

Roope Tapaninen

Uuden ominaisuuden toteuttaminen Symbion Competence Center -työkaluun

Uuden ominaisuuden toteuttaminen Symbion Competence Center -työkaluun

Roope Tapaninen
Opinnäytetyö
Kevät 2017
Tietojenkäsittely
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittely, Internet-palvelut ja digitaalinen media

Tekijä(t): Roope Tapaninen

Opinnäytetyön nimi: Uuden ominaisuuden toteuttaminen Symbion Competence Center -työkaluun

Työn ohjaaja: Liisa Auer

Työn valmistumislukukausi ja -vuosi: Kevät 2017

Sivumäärä: 25

Symbio on maailmanlaajuinen IT-alan yritys, joka tarjoaa asiakkailleen laadunvarmistus- ja ohjelmistokehityspalveluita. Olen ollut Symbiolla töissä opintojen ohella vähän yli vuoden, ja siitä suurimman osan ajasta asiakasprojektissa. Opinnäytetyön aiheen sain, kun kysyin esimieheltäni opinnäytetyön aiheita asiakasprojektin päätyttyä.

Symbiolla on käytössä sisäinen työkalu nimeltään Competence Center, eli CC-tool. CC-toolia käytetään henkilöstön osaamisen ja kompetenssin kartoittamiseen, ja siihen on myös liitetty muita ominaisuuksia kuten projektinhallinta, lomien merkkautus ja asiakkaat. Työn tarkoituksena on kehittää lomien merkkautus -järjestelmää ja ratkaista muutama keskeinen ongelma lomien merkkautukseen liittyen.

CC-tool on kehitetty Ruby-ohjelmointikieltä ja Ruby on Rails -ohjelmistokehystä käyttäen. Tietoperustassa kerrotaan Rubystä, Ruby on Railsista ja MVC-mallista. Lisäksi kerrotaan kuinka MVC-malli liittyy Ruby on Railsiin. Toteutusosiossa kerrotaan vaihe vaiheelta, kuinka muutokset CC-tooliin toteutettiin.

Työn tuloksena saatiin toimeksiantajalle toimivia ratkaisuja lomien merkkautus -järjestelmän ongelmiin. CC-tooliin jäi kuitenkin paljon kehitettävää, ja sitä tullaan kehittämään opinnäytetyön jälkeen vielä paljon.

Asiasanat: Web-sovellus, Verkkosivut, Model–View–Controller, Ruby, Ruby on Rails, Ohjelmointi

ABSTRACT

Oulu University of Applied Sciences
Business Information Systems, Internet-services and digital media

Author(s): Roope Tapaninen

Title of thesis: Developing a new feature to Symbio's Competence Center -tool

Supervisor(s): Liisa Auer

Term and year when the thesis was submitted: Spring 2017 Number of pages: 25

Symbio is a global IT company, that offers quality assurance and software engineering services to its customers. I have been working at Symbio while studying for a little over year, and most of it I have been in a customer project. I got the subject for the thesis when I asked for it from my supervisor after the customer project ended.

Symbio is using an internal tool called Competence Center, or CC-tool. CC-tool is used to determine personnel competence and expertise. There are also other features in CC-tool such as project management, leave requests and clients. The focus of this thesis is to develop the leave request system and solve a few fundamental issues related to it.

CC-tool has been developed with Ruby programming language and Ruby on Rails framework. The theoretical part will tell about Ruby, Ruby on Rails and MVC pattern. It will also tell how MVC pattern is used in Ruby on Rails. The practical part will tell step-by-step how the changes in CC-tool were implemented.

As the result of this thesis, the problems related to the leave request system were solved. There is still a lot of room for improvement in CC-tool, and there will be a lot of improvements made to CC-tool in the future.

Keywords: Web-application, Website, Model–View–Controller, Ruby, Ruby on Rails, Programming

SISÄLLYS

1	JOHDANTO	6
2	KEHITTÄMISKOHDE	7
2.1	Leaves-näkymä.....	7
2.2	Vanhan version ongelmat	8
3	KEHITYSYMPÄRISTÖ.....	10
3.1	MVC eli Model–View–Controller	10
3.2	Ruby.....	11
3.3	Ruby on Rails ja MVC.....	11
3.4	Railsin muut komponentit.....	12
4	TOTEUTUS	14
4.1	Kehitysympäristön valmistelu.....	14
4.2	Lomien hyväksyminen.....	14
4.3	Excel-tiedosto lomamerkkauksista	18
4.4	Sähköposti-ilmoitus lomamerkkauksen muutoksista.....	19
5	LOPPUTULOS	21
6	POHDINTA.....	23
	LÄHTEET.....	24

1 JOHDANTO

Tämän opinnäytetyön aiheena on toteuttaa uusi ominaisuus Symbion Competence Center - työkaluun eli CC-tooliin. Projektin toimeksiantajana toimi Symbio, joka on maailmanlaajuinen IT-alan yritys. Symbio tarjoaa asiakkailleen ohjelmistokehitys- ja laadunvarmistuspalveluita. Symbiolla on maailmanlaajuisesti noin 2000 ja Suomessa noin 200 työntekijää.

Olen ollut Symbiolla töissä opintojen ohella vähän yli vuoden, josta suurimman osan ajasta olen ollut asiakasprojektissa. Työtehtäviini on kuulunut asiakkaan laajan järjestelmän kehittäminen sekä frontendissä että backendissä uusimmilla web-teknologioilla, kuten Reactilla ja Spring-bootilla. Opinnäytetyön tekeminen tuli ajankohtaiseksi asiakasprojektin päättyttyä ja kun sain väliaikaiseksi tehtäväksi CC-toolin kehittämisen.

CC-tool on kehitetty Ruby on Rails -ohjelmistokehystä ja MVC-mallia apuna käyttäen. Ruby on Rails on yksi suosituimmista ohjelmistokehyksistä web-kehityksessä. Suosituimmissa ohjelmointikielissä on omat MVC-ohjelmistokehityksensä. Sain toimeksiantajalta tehtäväksi tehdä muutoksia CC-toolin lomien merkkkaus -järjestelmään.

Tässä opinnäytetyössä esitellään ensimmäiseksi kehittämiskohde, eli CC-tool ja Leaves-näkymä yleisesti, jonka jälkeen esitellään alkuperäisen toteutuksen ongelmat ja käyttötapaukset. Ensimmäisen osion tarkoituksena on antaa lukijalle käsitys siitä, millainen CC-tool on ja mitä sille lähdetään tekemään.

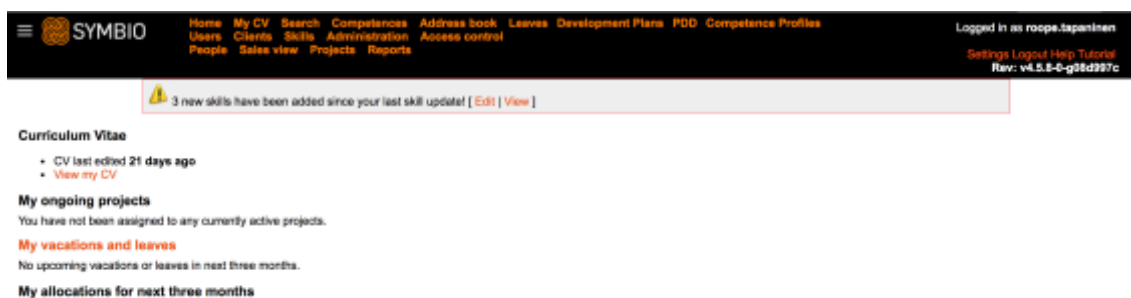
Kehittämiskohteen jälkeen esitellään kehitysympäristö, jotta lukija pystyy ymmärtämään mitä toteutusvaiheessa tehdään. Kehitysympäristössä selitetään yleisesti MVC-mallista, Rubystä ja Ruby on Railsista ja siitä, miten MVC-malli yhdistyy Ruby on Railsiin.

Lopuksi esitellään vaihe vaiheelta, miten muutokset toteutettiin sovellukseen ja sen koodiin. Koodista ja lopputuloksesta on paljon kuvakaappauksia, koska muutoksia on vaikeaa selittää pelkällä tekstillä. Viimeisenä pohditaan opinnäytetyön aikataulua ja lopputulosta. Lisäksi mietitään mikä meni hyvin ja missä jäi parannettavaa.

2 KEHITTÄMISKOHDE

Competence Center -työkalu eli CC-tool on Symbion kehittämä työkalu, joka on alun perin tarkoitettu Symbion henkilöstön osaamisen ja kompetenssin kartoittamiseen. Osaamisen kartoittamiseen on ensisijaisesti tarkoitettu CC-toolin CV-ominaisuus, jossa käyttäjä voi luoda oman CV:nsä helposti Symbion asettelujen mukaisesti. CC-tooliin on myös liitetty muita ominaisuuksia kuten projektinhallinta, lomien merkkäminen ja asiakkaat.

CC-toolin etusivulla (katso kuvio 1.) käyttäjä näkee tärkeimmät tiedot, eli sillä hetkellä meneillään olevat projektit, tulevat lomat ja CV:n tilanteen. Ylhäällä käyttäjä näkee navigoinnin, jossa peruskäyttäjälle näkyisi vain ylin valikkorivi. Tämän opinnäytetyön varsinainen kehittämisen kohde sijaitsee kuitenkin Leaves-näkymässä.



KUVIO 1. Competence Center -työkalun etusivu

2.1 Leaves-näkymä

Leaves-näkymässä (katso kuvio 2.) käyttäjä voi merkata lomiaan. Lomamerkkauksia voi selata, muokata, poistaa, suodattaa eri kriteerien mukaan (esim. loman alkamispäivä) ja järjestää tietyn kriteerin mukaan. Peruskäyttäjälle Leaves-näkymässä näkyy vain heidän omat lomamerkkauksensa.

Esimiehelle Leaves-näkymässä näkyy hänen alaisensa lomamerkkaukset. Esimies voi selata, muokata, poistaa, suodattaa niitä eri kriteerien mukaan ja järjestää niitä tietyn kriteerin mukaan. Esimies voi lisäksi tehdä lomamerkkauksen alaisensa puolesta ja hyväksyä alaisensa lomamerkkaukset.

Leaves and vacations

Enter new vacation or leave

Filters

Surname: Start after: Ends before: Type: Show only approved: Only direct subordinates: Per page:

Displaying Leave 1 - 10 of 2116 in total

Select	Name	Start date	End date	Type	Approved	Notes	Commands
<input type="checkbox"/>	Roopela, Roope	18.04.2011	21.04.2011	Holiday	Approved (2012-04-27)	Approved	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Roopela, Roope	19.05.2011	28.05.2011	Non-paid	Approved (2012-04-03)	Approved	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Roopela, Roope	21.05.2011	20.02.2012	Parental	Yes	Approved	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Roopela, Roope	27.05.2011	05.06.2011	Holiday	Approved (2012-04-20)	Approved	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Roopela, Roope	01.09.2011	30.04.2012	Parental	Approved (2012-04-27)	Approved	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Roopela, Roope	08.08.2011	01.07.2011	Holiday	Approved (2012-04-20)	Approved	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Roopela, Roope	13.05.2011	10.07.2011	Holiday	Approved (2012-04-20)	Approved	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Roopela, Roope	13.05.2011	26.06.2011	Holiday	Approved (2012-04-27)	Approved	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Roopela, Roope	16.06.2011	11.07.2011	Holiday	Approved (2012-04-27)	Approved	<input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	Roopela, Roope	17.05.2011	17.07.2011	Holiday	Approved (2012-04-27)	Approved	<input type="checkbox"/> <input type="checkbox"/>

-- Previous 1 2 3 4 5 6 7 8 9 ... 211 212 Next --
 With selected: approve Execute

KUVIO 2. Leaves-näkymä josta on poistettu henkilötiedot

2.2 Vanhan version ongelmat

Vanhassa versiossa oli useanlaisia ongelmia. Henkilöstövastaava ei pystynyt hyväksymään tai näkemään työntekijöiden lomamerkkauksia. Lomia pystyi käsittelemään vain CC-toolissa, eikä niitä pystynyt tuomaan esimerkiksi Exceeliin tarkasteltavaksi. Lomamerkkauksen tehnyt ei saanut minkäänlaista tietoa lomamerkkaustensa tilanteesta kirjautumatta CC-tooliin. Lisäksi leaves-näkymä oli integroitu erään toisen järjestelmän kanssa, ja samat toiminnallisuudet pystyi osittain toteuttamaan myös siellä.

Toiminnallisuutta haluttiin nyt yhdenmukaistaa, ja siirtää kaikki toiminnallisuus CC-tooliin. Samalla haluttiin poistaa lomamerkkauksiin liittyvät ongelmat. Muutostarpeet käyttötapauksina olivat seuraavat.

1. Esimiehenä voin antaa alustavan hyväksynnän alaisten lomamerkkauksille.
2. Henkilöstövastaavana voin listata kaikki lomamerkkaukset tietyltä aikaväliltä saadakseni korkeamman tason käsityksen lomista.
3. Henkilöstövastaavana voin tuoda listan lomamerkkauksista Exceeliin loman alku- ja loppupäivineen.
4. Henkilöstövastaavana voin antaa hyväksynnän lomamerkkauksille.
5. Käyttäjänä saan sähköpostiviestin hyväksytyistä ja hylätyistä lomista. (Aittola, sähköpostiviesti 3.3.2017.)

Muutostarpeiden kanssa piti kuitenkin pitää mielessä, että CC-tool on jo vanha työkalu ja että se todennäköisesti joutuu suuremman muutoksen kohteeksi lähitulevaisuudessa. Tästä johtuen muutosten pitää olla toimivia, mutta nopeita toteuttaa, joten mihinkään ylimääräiseen tyylyttelyyn ei välttämättä kannata käyttää aikaa.

3 KEHITYSYMPÄRISTÖ

CC-tool on kehitetty Ruby on Rails -ohjelmistokehyksen päälle MVC-mallin tapaan, tietokantana on PostgreSQL. Tässä luvussa kerrotaan MVC-mallista, Rubystä, Ruby on Rails -ohjelmistokehyksestä ja niiden ominaisuuksista. Lisäksi kerrotaan kuinka MVC-mallia sovelletaan Ruby on Railsissa.

3.1 MVC eli Model–View–Controller

MVC eli Model–View–Controller on arkkitehtuurimalli, joka jakaa ohjelman kolmeen eri komponenttiin: malliin eli malliin, viewiin eli näkymään ja controlleriin eli ohjaimen. (Tutorialspoint 2017, viitattu 23.4.2017.) MVC-mallia käytetään yleisesti perinteisistä työpöydän sovelluksista web- ja mobiilisovelluksiin. Yleisimmillä ohjelmointikielillä kuten Javalla, C#:lla, Rubyllä ja PHP:lla on omat MVC-ohjelmistokehykset, joita voidaan käyttää web-kehityksessä. (Wikipedia 2017, viitattu 23.4.2017.)

Model eli **malli** on MVC-mallin keskeisin komponentti. Malli vastaa kaikesta käyttäjän työstämään dataan liittyvästä logiikasta. Tämä voi olla esimerkiksi dataa jota liikutellaan ohjaimen ja näkymän välillä. (Wikipedia 2017, viitattu 23.4.2017; Tutorialspoint 2017, viitattu 23.4.2017.)

View eli **näkymä** määrittelee, miten tieto esitetään käyttäjälle. Näkymä voi olla esimerkiksi tekstikenttä, alavetovalikko tai koko näyttö. Se pitää sisällään kaiken käyttöliittymän logiikan, joiden kanssa käyttäjä on tekemisissä. Näkymä näyttää käyttäjälle, kun mallin tilassa tapahtuu muutoksia. (Wikipedia 2017, viitattu 23.4.2017; Tutorialspoint 2017, viitattu 23.4.2017.)

Controller eli **ohjain** prosessoi sovelluksen logiikan ja sovellukseen tulevat kutsut, manipuloi sovelluksen dataa käyttäen mallia ja keskustelelee näkymien kanssa tulostaakseen lopullisen näkymän. Ohjain voi muuttaa mallin tilaa ja muokata näkymää. (Wikipedia 2017, viitattu 23.4.2017; Tutorialspoint 2017, viitattu 23.4.2017.)

Vaikka MVC-malli onkin alun perin kehitetty työpöytäsovelluksia varten, se on otettu laajalti käyttöön web-kehityksessä suosituimmista ohjelmointikielissä, kuten Javassa, C#:ssa, Rubyssä ja

PHP:ssa. MVC-malliin pohjautuvia ohjelmistokehyksiä on useita, ja ne yleensä eroavat toisistaan siinä, miten MVC:n osa-alueet jaetaan palvelimen ja selaimen kesken. Vanhimmissa MVC-ohjelmistokehyksissä, kuten Ruby on Railsissa, suurin osa MVC:n logiikasta sijaitsee palvelimella. Tyypillistä näissä on, että asiakas lähettää kutsuja palvelimelle ja saa vastauksena kokonaisen päivittyneen sivun. (Wikipedia 2017, viitattu 23.4.2017.)

3.2 Ruby

Ruby on dynaaminen avoimen lähdekoodin ohjelmointikieli, jonka tarkoituksena on olla yksinkertainen ja tuottava. Rubyyn ovat vaikuttaneet Rubyn luojan, Yukihiro "Matz" Matsumoton, lempi ohjelmointikielät Perl, Smalltalk, Eiffel, Ada ja Lisp. Matz julkaisi Rubyn vuonna 1995. (Ruby community 2017a, viitattu 9.4.2017; Ruby community 2017b, viitattu 9.4.2017.)

Rubyssä kaikki asiat ovat olioita, jolle voidaan määritellä ominaisuudet ja metodit. Näin ollen se on joustava ja hyvin muokattavissa oleva ohjelmointikieli. Rubyn muita ominaisuuksia ovat mm. virheidenkäsittely, dynaaminen lisäosien lataaminen, monisäikeistys (multithreading) ja yhteensopivuus eri käyttöjärjestelmien kanssa. Ruby on täysin ilmainen käyttää, kopioida, muokata ja jakaa. (Ruby community 2017b, viitattu 9.4.2017.)

Ruby muistuttaa paljon muita ohjelmointikieliä, joten sen oppiminen on nopeaa, mikäli osaa ohjelmoida jo jollakin ohjelmointikielillä. Suurimmasta osasta muita ohjelmointikieliä eroten Rubyssä esimerkiksi kaikki arvot paitsi "nil" tai "false" on totta, metodien nimet voivat päättyä kysymys- tai huutomerkkiin ja luokat (class) ovat avoimia, eli niihin voi lisätä myöhemmin ominaisuuksia ja metodeja. (Ruby community 2017c, viitattu 9.4.2017.)

3.3 Ruby on Rails ja MVC

Ruby on Rails on Rubyllä kirjoitettu web-kehitykseen käytetty avoimen lähdekoodin ohjelmistokehys eli framework. Se on Githubin toiseksi suosituin web-kehitykseen käytetty ohjelmistokehys yli 35000 tähdellään. Sitä käyttävät esimerkiksi yritykset ja yhteisöt kuten Airbnb, Basecamp, Github ja Twitter. (Hartl 2015, viitattu 9.4.2017; Github, viitattu 7.5.2017.)

Active Record on MVC:n malli, ja se pitää sisällään ohjelman tiedot ja logiikan. Sen avulla voidaan luoda ja käyttää olioita, joiden data tarvitsee paikan tietokannasta. Se yhdistää sovelluksen oliot tietokannan taulujen kanssa niin, että data voidaan tallentaa ja hakea tietokannasta ilman SQL-kyselyitä. Active Record myös varmistaa olioiden tiedot ennen kuin ne lisätään tietokantaan. (Railsguides 2017a, viitattu 10.4.2017.)

Action View on MVC:n näkymä. Railsissa web-kyselyt käsittelee Action Controller ja Action View. Action Controller vastaa yhteyksistä tietokantaan ja Action View vastaa tiedon näyttämisestä käyttäjälle halutulla tavalla. Action View -näkyvät kirjoitetaan Rubyn ja HTML:n sekoituksella ".erb" -päätteiseen tiedostoon. Jokaiselle Action Controllerille on olemassa oma Action Viewinsä. Sovellus näyttää tietyn Action Viewin perustuen sovelluksen URL-osoitteeseen. (Railsguides 2017d, viitattu 10.4.2017.)

Action Controller on MVC:n ohjain. Sovellus käyttää tiettyä ohjainta perustuen sovelluksen URL-osoitteeseen. Ohjain käsittelee web-kyselyn ja tulostaa halutun vastauksen. Ohjainta voisi kuvailla välimieheksi mallin ja näkymän välillä. (Railsguides 2017e, viitattu 10.4.2017.)

3.4 Railsin muut komponentit

Active Record Migraatiot mahdollistavat valmiin tietokannan muokkauksen. Migraatiot kirjoitetaan Ruby-kielellä, joten puhtaan SQL:n käsin kirjoittaminen ei ole tarpeen. Jokainen migraatio on uusi versio tietokannasta. Tietokannan malli on aluksi tyhjä, mutta jokainen migraatio muokkaa sitä lisäämällä tai poistamalla tauluja tai sarakkeita. Migraatioita voi myös ajaa taaksepäin ja näin voi perua jo tehdyt muutokset. (Railsguides 2017b, viitattu 10.4.2017.)

Active Record Validaatiot varmistavat, että vain halutunlainen data menee tietokantaan. Sovellukselle voi olla esimerkiksi tarpeen, että jokaisen käyttäjän tulee antaa toimiva sähköpostiosoite. Rails tarjoaa apumetodeja yleiseen käyttöön ja mahdollistaa omien validaatiometodien luomisen. (Railsguides 2017c, viitattu 10.4.2017.)

Action Mailer mahdollistaa sähköpostien lähettämisen sovelluksesta käyttämällä siihen tarkoitettuja luokkia ja näkymiä. Mailerit ovat hyvin samanlaisia ohjaimien kanssa, sillä niillä on myös metodit

nimeltä "actions" ja omat Action Viewinsä, jotka kokoavat sähköpostin sisällön halutun näköiseksi.
(Railsguides 2017f, viitattu 10.4.2017.)

4 TOTEUTUS

4.1 Kehitysympäristön valmistelu

Vaadittujen ominaisuuksien toteuttamiseksi kehitysympäristö piti ensiksi saada kuntoon. Aiemmin kehitys on tapahtunut virtuaalikoneen sisällä, ja se tuntui pienen kokeilun jälkeen mielestäni erittäin aikaa vievältä, joten päätin siirtää kehitysympäristön omalle tietokoneelleni. Kehitysympäristön asentamiseen liittyvä dokumentaatio oli vanhentunutta, joten selviteltävää riitti.

Ensiksi latsin CC-toolin koodit Symbion sisäisestä Gitistä omalle tietokoneelleni, jonka jälkeen alkoi riippuvuuksien selvittäminen. Riippuvuudet ovat CC-toolin toimintaan vaadittavia moduuleja ja palveluita, kuten esimerkiksi redis, joka toimii tietokannan välimuistina. Päätin kuitenkin ensin asentaa CC-toolin tietokannan. Loin omalle koneelleni PostgreSQL tietokannan, jonka jälkeen loin VirtualBoxilla olevasta CC-toolin tietokannasta SQL-dumpin ja asensin sen omalle tietokoneelleni.

Tietokannan asentamisen jälkeen yritin käynnistää sovellusta komennolla `rails server`, jolla sain heti tietoon jo muutaman puuttuvan riippuvuuden virheviestien perusteella. Koska sovellus oli vanha ja dokumentaatio heikkoa, riippuvuuksien selvittämisessä ja asentamisessa meni useita päiviä. Riippuvuuksien asentaminen eteni kuitenkin virheviestistä seuraavan, kunnes kaikki riippuvuudet oli asennettu ja sovellus alkoi toimimaan.

4.2 Lomien hyväksyminen

Lomien hyväksymiseen piti lisätä toinen vaihe henkilöstövastaavan hyväksymiselle. Kehittämisen aloitin miettimällä tietokannan skeemaa uudelleen. Kaksivaiheisen hyväksynnän toteuttamiseksi piti tietokantaan Leaves-tauluun lisätä sarakkeita henkilöstövastaavan hyväksynnälle. Loin uuden migraation komennolla `rails generate migration addLeaveHrApproval`, jonka jälkeen lisäsin luotuun migraatiotiedostoon tarvittavat muutokset (katso kuvio 3). Tarvittavia sarakkeita olivat `hr_approver_id`, eli hyväksyneen henkilöstövastaavan id, `hr_approver_date`, eli henkilöstövastaavan hyväksymisen ajankohta. Lisäksi `hr_approver_id` piti yhdistää viiteavaimella käyttäjän id:hen.

```

class AddLeaveHrApproval < ActiveRecord::Migration
  def change
    add_column :leaves, :hr_approver_id, :integer, :not_null => false
    add_column :leaves, :hr_approved_date, :datetime, :not_null => false
    add_foreign_key :leaves, :users, column: "hr_approver_id", on_update: :restrict, on_delete: :restrict
  end
end

```

Kuvio 3. Migraatiodiedosto, joka lisää tietokantaan sarakkeita henkilöstövastaavan hyväksynnälle.

Jotta vain henkilöstövastaava voi antaa kyseisen hyväksynnän, piti tietokantaan luoda uusi oikeus henkilöstövastaavan hyväksynnälle. Loin taas uuden migraation komennolla `rails generate migration addHrLeaveConfirmationRight`, ja lisäsin luotuun migraatiodiedostoon tarvittavat muutokset (katso kuvio 4). Migraatiodiedostoon piti lisätä uusi oikeus `tt_hr_confirm_leaves`, joka mahdollistaa henkilöstövastaavan hyväksynnän antamisen. Ylläpitäjä voi myöhemmin antaa oikeuden henkilöstövastaavan roolin omaaville käyttäjille CC-toolissa.

```

1 class AddHrLeaveConfirmationRight < ActiveRecord::Migration
2   def change
3     Right.create( { :right => 'tt_hr_confirm_leaves', :right_category_id => 2, :description => "Right
4     to give final confirmation to leaves and vacations." })
5   end
6 end

```

Kuvio 4. Migraatiodiedosto, joka lisää tietokantaan oikeuden henkilöstövastaavan hyväksynnälle.

Migraation jälkeen henkilöstövastaavat pystyvät hyväksymään lomamerkkaukset. CC-tooliin piti kuitenkin vielä lisätä mahdollisuus hylätä lomamerkkaukset, joten loin sillekin uuden migraation komennolla `rails generate migration addLeaveDeclineStatus`. Koska esimiehen ja henkilöstövastaavan hylkäykset piti erotella jollakin tavalla, piti migraatiodiedostoon molemmille oma sarake (katso kuvio 5). Tarvittavat sarakkeet olivat `hr_declined` henkilöstövastaavan hylkäykselle ja `declined` esimiehen hylkäykselle.

```

1 class AddLeaveDeclineStatus < ActiveRecord::Migration
2   def change
3     add_column :leaves, :hr_declined, :boolean, default: false
4     add_column :leaves, :declined, :boolean, default: false
5   end
6 end
7

```

Kuvio 5. Migraatiodiedosto joka lisää tietokantaan sarakkeen henkilöstövastaavan ja esimiehen hylkäykselle.

Koska migraatioita ei niiden tietokantaan ajamisen jälkeen enää ole tapana muokata, piti jokaiseen muutokseen luoda erillinen migraatiotiedosto. Ajoin migraatiot komennolla `rake db:migrate`, jonka jälkeen muutokset olivat tietokannassa. Nyt oli mahdollista alkaa toteuttaa muutoksia käyttöliittymään ja logiikkaan.

Leave-malliin piti lisätä validointi henkilöstövastaavan hyväksynnälle, käytännössä matkien aikaisempaa toteutusta hyväksyjästä (katso kuvio 6). Validointi tarkastaa, että lomamerkkauksella voi olla hyväksyjä jonka luokkanimi on `User`, ja se liitetään tiettyyn käyttäjään viiteavaimella.

```
4 belongs_to :approver, :foreign_key => :approver_id, :class_name => 'User'
5 belongs_to :hr_approver, :foreign_key => :hr_approver_id, :class_name => 'User'
```

Kuvio 6. Leave-malliin lisätty validointi henkilöstövastaavalle.

Käyttöliittymän osalta Leaves-näkymään piti lisätä sarake henkilöstövastaavan hyväksynnälle, ja lisäksi piti jotenkin erottaa hyväksytty ja hylätty päätös. Lisäksi tarvittiin valinta hylkäämiselle. Päätin tehdä hyväksytyt ja hylätyt päätöksen erottelamisen lisäämällä vain sanan "approved" tai "declined" hyväksyjän nimen ja ajankohdan perään. Sarakkeen lisääminen ei vaatinut suuria muutoksia koodiin (katso kuvio 7). Kopioin kuvion 7 ylemmässä kuvassa näkyvän koodin ja loin uuden sarakkeen henkilöstövastaavan hyväksynnälle alemman kuvan tapaan. Lisäksi muokkasin hieman selitetekstiä, joka näkyy, kun osoittimen vie sarakkeen päälle.

```
27 <th><span title="Holiday approved status. Holidays must be approved in advance by supervisor.">
    >Approved</span></td>
28 <th><span title="Holiday approval status. Holidays must be approved by supervisor and HR.">
    Supervisor Approval Status</span></td>
29 <th><span title="Holiday approval status. Holidays must be approved by supervisor and HR.">HR
    Approval Status</span></td>
```

Kuvio 7. Sarakkeen lisääminen Leaves-näkymään.

Valinta hylkäämiselle vaati hylkäysvalinnan lisäämisen alavetovalikkoon (katso kuvio 8). Kun valinta tehdään, käyttää näkymä ohjaimen metodia `update_leaves`, joka käsittelee valinnan. Valintanäkymä myös varmistaa, että käyttäjällä on vaaditut oikeudet apumetodilla `@user.has_right`. Apumetodiin piti lisätä parametri `tt_hr_confirm_leaves`, jolla tarkastetaan, onko käyttäjällä oikeutta antaa henkilöstövastaavan hyväksyntä. Ruby-koodi sijoitetaan ".erb"-tiedostossa `<% ja -%>` -tagien sisään.


```

41 <select id="leave_action" name="leave_action">
42   <? if @user.has_right([:tt_confirm_leaves, :tt_hr_confirm_leaves]) ->
43     <option value="confirm">approve</option>
44   <? end ->
45   <? if @user.has_right([:tt_confirm_leaves, :tt_hr_confirm_leaves]) ->
46     <option value="decline">decline</option>
47   <? end ->
48   <? if @user.has_right([:tt_delete_leaves]) ->
49     <option value="delete">delete</option>
50   <? end ->
51 </select>

```

Kuvio 8. Alasvetovalikon koodi muutoksen jälkeen.

Ohjaimessa piti toteuttaa logiikkamuutoksia (katso kuvio 9), jotta henkilöstövastaavan hyväksyntä voidaan käsitellä ja hylkäysominaisuus toimisi. Käytännössä muutoksessa piti ottaa huomioon, onko hyväksyjällä tai hylkääjällä henkilöstövastaavan oikeudet tai onko hän loman merkanneen esimies. Jos hänellä on molemmat oikeudet, tulee hyväksyntä tai hylkäys molempiin sarakkeisiin.

```

161 # execute selected action
162 leaves.each do |leave|
163   if action == "confirm"
164     if @user.is_subordinate(leave.person.user)
165       leave.approver_id = @user.id
166       leave.approved_date = Time::now()
167       leave.declined = false
168     end
169     if @user.has_right(:tt_hr_confirm_leaves)
170       leave.hr_approver_id = @user.id
171       leave.hr_approved_date = Time::now()
172       leave.hr_declined = false
173     end
174     unless (leave.hr_approver_id.nil? or leave.approver_id.nil? or leave.hr_declined or leave.
175            declined)
176       leave.confirmed = true
177     end
178     send_leave_notification(leave, @user)
179     leave.save
180   elsif action == "decline"
181     if @user.is_subordinate(leave.person.user)
182       leave.approver_id = @user.id
183       leave.approved_date = Time::now()
184       leave.declined = true
185     end
186     if @user.has_right(:tt_hr_confirm_leaves)
187       leave.hr_approver_id = @user.id
188       leave.hr_approved_date = Time::now()
189       leave.hr_declined = true
190     end
191     leave.confirmed = false
192     send_leave_notification(leave, @user)
193     leave.save
194   elsif action == "delete"
195     leave.destroy()
196   else
197     flash[:warning] = "Unsupported action: " + action
198   end
199 end

```

Kuvio 9. Uusi hyväksynnän ja hylkäämisen toteutus ohjaimen update_leaves-metodissa.

4.3 Excel-tiedosto lomamerkkauksista

Lomamerkkauksen tuomiseksi Exceliin päätin ensin luoda näkymään linkin, joka ohjaa `export`-metodiin ohjaimessa (katso kuvio 10). Käytin linkin luomiseksi valmista apumetodia `link_to`, joka luo oikeanlaisen linkin oikeilla tyylimäärittelyillä. Sijoitin linkin näkymässä taulukon yläpuolelle mielestäni sopivaan kohtaan.

```
8 <%= link_to("Export to XLSX", { :action => "export", :format => "xlsx", :filter => @filter }, id: "export_link" ) ->
```

Kuvio 10. Linkki ohjaimen metodiin näkymässä.

Jotta linkki toimisi, piti minun tehdä ohjaimen `export`-metodi (katso kuvio 11). Metodi käyttää toista ohjaimen metodia `filter_by_user`, jota käytetään myös Leaves-näkymän tietoja haettaessa. `filter_by_user`-metodi suodattaa lomat käyttäjän määrittelemien kriteerien mukaan, jonka jälkeen `list_leaves`-metodi listaa lomat, ja näin ollen Exceliin tulostuu samat lomat kuin mitä Leaves-näkymässä näytetään. Lopuksi käytetään railsin metodia `respond_to`, joka vastaa kutsuun määritellyllä formaatilla ja asetuksilla, eli tässä tapauksessa xlsx-tiedostolla.

```
228 def export
229   filter_by_user(params)
230   list_leaves()
231   date = Time.now.to_date
232
233   respond_to do |format|
234     format.xlsx {
235       response.headers['Content-Dispostion'] = "attachment; filename=\"leaves_#{@date.strftime('%Y%m%d')}.xlsx\""
236     }
237   end
238 end
```

Kuvio 11. Export-metodi ohjaimessa.

Xlsx-formaatin käyttämistä varten piti kuitenkin luoda `.xlsx.xlsx`-päätteinen näkymä (katso kuvio 12), jolla määritellään tulostuvan tiedoston asetukset. Jokaiselle riville tulostetaan loman merkkauksen nimi, loman tyyppi, alkamispäivä, loppumispäivä, esimiehen hyväksyntä, henkilöstövastaavan hyväksyntä, muut merkinnät ja linkki lomamerkkaukseen CC-toolissa.

```

1  wb = xlsx_package.workbook
2  style_shout = wb.styles.add_style sz: 16, b: true, alignment: { horizontal: :center }
3  wb.add_worksheet(name: "Leaves") do |sheet|
4    cols = [ "Person", "Type", "Start Date", "End Date", "Supervisor Approval Status", "HR Approval Status", "Notes", "URL" ]
5    sheet.add_row cols
6
7    # Rows
8    for leave in @leaves
9      row = 11
10
11     if leave.person.full_name
12       row << leave.person.full_name
13     else
14       row << ""
15     end
16
17     if leave.leave_type.name
18       row << leave.leave_type.name
19     else
20       row << ""
21     end
22
23     row << leave.start_date.strftime("%Y-%m-%d")
24     row << leave.end_date.strftime("%Y-%m-%d")
25
26     str = ""
27     unless leave.approver.nil?
28       unless leave.declined
29         str = "Approved"
30       else
31         str = "Declined"
32       end
33     end
34     row << str
35
36     str = ""
37     unless leave.hr_approver.nil?
38       unless leave.hr_declined
39         str = "Approved"
40       else
41         str = "Declined"
42       end
43     end
44     row << str
45     row << leave.notes
46
47     row << url_for({ :controller => "tt/leaves", :action => "view_leave", :id => leave.id, :format => "html", :only_path =>
48       false })
49
50     sheet.add_row row
51   end
52 end

```

Kuvio 12. *.xlsx.xlsx*-päätteinen näkymä, joka asettelee *xlsx*-tiedoston halutulla tavalla.

4.4 Sähköposti-ilmoitus lomamerkkauksen muutoksista

Sähköposti-ilmoituksen toteuttamiseksi piti käyttää ResourceMailer-mallia, joka perii Ruby on Railsin ActionMailerin. Malliin lisäsin uuden metodin `leave_changes` (katso kuvio 13). Metodi lähettää sähköpostin käyttäen sähköpostin näkymää (katso kuvio 14), joka periaatteessa on samanlainen kuin muutkin näkymät.

```

95  def leave_changes(recipients, leave, changed_by, person)
96    logger.info("Subscribed notification sent to: "+recipients.to_s)
97    @recipients = add_domain_part(recipients)
98    @subject = "[cc-tool] Leave approval changes for " + person.full_name
99    @changed_by = changed_by
100   @leave = leave
101
102   mail(to: @recipients, subject: @subject)
103  end

```

Kuvio 13. *metodi*, joka lähettää sähköposti-ilmoituksen lomamerkkaukseen tapahtuneista muutoksista.

Kuvion 14 näkymä on tavallinen “.erb”-päätteinen tiedosto, jota vain käytetään sähköpostin muotoilemiseen nettisivun muotoilun sijaan. Myös tässä näkymässä näytetään esimiehen ja henkilöstövastaavan hyväksynnit ja hylkäykset, jos niitä on. Ensin tarkastetaan, onko lomamerkkaukselle annettu hyväksyntöjä tai hylkäyksiä ja sen jälkeen lisätään joko sana ”approved” tai ”declined” hyväksyjän tai hylkääjän perään. Näkymään lisätään myös loman muut tiedot ja linkki lomaan CC-toolissa apumetodilla `url_for`.

```

1  <p><i>Changed by: <%= @changed_by %></i></p>
2  <% url = url_for only_path: true, controller: 'tt/leaves', action: :view_leave, id: @leave.id %>
3
4  Leave approval status has been chooped:
5
6
7  <table>
8    <tr>
9      <td style="font-weight: bold;">Person</td>
10     <td>
11       <%= @leave.person.full_name %>
12     </td>
13   </tr>
14   <tr>
15     <td style="font-weight: bold;">Type</td>
16     <td><%= @leave.leave_type.name %>
17   </tr>
18   <tr>
19     <td style="font-weight: bold;">Duration</td>
20     <td><%= @leave.start_date %> to <%= @leave.end_date %></td>
21   </tr>
22   <tr>
23     <td style="font-weight: bold;">Supervisor Approval Status</td>
24     <td>
25       <% if not @leave.approver.blank? %>
26         <%= @leave.approver.full_name %> ( <%= @leave.approved_data.strftime("%Y-%m-%d") if not @leave.approved_data.blank? %> ]
27         <% if @leave.declined %>
28           <span style="color: red;">Declined</span>
29         <% else %>
30           <span style="color: lightgreen;">Approved</span>
31         <% end %>
32       <%= @leave %>
33     </td>
34   </tr>
35   <tr>
36     <td style="font-weight: bold;">HR Approval Status</td>
37     <td>
38       <% if not @leave.hr_approver.blank? %>
39         <%= @leave.hr_approver.full_name %> ( <%= @leave.hr_approved_data.strftime("%Y-%m-%d") if not @leave.hr_approved_data.blank? %> )
40         <% if @leave.hr_declined %>
41           <span style="color: red;">Declined</span>
42         <% else %>
43           <span style="color: lightgreen;">Approved</span>
44         <% end %>
45       <%= @leave %>
46     </td>
47   </tr>
48   <tr>
49     <td style="font-weight: bold;">Notes</td>
50     <td><%= @leave.notes %></td>
51   </tr>
52 </table>
53
54 <br>
55 <a href="<%= SERVER %><%= url %>"><%= SERVER %><%= url %></a>
56
57
58

```

Kuvio 14. Sähköpostin näkymä.

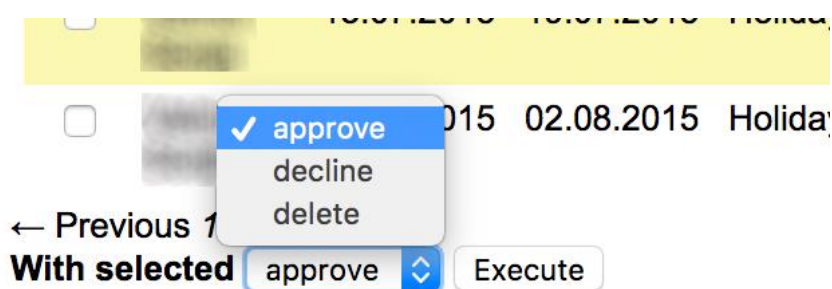
5 LOPPUTULOS

Muutosten jälkeen CC-tooliin saatiin uusia ominaisuuksia, joiden avulla lomamerkkausten käsittely helpottuu. Leaves-näkymään lisättiin sarake ”HR Approval Satus” henkilöstövastaavan hyväksynnälle (katso kuvio 15). Lisäksi näkymässä näkyy punaisella ”declined”, jos lomamerkkaus on hylätty tai vihreällä ”Approved”, jos lomamerkkaus on hyväksytty. Näkymässä on nyt myös linkki ”Export to XLSX” Excel tiedoston tulostamiseen. Lisäksi valinnalla ”With selected” on nyt mahdollista hylätä lomamerkkaus (katso kuvio 16).

The screenshot shows the 'Leaves and vacations' interface. At the top, there is a header 'Leaves and vacations' and a sub-header 'Enter new vacation or leave'. Below this, there are filters for Surname, Start date, End date, and Type. The main table displays a list of leave requests with the following columns: Select, Name, Start date, End date, Type, Supervisor Approval Status, HR Approval Status, Notes, and Commands. The table contains several rows of data, including entries for 'Holiday' and 'Overtime' with various dates and statuses like 'Approved' and 'Declined'. At the bottom, there are navigation buttons for 'Previous', 'Next', and 'Execute', along with a 'With selected' dropdown menu.

Select	Name	Start date	End date	Type	Supervisor Approval Status	HR Approval Status	Notes	Commands
<input type="checkbox"/>	Tapaninen, Roope (2017-03-16)	01.10.2013	07.10.2013	Holiday	Approved	Employee - testing, Role Employee (2017-03-15)	Declined	
<input type="checkbox"/>	Tapaninen, Roope (2017-03-16)	17.06.2014	23.06.2014	Holiday	Approved			
<input type="checkbox"/>	Tapaninen, Roope (2017-03-16)	18.07.2014	25.07.2014	Holiday	Approved			
<input type="checkbox"/>	Tapaninen, Roope (2017-03-16)	28.07.2014	29.07.2014	Overtime	Declined			
<input type="checkbox"/>	Tapaninen, Roope (2017-03-16)	30.07.2014	01.08.2014	Non-paid	Declined			
<input type="checkbox"/>	Tapaninen, Roope (2017-03-16)	19.05.2015	25.05.2015	Overtime	Declined			
<input type="checkbox"/>	Tapaninen, Roope (2017-03-16)	26.06.2015	01.08.2015	Holiday	Declined	Employee - testing, Role Employee (2017-03-15)	Approved	
<input type="checkbox"/>	Tapaninen, Roope (2017-03-16)	29.06.2015	05.07.2015	Holiday	Declined			
<input type="checkbox"/>	Tapaninen, Roope (2017-03-16)	13.07.2015	19.07.2015	Holiday	Declined			
<input type="checkbox"/>	Tapaninen, Roope (2017-03-16)	27.07.2015	02.08.2015	Holiday	Declined			

KUVIO 15. Leaves-näkymä muutoksen jälkeen.



KUVIO 16. Decline ominaisuuden toteutus.

Lomamerkkaukset on mahdollista tuoda Exceliin (katso kuvio 17). Exceliin tuoduissa lomamerkkauksissa näkyy kaikki samat tiedot kuin Leaves-näkymässä lukuun ottamatta

lomamerkkauksen hyväksyneen esimiehen ja henkilöstövastaavan nimeä. Lisäksi Excelliin tuoduissa lomamerkkauksissa on linkki lomamerkkaukseen CC-toolissa. Kuviosta 17 poiketen linkki ei tuotannossa osoita lokaaliin CC-tooliin.

Person	Type	Start Date	End Date	Supervisor Approval Sta	HR Approval Status	Notes	URL
	Holiday	2013-10-01	2013-10-07	Approved	Declined		http://localhost:3000/leaves/view_leave.html?id=
	Holiday	2014-06-17	2014-06-23	Approved			http://localhost:3000/leaves/view_leave.html?id=
	Holiday	2014-07-18	2014-07-25	Approved			http://localhost:3000/leaves/view_leave.html?id=
	Overtime	2014-07-28	2014-07-29	Declined			http://localhost:3000/leaves/view_leave.html?id=
	Non-paid	2014-07-30	2014-08-01	Declined			http://localhost:3000/leaves/view_leave.html?id=
	Overtime	2015-05-19	2015-05-25	Declined			http://localhost:3000/leaves/view_leave.html?id=
	Holiday	2015-05-28	2015-06-01	Declined	Approved		http://localhost:3000/leaves/view_leave.html?id=
	Holiday	2015-06-29	2015-07-05	Declined			http://localhost:3000/leaves/view_leave.html?id=
	Holiday	2015-07-13	2015-07-19	Declined			http://localhost:3000/leaves/view_leave.html?id=
	Holiday	2015-07-27	2015-08-02	Declined			http://localhost:3000/leaves/view_leave.html?id=

KUVIO 17. Excelliin tuodut lomien merkkaukset muutoksen jälkeen.

Lomamerkkauksista lähetetään sähköpostiviesti lomamerkkauksen tehneelle, kun se hyväksytään tai hylätään (katso kuvio 18). Sähköpostiviestissä kerrotaan muutoksen tehneen henkilön nimi, lomamerkkauksen hyväksymisen tilanne ja muut tiedot. Sähköpostiviestissä on myös linkki lomamerkkaukseen CC-toolissa.

[cc-tool] Leave approval changes for [REDACTED]



Wednesday, 15 March 2017 at 15:50
To: Roope Tapaninen

Changed by: Tapaninen, Roope

Leave approval status has been modified:

Person [REDACTED]
>Type Holiday
>Duration 2013-10-01 to 2013-10-07
>Supervisor Approval Status Tapaninen, Roope (2017-03-15) **Declined**
>HR Approval Status Employee - testing, Role Employee (2017-03-15) **Declined**
>Notes National Day

[https://cc-staging.symbio.com/leaves/view_leave?id=\[REDACTED\]](https://cc-staging.symbio.com/leaves/view_leave?id=[REDACTED])

KUVIO 18. Sähköpostiviesti muutoksen jälkeen.

6 POHDINTA

Sain CC-toolin kehitettäväksi, kun kysyin opinnäytetyöaihetta esimieheltäni. Kehittääkseni CC-toolia minun piti opetella minulle kokonaan uusi ohjelmointikieli ja ohjelmointikehys, eli Ruby ja Ruby on Rails. CC-toolin kehityksen aloittaminen oli hankalaa, koska CC-toolia oltiin aikaisemmin kehitetty virtuaalikoneella, ja sitä viimeksi kehittänyt kehittäjä ei ollut enää Symbiolla töissä. Jouduin siis yksin selvittämään, miten CC-tool ja sen kehitysympäristö toimii.

Opinnäytetyön lopullisen aiheen tarkentumisessa kesti kauan. Alkuperäisenä ajatuksena oli tehdä opinnäytetyönä käyttöliittymä- ja käyttökokemussuunnittelua CC-tooliin, ja käyttää suunnittelun tietoperustana tutkimusmenetelmät-kurssilla tehtyä tutkimusta CC-toolin käyttökokemuksesta. Tutkimusta tehdessä sain kuitenkin tehtäväksi toteuttaa muutoksia CC-toolin lomien merkkauksen järjestelmään. Lopulta opinnäytetyön aihe tarkentui näiden muutosten toteuttamiseen.

Opettelussa ja kehitysympäristön asentamisessa meni noin kolme viikkoa. Toteutin muutokset jo ennen raportin kirjoittamista, eli opinnäytetyön tekeminen tapahtui väärässä järjestyksessä. Tietoperustan kirjoittamisessa toteutusvaiheen valmiina oleminen auttoi, koska pystyin suoraan kohdentamaan tietoperustan toteutukseen liittyviin asioihin. Lähteiden hakeminen oli toisaalta hankalaa, koska esimerkiksi MVC-mallista kertovat lähteet olivat lähes aina liitetty johonkin tiettyyn ohjelmointikehykseen. Kirjoitusvaihe kesti moninkertaisesti pidempään kuin toteutus, mutta aikataulussa kuitenkin pysyttiin melko hyvin. Aloituskustelusta opinnäytetyön valmistumiseen meni siis noin neljä kuukautta.

Muutosten avulla toimeksiantajalle saatiin toimivia ratkaisuja CC-toolin lomien merkkaukseen liittyviin ongelmiin. CC-tooliin jäi kuitenkin paljon kehitettävää, ja sitä tullaankin kehittämään opinnäytetyön jälkeen paljon. Itse en kuitenkaan tulevaisuudessa tule kehittämään CC-toolia, sillä työsuhteeni Symbiolla päättyi.

Opinnäytetyö antoi minulle kokemusta vanhan projektin haltuun ottamisesta ja kehittamisestä. Opin kokonaan uuden ohjelmointikielen ja -kehiksen. Yksin kehittäminen antoi minulle myös kuvan siitä, kuinka paljon hyötyä tiimistä on tällä alalla. Toivonkin ehdottomasti pääseväni tulevaisuudessa projekteissa työskentelemään tiimissä.

LÄHTEET

Aittola, M. 2017. RE: CC tool Leaves. Head of Operational Excellence, Symbio Finland Oy. Sähköpostiviesti 3.3.2017.

Github. 2017. Web application frameworks. Viitattu 7.5.2017, <https://github.com/showcases/web-application-frameworks>

Hartl, M. 2015. Ruby on rails tutorial: learn web development with Rails. Viitattu 9.4.2017, <https://3rd-edition.railstutorial.org/book/beginning>

Railsguides. 2017a. Active Record Basics. Viitattu 10.4.2017, http://guides.rubyonrails.org/active_record_basics.html

Railsguides. 2017b. Active Record Migrations. Viitattu 10.4.2017, http://guides.rubyonrails.org/active_record_migrations.html

Railsguides. 2017c. Active Record Validations. Viitattu 10.4.2017, http://guides.rubyonrails.org/active_record_validations.html

Railsguides. 2017d. Action View Overview. Viitattu 10.4.2017, http://guides.rubyonrails.org/action_view_overview.html

Railsguides. 2017e. Action Controller Overview. Viitattu 10.4.2017, http://guides.rubyonrails.org/action_controller_overview.html

Railsguides. 2017f. Action Mailer Basics. Viitattu 10.4.2017, http://guides.rubyonrails.org/action_mailer_basics.html

Ruby community. 2017a. Ruby is... Viitattu 9.4.2017, <https://www.ruby-lang.org/en/>

Ruby community. 2017b. About Ruby. Viitattu 9.4.2017, <https://www.ruby-lang.org/en/about/>

Ruby community. 2017c. Ruby from Other Languages. Viitattu 9.4.2017, <https://www.ruby-lang.org/en/documentation/ruby-from-other-languages/>

Tutorialspoint. 2017. MVC Framework – Introduction. Viitattu 23.4.2017, https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

Wikipedia. 2017. Model–view–controller. Viitattu 23.4.2017, <https://en.wikipedia.org/wiki/Model–view–controller>