmas ja neljas punkti antud töö autoril oli vaja teostada viies punkt – kommentaaride kuvamine üliõpilase profiili lehel. Antud protsess on kujutatud graafiliselt joonisel 21.

Piletiga #156 seotud tabelid ja failid:

- *prefix\_course\_rels\_comments* - Tabel, kust saadakse *participant\_id* alusel üliõpilase kommentaarid
- *prefix\_comments* - üliõpilaste kommentaaride tabel
- *participant\_profile.php* - fail, mille abil käivitatakse käsklus *getParticipantComments* üliõpilase profiilil kommentaaride kuvamiseks kronoloogilises järjekorras
- *edusuckr.php* - fail, kus on defineeritud funktsioon *getParticipantComments* üliõpilase kommentaaride saamise päringu edastamiseks
- *ws.php* - fail, millest edastatakse päring EduFeedrist edasi EduSuckrisse
- *db.php* - fail, kus on defineeritud *getParticipantComments*, mis tagastab kõik üliõpilase kommentaarid, mida kuvada EduFeedris asuval üliõpilase profiililehel



**Joonis 21. Kommentaaride kronoloogiline kuvamine üliõpilase profiilil**

Failis `participant_profile.php` on defineeritud, et EduSuckrist toimuks kommentaaride päring kasutades funktsiooni `getParticipantComments` ning kommentaare kuvatakse juhul kui neid on rohkem kui null. Iga kommentaarile tuleb lisada kuupäev (päev, kuu, aasta), kuvades kasutaja profiilil kommentaari ajaveebi postituse autori nimega, kelle postitust kommenteeriti (vt koodinäide 13).

```

$es = new EduSuckr;
$comments = $es->getParticipantComments($vars['entity']->guid, $vars['participant']->id);
if (!$comments && is_array($comments)) {
$comments = array();
}
/*translation:Comments*/
$body .= '<h3>'.elgg_echo('edufedr:latest:comments').' ('.sizeof($comments).'):</h3>';
if (is_array($comments) && sizeof($comments)>0) {
$body .= '<table id="profile_comments"><tbody>';
foreach ($comments as $comment) {
$body .= "<tr>";
$body .= '<td>'.date('d.m.Y', $comment['date']).'</td>';
$body .= '<td>'.elgg_view('output/url', array('value' =>
$vars['url'].'pg/edufedr/view_post/'.$vars['entity']->getGUID().'/'.$comment['post_id'],
'text' => $comment['post_author'])).'</td>';
$body .= "</tr>";
}
$body .= '</tbody></table>';
}
}

```

### Koodinäide 13. Kommentaaride kuvamine üliõpilase profiililehel

Edasi pidi olema defineeritud EduFeedri poolses failis edusuckr.php funktsioon *getParticipantComments*, mis vastava päringu EduSuckrisse edasi saatis (vt koodinäide 14). Kui antud funktsiooni defineeritud ei olnud sai kasutaja üliõpilase profiilile minnes teateks „*Call to undefined method EduSuckr:: getParticipantComments in participant\_profile line 55*“.

```

function getParticipantComments($course_guid, $participant_id) {
return unserialize($this->client->call('getParticipantComments', array($course_guid,
$participant_id)));
}

```

### Koodinäide 14. Funktsiooni defineerimine edusuckr.php's

EduSuckri poole peal pidi failis ws.php meetod *getParticipantComments* registreeritud olema, mis tagastaks väärtused string kujul. Samuti pidi olema defineeritud ka EduSuckri funktsioon *getParticiPantsComments* (vt koodinäide 15). Kui ühte neist või kumbagi defineeritud ei olnud tagastati tulemuseks 0 ning üliõpilase profiililehel ei kuvatud ühtegi tema poolt kirjutatud kommentaari.

```

$server->register('getParticipantComments', array("course_guid"=>"xsd:int",
"participant_id"=>"xsd:int"), array("result"=>"xsd:string"), WS_URL);
/* Function getParticipantComments */
function getParticipantComments($course_guid, $participant_id) {
global $db;
return !isAuthenticated() ? 0 : $db->getParticipantComments($course_guid, $participant_id);
}

```

#### Koodinäide 15. ws.php's olev meetod ja funktsioon andmete päringuks db.php'st

Kui antud meetod ja funktsioon koos muutujatega oli defineeritud, käivitati failis db.php vastav päring, mis tagastaks kõik üliõpilase poolt tehtud kommentaarid. Selleks väljastati kõik väljad tabelist *prefix\_comments* ning nendele vastavad väljad *prefix\_course\_rels\_comments* tabelist. Nendele väljadele millele vastet ei leitud väljastati tulemuseks NULL. Tähtis oli et *prefix\_comments* tabelis link väli oleks sama, mis *prefix\_course\_rels\_comments* tabelis oleva lingi väli. Hiljem muudeti käsu *serialize*<sup>40</sup> abil funktsioon ümber stringiks (vt koodinäide 16). Antud päringu läbi viimisel saadud tulemust kuvati üliõpilase profiili lehel (vt joonis 22). Kui db.php's vastavat päringu tegemis funktsiooni ei sisaldanud, siis tagastati kasutajale samuti null tulemust ning ühtegi kommentaari üliõpilase profiili lehel ei kuvatud.

```

function getParticipantComments($course_guid, $participant_id) {
$query = "SELECT DISTINCT id, c.link AS link, base, title, date, content, author, blogger_id,
post_id, post_author FROM ".DB_PREFIX."comments c LEFT JOIN ".DB_PREFIX."course_rels_comments
r ON c.link=r.link WHERE r.course_guid=$course_guid AND r.participant_id=$participant_id AND
!r.hidden ORDER BY date DESC";
$result = $this->query($query);
$comments = array();
if($result && mysql_num_rows($result)) {
while($comment = mysql_fetch_array($result)) {
if ($comment['blogger_id']) {
if ($fn = $this->getFullNameByBloggerId($comment['blogger_id'])) {
$comment['author'] = $fn;
}
}
$comments []= $comment;
}
return serialize($comments);
}
}

```

#### Koodinäide 16. db.php's olev päring, mis pidi tagastama üliõpilase kirjutatud kommentaarid

---

<sup>40</sup> Serialize — muudab massiivi stringiks ümber

## Digitaalsete õppematerjalide koostamine 2014



Eve Saare

Blog: <http://eshtm13digitoppematerjalid.wordpress.com/>

### Blog posts (7):

- 01.05.2014 Kuues teema: Õppematerjalide koostamise protsess ja kvaliteet &#8211; kvaliteetse digitaalse õppematerjali 10 käsku
- 07.04.2014 Viies teema: Õppematerjalide autoriõigus
- 24.03.2014 Neljas teema: Uued tehnoloogiad õppematerjalide koostamiseks
- 11.03.2014 Kolmas teema: Arvutipõhine testimine
- 24.02.2014 Teine teema: Õppematerjalide koostamise vahendid
- 09.02.2014 Esimene teema: Õpiobjekti mõiste ja õppematerjalide levitamise vahendid
- 06.02.2014 Õpileping

### Comments (7):

- 01.05.2014 Tuuli Perolainen
- 01.05.2014 Katrin Uurman
- 23.04.2014 Nameless
- 20.04.2014 Eve Saare
- 23.03.2014 Eve Saare
- 11.03.2014 Kristel Tereping
- 16.02.2014 Kadi Ennok

## Joonis 22. Kommentaaride kuvamine üliõpilase profiilil

Seni on ajaveebipõhistes õpikeskkondades puudunud mugav viis kõiki õppija poolt erinevates ajaveebides tehtud kommentaare korraga näha. Uus õppija profiil EduFeedris lahendab selle probleemi tuues välja lingid kõigile õppija poolt kirjutatud kommentaaridele. Lisaks sellele kuvatakse profiilis linke õppija postitustele ning Gravatar<sup>41</sup> teenuse kaudu saadud profiilipilti.

---

<sup>41</sup> <http://en.gravatar.com>

## 7 Testimine

Scrumi toote omanik (ingl *Scrum product owner*) viis läbi testimise piletite #154, #155 ja #156, kus pilet #154 kirjeldas ajaveebi postituste sidumist õppejõu poolt püstitatud ülesannetega, pilet #155 kommentaari sidumist autoriga ning pilet #156 kasutaja profiili kuvamist. Antud piletite kasutajalugudes olid kirjeldatud täpsed funktsioonide kirjeldused, nende nõuded ja kujundusparameetrid. Toote omanik kasutas tehtud täienduste testimiseks musta kasti testimismetoodikat (ingl *black box testing*). Kui testimise käigus tulid välja funktsioonide ebatäpsused või puudulikkus, siis tehti nende kohta veapiletid järgmise arendussprindi jaoks. Antud testimine oli sobilik toote omaniku jaoks, kuna läbiviidud funktsionaalsete täienduste juures ei pidanud toote omanik mõistma programmeerimiskeelt, millega antud tarkvara funktsionaalsust täiendati. Testija andis ainult sisendparameetreid klikkides nupul, mis vastava funktsiooni käivitas ning kontrollis, kas väljundtulemus on õige.

Töö autor rakendas arenduse käigus läbi viidud täienduste funktsionaalsuse testimiseks dünaamilise testimise kahte erinevat meetodikat — halli kasti (ingl *grey box testing*) ja visuaalse testimise (ingl *visual testing*) testimismetoodikat. Testid aga sooritati musta kasti tasemel, uurides sisse antud sisenditest saadud väljund tulemusi. Selle juures jälgis autor andmebaasis toimunud muudatusi, et kontrollida läbi viidud funktsionaalsete täienduste omavahelist koostööd. Esimest meetodikat kasutas antud töö autor, kuna see võimaldas viia testimist läbi lõppkasutaja vaatenurga ning teist testimismetoodikat, et autor näeks probleemi tekkimist, selle asemel, et keegi teine oleks seda talle kirjeldanud. See aitas paremini probleemi selgusele jõudmise ja arusaamiseni. Järgnevas alampeatükis on välja toodud erinevad testlood, mille põhjal testimine läbi viidi ning testimise tulemusena saadud veaaruaanded, kui vigu esines.

### 7.1 Testlood

Koostati 4 testlugu, mis koosnevad kirjeldusest, eeltingimustest, põhivoost, järeltingimustest ja veaaruaandest

#### 7.1.1 Testlugu 1: Postituse ühendamise ülesandega

**KIRJELDUS:** Õppejõud seob üliõpilase poolt tehtud postituse tema poolt loodud ülesandega

**EELTINGIMUSED:**

1. Õppejõud on end EduFeedr keskkonda sisse loginud;
2. Õppejõud on loonud kursuse;

3. Üliõpilane on end kursusele registreerinud või on õppejõu poolt registreeritud;
4. Õppejõud on loonud kursusele ülesande;
5. Üliõpilane on teinud ajaveebi postituse kasutades enda ajaveebi;
6. Andmed on agregeeritud;
7. Õppejõud asub kursuse avalehel.

#### PÕHIVOOG:

1. Õppejõud vajutab ajaveebi postituse pealkirjale ;
2. Õppejõud suunatakse postitusele edasi;
3. Õppejõud valib ülesannete rippmenüüst loodud ülesande;
4. Õppejõud vajutab nupul edufeedr: Connect.

#### JÄRELTINGIMUSED:

1. Üliõpilase tehtud postitus on õppejõu poolt loodud ülesandega seotud;
2. Õppejõud suunatakse vastava postituse lehele tagasi;
3. Tekib lahti ühendamise ikoon.

#### VEAARUANNE:

- 1) Postituse ülesandega sidumise nupu tekst „ühenda (ingl *connect*)“ on puudu;
- 2) Postituse ülesandega sidumise menüüd ja nuppu kuvatakse sisse logimata kasutajatele;
- 3) Postituse ülesande küljest lahtiühendamise ikooni kuvatakse sisselogimata kasutajatele;
- 4) Esimene valik ülesannete menüüs peab olema „Vali ülesanne (ingl *Select an assignment*)“;
- 5) Postituse loomisaega kuvatakse topelt;
- 6) Osadel lehtedel on `<div>` element lõpetamata.

### 7.1.2 Testlugu 2: Kommentaari ühendamine autoriga

KIRJELDUS: Õppejõud seob kirjutatud kommentaari autoriga

#### EELTINGIMUSED:

1. Õppejõud on end EduFeedr keskkonda sisse loginud;
2. Õppejõud on loonud kursuse;

3. Üliõpilased on end kursusele registreerinud või on õppejõu poolt registreeritud;
4. Üliõpilased on teinud postitusi kasutades endi ajaveebe;
5. Üliõpilased on üksteise postitusi kommenteerinud;
6. Andmed on agregeeritud;
7. Õppejõud asub kursuse avalehel.

#### PÕHIVOOG:

1. Õppejõud vajutab postituse pealkirjal;
2. Õppejõud suunatakse postitusele edasi;
3. Õppejõud kerib lehekülje lõppu;
4. Õppejõud valib autorite rippmenüüst autori nime;
5. Õppejõud vajutab nupul edufeedr:connect\_comment.

#### JÄRELTINGIMUSED:

1. Kommentaar seotakse autoriga;
2. Õppejõud suunatakse vastava postituse lehele tagasi;
3. Tekib lahti ühendamise ikoon.

#### VEAARUANNE:

- 1) Kommentaari autoriga sidumise nupu tekst „ühenda (ingl *connect*)“ on puudu;
- 2) Kommentaari autoriga sidumise menüüd ja nuppu kuvatakse sisse logimata kasutajatele;
- 3) Kommentaari lahtiühendamise ikooni kuvatakse sisselogimata kasutajatele;
- 4) Esimene valik ühendamata kommentaari autorite jaoks peaks olema „Vali autor (ingl *Select an author*)“;
- 5) Kommentaaride autoreid menüüs tuleks kuvada tähestikulises järjekorras autori perekonnanime järgi;
- 6) TSV fail ei sisalda ühendatud kommentaare ja autoreid.

### 7.1.3 Testlugu 3: Üliõpilase profiili kuvamine

KIRJELDUS: Õppejõule kuvatakse üliõpilase profiil, kelle nime peale ta vajutas

#### EELTINGIMUSED:



1. Õppejõud on end EduFeedr keskkonda sisse loginud;
2. Õppejõud on loonud kursuse;
3. Üliõpilased on end kursusele registreerinud või on õppejõu poolt registreeritud;
4. Üliõpilased on teinud ajaveebi postitusi kasutades endi ajaveebe;
5. Üliõpilased on üksteise postitusi kommenteerinud;
6. Õppejõud on kommentaare autoritega sidunud;
7. Andmed on agregeeritud;
8. Õppejõud asub kursuse avalehel.

#### PÕHIVOOG:

1. Õppejõud vajutab valiku „osavõtjad (ingl *participants*) peal;
2. Õppejõule kuvatakse kõiki kursuselt osavõtjaid ees- ja perekonnanime järgi;
3. Õppejõud vajutab vastava üliõpilase nime peal, kelle profiili ta soovib vaadata;
4. Õppejõud suunatakse valitud kasutaja profiilile.

#### JÄRELTINGIMUSED:

- 1) Õppejõule kuvatakse üliõpilase profiilipilti;
- 2) Õppejõule kuvatakse üliõpilase ajaveebi aadressi;
- 3) Õppejõule kuvatakse üliõpilase e-maili aadressi;
- 4) Õppejõule kuvatakse üliõpilase poolt tehtud postituste pealkirju;
- 5) Õppejõule kuvatakse üliõpilase kommentaare, mis on õppejõud autoriga sidunud.

#### **7.1.4 Testlugu 4: Üliõpilase progressi kuvamine**

**KIRJELDUS:** Õppejõul on võimalik graafiliselt jälgida, millised ülesanded on üliõpilase poolt esitatud

#### **EELTINGIMUSED:**

1. Õppejõud on loonud kursuse;
2. Üliõpilane on end kursusele registreerinud või on õppejõu poolt registreeritud;
3. Õppejõud on sidunud üliõpilase postituse enda poolt loodud ülesandega;
4. Üliõpilane on teinud postituse, mis sisaldab viidet õppejõu poolt loodud ülesandele;
5. Üliõpilane on teinud postituse õppejõu poolt loodud ülesande esitamise tähtaja vahemikus;
6. Andmed on agregeeritud;

7. Õppejõud asub kursuse avalehel.

#### PÕHIVOOG:

1. Õppejõud vajutab valiku „Progress (ingl *progress*) peal;
2. Õppejõule suunatakse progressi lehele.

#### JÄREL TINGIMUSED:

- 1) Kuvatakse tumesiniselt postitust, mis on õppejõu poolt loodud ülesandega seotud;
- 2) Kuvatakse tumesiniselt postitust, mis sisaldab viidet õppejõu poolt loodud ülesandele;
- 3) Kuvatakse helesiniselt postitust, mis on tehtud ülesande perioodi ajal.

#### VEAARUANNE:

Ülesande perioodi järel tehtud postitust kuvatakse õige ülesandega ühendamise järel progressi tabelis topelt

#### NÄIDE:

Üliõpilane tegi 1.ülesande kohta postituse 2.ülesande ajal ning 2.ülesande kohta pole veel postitust teinud. Kui õppejõud ühendas üliõpilase poolt tehtud postituse 1.ülesandega, siis peale ühendamist postituse ühendamist ülesandega kuvatakse seda tabelis nii 1. kui ka 2. ülesande kohal.

## **7.2 Ettepanekud EduFeedri edasiarenduseks**

Antud peatükis on kirja pandud mõtted, milliseid funktsionaalsusi võiks EduFeedr veel sisaldada, et antud õpikeskkonna kasutamine oleks mugavam.

### **1) Postituse automaatne ühendamine ülesandega**

Hetkel kuvatakse postituse ülesandega sidumise menüüd ka nende postituste juures, mille puhul õpilane on lisanud lingi ülesandele kursuse blogis (kuvatakse progressi tabelis tumesinise värviga). Nende postituste juures võiks ülesannete rippmenüü kuvamise asemel näidata kohe, et postitus on ülesandega seotud ning kuvada igaks juhuks lahtiühendamise nuppu.

### **2) Ülesande pakkumine postituse aja järgi**

Postituse ülesandega sidumise rippmenüüs kuvatakse hetkel vaikimisi esimesena kõige esimest ülesannet. Selle asemel võiks EduFeedr kuvada vaikimisi esimesena seda ülesannet, mille perioodi jooksul postitus on tehtud.

### **3) *Pingback* kommentaaride eristamine kursuslaste kirjutatud kommentaaridest**

EduFeedr võiks eristada *pingback* kommentaare kursuslaste poolt kirjutatud kommentaaridest. *Pingback* kommentaar tekkib juhul kui lingitakse mõnele kursusel tehtud postitusele. Kõige sagedamini juhtub see siis, kui õpilased lingivad ülesande postitusele kursuse ajaveebis. Seetõttu peab õppejõud regulaarselt peitma kursuse avalehel *pingback* kommentaare, et kommentaaride all oleks esikohal need kommentaarid, kus toimub tegelik arutelu. *Pingbackide* ja kommentaaride eristamise korral võiks õppija profiilil kuvada eraldi tema poolt tehtud linkimisi kursuslaste postitustele.

### **4) Õppejõudude kuvamine kommentaaride menüüs**

Hetkel saab kommentaare siduda ainult üliõpilastega. Kommentaaride autorite rippmenüü nimekirjast puuduvad õppejõud. Viimased annavad aga tihti üliõpilastele tagasisidet kommentaari kujul. Seepärast võiks olla kommentaaride autorite rippmenüüs kuvatud ka õppejõudude nimed, kes kursust läbi viivad.

### **5) Kommentaari sidumine isikuga, kes ei võta kursuselt osa**

Hetkel saab kommentaare saab siduda ainult ühe kursuse raames olevate üliõpilastega. Kursuse läbiviimisel on tulnud ette mõningaid olukordi, kus kommenteerijaks on inimesed, kes kursusest osa ei võta. Selliste kommentaaride puhuks võiks olla valik “Väljapoolt kursust kommenteerija”.

## Kokkuvõte

Antud bakalaureusetöö eesmärgiks oli täiendada ajaveebipõhiste kursuste haldusvahendi EduFeedr funktsionaalsust, et oleks võimalik ühendada õppija postitusi ülesannetega ning kirjutatud kommentaare autoritega. Käesoleva eesmärgi saavutamiseks analüüsis töö autor esmalt ajaveebide õppetöös kasutamise seonduvat kirjandust ning sellele järgnes EduFeedri koodbaasi migreerimine GitHub repositooriumisse, kuna Trac asus Tallinna Ülikooli Haridustehnoloogia keskuse serveris, kuhu oli juurdepääs ainult isikutel, kes omavad Haridustehnoloogia keskuse serveris kasutajakontot. Seejärel täiendati EduFeedr agregeerimislahendusi ning lõpetuseks testiti tehtud täiendusi Scrumi toote omaniku (ingl *Scrum product owner*) ja käesoleva bakalaureusetöö autori poolt.

Testimise käigus tulid välja läbiviidud funktsionaalsete täienduste tulemusena uued vead, mille kohta tehti kasutajalugudel põhinevad arenduspiletid. Samuti tuli ilmsiks viga, mis esines enne praeguste funktsionaalsete täienduste läbiviimist EduFeedris. Antud keskkonnas kuvati tehtud postitust progressi tabelis topelt, kui see oli loodud õppejõu poolt püstitatud ülesande perioodi järel ning kui antud postitus oli õige ülesandega ühendatud.

Samuti tõi autor välja ideed EduFeedr ajaveebide agregeerimistarkvara edasiarendamiseks, mis muudaksid antud tarkvara kasutamist tõhusamaks ja mugavamaks.

Antud bakalaureusetöö raames uuris töö autor ajaveebide kasutamisevõimaluste kohta õppetöös, omandas teadmisi Scrum arendusmeetodi kohta, õppis läbi viima arendustöid kasutades Elgg raamistiku ning õppis viima läbi testimist, kasutades erinevaid testimismetoodikaid.

Käesolevas töös on 56 lehekülge, töö sisaldab 22 joonist, viite tabelit, 41 joonealust viidet ja 26 kasutatud kirjandusallikat.

## Kasutatud kirjandus

Antriainen, O. (2012). *Ajaveebid*. Loetud 12. märts 2014 aadressilt [http://eope.ee/\\_download/euni\\_repository/file/2589/4.moodul\\_2\\_ajaveebid.pdf](http://eope.ee/_download/euni_repository/file/2589/4.moodul_2_ajaveebid.pdf)

Dadeau, F., & Peureux, F. (2011). Grey-Box Testing and Verification of Java/JML. *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. Los Alamitos, CA: IEEE Computer Society. doi:10.1109/ICSTW.2011.30

Dillenbourg, P., Schneider, D., Paraskevi, S. (2002). Virtual Learning Environments. A. Dimitracopoulou (toim.), *Proceedings of the 3rd Hellenic Communication Technologies in Education* (lk 3–18). Rhodes: Kastaniotis Editions.

Downes, S. (2010). New Technology Supporting Informal Learning. *Journal of Emerging Technologies in Web Intelligence*, 2(1), 27–33. doi:10.4304/jetwi.2.1.27-33

Fiedler, S. H. D., & Väljataga, T. (2013). Challenging Learning Myths through Intervention Studies in Formal Higher Education. T. Ley, M. Ruohonen, M. Laanpere, & A. Tatnall (toim), *Open and Social Technologies for Networked Learning* (lk 141–146). doi:10.1007/978-3-642-37285-8\_15

Goktas, Y., & Demirel, T. (2012). Blog-enhanced ICT courses: Examining their effects on prospective teachers' ICT competencies and perceptions. *Computers & Education*, 58(3), 908–917. doi:10.1016/j.compedu.2011.11.004

Johnson, S. M., Osmond, A., & Holz, R. J. (2009). Developing a current awareness service using really simple syndication (RSS). *Journal of the Medical Library Association*, 97(1), 52–54. doi:10.3163/1536-5050.97.1.011

Jones, D. (2013). BIM – Feed Aggregation Management and Marking | The Weblog of (a) David Jones on WordPress.com. Loetud 25. aprill 2014 aadressil <http://davidtjones.wordpress.com/research/bam-blog-aggregation-management/>

Laanpere, M., Pata, K., Normak, P., & Põldoja, H. (2014). Pedagogy-driven Design of Digital Learning Ecosystems. *Computer Science and Information Systems*, 11(1), 419–442. doi:10.2298/CSIS121204015L

Linkback. (kuupäev puudub). *Wikipedia*. Loetud 2. mai 2014 aadressil <http://en.wikipedia.org/wiki/Linkback>

Lönnberg, J. (2003). *Visual testing of software* (magistritöö). Loetud aadressil <http://www.cs.hut.fi/~jlonnber/VisualTesting.pdf>

Mikoluk, K. (2013, 20. juuni). Best Blogging Platform: Wordpress, Blogger, Tumblr, SquareSpace, or Typepad? Loetud aadressil <https://www.udemy.com/blog/best-blogging-platform/>

Nottingham, M., & Sayre, R. (2005). *The Atom Syndication Format*. Loetud 30. märts 2014 aadressil <https://www.ietf.org/rfc/rfc4287>

O'Reilly, T. (2005, 30. september). *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. Loetud aadressil <http://oreilly.com/web2/archive/what-is-web-20.html>

Ostrand, T. (2002). Black-Box Testing. *Encyclopedia of Software Engineering*. doi:10.1002/0471028959.sof022

Paulsen, M. F. (2003). *Online Education and Learning Management Systems. Global E-learning in a Scandinavian Perspective*. Bekkestua: NKI Forlaget.

Põldoja, H., Savitski, P., & Laanpere, M. (2010). Aggregating Student Blogs with EduFeedr. Lessons Learned from First Tryouts. F. Wild, M. Kalz & M. Palmér (toim), *Mashup Personal Learning Environments 2010. Proceedings of the 3rd Workshop on Mashup Personal Learning Environments. In conjunction with the 5th European Conference on Technology-Enhanced Learning (ECTEL'10): Sustaining TEL, Barcelona, Spain, September 29, 2010*. Aachen: CEUR-WS.

Schwaber, K., & Sutherland, J. (2009). *The Scrum Guide*. Loetud aadressil <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf>

Scrum.ee. (kuupäev puudub). *Miks Scrum?* Loetud 25. aprill 2014 aadressil <http://scrum.ee/miks-scrum>

Sharma, M. (2008). *Elgg Social Networking: Create and Manage Your Own Social Network Site Using This Free Open-source Tool*. Birmingham: Packt Publishing Ltd.

Soylu, A., Wild, F., Mödritscher, F., Desmet, P., Verlinde, S., & De Causmaecker, P. (2011). Mashups and Widget Orchestration. W. I. Grosky, Y. Badr, & R. Chbeir (toim), *MEDES '11: Proceedings of the International Conference on Management of Emergent Digital EcoSystems* (lk 226–234). New York, NY: ACM. doi:10.1145/2077489.2077532

Zhu, L., Sun, A., & Choi, B. (2011). Detecting spam blogs from blog search results. *Information Processing and Management*, 47(2), 246–262. doi:10.1016/j.ipm.2010.03.006

Tomberg, V., Laanpere, M., Ley, T., & Normak, P. (2013). Sustaining Teacher Control in a Blog-Based Personal Learning Environment. *The International Review of Research in Open and Distance Learning*, 14(3), 109–133.

Wild, F., Mödritscher, F., & Sigurdarson, S. (2008). Designing for Change: Mash-Up Personal Learning Environments. *eLearning Papers*, 9, 1–15.

WordPress. (kuupäev puudub). *Embeds*. Loetud 30. märts 2014 aadressil <https://codex.wordpress.org/Embeds>

Väljataga, T., & Fiedler, S. (2009). Supporting students to self-direct intentional learning projects with social media. *Educational Technology & Society*, 12 (3), 58–69.

## Summary

### **Aggregating Blogs in an Online Course: the Case of EduFeedr**

This bachelor's thesis studies different approaches for aggregating blog posts and comments in online courses where learners use blogs. It is related to the development of EduFeedr feed reader that has been developed in Tallinn University. The goal of this thesis was to improve the functionality of EduFeedr software so that it would be possible to connect blog posts with course assignments and comments with the participants who have posted the comments. Although there are some automatic methods for connecting posts with assignments and comments with authors, these methods are not always reliable. Therefore this thesis aims to develop a manual method for connecting blog posts with assignments and comments with authors.

In the first chapter the author is analysing literature related to use of blogs in online courses. In the second chapter, the author gives an overview of some relevant aggregation tools. The third chapter focuses on the methodology that is based on some elements of Scrum and some methods of software testing. The fourth chapter provides an overview of the functionalities and technical architecture of the EduFeedr system. The fifth chapter describes the migration of user stories and development tickets from Trac to GitHub. In the sixth chapter the author presents the development of three new functionalities:

- connecting blog posts with assignments;
- connecting comments with authors;
- displaying comments in user profiles.

In the final chapter the new functionalities are tested using both black box, grey box, and visual testing approaches.

Final result of this bachelor's thesis is a new working prototype of EduFeedr software that allows to connect blog posts with assignments, comments with participants, and displays correct data on the participant profile. Also, author provides a number of recommendations for further development of EduFeedr.