

CombiNation – konsepti vieraan kielen oppimiseen

Juha Ahola, Leo Jäppinen



Tekijät Juha Ahola, Leo Jäppinen	
Koulutusohjelma Tietojenkäsittely	
Opinnäytetyön nimi CombiNation – konsepti vieraan kielen oppimiseen	Sivu- ja liitesivumäärä 33 + 11
<p>Tämän toiminnallisen opinnäytetyön tavoitteena oli luoda ohjelma, joka auttaa Haaga-Helian opiskelijoita sekä henkilökuntaa oppimaan vieraita kieliä. Työssä kehitetty CombiNation on konsepti, johon rakennettiin automatisoitu ohjelma. Käyttäjä kirjoittaa ohjelmaan, mitä kieltä haluaa opettaa ja mitä kieltä haluaa oppia. Kun ohjelma löytää vastakkaisparit, se lähettää molemmille käyttäjille sähköpostin, jonka jälkeen käyttäjät ovat keskenään yhteydessä ja vastuu kielen opettamisesta ja oppimisesta siirtyy heille.</p> <p>Työhön sisältyy ”vaihe-vaiheelta –ohje”, jossa kerrotaan, miten omasta ideasta tehdään pilvessä toimiva ohjelma ja mitä vaatimuksia sille on. Lisäksi työssä verrataan PHP:n ja Javan suosiota ja toiminnallisuutta ohjelmointikielinä sekä selitetään, miten ohjelman koodi toimii.</p> <p>Ohjelman käyttötapauskaavioista, sekvenssidiagrammeista ja ulkoasujen rautalangoista löytyvät selostuksilla varustetut kuvat. Työssä käydään läpi myös ohjelman elinkaarta, eli mistä saatiin idea ohjelmalle ja miten sitä lähdettiin toteuttamaan, sekä kuvataan mahdollisia jatkokehityssuunnitelmia.</p> <p>Dokumentin lopusta löytyy vielä pohdintaa haasteellisista tilanteista ja niiden vaikutuksista opinnäytetyöhön. Kerrotaan myös, kuinka opinnäytetyö kasvatti meitä kehittäjinä.</p>	
Asiasanat Tietojenkäsittely, Ohjelmointi, PhP, Haaga-Helia, CombiNation	

Sisällys

1	Johdanto	1
2	Sanasto ja Käsitteet	3
2.1	Sanasto.....	3
2.2	Käsitteet.....	3
3	PHP ohjelmointikielenä	7
3.1	PHP:n suosio	7
3.2	PHP versiot.....	9
3.3	PHP vs Java	9
4	Projektin kulku.....	11
4.1	Idea opinnäytetyölle	11
4.2	Vaatimukset ja niiden saavutukset	11
4.3	Rautalangat	12
4.4	Ulkoasu.....	12
4.5	Ohjelmointikieli.....	12
4.6	Tietokanta.....	13
4.7	Säilytyspalvelu	13
4.8	Toteutus.....	14
5	Käyttötapaus ja koodaaminen	15
5.1	Käyttötapauskaavio.....	15
5.2	Tiedosto arkkitehtuuri.....	16
5.3	Sekvenssikaavio	17
5.4	Hakemusten käsittely	20
6	Web-sovellus konseptista tuotteeksi.....	22
6.1	Ohjelmointikielten valitseminen	22
6.2	Tietokannan valitseminen	22
6.3	Ohjelmointiympäristö.....	23
6.4	Domainin ja webhotellin hankinta.....	24
6.5	Tiedostojen vienti palvelimelle.....	24
7	Pohdinta.....	27
7.1	Esteet ja ratkaisut	27
7.2	Käyttäjätestaus	27
7.3	Jatkokehitysehdotukset.....	27
7.4	Päätelmät.....	28
	Lähteet	30
	Liitteet.....	32
	Liite 1. COmbiNationin MyNet esittelysivusto	32
	Liite 2. Ohjelman lomakesivu (combination.php)	33

Liite 3. Ohjelman tilastosivu (statistics.php)	34
Liite 4. Kuva ohjelman lähettämästä sähköpostiviestistä	35
Liite 5. CombiNation parin löydyttyä saatava sähköposti (molemmat osapuolet)	36
Liite 6. Admin kirjautumissivu (login.php).....	37
Liite 7. Admin näkymä 1 (admin.php)	37
Liite 8. Admin näkymä 2 (adminDeleted.php)	38
Liite 9. Admin näkymä 3 (adminMatched.php).....	38
Liite 10. Kuva Postmanista	39
Liite 11. Tandem Programme rekisteröintilomake.....	40
Liite 12. Kuva ohjelman rautalankamallista (MyNet näkymä).....	41
Liite 13. Kuva lomakesivun rautalankamallista	42
Liite 14. Kuva admin näkymän rautalankamallista	43
Liite 15. Helpdeskin vastaus protopyyntöön	44
Liite 16. Palautelomake 1	45
Liite 17. Palautelomake 2	46
Liite 18. Palautelomake 3	47

1 Johdanto

Tämän toiminnallisen opinnäytetyön tavoitteena oli luoda ohjelma, joka auttaa Haaga-Helian opiskelijoita sekä henkilökuntaa oppimaan vieraita kieliä. Tavoite ohjelman valmistumiselle oli kesä 2018, ennen syyslukukauden vaihto-oppilaiden saapumista. Pintapuolisesti tavoitteen saavuttaminen tarkoitti sitä, että ohjelma toimii MVP-tyylillä kevyesti ja helposti toteutettuna, mutta täyttäen kaikki tekniset kriteerit. Näihin kriteereihin kuului ohjelman automatisointi, validointi, performanssi ja ohjelman yksinkertainen käyttö. Ohjelma nimettiin CombiNationiksi, ja sen toivotaan menestyvän niin, että se otetaan ammattikorkeakoulujen yhteiseen käyttöön introjen yhdistyessä PEPPI:iin. Toivotaan myös, että kaikki assistenttilinjalaiset rekisteröityisivät joka lukukausi ohjelmaan, sillä heidän koulutusohjelmaan kuuluu yhden ylimääräisen kielen opiskelu.

Dokumentin alusta löytyy sanastoa ja käsitteitä projektissa käytetyistä termeistä, työkaluista, ympäristöistä ja ohjelmointikielistä. Jokaisessa kappaleessa on pienimuotoinen johdanto, jossa pohjustetaan lukijaa kappaleen sisällöstä. Mukana on ”vaihe-vaiheelta –ohje”, jossa kerrotaan, miten omasta ideasta tehdään pilvessä toimiva ohjelma ja mitä vaatimuksia sille on. Tässä osiossa selitetään, mitä työkaluja ohjelman pilveen viemiselle vaaditaan ja käytetään ohjelmalle valittua ”hostingpalvelu.fi” - palveluntarjoajaa havainnollistamiseen. Osiossa verrataan lisäksi PHP:n ja Javan suosiota ja toiminnallisuutta ohjelmointikielinä, sekä selitetään, miten ohjelman koodi toimii. Työn koodiosuus sisältää valitun ohjelmointikielen metodeja ja menetelmiä, joita ohjelmassa käytetään. Ohjelman käyttötapauskaavioista, sekvenssidiagrammeista ja ulkoasujen rautalangoista löytyvät selostuksilla varustetut kuvat. Työssä käydään läpi ohjelman elinkaarta, eli mistä saatiin idea ohjelmalle ja miten sitä lähdettiin toteuttaa. Ohjelman rakentaminen dokumentoidaan työssä kronologisessa järjestyksessä. Lopuksi kirjataan havainnot ylös ja kerrotaan mahdollisista jatkokehityssuunnitelmista, sekä liitetään käyttäjäpalautteet liitteinä.

Opinnäytetyön toimeksiantajana toimii Haaga-Helia, jota edustaa Haaga-Helian hallintojohtaja, Ari Hälikkä. Haaga-Helia on kahden aikaisemman korkeakoulun yhdistymä, joka aloitti toimintansa vuonna 2007. Korkeakoulu kouluttaa mm. myynnin-, palvelun-, tietojenkäsittelyn- ja liiketalouden ammattilaisia.

Tässä opinnäytetyössä kehitetty **CombiNation** on ohjelma, johon voi osallistua kuka tahansa Haaga-Helian opiskelija tai työntekijä. Se on konsepti, johon rakennettiin automatisoitu ohjelma, johon käyttäjä kirjoittaa, mitä kieltä haluaa opettaa ja mitä kieltä haluaa oppia. Kun ohjelma löytää vastakkaisparit, se lähettää molemmille käyttäjille sähköpostin, jonka jälkeen käyttäjät ovat keskenään yhteydessä ja vastuu kielen opettamisesta ja oppimisesta siirtyy heille. Tarkempi kuvaus ohjelman toiminnasta löytyy kuvitettuna luvusta 5.1, "Käyttötapauskaavio".

2 Sanasto ja Käsitteet

2.1 Sanasto

Open Source = Kaikille avoin lähdekoodi.

Loppukäyttäjä = Henkilö, joka käyttää luotua sovellusta.

Työkalut = Projektiin vaadittavat ohjelmat.

Intra = Intranet, eli organisaation sisäinen kommunikaatio-ohjelma.

Portattu = sijoitettu myös toiseen ympäristöön.

PEPPI = Kolmen korkeakoulun yhteinen intranet, joka ei ole vielä tuotannossa.

CombiNation pari = Kaksi ihmistä, jotka CombiNation –ohjelma on yhdistänyt.

Lähdekoodi = Lähdekoodilla tarkoitetaan ohjelmoinnissa tietokoneohjelman tekstimuotoista ohjelmointikielistä listausta.

Tietokanta = Säiliö muuttuville ja listattaville arvoille, jota ohjelma käyttää.

Palvelin = Tietokone jossain muualla, jossa tietoa säilytetään ja kutsutaan.

Skriptikieli = Kommentosarjakieli, jolla kirjoitetaan komentosarjoja, eli skriptejä. Näillä automatisoidaan koodia, ettei ohjelmointikieltä tarvitse kirjoittaa.

Responsiivisuus = Eri tyylien skaalautuminen näytön resoluutiota vaihdettaessa.

Rautalangat = Piirros taikka suunnitelma tulevan ohjelman ulkoasusta ja käytettävyydestä. Voidaan tehdä useilla eri alustoilla, kuten tietokoneella tai paperilla.

UI ja UX = Ulkoasu, eli user interface ja käytettävyys, eli user experience.

2.2 Käsitteet

Eclipse

Eclipse on Javaan pohjautuva open source -ohjelmointiympäristö, joka on suunniteltu integroitujen kehitysympäristöjen luomiseen. Se on tunnettu useista liitännäisistä, jotka tukevat ohjelmistokehittäjiä kirjoittamaan ja testaamaan koodia eri kielillä. Eclipse on myös Haaga-Helian standardoitu ohjelmointiympäristö.

Brackets

Brackets on Adoben kehittämä kevyt open source -ohjelmointiympäristö. Brackets täyttää yksinkertaisen ohjelmoinnin vaatimukset ja se on suunniteltu HTML-, CSS- ja JavaScriptin nopeaan käyttöön, mutta kykenee myös lukemaan muita kieliä, kuten PHP:tä.

MySQL

MySQL (My Structured Query Language) on open source relaatiokannan hallintajärjestelmä. MySQL on keskeinen asia melkein kaikissa open source PHP ohjelmissa.

MariaDB

Maria database on yksi maailman suosituimmista tietokantapalvelimista. Se on alkuperäisten MySQL kehittäjien luoma open source ohjelma. MariaDB muuntaa datan jäsenneltyihin tietoihin erinäisille ohjelmille.

phpMyAdmin

phpMyAdmin on PHP:lla kirjoitettu ilmainen ohjelmatyökalu, jolla voi helposti hallinnoida operaatioita MySQL:ssa ja MariaDB:ssa. phpMyAdmin antaa käyttäjälle visuaalisen työkalun, joka generoi käyttäjän haluamat tietokantauseet ja -taulut. phpMyAdminia käytetään lokaalina tietokantana, johon syötetään tunnuksia ja tauluja, joiden toimivuutta voi testata, ennen pilvitietokantaan tai muuhun vastaavaan siirtymistä.

XAMPP

XAMPP on open source ohjelma, joka toimii lokaalina isäntänä tai palvelimena. Se on yhteensopiva Apache palvelimen, MariaDB:n ja PHP:n kanssa.

Postman

Postman API (Application programming interface) pystyy lähettämään kokoelmia tietokantaan ja käsittelemään dataa Postmanista käsin. Kantoihin pystyy siis lisäämään dataa käyttämättä itse tietokannanhallintajärjestelmää.

GitHub

GitHub on pilvessä sijaitseva säilytyspalvelu. Sitä käytetään yleisimmin oman koodin säilömiseen, mutta siellä voi koodin lisäksi säilyttää muitakin tiedostoja, kuten

dokumentteja ja PDF-tiedostoja. GitHubiin pääsee käsiksi GIT-kielellä ja sieltä saa ladattua koodia usealle eri koneelle. Se on yksi yleisimpiä koodin säilöntäpalveluita, jonka avulla ohjelmoijat jakavat koodiaan muiden kanssa tehdessään yhteistyötä.

HTML

HTML (Hyper Text Markup Language), on standardi merkintäkieli, jota käytetään web-sivujen luomiseen. HTML selittää selaimelle koodisivun rakenteen, jonka selaimen käyttäjä näkee verkkosivuna.

CSS

Css (Cascading Style Sheets) on ohjelmointikieli, joka kertoo HTML-sivulle, miten data tulee näyttää loppukäyttäjälle.

JavaScript

JavaScript on skriptikieli, joka pystyy määrittelemään HTML-sivun käyttäytymisen. Esim. klikkaamalla HTML-elementtiä, pystytään muuttamaan sen arvoa halutulla tavalla.

JSON

Palvelimen ja selaimen keskeinen kommunikaatio on mahdollista vain tekstinä. JSON (JavaScript Object Notation) muuttaa JavaScript-objektit tekstimuotoon, jotta palvelin osaa tulkita niitä. JSON-data pystytään myös muuttamaan tekstimuodosta JavaScript-objekteiksi.

AJAX

Ajax pystyy lukemaan dataa palvelimelta senkin jälkeen, kun selain on kokonaan ladannut sivun. Muutoksia pystyy tekemään sivua uudelleenlataamatta ja dataa pystyy lähettää ja hakea taustalla palvelimen toteuttaessa muita komentoja.

SQL

SQL (Structured Query Language) on kaiksita suosituin kyselykieli, jolla lisätään, muokataan ja käsitellään tietokantaan menevää dataa.

Java

Java on korkeatasoinen Sun Microsystemsin kehittämä ohjelmointikieli, joka on olio-ohjelmointia. Javassa yleensä luokka määrittää oliot ja metodit omiin luokkiinsa. Java luetaan JVM:n (Java Virtual Machine) läpi, jonka ansiosta Java toimii useilla eri laitteilla.

PHP

PHP on kovassa käytössä oleva open source -ohjelma, joka on suunniteltu monikäyttökoodaukseen. Tämä skriptikieli on erityisen yhteensopiva web-kehityksen kanssa ja sen pystyy upottamaan HTML: n.

Bootstrap

Bootstrap on open source koodikirjasto. Bootstrapissä on useita eri valmistryylejä, joita voidaan kutsua yhdellä komennolla, sen sijaan että kirjoitetaan koko koodia alusta loppuun. Kirjasto tarjoaa erilaisia ratkaisuja HTML-, CSS- ja JavaScript kielille, sekä vahvoja liitännäisiä, jotka on rakennettu JQuerylle. Bootstrapin vahvuuksiin kuuluu sovellusten responsiivisuuden helpotettu ohjelmointi.

JQuery

JQuery on kevyt JavaScript kirjasto, jonka tarkoitus on antaa enemmän toimintoja vähemmällä koodilla. Kirjaston avulla pystyy tekemään usean rivin JavaScriptiä jopa yhdellä rivillä koodia. JQueryä käytetään mm. Elementtien efekteihin ja animointiin.

3 PHP ohjelmointikielenä

PHP:n (Personal Home Page) syntytarina sai alkunsa vuonna 1994 kun Rasmus Lerdorf julkaisi ensimmäisen version nimeltä PHP/FI. Ensimmäiset PHP versiot erosivat suuresti nykyisestä PHP:stä, mutta vuonna 1997 julkaistu PHP 3.0 muistuttaa hyvin paljon nykyistä PHP:tä. Koodia voidaan laittaa HTML koodin sekaan, tai tehdä omia tiedostoja joita kutsutaan toisesta PHP-tiedostosta. Aluksi se oli vain skriptauskieli, mutta ajansaatossa siitä kehittyi monipuolinen olio-ohjelmointia tukeva ohjelmointikieli. (Polcode Team 2016)

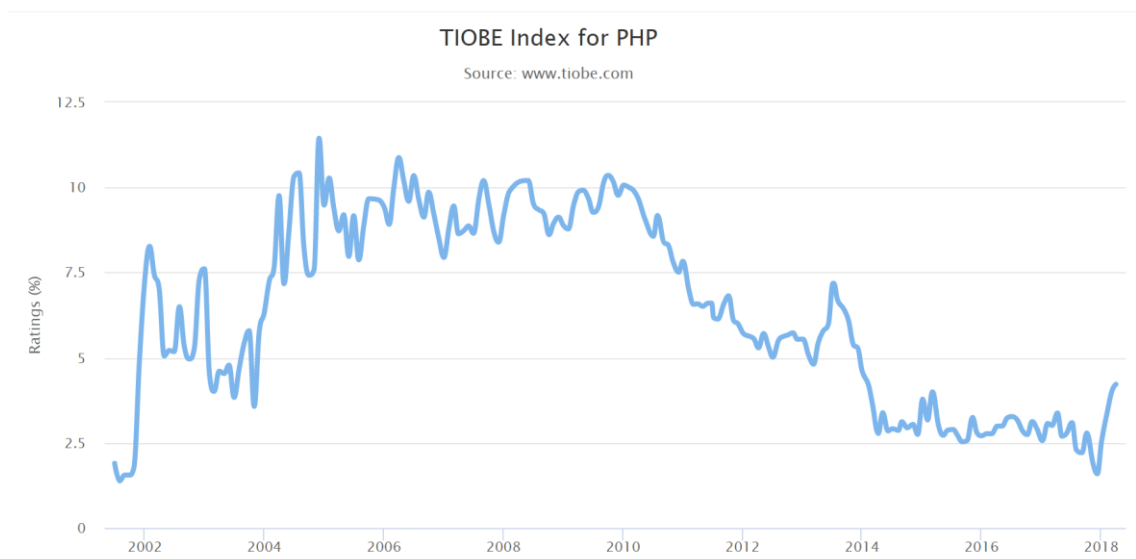
```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>
```

3.1 PHP:n suosio

PHP on erittäin suosittu ohjelmointikieli dynaamisten web-sivustojen luontiin. TIOBE indexin mukaan se on tällä hetkellä (07.05.2018) 7. käytetyin ohjelmointikieli 4.2 % osuudella. PHP:n suosio oli huipussaan vuonna 2010, jolloin se oli 3. käytetyin kieli noin 10 % osuudella.



Kuva 1. Tiobe Index for PHP (Tiobe, 2018)

Vaikka TIOBE:n indexissä PHP:n suosiossa on havaittavissa selvää laskua, noin 83 % nettisivustoista käyttää PHP:ta back-end kielenä.

Suosittuja PHP:lla rakennettuja sivustoja ovat mm.

- Facebook
- Wikipedia
- Wordpress.com
- Yahoo
- Tumblr

(Goyal 2017)

PHP:n hyvät puolet

PHP koodia on helppo ymmärtää ja se helpottaa sivujen luontia ja ylläpitoa. Kaikki koodi sijaitsee .php -tiedostoissa, jotka linkitetään toisiinsa "require" -komennolla. Sen avulla funktiot saa erilleen HTML koodista. Muissa ohjelmointikielissä, kuten Javassa joudutaan menemään luokkien ja alaluokkien läpi ymmärtääkseen, kuinka ohjelma toimii.

```
<?php
require('funktiot.php');
?>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      tulostaJotain();
    ?>

  </body>
</html>
```

Yksi PHP:n parhaista puolista on sen suosio. Sen takana on valtava yhteisö ohjelmoijia, jotka tekevät lisää frameworkkeja PHP:lle. Suosion ansiosta ongelmiin on helppo löytää ratkaisu esimerkiksi Stackoverflowsta. (Logic Heart Pvt Ltd 2016)

PHP:n huonot puolet

PHP erottuu muista kielistä epäjärjestelmällisyydellään. Funktiot voidaan kirjoittaa eri tavoilla ja niille annettavat parametrit voivat olla satunnaisessa järjestyksessä. Tämä vaikeuttaa ohjelman jatkokehittämistä ja kehittäjä saattaa joutua tutkimaan ohjelman dokumentaatiota. Epäjärjestelmällisyyttä lisää se, ettei parametreille vaadita tietotyyppejä, kuten: String, Integer ja Double. Tämä hankaloittaa ohjelman jatkokehittämistä.

Taulukko 1. Esimerkki tietotyypeistä

Java	PHP
int luku=5	\$luku = 5
String sana="esimerkki"	\$sana = "esimerkki"

Koska PHP on helppo kieli, siihen on helppo kirjoittaa haavoittuvaista koodia.

Tyypillisimpiä tietoturva-aukkoja ovat SQL-injektiot. Helppouden takia PHP-koodista löytyy usein huonosti tehtyä koodia. (Polcode Team 2017)

3.2 PHP versiot

PHP:sta on julkaistu useita eri versioita, joista suosituin on PHP 5. Version 5 päivittäminen versioon 7 saattaa poistettujen ominaisuuksien takia aiheuttaa yhteensopivuusongelmia vanhan koodin kanssa, minkä takia versio 5 on yhä suositumpi kuin 7 (W3techs.com 2018a). Opinnäytetyö projektissa käytetään PHP 7.1.9. versiota.

Taulukko 2. Viimeisimmät PHP versiot (Php.net 2018)

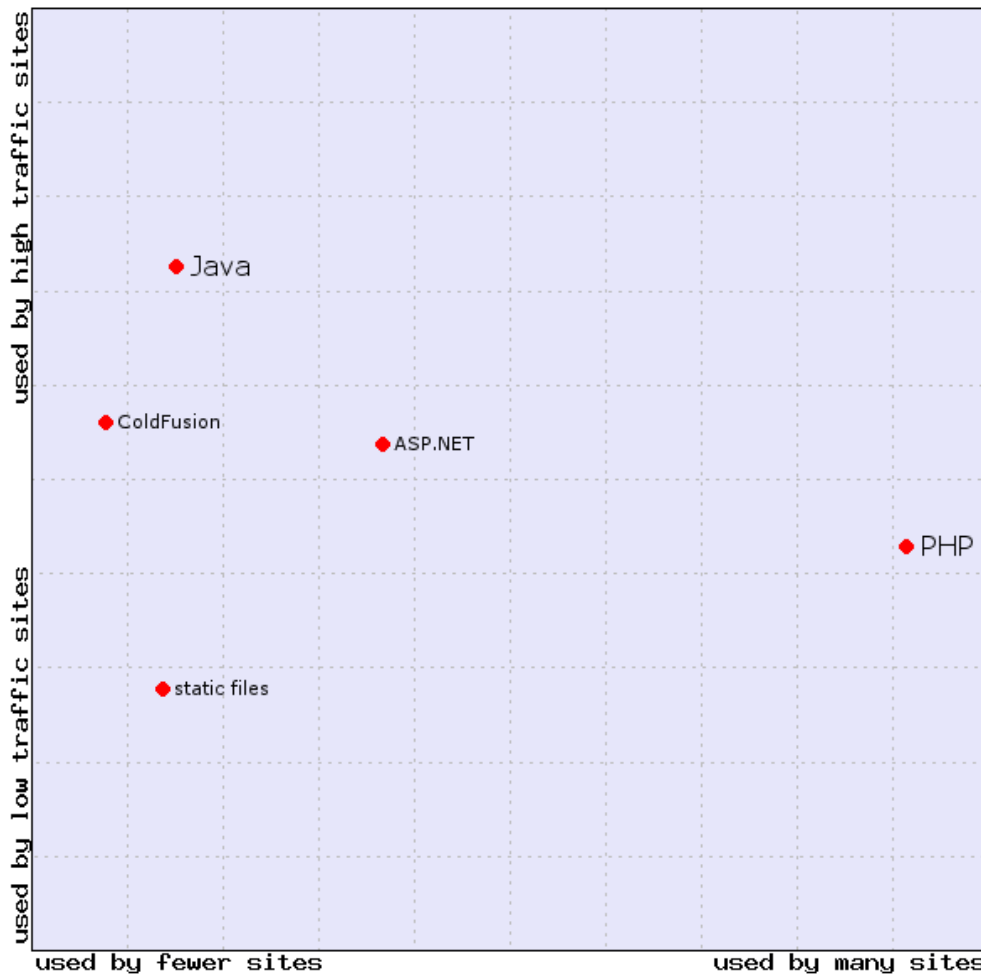
Versio	Julkaistu	Suurimmat kehitysaskleet
PHP 7	03.12.2015	kaksi kertaa parempi suorituskyky tietotyypit muuttujille 64-bittinen Windows tuki uusia featureita lisätty ja vanhoja poistettu
PHP 5	13.07.2004	yhtenäiset konstruktorit poikkeuksien hallinta, rajapintojen käyttö, kolme eri näkyvyyttä metodeille: Public, Private ja Protected
PHP 4	22.05.2000	tuki loppui 31.12.2007

3.3 PHP vs Java

PHP:n helppouden takia se on usein parempi vaihtoehto pienien web-aplikaatioiden tekemiseen. PHP oli alun perin skriptikieli ja Java ohjelmointikieli, mutta nykypäivänä niillä ei ole suurta eroa. Javalla yksinkertaisen toiminnallisuuden toteuttaminen vaatii paljon enemmän koodin kirjoittamista, kuin PHP:lla. Jos esimerkiksi haluat tuoda tekstiä palvelinpuolelta selaimelle, vaatii tämä Javalla kahden tiedoston muokkaamista. PHP:lla voit laittaa koodin suoraan HTML:n sekaan.

Isoissa ja monimutkaisemmissa projekteissa Javan vahvuudet tulevat paremmin esiin. Javaa pidetään luotettavampana ohjelmointikielenä sen vakauden ja skaalautuvuuden takia.

Market position, selected server-side programming languages, 8 May 2018, W3Techs.com



Kuva 2. Ohjelmointikielten suosio korkean ja matalan tietoliikenteen sivustoissa (W3techs.com, 2018b)

Kaavio havainnollistaa, kuinka PHP:ta käytetään useimmissa pienissä sivustoissa. Java sen sijaan on suosittu suuren tietoliikenteen omaavissa sivustoissa.

Työllistymisen kannalta Java on parempi kieli kuin PHP. TE-palveluiden haku antaa 160 työpaikkailmoitusta Javalle ja 45 PHP:lle (Te-palvelut.fi 2018). Myös Java-kehittäjien palkka on huomattavasti korkeampi (Challenge Rocket 2017).

4 Projektin kulku

Alussa määriteltiin vaatimukset ohjelmalle ja päätettiin ohjelmointikieliksi Java, HTML, CSS ja JavaScript. Kaikki ohjelmat alkavat aina ideasta, jonka jälkeen sille tehdään suunnitelma ohjelman työkaluista, ohjelmointikielistä, ulkoasusta, tarvittavista välineistä, työmenetelmistä ja aikataulusta. Opinnäytetyön kokonaisuuden suunnittelu tehdään käynnistyskokouksessa ja tässä luvussa syvennyttään työn eri vaiheisiin.

4.1 Idea opinnäytetyölle

Saksalaisessa korkeakoulussa, Hochschule Reutlingenissa, on käytössä konsepti, joka yhdistää vaihto-opiskelijat paikallisten opiskelijoiden kanssa. Konseptin nimi on ”Tandem Programme” ja sen tarkoitus on saattaa kaksi henkilöä yhteen, jotta he voivat tehdä jokapäiväisiä asioita yhdessä ja samalla opettaa toisilleen äidinkieltänsä (Hochschule Reutlingen 2018).

Ohjelma toimii koulun intran kautta, missä opiskelijat täyttävät lomakkeen ja lähettävät sen eteenpäin (liite 11). Tämän jälkeen kansainväliset koordinaattorit etsivät hakemuksille vastaparia manuaalisesti.

Opinnäytetyön tavoitteeksi muodostui luoda samanlainen ohjelma, mutta automatisoituna. Haaga-Heliolla ei ollut olemassa mitään vastatyypistä palvelua, joten sellainen päätettiin kehittää alusta alkaen. Konseptin nimeksi päätettiin CombiNation.

4.2 Vaatimukset ja niiden saavutukset

Toimeksiantaja esitti projektille jonkin verran vaatimuksia. Ohjelman automatisoinnin lisäksi toimeksiantaja vaati, lisäkustannuksilta välttyäkseen, ohjelmalle ylläpitäjää Haaga-Helian henkilöstöstä. Tähän löydettiin vastuuhenkilö Haaga-Helian tutoreista. Sopimuksen mukaan kaikki oikeudet ohjelmasta siirtyvät toimeksiantajalle. Toiveena oli myös mainostaa ohjelmaa Porvoon kampukselle, koska monikielisyys on siellä erittäin kovassa huudossa. Valmistettiin PowerPoint-esitys, jota näytetään kaikissa Haaga-Helian kampuksissa.

Käytiin läpi myös vaatimukset projektin suhteen. Ohjelman kehittämiseen tarvittiin admin-tunnukset MyNettin, johon ohjelmaan johtava linkki ja esittelyteksti kirjoitettiin. Selvitettiin vastuuhenkilö, joka ylläpitää MyNettiä ja pyydettiin käyttöoikeudet koulun opiskelijatunnuksille. Tarvittiin palvelin, joka isännöi tietokantaa ja ohjelmaa. Pyydettiin

Proto-palvelinta Haaga-Helian helpdeskistä, mutta päädyttiin ostamaan oma domain hostingpalvelu.fi:ltä.

Tarvittiin lisää PHP:n tuntemusta. Käytiin Haaga-Helian PHP-kurssi ja käytettiin hyväksi tunnilla opittuja metodeja. Haluttiin, että ohjelma toimii itsenäisesti. Tätä varten opeteltiin SQL-kieltä, jolla luotiin tietokantapäivityslauseita, mikä mahdollisti ohjelman automatisoinnin. Tarvitaan käyttäjiä ohjelmalle, jotta sen ylläpitäminen olisi kannattavaa. Tehtiin PowerPoint-esityksiä opettajille, jossa kerrotaan CombiNationin konseptista, jonka jälkeen kyseisten opettajien olisi tarkoitus tehdä sama aloittaville opiskelijoille.

4.3 Rautalangat

UI ja UX suunniteltiin käyttämällä yksinkertaisinta lähestymistapaa eli konsepti piirrettiin paperille. Rautalangat piirrettiin ennen minkäänlaista koodaamista ja niitä mietittiin viimeisen päälle, jonka mukaan ohjelma rakennettiin. Ne helpottavat ohjelman rakentamista ja säästävät aikaa, koska paperille uudelleenpiirtäminen vie huomattavasti vähemmän aikaa kuin koodin uudelleenkirjoittaminen (liitteet 12, 13 ja 14).

4.4 Ulkoasu

Ulkoasu tehtiin rautalankamallien mukaan HTML ja CSS:llä käyttäen Bootstrapia. Se rakennettiin pienistä palasista, jotka loivat yhden kokonaisuuden. Ei kuitenkaan osattu vielä projektia aloittaessa alustaa pohjaa ulkoasulle, joten kyseisiä palasia on vaivalloista siirtää jälkikäteen. Ongelma ilmeni vasta projektin loppupuolella, kun ulkoasua muutettiin.

Ulkoasua tehdessä otettiin kuitenkin huomioon toimeksiantaja ja kohdeyleisö. Koska tuote otetaan käyttöön aluksi vain Haaga-Heliassa, käytettiin koulun värejä ja tyylejä MyNetin CombiNation sivulla sekä ohjelmassa itsessään. Molemmat edellä mainituista kirjoitettiin myös englanniksi, koska käyttäjien oletetaan pääsääntöisesti kommunikoivan keskenään englanniksi.

4.5 Ohjelmointikieli

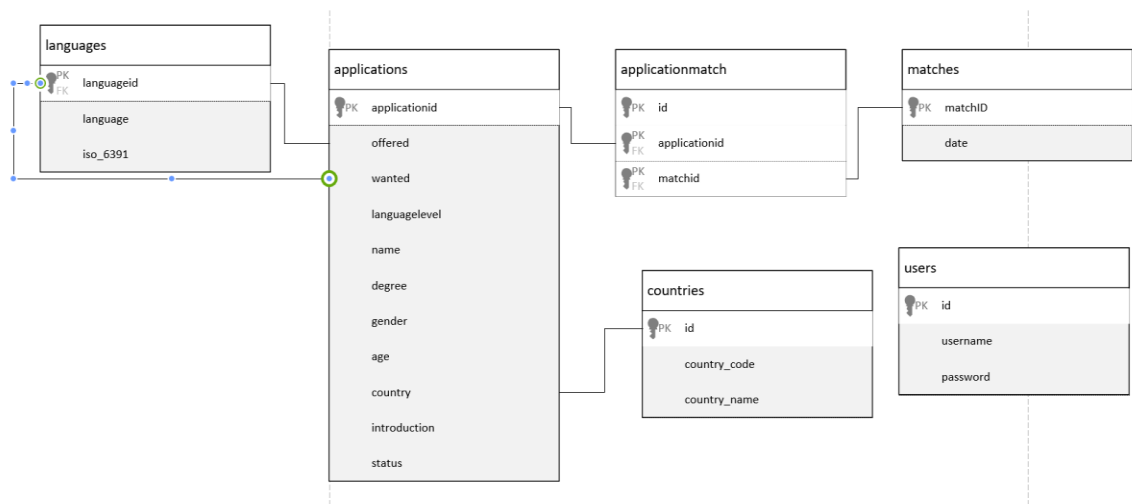
Aloituskokouksessa päätettiin rakentaa ohjelmaa Javalla, koska kaikki projektit oli myös ennen tehty Javalla. Noin kuukauden jälkeen ohjelman rakentamisesta sovittiin tapaaminen it-järjestelmäpäällikön, Olavi Kurttion kanssa, jossa analysoitiin CombiNationin yhteensopivuutta koulun tietokannan ja muiden ohjelmien kanssa. Jatkoa, toiminnallisuutta ja informaation saatavuutta ajatellen vaihdettiin Olavin kehotuksesta ohjelmointikieli Javasta PHP:hen.

Osallistuttiin Haaga-Helian PHP-kurssille samoihin aikoihin opinnäytetyön kanssa ja huomattiin, että kurssilla käytiin läpi projektille kriittisiä toimintoja. Näitä uusia tekniikoita ja ohjelmointikieltä hyväksikäytettiin suoraan opinnäytetyöprojektissa.

4.6 Tietokanta

Olavi Kurttion kanssa pidetyssä kokouksessa selvisi, että tietokanta saadaan toimeksiantajalta. Tietokanta oli MariaDB, joka toimii Haaga-Helian palvelimella ja käyttää SQL-kieltä. Lähetettiin sähköposti Haaga-Helian helpdeskiin ja pyydettiin omaa protopalvelinta Olavin suostumuksella varustettuna. Vastauksena saatiin projektille oma proto rajallisilla käyttöoikeuksilla (liite 15). Teknisten ongelmien takia päädyttiin kuitenkin käyttämään ulkoista palvelinta, johon loimme oman tietokannan.

Projektiin tarvittiin kuitenkin myös lokaali tietokanta, johon voitiin nopeasti tehdä muutoksia ja testata yhteensopivuutta koodin kanssa. Ohjeet tähän tietokantaan, phpMyAdminiin, saatiin PHP-kurssilta. phpMyAdmin toimii XAMPP:in yhteydessä lokaalisti ja sillä luotiin CombiNationin ensimmäiset tietokannat alla olevan relaatiokaavion mukaan.



Kuva 3. Ohjelman tietokannan rakenne relaatiokaaviona

4.7 Säilytyspalvelu

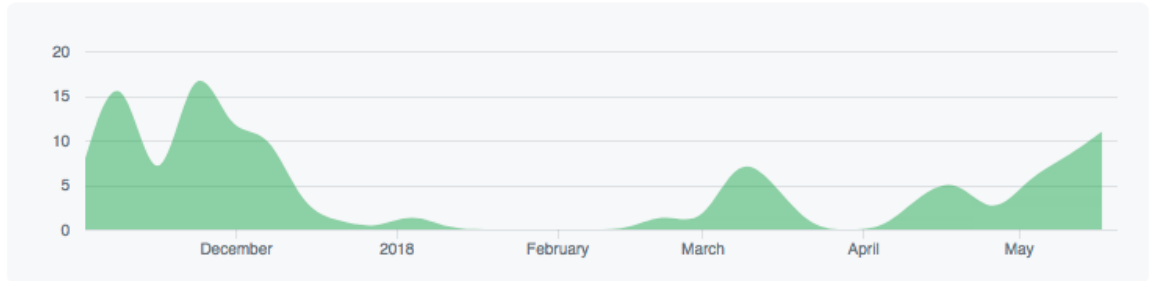
Oli erittäin tärkeää, että yhtä projektia pystyi tekemään useampi henkilö, ja siksi tarvittiin pilvipalvelussa toimiva säilytyspaikka. Haaga-Helian tarjoaa opiskelijoilleen ykistyisen käyttäjän GitHub-palveluun, eli projekteja voi tehdä yksityisesti muiden näkemättä koodia. Tämän takia päädyimme GitHubiin.

Alla näkyvät CombiNationin yhteis-committit, eli töiden viennin määrä GitHubiin, diagrammina.

Nov 5, 2017 – May 23, 2018

Contributions: Commits ▾

Contributions to master, excluding merge commits



Kuva 4. GitHubiin tehdyt committit (Kuvakaappaus: github.com)

4.8 Toteutus

Projektia tehtiin säännöllisesti ja edettiin suunnitelmien mukaan MVP tyylillä. Dokumentaatio tehtiin aina parikoodaussessioiden yhteydessä opinnäytetyöpäiväkirjaan. Tähän päiväkirjaan piirrettiin myös ensimmäiset rautalangat ohjelman ulkoasusta ja käytettävyydestä, sekä vaatimusmäärittelyt.

Koodia kirjoitettiin Eclipselle ja Bracketsillä, joissa käytettiin HTML, CSS, JavaScripti, PHP ja SQL –kieliä. Käytetään XAMPP:ia lokaalina palvelimena ja luodaan sinne tietokannat phpMyAdminilla. Kun ympäristöt oli pystytetty ja kielet valittu pystyttiin ohjelma kehittää valmiiksi.

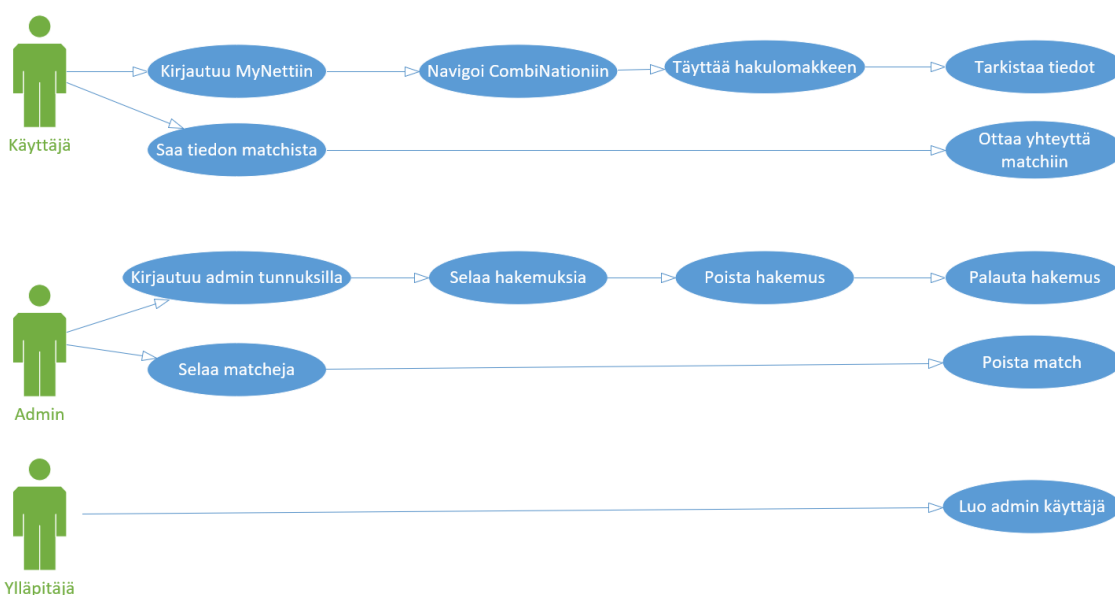
Kun tavoitteet ohjelmalle saavutettiin, vietiin se suunnitelmasta poiketen hostinpalvelu.fi – palveluun Haaga-Helian oman proto-palvelimen sijaan. Koodi vietiin hostinpalgelu.fi – palvelimelle, jossa sille luotiin samanlainen tietokanta. Ohjelman domainiksi valittiin www.combination.fi.

5 Käyttötapaus ja koodaaminen

Käyttötapaukset kerrotaan kuvien avulla. Ohjelman arkkitehtuurista on kuva kaikkinen kansioineen. Käydään myös läpi sekvenssidiagrammia ja samalla avataan ohjelman toimintaa ohjelman omasta näkökulmasta ja kerrotaan, mitä koodi tekee ohjelmassa.

5.1 Käyttötapauskaavio

Seuraavassa kaaviossa kuvataan käyttäjän askeleet alusta loppuun. Kuvan jälkeen avataan jokaiset kohdat selityksillä ja sivukohtaisilla liitteillä varustettuna.



Kuva 5. Käyttötapauskaavio

Käyttäjä: käyttötapaus 1

Käyttäjä kirjautuu MyNettiin ja navigoi International välilehden alle CombiNationin esittely sivulle (liite 1).

Käyttäjä klikkaa eri sivulla olevaa "APPLY HERE" –linkkiä, joka vie hänet uudelle sivulle, johon aukeaa CombiNation-lomake (liite 2).

Käyttäjä täyttää lomakkeen ja painaa "Let's go!" –nappulaa, jonka jälkeen hänelle aukeaa uusi sivu käyttäjän täyttämällä tiedoilla ja jatko-ohjeistuksilla (liite 3).

Käyttäjä saa sähköpostiinsa rekisteröintivahvistuksen (liite 4).

Käyttäjä sulkee ikkunan.

Käyttäjä: käyttötapaus 2

Käyttäjää saa sähköpostiin viestin CombiNation partnerin löytymisestä, joka sisältää uuden ystävän esittelyviestin ja hänen täyttämänsä tiedot (liite 5).

Admin: käyttötapaus

Admin kirjautuu ohjelmaan (liite 6).

Admin selaa vastaanotettuja hakemuksia, joista hän voi selata ja poistaa väärinlähetetyt hakemukset (liite 7).

Nämä hakemukset siirtyvät "Deleted" –sivulle, josta admin voi halutessaan vielä palauttaa poistetun hakemuksen taikka poistaa sen täysin tietokannasta (liite 8).

Kun kaksi ohjelmaa yhdistää kaksi käyttäjää keskenään, siirtää se heidän hakemuksensa "Matched" –sivulle, josta admin voi myös poistaa hakemuksia (liite 9).

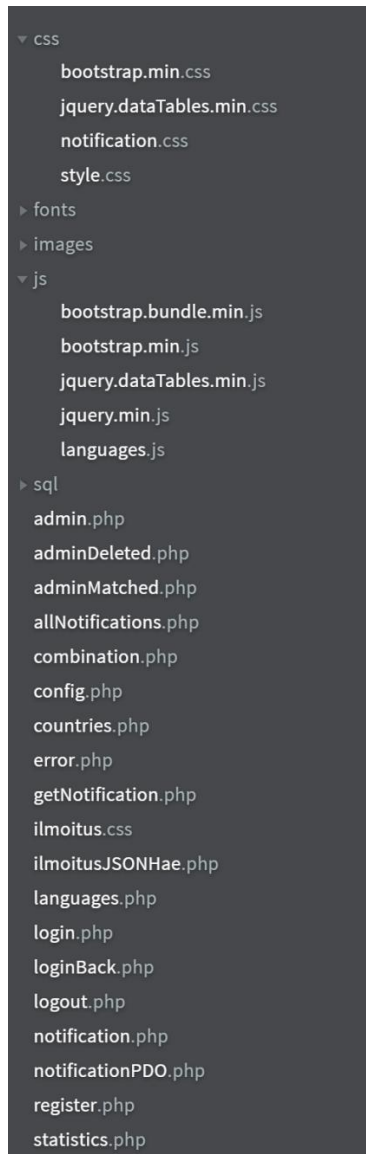
Admin kirjautuu ulos "Logout"-napista.

Ylläpitäjä: käyttötapaus

Ylläpitäjä lisää pyydettäessä admintunnuksia Postmanilla (liite 10).

5.2 Tiedosto arkkitehtuuri

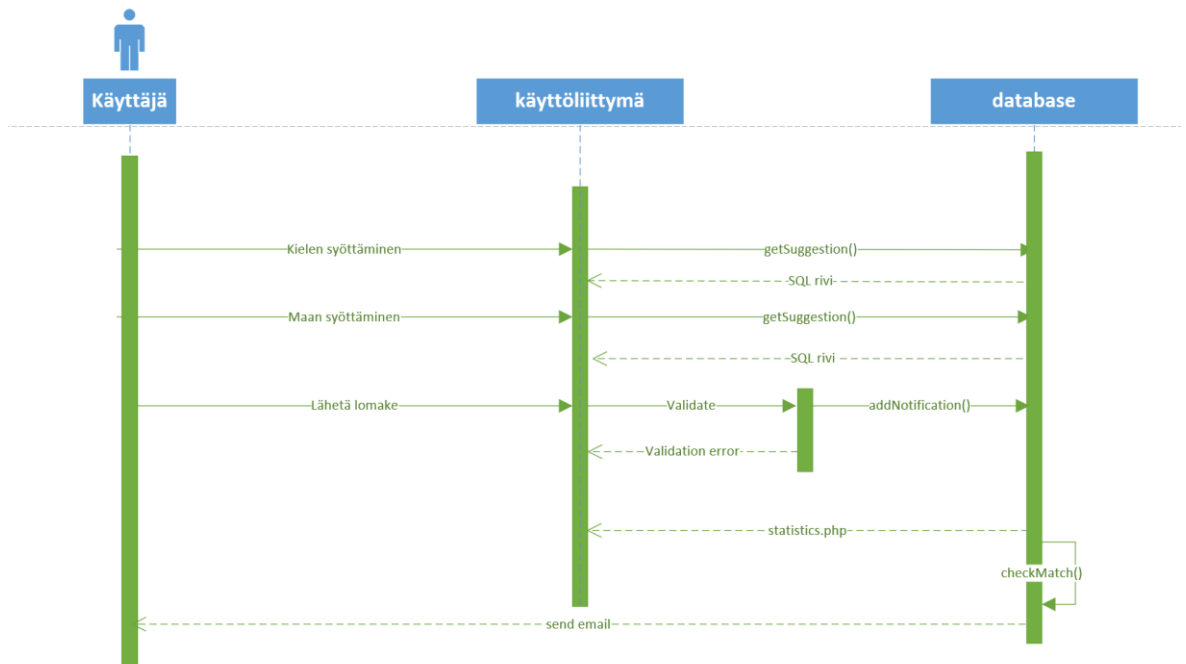
Sijoitimme kaikki .php -tiedostot projektiokansion juureen. Tiedostojen lisääntyessä ne olisi kannattanut jakaa eri kansioihin rakenteen selkeyttämisen takia. Kuvat, javascriptit ja tyylitiedostot ovat sijoiteltu omiin kansioihin



Kuva 6. Opinnäytetyöprojektin tiedostorakenne (Kuvakaappaus Brackets -ohjelmasta)

5.3 Sekvenssikaavio

Lomakkeen täyttämisestä tehtiin sekvenssikaavio, joka kertoo ohjelman ja palvelimen kommunikoinnista lomaketta täytettäessä ja lähettäessä.



Kuva 7. Sekvenssikaavio lomakkeen täyttämisestä

Ohjelma kommunikoi palvelimen kanssa jo lomaketta täyttäessä. Ohjelma kutsuu jQuery:llä `getSuggestions` -funktioita aina kun kenttään syötetään tekstiä. Funktio suoritetaan Ajaxin avulla sivua uudelleen lataamatta. Funktio palauttaa tietokantahaulla löytyneet kielet ja ehdottaa niitä käyttäjälle.

```

$("#offeredLanguage").keyup(function () {
    language = this.value;
    if (language.length > 1) {
        getSuggestions(language, "#optionsOffered", "languages.php");
    }
});

function getSuggestions(term, id, url) {
    $.ajax({
        type: "POST",
        dataType: "json",
        url: url,
        data: {"term": term},
        success: function (data) {
            $(id).empty();
            data.records.forEach(function (value) {
                $(id).append("<option value='" + value + "'>");
            });
        }
    });
}
  
```

Kun lomake lähetetään, se validoidaan palvelinpuolella. Jokaiselle vaaditulle kentälle on oma validointifunktio. Virheen löytyessä ohjelma palauttaa virheviestin lomakkeen kohtaan, jossa virhe tapahtui. Jokaiselle kentälle on tehty oma virheilmoituksensa.

```

public function checkName($required = true, $min = 2, $max = 50)
{
    // Jos saa olla tyhjä ja on tyhjä
    if ($required == false && strlen($this->name) == 0) {
        return 0;
    }
    // Jos ei saa olla tyhjä ja on tyhjä
    if ($required == true && strlen($this->name) == 0) {
        return 11;
    }
    // Jos nimen muoto ei ole oikea
    if (preg_match("/^[^a-zääöüA-ZÄÄÖÜ\ - ]/", $this->name)) {
        return 12;
    }
    // Jos name on liian lyhyt
    if (strlen($this->name) < $min) {
        return 13;
    }
    // Jos name on liian pitkä
    if (strlen($this->name) > $max) {
        return 14;
    }
    // Kentässä ei ole virhettä
    return 0;
}

-1 => "Unknown error",
0 => "",
11 => "Name cannot be empty",
12 => "Name must include letters only",
13 => "Name doesn't have enough characters",
14 => "Name has too many characters",
21 => "Email cannot be empty",
22 => "Email is not in the right form",
23 => "Email is too short",
30 => "Offered language cannot be empty",
31 => "Wanted language cannot be empty",
32 => "Must be a valid language",
33 => "Offered and wanted language cannot be the same",
41 => "Language level has to be one from the list",
42 => "Write an introduction of yourself",
43 => "Degree has to be one from the list",
72 => "Invalid gender",
73 => "Invalid age",
98 => "Age cannot be negative",
99 => "Program isn't meant for turtles",
100 => "You must agree on terms of service"

```

Jos validointi menee läpi, tiedot lähetetään tietokantaan ja käyttäjä ohjataan vahvistussivulle. Samalla hakemukselle etsitään vastaparia tietokannasta. Mikäli vastapari löytyy, lähetetään molemmille osapuolille vahvistusviesti sähköpostiin parin löytymisestä.

```
function addNotification($notification) {
```

```

$sql = "insert into applications (offered, wanted, languagelevel, name,
email, degree, gender, age, country, introduction, status, date)
values (:offered, :wanted, :languagelevel, :name, :email, :degree,
:gender, :age, :country, :introduction, :status, :date)";

// Preparing SQL-statement
if (! $stmt = $this->db->prepare ( $sql )) {
    $error = $this->db->errorInfo ();
    throw new PDOException ( $error [2], $error [1] );
}

// Binding parameters
$stmt->bindValue ( ":offered", utf8_decode ( $notification->getOffered
() ), PDO::PARAM_STR );
$stmt->bindValue ( ":wanted", utf8_decode ( $notification->getWanted ( )
), PDO::PARAM_STR );
...

$this->db->beginTransaction();

// Execute SQL-statement (insert)
if (! $stmt->execute ()) {
    $error = $stmt->errorInfo ();
    if ($error [0] == "HY093") {
        $error [2] = "Invalid parameter";
    }
    // Cancell event
    $this->db->rollBack();
    throw new PDOException ( $error [2], $error [1] );
}
$id = $this->db->lastInsertId ();
$this->db->commit();
return $id;
}

```

5.4 Hakemusten käsittely

Kun hakemus on lähetetty tietokantaan, etsitään sille vastapari kaikista kannassa jo olevista hakemuksista. Tämä kantakutsu tehdään laskevassa järjestyksessä, eli etsimissuunta on viimeisestä hakemuksesta ensimmäiseen. Näin annetaan jonossa pisimpään odottaneelle vastapari, mikäli vastapareja löytyy useampia.

```

SELECT * FROM applications
WHERE offered=:wanted AND wanted=:offered AND status='Searching'
ORDER BY applicationid DESC

```

Ohjelmassa on myös admin näkymä, josta admin voi käsitellä hakemuksia. Kun admin kirjautuu sisään ohjelmaan, hänet uudelleenohjataan admin.php sivulle, jossa on taulukko kaikista hakijoista, joiden status on "searching". Sivun avautuessa suoritetaan sql haku tietokannasta "applications" taululta.


```
SELECT applicationid, offered, wanted, languagelevel, name,  
email, degree, gender, age, country, introduction, status, date  
FROM applications  
WHERE status = 'searching'
```

Jos admin poistaa hakemuksen, sen statukseksi muuttuu "deleted", jolloin hakemus siirtyy poistettujen hakemusten taululle (adminDeleted.php).

```
UPDATE applications  
SET status = 'deleted'  
WHERE applicationid = :applicationid
```

Kun ohjelma löytää parin, se muuttaa molempien osapuolten statukseksi "matched".
Kaikki paritetut hakijat löytyvät matched taulusta (adminMatched.php).

```
UPDATE applications  
SET status='Matched'  
WHERE applicationid=:applicationid
```

6 Web-sovellus konseptista tuotteeksi

Tässä luvussa kerrotaan, kuinka web-sovellus tehdään alusta loppuun ja kuinka se saadaan Internetiin muiden käytettäväksi. Luvussa käsitellään ohjelmaan tarvittavat yleiset asiat, kuten ohjelmointikieli, tietokanta, ohjelmointiympäristö ja domain. Combination-ohjelma vietiin nettiin käyttäen seuraaksi kuvattuja vaiheistuksia.

6.1 Ohjelmointikielien valitseminen

Ohjelmointikieltä valittaessa täytyy ottaa huomioon ohjelman käyttötarkoitus, laajuus ja haluaako ylipäättään oppia ohjelmoimaan. Nykyään internetsivuja pystytään kehittämään kirjoittamatta riviäkään koodia esimerkiksi Wordpressin tai Wixin avulla. Tällaisissa palveluissa on mahdollista sijoitella valmiita komponentteja haluamilleen paikoille, jonka ansiosta täysin kokematonkin henkilö pystyy nopeasti toteuttamaan näyttävät internetsivut.

Ohjelmistokehittäjän urasta haaveilevien kannattaa ehdottomasti koodata sivustot itse. Sovellukseen tarvitaan HTML: n lisäksi back-end-kieli, mikäli ohjelmalla on oma tietokanta. Yleisin back-end-kieli web-ohjelmointiin on PHP. Se on helppoutensa ansiosta hyvä vaihtoehto aloittelijalle (Logic Heart Pvt Ltd 2016).

Suosittujen ohjelmointikielien etuna on se, että niihin löytyy helposti apua internetistä. Uudemmissa ohjelmointikielissä ongelmana on, ettei kysymyksiin välttämättä löydy vastausta ja ongelmiin tarvitaan omaa päättelykykyä. Aloittelevan ohjelmoijan kannattaa siis valita kieli, jota käytetään laajasti.

Kielen valinnassa kannattaa myös miettiä omaa ohjelmointiuraa. Työpaikkailmoituksia selatessa huomaa nopeasti, minkä kielten osaamista hakijoilta toivotaan. Esimerkiksi Java-kehittäjillä tulee olemaan vielä pitkään suotuisat työmarkkinat, koska se on maailman suosituin ohjelmointikieli (Tiobe 2018).

6.2 Tietokannan valitseminen

Jos ohjelma tallentaa esimerkiksi käyttäjän syöttämiä tietoja, tarvitaan sovellukselle tietokanta. Tietokannan valinnassa suurin päätös on valita SQL ja NoSQL tietokannan välillä.

SQL tietokannat ovat monipuolisia ja niillä voi tehdä komplekseja hakuja. Niissä tietokantojen rakenne pitää suunnitella ennen tietojen lisäämistä. Tämän takia tiedon

lisääminen SQL kantaan on rajoitettua. Rakenteen muuttaminen voi olla monimutkaista, jos tietoja on jo tallennettu kantaan. SQL on yleensä parempi vaihtoehto järjestelmiin, jotka vaativat monimutkaisia transaktioita. Tällaisia järjestelmiä ovat esimerkiksi pankkijärjestelmät ja verkkokaupat.

NoSQL on dynaaminen tietokanta, joka ei vaadi ennalta määrättyä rakennetta. Jokaiselle dokumentille voidaan luoda oma uniikki rakenne. NoSQL on suorituskyvyltään nopeampi kuin SQL ja sen skaalautuvuus on parempi. Suosituin SQL tietokanta on MySQL, jota käytetään myös tässä opinnäytetyöprojektissä. NoSQL kannoista MongoDB on ylivoimaisesti käytetyin tietokanta (Xplenty 2017).

6.3 Ohjelmointiympäristö

Ohjelmia voidaan koodata esim. Notepadilla, mutta ohjelmointiympäristöt tekevät siitä paljon helpompaa. Ohjelmointiympäristöt helpottavat koodin lukemista mm. värittämällä muuttujat ja funktiot eri väreillä. Ohjelmistoympäristöihin voi yleensä liittää haluamiaan plugineja, jotka esimerkiksi tunnistavat tiettyä ohjelmointikieltä, tai sisentää ja kaunistaa koodin puolestasi.

Isot kehitysympäristöt helpottavat ohjelmointia monella tavalla. Tällaisia ominaisuuksia ovat muun muassa:

- funktiokutsut ovat linkkejä, joita klikkaamalla pääsee funktion luokse
- kirjoitettavan koodin ennakoiminen koodia kirjoittaessa (funktiot, muuttujat)
- automaattinen koodin generointi
- kertoo virheestä punaisella alleviivauksella jo ennen kuin ohjelma suoritetaan
- koodin debuggaaminen
- globaali haku koko projektin sisältä, jolla voi hakea tiettyä tekstin osaa tai tiedostoa
- konsoli, johon tulee mahdollisia erroreita

Pienemmissä HTML projekteissa kevyet tekstieditorit kuten Brackets tai Notepad++ ovat hyviä valintoja, mutta suuremmissa back-end -koodia sisältävissä projekteissa kannattaa harkita suurempia ohjelmistoympäristöjä, kuten Eclipse tai IntelliJ. (Louis 2016)

Kun ohjelmointikielet ja ohjelmistoympäristö on valittu, tarvitaan vielä lokaali WWW-palvelinohjelma ja tietokannanhallintaohjelma ohjelman testaamiseen. Tähän tarkoitukseen löytyy monia erilaisia ohjelmia, joista valitsimme XAMPP ohjelmiston, koska se tukee parhaiten tarpeitamme. XAMPP sisältää Apache palvelimen, MariaDB tietokannan ja PHP tuen.

6.4 Domainin ja webhotellin hankinta

Domainilla tarkoitetaan web-sivuston osoitetta, joka kirjoitetaan selaimen osoiteriville. Domainille on saatavilla useita eri päätteitä, kuten .com, .fi, ja org. Domainin nimi tulee myös palvelimelta lähetettäviin sähköpostien osoitteisiin. (Wikipedia 2017)

Taulukko 3. Esimerkkejä erilaisista domaineista

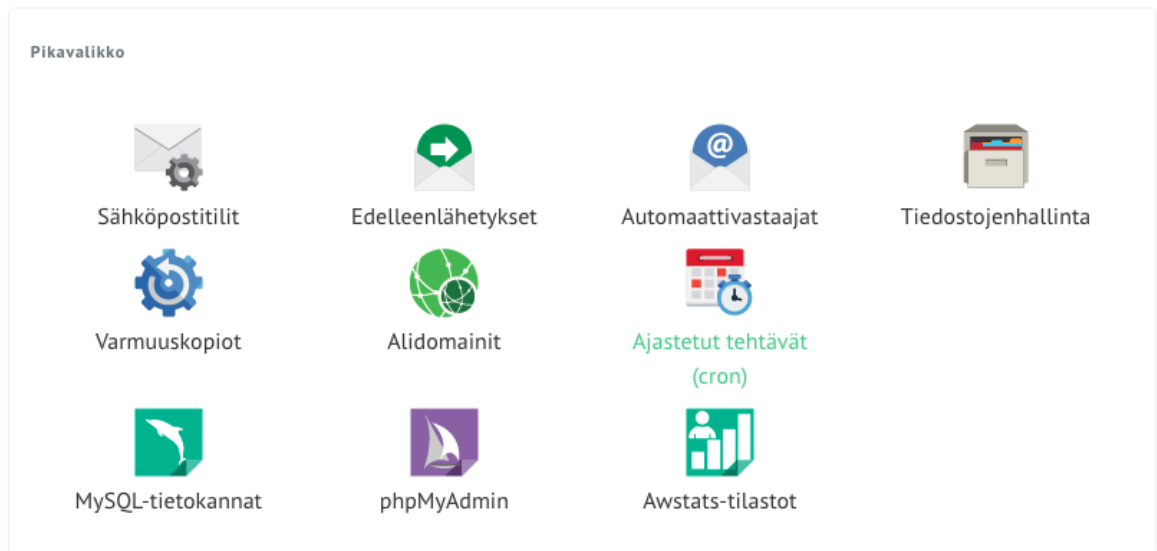
Pääte	
.fi (finland)	eduskunta.fi
.com (commercial)	facebook.com
.org (organization)	wikipedia.org
.de (deutschland)	oktoberfest.de

Palvelimen pystyttäminen omalle kotikoneelle on haasteellista, jonka takia on olemassa Webhotelleja. Ne rekisteröivät sinulle haluamasi domainin ja ylläpitävät palvelinta puolestasi. Ohjelmallemme valittiin palveluntarjoajaksi hostingpalvelu.fi. Domainin nimeksi valittiin combination.fi, jonka vuosimaksu on 15 €. Palvelimen pitäminen kolme kuukautta maksaa noin 16 €. (Hostingpalvelu 2018)

6.5 Tiedostojen vienti palvelimelle

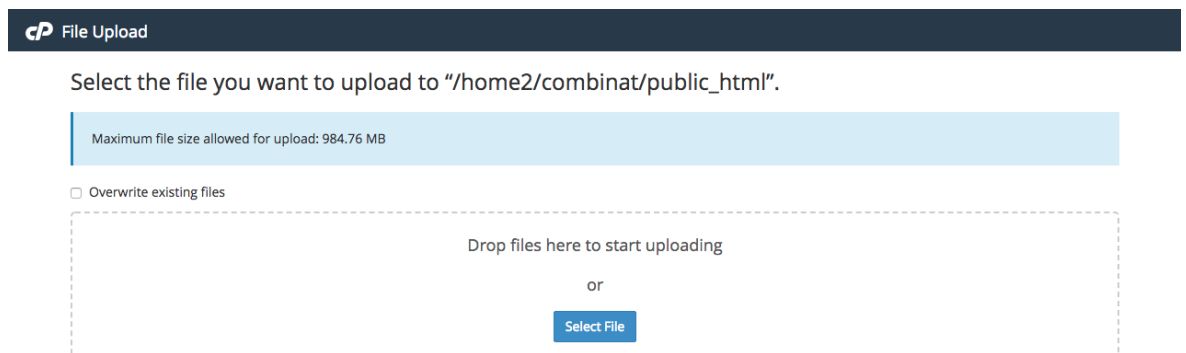
Ohjelma tehdään lokaalisti omassa työympäristössä, jonka jälkeen se viedään pilvipalvelimelle tai sille pystytetään oma palvelin. Pilvipalvelut ovat kuitenkin yleisempi vaihtoehto varisinkin pienemmille yrityksille ja ohjelmille, koska niistä tarvitsee usein maksaa vain kuukausimaksu, ja domainin, palvelimen ylläpitäjä hoitaa kaiken lopun.

Kun palvelimen ylläpitäjältä on ostettu palvelu käyttöön, oman projektin voi viedä palvelimelle tiedosto kerrallaan tiedostojenhallinnan avulla. Tämä ominaisuus on tarkoitettu yksittäisten tiedostojen lisäämiseen, jo valmiiksi tehtyyn projektiin.



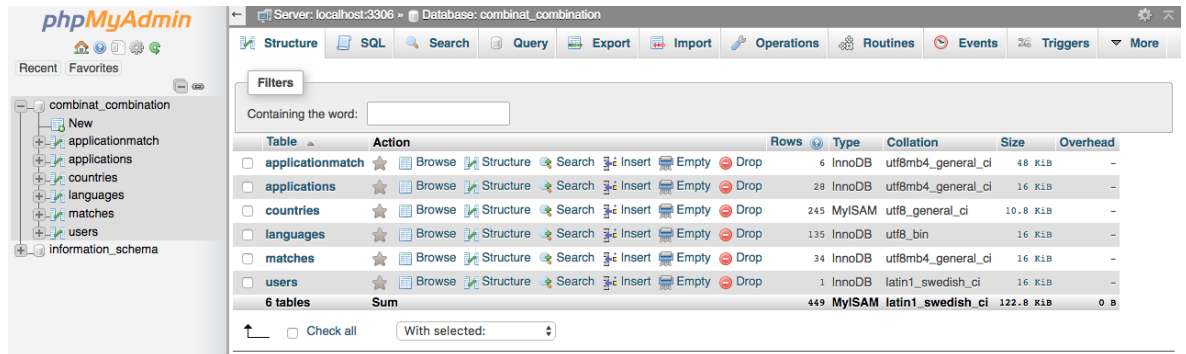
Kuva 8. hostingpalvelut.fi pikavalikko (Kuvakaappaus: hostingpalvelu.fi)

Paketoimalla koko projektin ja lähettämällä sen tiedostojenhallinnan kautta palvelimelle, pystytään hetkessä tuomaan kokonaiset projektit palvelimelle.



Kuva 9. Tiedostojen lataus –toiminto (Kuvakaappaus: hostingpalvelu.fi)

Palvelimella voi myös rakentaa tietokannan ohjelmalle. Hostinpalvelu.fi tarjoaa phpMyAdmin –ohjelman, jolla tehtiin tietokanta muun muassa CombiNationille. Palvelimeen viedyn koodit paritetaan rakennetun tietokannan kanssa, jonka jälkeen ohjelma toimii valitussa internetosoitteessa samalla lailla kuin lokaalissa ympäristössä.



Kuva 10. phpMyAdmin –ohjelmalla tehty tietokanta (Kuvakaappaus: phpMyAdmin web-sovellus)

7 Pohdinta

Tässä luvussa reflektoidaan projektin kulkua. Lisäksi luvussa kuvataan käyttäjätestaamista ja ohjelmasta saatua palautta ja tarjotaan jatkokehityssuunnitelmia tuleville kehittäjille.

7.1 Esteet ja ratkaisut

Tekniset ongelmat ratkottiin Googlen avulla. Parhaimmiksi sivuiksi todettiin W3school, ja Stack Overflow. W3school tarjoaa esimerkeillä varustettua apua HTML, CSS, JavaScriptiin ja jQueryyn. Näitä esimerkkejä oli helppo soveltaa omaan käyttöön, ja ne jäivät nopeasti mieleen. Stack Overflow oli taas haastavampi, mutta antoi tarkemman kuvauksen, esimerkin ja vastauksen kuin w3school. Stack Overflowssa voi olla useita erilaisia vastuksia usealla eri tavalla ilmaistuna.

Myös php.net oli kovassa käytössä, koska se käsittelee vain ja ainoastaan PHP:tä. Sieltä löytyy PHP-kielen perusasiat esimerkkeineen ja niitä on helppo testata. Php.net antaa vastukset yleisimpiin PHP:ssä käytettyihin metodeihin.

Yhtäkkinen työn alkaminen hidasti projektin edistystä, kunnes se kokonaan pysähtyi kolmeksi kuukaudeksi. Huhtikuussa projekti elpyi ja kehitettiin ohjelma työssä opitulla tiedolla ja taidolla tuplatempolla valmiiksi. Työn mukana tuli suuri määrä osaavia työkavereita, joita konsultoitiiin koodista, ulkoasusta ja käytettävyydestä. Näiden konsultaatioiden tuloksena projektiin saatiin parempaa toimivuutta, käyttäjäystävällisempää ulkoasua ja käytettävyyttä.

7.2 Käyttäjätestaus

Valittiin kourallinen testikäyttäjiä, jolle kerrottiin, mistä CombiNationiin pääsee kirjautumaan. Tämän jälkeen annettiin testaajien itse selvittää ohjelman alusta loppuun saakka. Testitapauksen jälkeen testaajat täyttivät palautelomakkeen, johon on listattu ohjelman tärkeimpiä ominaisuuksia ja joista pyydettiin mielipiteitä asteikolla 1 – 5 (liitteet 16 - 18).

7.3 Jatkokehitysehdotukset

CombiNationin toivotaan menestyvän pitkään Haaga-Heliassa, joka on tällä hetkellä ainoa koulu, jossa konsepti otetaan käyttöön. Mutta PEPPI-intran yhdistäessä kolme PK-seudun

ammattikorkeakoulua, sillä on varaa ja potentiaalia kasvaa ja tarjota palveluaan myös näille kouluille. Suositellaan CombiNationia opinnäytetyöaiheeksi tuleville sukupolville.

UI ja UX on syytä muuttaa responsiiviseksi "mobile first" periaatteella. Myös ohjelmien ulkoasut muodin lailla muuttuvat ajan mittaan, eli tarpeen vaatiessa UI:ta voi muuttaa tekohetken preferenssien tyyliseksi. Myös harkittuja animaatioita voi ohjelmaan laittaa.

Implementaatio Haaga-Helian (tai PEPIN) verkostoon, jotta saataisiin uhka koulujen ulkopuolelta tulevilta käyttäjiltä minimoitua. Tällä hetkellä ohjelmaa ylläpitää hostingpalvelu.fi, jonka toimii omakustanteisesti. Implementaation voi suorittaa yhdessä Haaga-Helian helpdeskin kanssa.

Adminille voi lisätä ominaisuuksia, kuten käyttäjän lisääminen, hakemusten lukeminen ja sähköpostista painettaessa avaa suoraan mahdollisuuden kirjoittaa käyttäjälle viestin. Adminin/CombiNationin tukipalvelun sähköposti voisi olla ohjelmassa siten, että admin pääsee CombiNationin sivuilta niitä suoraan lukemaan. Matched-tauluun voisi kehittää ominaisuuden, jolla admin pystyy muokkaamaan ja vaihtamaan pareja keskenään, mikäli pareista ilmaantuu ongelmia/toiveita.

Toiminnallisuuksia CombiNationiin voi kehittää niin paljon, kun mielikuvitusta riittää. Oma yhteydenpitokanava ohjelmalle olisi ideaali. Tämä tarkoittaisi CombiNationin sisäistä "Facebookia", jossa käyttäjät kirjautumisen jälkeen voivat keskustella pariensa kanssa taikka yleisellä seinällä lisätä näkyvyyttä parin etsimiseen. Ohjelma voisi jatkossa käyttää MyNetin tunnustautumista, eli vain käyttäjät, jotka ovat kirjautuneena MyNettiin voivat täyttää hakemuksen.

7.4 Päätelmät

Projektia aloittaessa ei osattu kirjoittaa riviäkään PHP:ta, mutta nyt sillä osattaisiin ja osattiinkin tehdä toimiva ohjelma. Ohjelmaa kehittäessä opittiin paljon esteistä ja niiden ylittämisestä, erilaisista metodeista ja niiden soveltamisesta sekä tärkeimpiin taitoihin kuuluvaa palvelemille viemistä. Kokemus ohjelman ja projektin kehityksestä on hyvä, ja lisäksi omien taitojen kehitys on ollut valtava.

Tavoitteenamme oli saada toimiva MVP-mallin ohjelma. Suurimmaksi esteeksi aiheutui työpaikka ja siihen keskitetty aika, mutta ohjelman toiminnallisuus ei tauosta huolimatta kärsinyt vaan vastaa täysin odotuksia. Jälkeenpäin katsottuna olisi ollut parempi yrittää tasapainotella työn ja koulun kanssa, jolloin olisi ollut enemmän aikaa keskittyä myös

konseptin vähemmän kriittisiin ominaisuuksiin, kuten esitelmien tekemiseen tai testikäyttäjien etsimiseen.

Suunnitelmaan kuului tehdä opinnäytetyöpäiväkirjaa samalla kun koodia kehitettiin. Tätä suunnitelmaa noudatettiin kaksi ensimmäistä kuukautta, ennen tauon alkamista. Kun projektia jatkettiin, tehtiin vain koodia, koska se priorisoitiin päiväkirjan edelle. Alusta saakka sovittiin myös, että ensin tehdään ohjelma ja sitten vasta dokumentointi. Jälkeen päin tiedetään paremmin, että nämä kaksi asiaa kulkevat käsi kädessä ja ne on parempi tehdä myös samaan aikaan.

Ohjelmaan oli jo tehty UI ja UX, kun saatiin konsultaatiota ammattilaiselta, jonka jälkeen UI ja UX vaihtui radikaalisti. Projektin alkaessa ei myöskään osattu koodata yhtä hyvin kuin nyt, joten ohjelman julkaisun jälkeen joudutaan tekemään korjausliikkeitä koodissa, mihin kuluu paljon aikaa. Menetelmä koodata oli ”Tee ensin itse, tutki sen jälkeen”. On vaikea verrata erilaisia toimintatapoja, sillä ne toimivat eri lailla eri henkilöille, mutta tämän ohjelman tapauksessa koodin esitutkinta olisi voinut toimia paremmin. Onneksi minkään ohjelman kehitys ei lopu julkaisuun, vaan sitä kehitetään niin kauan, kuin sille löytyy käyttäjiä. Työelämässä opittiin, että aina kannattaa tehdä ohjelmat ”mobile first” -menetelmällä, jolloin ohjelma saadaan responsiiviseksi. Projektissa ei kuitenkaan käytetty kyseistä menetelmää, joten sen muuttaminen responsiiviseksi on ylimääräistä työtä.

Osasta suunnitelmaa pidettiin kuitenkin kovasti kiinni ja taidot riittivät tuottamaan rautalankojen mukaista koodia, josta oli helppo lähteä muokkaamaan ulkoasua ammattilaisen suosimaksi. Opittiin myös, kuinka ostaa oma domain ja viedä sen alle ohjelma siten, että se näkyy muulle maailmalle. Pintaraapaistiin myös monia ohjelmia, joita käytetään paljon työelämässä, joten työ pelkästään ei auttanut valmistamaan opinnäytetyötä vaan opinnäytetyö auttoi valmistumaan työelämään.

Alussa oli idea, joka haluttiin jakaa maailmalle. Haaste otettiin vastaan, vaikkei projektin toteutuksesta ollut juurikaan lainkaan kokemusta. On kuljettu pitkä matka konseptin suunnittelusta toteutukseen ja se oli enemmän opettavaista kuin osattiin kuvitella. Intohimo työhön ja sen toteutukseen oli aito, minkä takia päästiin tavoitteisiin. Henkilökohtaisen kasvun kannalta CombiNation tarjosi hyvän hiekkalaatikon, jossa uskallettiin oppia, epäonnistua, onnistua ja ennen kaikkea kasvaa ohjelmistokehittäjinä ja ihmisinä.

Lähteet

Apache 2018. HTTP SERVER PROJECT. Luettavissa: <http://httpd.apache.org/> Luettu: 13.5.2018.

Belitsoft.com 2017. PHP vs Java: Terminology differences. Luettavissa: <https://belitsoft.com/php-development-services/php-vs-java> Luettu: 15.5.2018.

Challenge Rocket 2017. Top 10 of Programming Languages With The Highest Salaries In 2017. Luettavissa: <https://medium.com/@ChallengeRocket/top-10-of-programming-languages-with-the-highest-salaries-in-2017-4390f468256e> Luettu: 15.5.2018

Goyal, A. 2017. Top 10 Successful Websites Developed in PHP. Luettavissa: <https://www.octalsoftware.com/blog/top-10-successful-websites-developed-php> Luettu 15.5.2018

Hostingpalvelu 2018. Luettavissa <https://www.hostingpalvelu.fi/> Luettu: 15.5.2018

Logic Heart Pvt Ltd 2016. Top 6 Reasons Why PHP Better for Web Development. Luettavissa: <https://www.linkedin.com/pulse/top-6-reasons-why-php-better-web-development-logic-heart-pvt-ltd/> Luettu 15.5.2018

Louis, S. 2016. 5 Reasons to use an IDE Instead of an Editor. Luettavissa: <https://www.activestate.com/blog/2016/06/5-reasons-use-ide-instead-editor> Luettu: 16.5.2018

Php.net, 2018. Php releases. Luettavissa <http://php.net/releases/> Luettu: 14.5.2018

Polcode Team, 2016. Is PHP the most hated programming language? Luettavissa: <https://polcode.com/en/is-php-the-most-hated-programming-language/> Luettu 14.5.2018.

Hochschule Reutlingen. International communication. Luettavissa: <https://www.reutlingen-university.de/en/international/intercultural-communication/tandem-programme/> Luettu 23.5.2018

Tiobe.com 2018. Tiobe index for PHP. Luettavissa: <https://www.tiobe.com/tiobe-index/php/> Luettu 15.5.2018

W3Techs.com 2018a. Usage statistics and market share of PHP for websites. Luettavissa: <https://w3techs.com/technologies/details/pl-php/all/all> Luettu: 15.5.2018

W3techs.com 2018b. Comparison of the usage of PHP vs. JAVA for websites. Luettavissa: <https://w3techs.com/technologies/comparison/pl-java.pl-php> Luettu: 14.5.2018

Wikipedia, 2017. Verkkotunnus. Luettavissa: <https://fi.wikipedia.org/wiki/Verkkotunnus> Luettu: 15.5.2018

WP Engine. Why upgrade to PHP 7? Luettavissa: <https://wpengine.com/resources/upgrading-to-php-7/> Luettu 14.5.2018.

Xplenty, 2017. The SQL vs NoSQL Difference: MySQL vs MongoDB. Luettavissa: <https://medium.com/xplenty-blog/the-sql-vs-nosql-difference-mysql-vs-mongodb-32c9980e67b2> Luettu: 15.5.2018.

Liitteet

Liite 1. CCombiNationin MyNet esittelysivusto

SharePoint Yammer OneDrive Sites Jäppinen Leo ?

BROWSE PAGE PUBLISH SHARE FOLLOW EDIT

MyNet ENGLISH WHAT'S NEW STUDIES SERVICES CAREER AND ENTREPRENEURSHIP STUDENT ACTIVITIES

INTERNATIONAL CAMPUSES SUOMEKSI

International
Contact Information
Outgoing Students
Incoming Students
International Student Activities
CombiNation
Site Contents

Mynet » English » International » CombiNation

Not published yet Visitors can't see this page. [Publish it.](#)

CombiNation

Are you having trouble learning languages on lectures or just wish to learn more and faster than the university offers?

CombiNation is a programme, which helps you learn languages in an intensive and practical way. It combines you with a partner who speaks the language you wish to learn and sends you both off to your combined journey.

Example: A German exchange student comes to study in Haaga-Helia and wishes to learn Finnish. He then fills in the CombiNation form and is paired with a Finn, who wishes to learn German. They both meet face to face and learn the language faster through conversations and handling everyday situations.

Who can apply?

Everyone! You can be a student, professor or a staff member in Haaga-Helia faculty to join the programme. CombiNation uses the same credentials as MyNet, so no additional login is required. Easy, right?

How to apply?

- Open the application form and fill in the compulsory fields.
- Once your partner has been found, you will both receive an email with each other's contact information in addition with the information they fill in the form. (We cannot guarantee a partner for everyone, but we will do our best to find you one!)
- If you have any questions about the Tandem programme, you can contact us here <-link

What to do after I find a CombiNation partner?

Once a match is found, you will receive an email with each others info partner:ration. After that you are responsible of your interaction with your CombiNation partner.

Keep in mind that you are both in the role of a teacher and a student, so put equally as much effort in teaching as you do in learning.

Tandem partners are not necessarily qualified language teachers, so don't demand too much from your partner.

Things to do with my CombiNation partner?

Start by sending a greeting message to your partner via email or Facebook! After that you can plan to meet in person or decide just keep your newly found friendship on a written level.

If you're out of ideas what to talk or do with your partner, here's a few options:

Topics to discuss:

- Favourite food/color/kardashian
- Games
- Hobbies
- Pets
- Culture

Things to do:

- Meet for a cup of coffee
- Go hiking in the vicinity
- Sightseeing
- Bowling, Boulderling, Swimming, Ice skating, Sledge sliding during winter
- Homework together
- Invite for a night out with your friends

APPLY HERE

Last modified at 5/8/2018 20:12 by Jäppinen Leo

Yhteystiedot	Hakijapalvelut	Ratapihantie 13 00520 HELSINKI Puh. 0400 230 409	Laskutusosoite	Linkkejä / Hyödyllistä
Ota yhteyttä Puhelinvaihte (09) 229 611	Ammattilinen opettajakorkeakoulu	Ratapihantie 13 00520 HELSINKI Puh. (09) 229 611	Haaga-Helia ammattikorkeakoulu Oy Taloustoimisto Ratapihantie 13 00520 HELSINKI	• Copyright © 2015 Haaga-Helia
Helsinki Haaga Pajuniityntie 11 00320 Helsinki Puh. (09) 229 611	Porvoo	Talokatu 1 06100 Porvoo Puh. 040 488 7444	Verkkolaskun vastaanotto Haaga-Helia ammattikorkeakoulu Oy Verkkolaskuosoite / OVT-TUNNUS 003720291888 Operaattori Basware Oy Välittäjän tunnus BAWCFI22	
Helsinki Malmi Hietakummuntie 1 A 00700 Helsinki Puh. (09) 229 611	Vierumäki	Kaskelantie 10 19120 Vierumäki Puh. (09) 229 611	Y-tunnus 2029188-8	
Helsinki Pasila Ratapihantie 13 00520 Helsinki Puh. (09) 229 611	Intra-palaute	viestinta@haaga-helia.fi		

Liite 2. Ohjelman lomakesivu (combination.php)

CombiNation

Name *	<input type="text" value="Erkki Esimerkki"/>
Email *	<input type="text" value="erkki.esimerkki@myy.haaga-helia.fi"/>
Age	<input type="text" value="20"/>
Gender	<input type="text" value="Male"/>
Degree	<input type="text" value="Bachelors"/>
Home country	<input type="text" value="Finland"/>
My language *	<input type="text" value="Finnish"/>
My language level *	<input type="text" value="C2 Distinguished"/>
Language I'd like to learn *	<input type="text" value="Welsh"/>
Introduction * (Tell your future partner something about yourself :))	<div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">I like turtles!</div>

985

I agree to the [terms of service](#)

Lets go!

Liite 3. Ohjelman tilastosivu (statistics.php)

Thank you for joining CombiNation!

You will shortly receive a confirmation email of your application being submitted.

Your application

Offered language: Finnish
Wanted language: Welsh
Language level: C2 Distinguished
Name: Erkki Esimerkki
Email: erkki.esimerkki@myy.haaga-helia.fi
Degree: Bachelors
Gender: Male
Age: 20
Country Country: Finland
Introduction: I like turtles!

Statistics

Native Finnish speakers looking for	
Language	Number
French	2
English	3
German	5

Native Finnish speakers wanted by	
Language	Number
French	2
English	3
German	5

Other language combinations	
Language	Number
French	2
English	3
German	5

Thank you for joining CombiNation!

We will send you a confirmation email, once a CombiNation match for you has been found!

(Be sure to check your spam folder once in a while.)

Your application

Name: Arnold Schwarzenegger

Email: leomarjappinen@gmail.com

Age: 70

Gender: Male

Degree: Masters

Home country: Austria

My language: German

My language level: C2 Distinguished

I want to learn: Finnish

Introduction: Put the cookie down!

Liite 5. CombiNATION parin löydyttyä saatava sähköposti (molemmat osapuolet)

Google

Gmail

1 of 882

COMPOSE

Inbox (117)

Starred

Sent

Drafts (2)

More

Leo

No recent chats
Start a new one

We have found a match for you! Inbox x

1:54 AM (0 minutes ago)

We have found a match for you!

Congratulations! We have found someone, who would like to start a combined journey with you!

Your application

Name: Sylvester Stallone
Email: a1402935@my.haaga-helia.fi
Age: 71
Gender: Male
Degree: Masters
Home country: Italy
My language: Finnish
My language level: C2 Distinguished
I want to learn: German
Introduction: I'm putting together a team, wanna join?

info@combination.fi via cloud30.hostingpalvelu.fi to me

We have found a match for you!

Congratulations! We have found someone, who would like to start a combined journey with you!

Your CombiNATION partners info

Name: Arnold Schwarzenegger
Email: leomarjappinen@gmail.com
Age: 70
Gender: Male
Degree: Masters
Home country: Austria
My language: German
My language level:
I want to learn: Finnish
Introduction: Put the cookie down!

info@combination.fi via haagahelia.onmicrosoft.com to a1402935

1:54 AM (0 minutes ago)

Click here to [Reply](#), [Reply to all](#), or [Forward](#)

0.56 GB (3%) of 15 GB used [Manage](#) [Terms - Privacy](#) Last account activity: 1 hour ago [Details](#)

Liite 6. Admin kirjautumissivu (login.php)

Admin page

Log in

Username

Password

Liite 7. Admin näkymä 1 (admin.php)

Admin page

Searching

Deleted

Matched

Logout

Searching

Show entries Search:



Offered language	Wanted language	Language level	Name	Email	Degree	Gender	Age	Home Country	Date	Delete
Arabic	Welsh	C2 Distinguished	Leo Jäppinen	leomarjappinen@gmail.com	Bachelors		0		2018-05-08	<input type="button" value="X"/>
Arabic	Welsh	C2 Distinguished	Leo Jäppinen	leomarjappinen@gmail.com	Bachelors		0		2018-05-08	<input type="button" value="X"/>
Arabic	Welsh	C2 Distinguished	Leo Jäppinen	leomarjappinen@gmail.com	Bachelors		0		2018-05-08	<input type="button" value="X"/>
Finnish	Welsh	C2 Distinguished	Leo Jäppinen	leomarjappinen@gmail.com	Bachelors	Male	29	Finland	2018-05-08	<input type="button" value="X"/>
Finnish	Welsh	C2 Distinguished	Erkki Esimerkki	erkki.esimerkki@myy.haaga-helia.fi	Bachelors	Male	20	Finland	2018-05-08	<input type="button" value="X"/>
Finnish	Welsh	C2 Distinguished	Erkki Esimerkki	erkki.esimerkki@myy.haaga-helia.fi	Bachelors	Male	20	Finland	2018-05-08	<input type="button" value="X"/>
Arabic	Finnish	A1 Novice	Asd	a@aha.fi	Bachelors		0		2018-05-07	<input type="button" value="X"/>
Arabic	German	A1 Novice	Asd	leomarjappinen@gmail.com	Bachelors		0	Finland	2018-04-26	<input type="button" value="X"/>
Japanese	Chinese	A1 Novice	Proto	a@aha.fi	Bachelors		12		2018-04-26	<input type="button" value="X"/>
Arabic	Finnish	A1 Novice	Asd	a@aha.fi	Bachelors		0		2018-04-26	<input type="button" value="X"/>

Showing 21 to 30 of 40 entries

Previous 1 2 4 Next

Liite 8. Admin näkymä 2 (adminDeleted.php)





Deleted

Offered language	Wanted language	Language level	Name	Email	Date	Restore	Delete permanently
Assamese	Welsh	A1 Novice	Dsa	sariilintumaa@gmail.com	2018-03-17		

Liite 9. Admin näkymä 3 (adminMatched.php)

Matched

Show entries Search:

Match ID	Name	Wanted	Offered	Language level	Email	Degree	Gender	Age	Home Country	Date	Delete
6	Anna Vettel	Finnish	German	A1 Novice	anna@gmail.com	Bachelors	Female	22	Germany	2018-03-02	
6	Seppo Taalasmaa	German	Finnish	A1 Novice	seppo@taalasmaa.fi	Masters	Male	67	Helsinki	2018-03-02	
7	Das	Finnish	Swedish	A1 Novice	leomarjappinen@gmail.com	Bachelors		0	Zimbabwe	2018-03-17	
7	Das	Swedish	Finnish	A1 Novice	leomarjappinen@gmail.com	Bachelors		0	Zimbabwe	2018-03-17	
8	Das	Finnish	Swedish	A1 Novice	leomarjappinen@gmail.com	Bachelors		0	Zimbabwe	2018-03-17	
8	Das	Swedish	Finnish	A1 Novice	leomarjappinen@gmail.com	Bachelors		0	Zimbabwe	2018-03-17	
9	Asda	Finnish	German	A1 Novice	sariilintumaa@gmail.com	Bachelors		0		2018-03-19	
9	Asd	German	Finnish	A1 Novice	leomarjappinen@gmail.com	Bachelors		0		2018-03-19	

Showing 1 to 8 of 8 entries

Previous Next

Liite 10. Kuva Postmanista


The screenshot shows the Postman interface with a POST request to `http://localhost/ont_p/register.php`. The response status is 200 OK, with a time of 290 ms and a size of 480 B. The response body is displayed in HTML format, showing a warning message:

```
1 <br />
2 <b>Warning</b>: mysql_fetch_array() expects parameter 1 to be mysql_result, boolean given in
3 <b>
4 <b>40</b> on line
5 <br />
```

Liite 11. Tandem Programme rekisteröintilomake

[Home](#)
[Contact & Directions](#)
[Login](#)
[Deutsch](#)


[Our Profile](#)
[Degree Programmes](#)
[International](#)
[For Companies](#)



Hochschule Reutlingen
Reutlingen University

BEFORE STUDYING
WHILE STUDYING
AFTER STUDYING

Choose School v



REUTLINGEN
INTERNATIONAL OFFICE

CONTACT

OUR PARTNER
UNIVERSITIES

DESTINATION: THE REST OF
THE WORLD

DESTINATION REUTLINGEN

COOPERATION PARTNERS
COMMUNICATION

Intercultural training
Workshops on specific
countries
Study strategies and
learning techniques
Tandem programme
Students4Students

PROJECTS

TANDEM PROGRAMME REGISTRATION

Language*

Target language*

Language level*

First name*

Last name*

My registration is
valid until.*

Your school?

Gender

Age

Home country

Other comments

OVERVIEW

- Before Studying
- While Studying
- After Studying
- Our Profile
- Degree Programmes
- Research
- International
- For Companies

GENERAL


- Home
- Legal Notice
- Site map

SCHOOLS


- Applied Chemistry
- ESB Business School
- Informatics
- Engineering
- Textiles & Design

CROSS-LINKED


[f](#) [e](#)




Mitglied der EUA -
European University
Association




**VIELFALT®
GESTALTEN**
DIVERSITY ADOPTS
DES DIFFERENTIALS



PRME



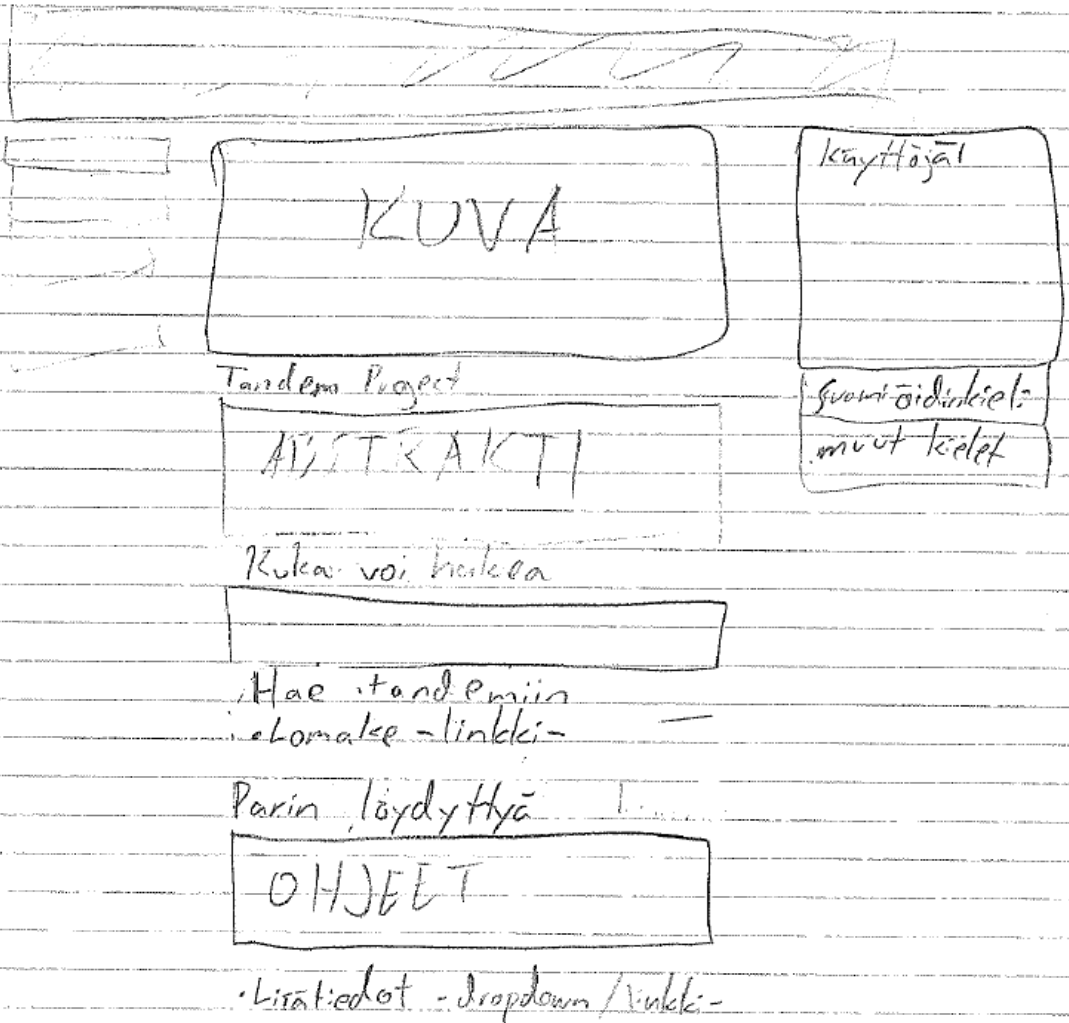
Zertifiziert seit 2010
als weltweite Partnerin
hochschule



**WELTOFFENE
HOCHSCHULEN**
GEGEN FREIHEIT
FEINDLICHKEIT

Liite 12. Kuva ohjelman rautalankamallista (MyNet näkymä)

Mockup etusivu



Liite 13. Kuva lomakesivun rautalankamallista

Lomake

Hand-drawn wireframe of a form on lined paper. The form is enclosed in a rounded rectangle and contains the following fields from top to bottom:

- Kieli * (with a dropdown arrow)
- Haluttu kieli (with a dropdown arrow)
- Taso (with a dropdown arrow)
- Nimi (with a text input field)
- email (with a text input field)
- kehutusaste (with a dropdown arrow)
- sukupuoli (with a dropdown arrow)
- ikä (with a text input field)
- kotimaa (with a text input field)
- kehoitus teksti (with a large text area)
- Lähetä (with a submit button)

Annotations and notes:

- Handwritten note: "pöytäkirja" with an arrow pointing to the top right of the form.
- Handwritten note: "A, B" with an arrow pointing to the "Kieli" field.
- Handwritten note: "262" next to the "Lähetä" button.
- Handwritten note: "lähettä kopion mailiin" with an arrow pointing to the "Lähetä" button.

fade out
fade in

kiitos + infotext

Hakemuksesi on voimassa... asti

Liite 14. Kuva admin näkymän rautalankamallista

Hokamukset Matchit Poistettut

Search bar

opettajien numerot	nimi	osasto kieli	haluttu kieli	PVM	matchit	Poista
=	nimi					X
≠	nimi					X
≠	nimi					X

hokamuksia: (määrä)

Matchit

match 1
match 1

match 2
match 2

Poistettut

			Poista
±1	nimi	kielot, ne	✓
			✓
			✓
			✓

Liite 15. Helpdeskin vastaus protopyyntöön



TUNNUSHAKEMUS
VIRTUAALIKONE

1 (1)

1.1.2013

Opettaja täyttää

Täytä jokaiselle toteutuksen ryhmälle oma lomake

Kurssin tunnus ja toteutus	Opinnäytetyö
Kurssin nimi	combiNation
Virtuaalikoneen aloitus pvm (pp.kk.vvvv)	Heti
Virtuaalikoneen lopetus pvm (pp.kk.vvvv)	31.12.2018 (ohjelma jää talolle)
Vastuullisen opettajan tunnus	A1402935, a1402956
Vastuullisen opettajan nimi	Leo Jäppinen, Juha Ahola
Virtuaalikoneen levyn määrä (default 2GB jos jätetään tyhjäksi)	
Portti 80 näkyviin ulkoverkkoon (kyllä/ei), oletus ei.	kyllä
Ohjelmistokehityskurssin ict2tn007 virtuaalikone (kyllä/ei)	ei
Koneelle laitettavien opiskelijoiden opiskelijanumerot (ja haluttaessa nimet)	A1402935 Leo Jäppinen A1402956 Juha Ahola

Ylläpito täyttää

Virtuaalikoneen nimi	proto351.haaga-helia.fi
-----------------------------	-------------------------

Tämä lomake toimitetaan ylläpidolle ja ylläpito palauttaa lomakkeen takaisin opettajalle tehtyään toimenpiteet ja täytettyään omat kohtansa. Ylläpidolle toimittaminen tapahtuu lähettämällä lomake osoitteeseen helpdesk@haaga-helia.fi ja cc:nä jiri.sariola@haaga-helia.fi
Virtuaalipalvelimien toimitusaika 1-2 arkipäivää.

Vanhalla pusku palvelimella ei enää käytetä kurssitarkoituksessa tomcat-palvelimia alkaen 1.1.2013.

CombiNation-palautelomake

☐

☐

1. CombiNationiin oli helppo navigoida.

☐

Eri mieltä ☐ ~~777~~ ☐ 2 ☐ 3 ☐ 4X ☐ 5 ☐ Samaa mieltä ☐

☐

2. Ohjeistus oli selkeää ymmärsin konseptin täysin.

☐

Eri mieltä ☐ ~~777~~ ☐ 2 ☐ 3 ☐ 4 ☐ 5X ☐ Samaa mieltä ☐

☐

3. Lomake oli selkeää ymmärsin kaikki kohdat.

☐

Eri mieltä ☐ ~~777~~ ☐ 2 ☐ 3 ☐ 4 ☐ 5X ☐ Samaa mieltä ☐

☐

4. Tiedän, miten edetä lomakkeen täyttämisen jälkeen.

☐

Eri mieltä ☐ ~~777~~ ☐ 2X ☐ 3 ☐ 4 ☐ 5 ☐ Samaa mieltä ☐

☐

☐

☐

Risut ja ruusut: ☐

-☐

☐

☐

☐

Kerro minun sanoin, miten parantaisit ohjelmaa: ☐

-Ohjeistus vahvistussivun sulkemiseen ☐

CombiNation-palautelomake

☐

☐

1. CombiNationiin oli helppo navigoida.

☐

Eri mieltä ☐ ~~777~~ ☐ 2 ☐ 3X ☐ 4 ☐ 5 ☐ Samaa mieltä ☐

☐

2. Ohjeistus oli selkeää ja ymmärsin konseptin täysin.

☐

Eri mieltä ☐ ~~777~~ ☐ 2 ☐ 3X ☐ 4 ☐ 5 ☐ Samaa mieltä ☐

☐

3. Lomake oli selkeää ja ymmärsin kaikki kohdat.

☐

Eri mieltä ☐ ~~777~~ ☐ 2 ☐ 3 ☐ 4X ☐ 5 ☐ Samaa mieltä ☐

☐

4. Tiedän, miten edetä lomakkeen täyttämisen jälkeen.

☐

Eri mieltä ☐ ~~777~~ ☐ 2 ☐ 3 ☐ 4X ☐ 5 ☐ Samaa mieltä ☐

☐

☐

☐

Risut ja ruusut: Kaikki meni hyvin ☐

☐

☐

☐

☐

Kerro minun sanoin, miten parantaisit ohjelmaa: ☐

CombiNation-palautelomake

☐

☐

1. CombiNationiin oli helppo navigoida.

☐

Eri mieltä ☐ ~~777~~ ☐ 2x ☐ 3 ☐ 4 ☐ 5 ☐ Samaa mieltä ☐

☐

2. Ohjeistus oli selkeää ja ymmärsin konseptin täysin.

☐

Eri mieltä ☐ ~~777~~ ☐ 2 ☐ 3 ☐ 4x ☐ 5 ☐ Samaa mieltä ☐

☐

3. Lomake oli selkeää ja ymmärsin kaikki kohdat.

☐

Eri mieltä ☐ ~~777~~ ☐ 2 ☐ 3 ☐ 4 ☐ 5x ☐ Samaa mieltä ☐

☐

4. Tiedän, miten edetä lomakkeen täyttämisen jälkeen.

☐

Eri mieltä ☐ ~~777~~ ☐ 2x ☐ 3 ☐ 4 ☐ 5 ☐ Samaa mieltä ☐

☐

☐

☐

Risut ja ruusut: ☐

☐

☐

☐

☐

Kerro minun sanoin, miten parantaisit ohjelmaa: ☐