

## NFC -tekniikkaan perustuvan tiedonhaku- ja uudelleenohjauspalvelun jatkokehittäminen - Case Tagglem

Ari Töyräänvuori



Tietojenkäsittelyn koulutusohjelma

<b>Tekijät</b> Ari Töyräänvuori	<b>Ryhmä</b> 2009
<b>Opinnäytetyön nimi</b> NFC-tekniikkaan perustuvan tiedonhaku- ja uudelleenohjauspalvelun jatkokehittäminen - Case Tagglem	<b>Sivu- ja liitesivumäärä</b> 21 +2
<b>Ohjaajat</b> Tanja Bergius	
<p>Tämän opinnäytetyön tarkoituksena oli jatkaa jo olemassa olevaa Tagglem-nimisen verkkosovelluksen kehittämistä. Tagglem on verkkoselainpohjainen markkinoinnin- ja mainoskampanjoiden hallintatyökalu, jonka on kehittänyt suomalainen yritys Tagglem Oy. Tagglem Oy toimi opinnäytetyön toimeksiantajana.</p> <p>Opinnäytetyössä tutkittiin käytettävyyttä, pilvipalveluita, Near Field Communication -tekniikkaa ja niiden mahdollisia hyötyjä Tagglem-ohjelmalle sekä yrityksen liiketoiminnalle että yleisesti ohjelmistokehityksessä. Tarkoituksena oli parantaa Tagglem-ohjelman käytettävyyttä ja tutkia mahdollisuuksia toteuttaa siihen NFC-tekniikkaan perustuva dynaaminen tiedonhaku- ja uudelleenohjausominaisuus.</p> <p>Opinnäytetyö toteutettiin kahdessa osassa - kehitys- ja teoriaosassa. Kehitysosuus toteutettiin ohjelmistokehitysprojektina aikavälillä 30.5.2014 - 25.8.2014. Teoriaosana toimii tämä opinnäytetyö. Sovelluskehitysosuuden aikana Tagglem-ohjelma päivitettiin versiosta 0.5.1 uuteen versioon 0.5.2. Teknisinä menetelminä käytettiin Twitter Bootstrap-, AngularJS- ja CSS-tekniikoita.</p> <p>Opinnäytetyön kehitysosuuden lopputuloksena syntyi ohjelman konekielinen versio ja lähdekoodi, joka palautettiin toimeksiantajan ylläpitämään versionhallintajärjestelmään. Merkittävimpänä hyötynä opinnäytetyöstä toimeksiantajalle oli se, että he saivat selkeämmän käyttöliittymän Tagglem-ohjelmaan.</p> <p>Tämä opinnäytetyö tarjoaa paljon tietoa käytettävyydestä, pilvipalveluista ja NFC-teknologiasta, jota toimeksiantaja voi mahdollisesti hyödyntää omassa liiketoiminnassaan.</p>	
<b>Asiasanat</b> käytettävyys, pilvipalvelut, NFC	

Degree Programme in Information Technology

<p><b>Authors</b> Ari Töyräänvuori</p>	<p><b>Group</b> 2009</p>
<p><b>The title of thesis</b> Further Development of a Data and Redirect Service Based on NFC-Technology - Case Tagglem</p>	<p><b>Number of pages and appendices</b> 21+2</p>
<p><b>Supervisors</b> Tanja Bergius</p>	
<p>The purpose of this thesis was to continue developing an already existing Web-application called Tagglem. Tagglem is a Web-browser based marketing- and advertising campaign management -application that has been developed by a Finnish company Tagglem Ltd. This thesis was made as an assignment for Tagglem Ltd.</p> <p>The objective of this thesis was to study usability-, cloud services- and Near Field Communication-techniques, and their potential advantages for Tagglem-application, the company's business, and generally in Software Development. The goal was to improve the usability of the Tagglem-application, and to research possibilities to implement a dynamic data search and redirect-feature.</p> <p>This thesis was done in two phases, the development and the theory phase. The development phase was carried out as a software development project during may 30<sup>th</sup> 2014 to august 8<sup>th</sup> 2014. This thesis is the result of the theory phase. During the development-phase the Tagglem-application was updated from version 0.5.1 to a new version 0.5.2. As technical tools Twitter Bootstrap, AngularJS and CSS-techniques were used.</p> <p>The result of the development phase was a machine-readable version of the application, and the applications source code that were delivered to the employer's version management system. The biggest advantage for the employer was getting a clearer user interface for Tagglem-application.</p> <p>This thesis provides a lot of information about usability, cloud services and NFC-technology that the employer can utilize in their own business.</p>	
<p><b>Key words</b> usability, cloud services, near field communication</p>	

# Sisällys

1	Johdanto .....	1
1.1	Termit .....	1
1.2	Opinnäytetyön toteutustapa .....	2
1.3	Tagglem-sovellus .....	2
1.3.1	Kampanjan luonti - käyttäjätarina .....	3
1.3.2	Raportoinnin hallinta - käyttäjätarina .....	4
2	Teknologiat ja työvälineet .....	6
2.1	Käytettävyys .....	6
2.2	Pilvipalvelut .....	7
2.3	NFC-tekniikka .....	10
2.4	AngularJS .....	12
2.5	Amazon AWS .....	13
2.6	Java .....	14
2.7	Scrum .....	14
2.8	Twitter Bootstrap .....	14
3	Tulokset .....	16
3.1	Lopputulos .....	16
3.2	Jatkokehitys .....	17
4	Johtopäätökset .....	18
4.1	Hyödyt toimeksiantajalle .....	18
4.2	Tekijän omat kokemukset ja oppiminen .....	18
	Lähteet .....	20
	Liitteet .....	22

# 1 Johdanto

Tagglem on jo ennestään olemassa oleva ohjelma, jonka on kehittänyt Tagglem Oy. Tagglem Oy on suomalainen ohjelmistoalan yritys. Tagglem-ohjelma on hallintatyökalu, joka toimii verkkoselaimella. Sovelluksen avulla käyttäjät voivat luoda ja hallita kampanjoita markkinointi- ja mainostarkoituksiin sekä tilata QR- ja NFC-tunnisteita, joita luetaan mobiililaitteilla ja näillä tunnisteilla voidaan ohjata käyttäjät tiettyyn verkko-osoitteeseen tai digitaaliseen dokumenttiin.

Opinnäytetyössä pyrittiin parantamaan Tagglem-ohjelman käytettävyyttä ja tutkia mahdollisuuksia toteuttaa siihen NFC (Near Field Communication) -tekniikkaan perustuva tiedonhaku- ja uudelleenohjausominaisuus. Ennen opinnäytetyön aloittamista ohjelmaan oli jo tehty määrittäydokumentti toimeksiantajan puolelta sekä ohjelma pystyi jo tekemään uudelleenohjauksen. Toimeksiantaja toivoi, että ohjelman toimintaa vielä parannettaisiin esimerkiksi siten, että uudelleenohjauksen sisältö olisi dynaamisesti muokattavissa.

Opinnäytetyön kehitysoisuuden tuloksena toimeksiantaja toivoi saavansa pidemmälle kehitetyn työkalun apuvälineeksi yrityksen sisäiseen sovelluskehitystyöhön. Mahdollisena tulevana hyötynä olisi, että löydetäisiin suorituskykyisempi ja toimivampi tapa toteuttaa uudelleenohjausominaisuus, jotta pystyttäisiin samanaikaisesti palvelemaan suuria määriä käyttäjiä ja tietomääriä nopeasti.

## 1.1 Termit

AngularJS	Googlen ylläpitämä Javascript-rajapinta, jota käytetään Tagglem-ohjelmassa
Amazon AWS	Amazon Web Services on pilvipalvelu- ja verkkopalvelinjärjestelmä, joka toimii Tagglem-ohjelman taustalla
CSS	Cascading Style Sheet on verkkosivujen tyyli-tiedosto ja tekniikka, jolla hallitaan verkkosivujen ulkoasua ja formatointia
Maven	Työkalu joka helpottaa Java-ohjelmien lisäosien hallintaa ja rakentamista. Tagglem-ohjelma on tehty Maven-projektina

NFC	Near Field Communication on tekniikka, joka mahdollistaa mobiililaitteiden välisen kommunikaation lähietäisyydellä ilman, että laitteet vaativat fyysistä kontaktia
QR-tunniste	Quick Response Code on tunniste, joka sisältää tietoa, joka voidaan lukea esimerkiksi älypuhelimella
Scrum	Ohjelmistokehityksen projektinhallintamenetelmä, jota käytettiin opinnäytetyön sovelluskehitysosuudessa
Tagglen	Tässä opinnäytetyössä kehitettävä sovellus
Twitter Bootstrap	Twitterin käyttöliittymärajapinta, jota käytetään Tagglen-ohjelman ulkoasussa

## 1.2 Opinnäytetyön toteutustapa

Opinnäytetyö jaettiin kahteen osaan - kehitys- ja teoriaosaan. Opinnäytetyön kehitysosuus toteutettiin soveltaen Scrum-projektinhallintamenetelmää. Kehitysosuus toteutettiin kuudessa kahden viikon jaksossa. Tätä jaksoa Scrumissa kutsutaan sprintiksi. Opinnäytetyön tekijä ja toimeksiantaja pitivät sprinttien päätyttyä Scrumin käytäntöjen mukaisen katselmoinnin.

Kehitysosuus toteutettiin aikavälillä 30.5.2014 - 25.8.2014. Projektissa ohjelmointi toteutettiin Eclipse-kehitysympäristössä. Sovelluskehitysosuuden päämääränä oli päivittää Tagglen-ohjelma versiosta 0.5.1 uuteen versioon 0.5.2 eli tehdä yksi ohjelmistokehityksen iteraatio.

## 1.3 Tagglen-sovellus

Tagglen on selainpohjainen sovellus, joka on kehitetty Java-ohjelmointikielellä ja siihen on käytetty useita muitakin tekniikoita (kuva 1.) Se on organisaation sisäinen hallintatyökalu, jolla voidaan hallita käyttäjien tietoja, luoda markkinointikampanjoita valituille aikaväleille sekä luoda kampanjoiden datan perusteella raportteja ja esittää ne käyttäjälle graafisessa- tai Excel- taulukkomuodossa.

Kuva 1. Tagglem-ohjelman päävalikko

Julkisessa versiossaan Tagglem-ohjelma käyttää MySQL-tietokantaa ja kehitysversiossaan HyperSonic2-tietokantaa. Tagglem-ohjelma on toteutettu Maven-projektina, joka helpottaa Java-ohjelmien lisäosien hallintaa.

Ohjelman lähdekoodi sijaitsee Git-versionhallinnassa. Versionhallinnalla tarkoitetaan palvelua - käytännössä palvelinta, joka mahdollistaa ohjelman lähdekoodin eri versioiden hallitsemisen erillistä ohjelmaa käyttäen muun muassa useiden kehittäjien välillä internetin välityksellä. Versionhallinta pitää huolen, että ohjelmoidessa koodaajat työskentelevät oikean lähdekoodin version parissa. Git on alun perin kehitetty vuonna 2005 Linux-käyttöjärjestelmälle, ja se on nykyään eniten käytetty versionhallintajärjestelmä ohjelmistonkehityksessä.

### 1.3.1 Kampanjan luonti - käyttäjätarina

Tagglem-ohjelman keskeisenä ominaisuutena on markkinointikampanjoiden luominen ja hallinta. Seuraavan käyttäjätarinan tarkoituksena on selvittää, miten kampanjan luonti tapahtuu Tagglemissä.

#### **Esimerkki A**

Kuvitellaan, että yritys X haluaa mainostaa jotain tuotettaan, palveluaan, nettisivuaan tai digitaalista dokumenttia. Yrityksellä X on verkkokauppa, jossa myydään elektroniikkatarvikkeita ja on meneillään alennusmyyntikampanja. Tällöin yritys X:n käyttäjä, jolla on tunnukset Tagglem-ohjelmaan voi kirjautua sisään ohjelmaan ja luoda uuden kampanjan painamalla ”Campaign” - ja ”New Campaign” -painikkeita.

Seuraavaksi ohjelma esittää käyttäjälle kolmivaiheisen lomakkeen, johon on syötettävä pyydytyt tiedot. Vaaditut tiedot ovat kampanjan nimi, kampanjan aloitus- ja lopetuspäivämäärä, sekä se onko kyseessä verkko-osoite vai digitaalinen dokumentti, jolloin valinnasta riippuen käyttäjä voi syöttää verkko-osoitteen tai lisätä omalta koneeltaan tiedoston, joka lopuksi siirretään talteen pilvipalvelimelle.

Tämän jälkeen käyttäjää pyydetään syöttämään kampanjaan liittyvät etuudet (engl. Assets) ja niiden sijainnin ja määrän. Näillä eduilla tarkoitetaan mitä tahansa markkinoinnissa käytettyä materiaaleja, jotka sisältyvät ohjelman lopulta generoimiin QR- ja NFC-tunnisteisiin. Yritys X:n tapauksessa etuudeksi laitetaan yrityksen nimi, sijainniksi yritys x:n toimiosoite ja tunnisteen määräksi 100 kappaletta. Tämän jälkeen käyttäjä painaa ”Complete”-näppäintä, jonka jälkeen käyttäjä voi tarkistaa tiedot, tehdä tilauksen ja valita haluaako asiakas generoidut tunnistet toimitettuna yritykselle sähköisenä kuvana tai tulosteina.

### **1.3.2 Raportoinnin hallinta - käyttäjätarina**

Tagglen-ohjelman toisena keskeisenä ominaisuutena on aikaisemmin luotujen kampanjoihin perustuvien raporttien muodostaminen ja hallinta. Seuraavan käyttäjätarinan tarkoituksena on selventää, miten raporttien luonti tapahtuu Tagglen-ohjelmassa.

#### **Esimerkki B**

Kuvitellaan tilanne, että yritys X:n verkkokaupan alennuskampanja on loppunut ja olisi tarve saada tietoja kampanjasta. Yritys X:n käyttäjä kirjautuu tunnuksillaan Tagglen-ohjelmaan ja painamalla päävalikosta ”Reporting”-painiketta avautuu ohjelman raportointinäkymä.

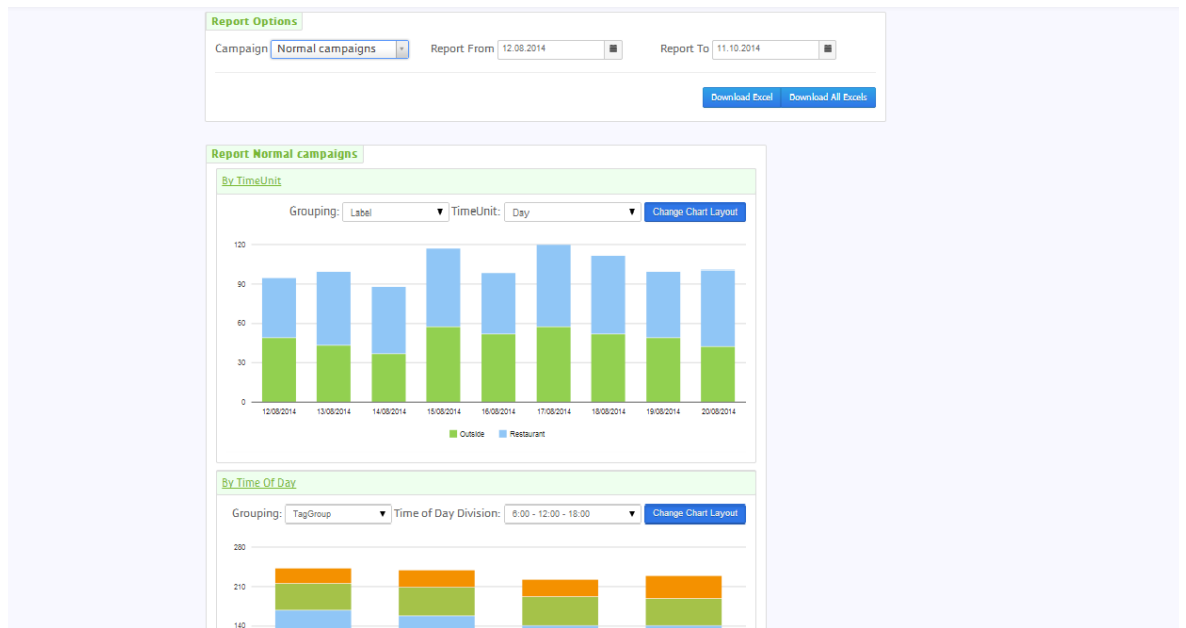
Tämän jälkeen käyttäjältä kysytään minkä kampanjan tiedot raportoidaan ja käyttäjä valitsee pudotusvalikosta haluamansa kampanjan eli tässä esimerkkitapauksessa yrityksen nimellä olevan kampanjan. Sitten käyttäjä valitsee kampanjan aikavälin, jolta raportti halutaan koostaa ja jos valitaan kampanjan aloitus - ja lopetuspäivämäärät, niin saadaan luotua raportti koko kampanjan ajalta.

Jos käyttäjä haluaa raportin taulukkomuodossa, painetaan ”Download Excel” -nappulaa, jolloin ohjelma lataa käyttäjän koneelle Excel-taulukon. Taulukko sisältää tiedot käyttäjän kampanjasta. Käyttäjä voi halutessaan myös tarkastella kampanjan graafisia kaavioita suoraan raportointinäkymässä.

Raportointinäkymässä graafiset kaaviot esitetään kolmessa eri muodossa. Tietoja voidaan tarkastella pylväskaavioina aikayksikön mukaisesti, kuten päivä, viikko tai kuukausi tai ryhmitellä ne NFC-tunnisteryhmien mukaan. Tietoja voidaan tarkastella myös vuorokauden eri kellonaikojen mukaisesti jaoteltuna, esimerkiksi kello 6:00-11:00-16:00-21:00. Tiedot esitetään pylväskaavioina, joiden esitystapaa voidaan muuttaa napinpainalluksella. Kolmannessa graafi-



nessä raportointinäkylässä tiedot esitetään yksinkertaisessa piirakkakaaviossa, jossa näkee kampanjan sisältämien tunnisteiden osuudet prosenttisarvoina. (Kuva 2.)



Kuva 2. Tagglem -ohjelman raportointinäkymä

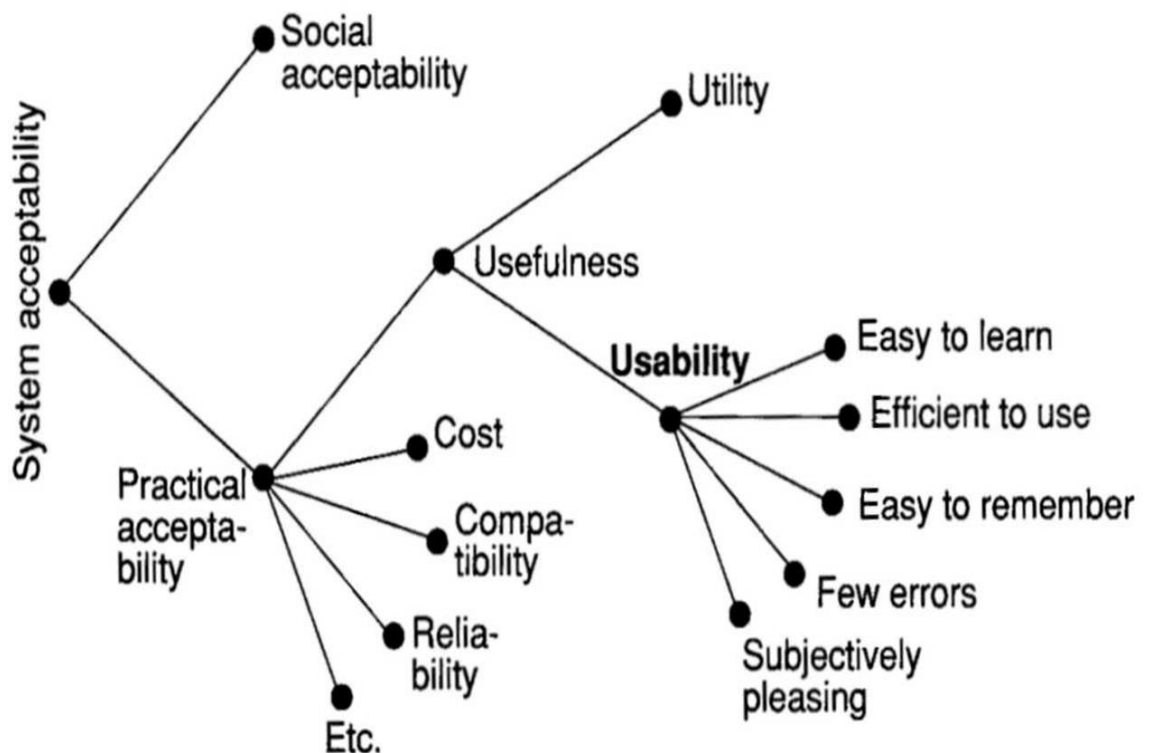
Tagglem-ohjelman generoima Excel-taulukko sekä graafinen esitys sisältävät tietoja muun muassa tunnisteiden nimistä, tunnisteiden ryhmästä, aikatiedoista, tunnisteiden tyypistä eli onko se QR-tunniste vai NFC-tunniste sekä tunnisteiden lukemiseen käytetyn verkkoselaimen tyyppin.

## 2 Teknologiat ja työvälineet

Tämä luku käsittelee tähän opinnäytetyöhön liittyviä teknologioita ja työvälineitä. Taggelm-ohjelma koostuu monista eri tekniikoista, joita tässä opinnäytetyössä tarvittiin. Opinnäytetyöhön liittyy keskeisesti käytettävyys, pilvipalvelut ja NFC-teknologia ja tässä luvussa käsitellään niihin liittyvää teoriaa.

### 2.1 Käytettävyys

Käytettävyydelle on olemassa monia erilaisia määritelmiä. Tanskalaisen käytettävyyskonsultti Jacob Nielsenin mukaan käytettävyydellä tarkoitetaan minkä tahansa laitteen tai sovelluksen käyttökelpoisuutta. Nielsenin mukaan käytettävyys voidaan jakaa viiteen tekijään: opittavuus, tehokkuus, muistettavuus, virheettömyys ja miellyttävyys. (Kuvio 1.) Jos käytettävyyden taso ei ole riittävä, ei laitetta tai sovellusta voida käyttää. (Parkkinen. 2002. 28.)



Kuvio 1. Nielsenin käytettävyyskaavio (Usability, Learnability, Memorability. 2009.)

Opittavuudella tarkoitetaan sitä, kuinka helppoa sovelluksen käyttö on ensimmäisellä kerralla. Tehokkuudella tarkoitetaan sitä, kun sovellusta on opittu käyttämään, sillä halutaan saavuttaa enemmän. Muistettavuudella tarkoitetaan kuinka helppoa järjestelmää on käyttää, kun sen on oppinut. Virheettömyydellä tarkoitetaan kun käyttäjää ei saada ohjata harhaan, virheiden tekemisen täytyy olla mahdollisimman vaikeaa. Miellettävyydellä tarkoitetaan, ettei sovelluksen käyttö saa olla ainakaan epämiellyttävää. (Parkkinen. 2002. 28-29.)

WWW-palvelun käytettävyyden määritelmä alkaa opittavuudesta. Kun käyttäjä tulee ensimmäistä kertaa palvelun sivulle, viipyy hän sillä eri tutkimusten mukaan kuudesta sekunnista kahteen minuuttiin. Opinnäytetyössä käytettävyyden kehittäminen rajattiin vain kirjautuneen käyttäjän näkymiin, joten työn aikana ei pyritty vaikuttamaan tähän osatekijään.

Taggelm-ohjelmaa kehitettäessä käytettävyyden parantaminen oli tärkein prioriteetti. Toimeksiantaja toivoi ohjelmaan yksinkertaisempaa ja selkeämpää käytettävyyttä. Esimerkiksi kaikki kampanjasivut siirrettiin saman pudotusvalikon alle, jotta ne olivat yhdenmukaiset ja helpompi ymmärtää kuuluvan loogisesti samaan kokonaisuuteen.

Nielsenin mukaan sivustojen pitäisi tehdä käyttäjien päätavoitteiden toteuttamisesta yksinkertaista. Sivuilla voi tietenkin olla muitakin toimintoja ja pitkällekin vietyjä ominaisuuksia, mutta yksinkertaiset toimenpiteet pitäisi pystyä suorittamaan mahdollisimman helposti ja tämän lisäksi suurin osa käyttäjistä haluaa sivujen olevan yksinkertaisia. (Parkkinen. 2002.)

Internetissä ihmiset käyttäytyvät hyvin tavoitehakuisesti eli heillä on jokin tavoite, joka täytyy saavuttaa. Tästä johtuen www-suunnittelun pääajatuksena on se, että pysyttäisiin pois käyttäjien tieltä ja autettaisiin heitä pääsemään päämääränsä mahdollisimman nopeasti. (Nielsen 2000, 380)

## **2.2 Pilvipalvelut**

Pilvipalveluilla (Cloud Services) tarkoitetaan internetissä toimivia verkkopalveluita, jotka mukautuvat käyttäjien tarpeisiin. Ne koostuvat palvelimista ja palveluista, jotka sallivat tietojen käsittelyn. Pilvipalvelut sisältävät keskitetyn tietojen tallennuksen, johon pääsee käsiksi internetin avulla. (What is Cloud Computing? 2014.)

Käytännössä siis käyttäjän tietoja säilytetään pilvipalvelimella, joten tiedot eivät voi taten hävitä käyttäjän koneelta, koska ne ovat tallessa etäpalvelimella. Pilvipalveluissa tietoihin pääsee kä- siksi omalla käyttäjätunnuksella ja salasanalla ja tämän johdosta pilvipalveluiden käyttäminen vaatii käyttäjätilin.

Käsitettä ”pilvi” käytetään kuvaamaan eri järjestelmistä koostuvaa infrastruktuuria. Usein ku- vattaessa tätä infrastruktuuria graafisesti, käytetään pilvenhattaran kuvaa havainnollistamaan palveluiden suhteita toisiinsa. (Kuva 4.)



Kuva 4. Graafinen esitys pilvipalveluista (JavaTpoint. 2014.)

Tagglem-ohjelma on kehitetty pilvipalvelupohjaiseksi, siten että se toimii Amazon Web Servi- ces -pilvipalvelinjärjestelmässä. Tagglemin lopullisessa versiossa ylläpidetään käyttäjien tietoja ja tiedostoja pilvipalvelimella.

Tagglemin käyttämä Amazonin pilvipalvelualusta tarjoaa yksinkertaisen verkkopalveluiden käyttöliittymän, jonka avulla voidaan käsitellä tietoja milloin vain, mistä tahansa internetissä. Sen etuna verkkosovelluskehittäjille on, että se tarjoaa pääsyn samalla korkeasti skaalautuvaan, luotettavaan, nopeaan ja edulliseen tietojentallennus-infrastruktuuriin, jota myös Amazon itse käyttää pyörittäessään omia verkkosivustojaan ja samalla maksimoiden näiden palveluiden tarjoamat edut verkkosovelluskehittäjille.

Pilvipalvelut tarjoavat nopean pääsyn joustaviin ja edullisiin tietoteknisiin resursseihin yrityksille ja liiketoiminnalle. Hyödyntämällä pilvipalveluja, ei tarvitse välttämättä tehdä suuria investointeja esimerkiksi laitteistoihin eikä käyttää vaivaa niiden asennuksiin ja ylläpitämiseen. Hyödynnä on myös se, että voidaan räätälöidä halutut palvelut oman liiketoiminnan tarpeiden mukaisiksi ja hyödyntää haluttuja resursseja ja maksaa vain niistä palveluista joita käytetään.

Pilvipalvelut tarjoavat helpon pääsyn palvelimille, tietokantoihin ja lukuisiin muihin palveluihin internetissä. Pilvipalveluiden tarjoajat kuten Amazon Web Services ylläpitää laitteistoja, joita tällaiset palvelut vaativat. Käyttäjät hallitsevat niitä verkkosovelluksen avulla. Pilvipalveluiden käytöllä on lukuisia hyötyjä muun muassa, ettei tarvitse investoida omiin palvelimiin ennen kuin tiedetään niiden tarve ja maksetaan vain niistä mitä tarvitaan. (What is Cloud Computing? 2014.)

Käyttämällä pilvipalveluja liiketoiminnassa voidaan saavuttaa alhaisemmat muuttuvat kustannukset kuin toimiessa ilman niitä. Samalla voidaan eliminoida se, ettei tarvitse arvioida liiketoiminnan tarpeita etukäteen. Kun tehdään päätöksiä kapasiteetin tarpeesta ennen sovellusten käyttöönottoa, päädytään usein ylimääräisiin resursseihin, jotka ovat turhia kuluja tai niiden kapasiteetti on rajoittunut. Pilvipalveluissa näitä ongelmia ei ole, koska resursseja voidaan käsitellä ja skaalata tarpeen mukaan vain muutaman minuutin varoitusajalla. Tämä lisää organisaation ketteryyttä, koska kokeilujen hinta ja aika on merkittävästi alhaisempi pilvipalveluita käytettäessä. (What is Cloud Computing? 2014.)

Voidaan myös keskittyä enemmän itse liiketoimintaan kuin laitteistoihin ja palvelinten ylläpitämiseen. Myös omien asiakkaiden palvelemiseen voidaan keskittyä paremmin. Sovellusten käyttöönotto voidaan nopeasti saada maailmanlaajuisesti. Tämä tarjoaa mahdollisuuden tarjota asiakkaille entistä paremman kokemuksen alhaisemmalla latenssilla eli viiveellä ja alhaisella hinnalla.

### 2.3 NFC-tekniikka

NFC-teknologia liittyy merkittävästi Tagglem-ohjelmaan siten, että ohjelmalla voi luoda QR-koodeja ja NFC-tunnisteita eli se on ohjelman keskeisin toiminto.

NFC on tekniikka, joka mahdollistaa laitteiden, kuten älypuhelimien ja tablettitietokoneiden välisen kommunikaation siten, että laitteet toimivat lähietäisyydellä toisistaan ilman fyysistä kontaktia. Laitteet, joissa on NFC-tunniste tai ns. tagi voivat lähettää informaatiota toisilleen. NFC hyödyntää RFID-tekniikkaa (Radio Frequency Identification), joka on menetelmä tietojen käsittelyyn käyttäen RFID-tunnisteita. (About Near Field Communication. 2014.)

RFID-tekniikka kehitettiin toisen maailmansodan aikana, jolloin englantilaiset käyttivät RFID-laitteita erottamaan omat koneet vihollisista, koska tutka pystyi havainnoimaan vain lentokoneen, mutta ei erottamaan sen tyyppiä. NFC:n suurin ero RFID-tekniikkaan on se, että NFC-laitte voi toimia sekä lukijana- että tunnisteena toisin kuin perinteiset RFID-laitteet. (RFID. Osa 1: Opas. Johdatus tekniikkaan. 2002. 11-12.)

Erona esimerkiksi Bluetooth- ja Wi-Fi-teknologiaan, jotka perustuvat radiotiedonsiirtoon on se, että NFC hyödyntää elektromagneettisia radiokenttiä. On olemassa kolme erityyppistä NFC-teknologiaa: Type A, Type B ja FeliCa. Kaikki NFC-teknologiat ovat samankaltaisia, mutta ne kommunikoivat hieman eri tavoin.

NFC-teknologiassa on olemassa standardit, joiden tulee täyttyä, jotta laitteet toimivat oikein keskenään ja näin taataan myös laitteiden yhteensopivuus tulevaisuudessa. On olemassa kaksi tärkeintä standardimäärittystä: ISO/IEC 14443 ja ISO/IEC 18000-3, joista ensimmäinen määrittää ID-kortit, jotka varastoivat tietoa, kuten NFC-tunnisteet ja jälkimmäinen määrittää RFID-kommunikaation, jota NFC-laitteet käyttävät. (About Near Field Communication. 2014.)

ISO/IEC 18000-3 on kansainvälinen standardi kaikille laitteille, jotka toimivat langattomasti 13.56MHz taajuudella ja käyttävät Type A tai Type B kortteja. Laitteiden pitää olla enintään neljän senttimetrin päästä toisistaan, jotta ne pystyvät käsittelemään tietoja. Standardit kertovat kuinka laitteen ja NFC-tunnisteen pitäisi kommunikoida keskenään. Laitetta voidaan kutsua ”kuulustelijaksi” ja NFC-tunnistetta ”tagiksi”. (About Near Field Communication. 2014.)

NFC-laitteet voivat olla aktiivisia tai passiivisia. Esimerkiksi NFC-tunniste on passiivinen laite, joka sisältää tietoja, jonka toinen laite voi lukea, mutta se ei itse pysty lukemaan tai käsittelemään tietoa. Aktiivinen NFC-laite, kuten älypuhelin pystyy lukemaan tietoja NFC-tunnisteesta ja lähettämään tietoja muiden yhteensopivien laitteiden välillä ja muuttamaan tietoja, jos NFC-tunniste on määritetty sallimaan muutosten tekemisen.

NFC yhteys toimii siten, että laite lähettää signaalin tunnistelle ja jos laitteet ovat tarpeeksi lähellä, niin tunniste vastaanottaa ”kuulustelijan” signaalin, joka käynnistää tunnisteen ja mahdollistaa sen toimimisen ilman omaa virtalähdettä. Laitteet yhdessä luovat korkean taajuuden magneetikentän välilleen, jolloin yhteys on saavutettu ja informaatiota voidaan käsitellä laitteiden välillä.

Ensimmäiseksi ”kuulustelija” lähettää viestin tunnistelle selvittääkseen, minkä tyyppistä viestintää tunniste käyttää, kuten Type A tai Type B. Kun tunniste on vastannut, niin ”kuulustelija” osaa lähettää tietoja oikeilla määrityksillä. Tämän jälkeen, tunniste vastaanottaa ohjeet laitteelta ja tarkistaa onko tiedot oikein. Jos ne eivät ole, niin mitään ei tapahdu, mutta jos ne on, niin tunniste vastaa pyydetyllä tiedoilla. (About Near Field Communication. 2014.)

NFC-teknologiassa turvallisuus on keskeinen kysymys ja NFC-tekniikassa käytetään turvattuja kanavia ja salausmenetelmiä, kun käsitellään arkaluontoisia tietoja, kuten pankkikortin numeroita.

FeliCa (Felicity Card) on RFID-teknologiaan perustuva maksutapa, joka on yleinen Japanissa. Sen pääasiallisena käyttötarkoituksena on toimia maksutapana esimerkiksi kaupoissa ja metroissa. FeliCa kortit ovat passiivisia laitteita, eli niillä ei ole omaa virtalähdettä. Kortinlukija, joka käsittelee FeliCa-kortteja luo signaalin, joka käynnistää ja lukee informaatiota siitä. Tällöin käytetään salausmenetelmiä.

Kortin pitää olla enintään kymmenen senttimetrin päästä lukijasta toimiakseen, mikä on vähän enemmän kuin NFC-laitteissa. Tulevaisuudessa uskotaan, että NFC-laitteiden ja FeliCan yhteensopivuus tulee vielä parantumaan ja siitä uskotaan tulevan kosketusyhteydettömän maksamisen globaali standardi.

NFC-teknologiassa FeliCa:lla on merkittävä rooli, koska teknologioiden pitää olla yhteensopivia ja koska FeliCa on toimiva esimerkki siitä, miten suosittu kosketusyhteydettömän maksa-

misen järjestelmä toimii, sillä NFC-teknologia ei ole vielä yltänyt saman mittaluokan suosioon. (About Near Field Communication. 2014.)

NFC-tekniiikan ja Tagglem-ohjelman edelleen kehittyessä Tagglemilla on potentiaalia tarttua teknologian tuomiin uusiin mahdollisuuksiin ja kehittää keinoja ratkaista kuluttajille tärkeitä arkipäivän käytäntöjä. Esimerkiksi nykyaikainen pesukone tai auto on mahdollista käynnistää NFC-laitteella ja samoin myös kaupan liiketoiminnassa ja käytännöissä on monia erilaisia käyttömahdollisuuksia. (Kuva 5.)



Kuva 5. Esimerkki NFC-teknologian käyttömahdollisuuksista (Global Tag. 2014.)

## 2.4 AngularJS

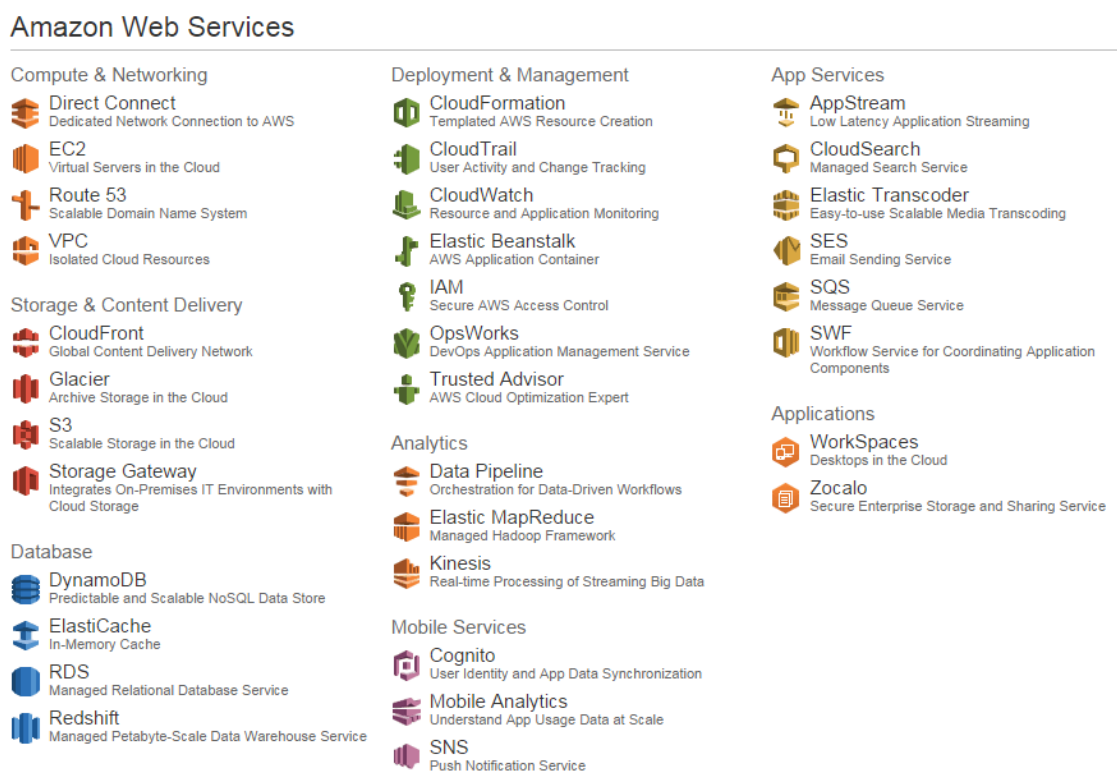
AngularJS on avoimen lähdekoodin rajapinta, jota kehittää Google. Se mahdollistaa yhden sivun sovellusten kehittämisen, jotka koostuvat HTML:stä, CSS:stä ja Javascriptistä. Se mahdollistaa käyttäjäpuolen MVC-mallin (Model-View-Controller), joka toimii siten, että Angu-



larJS lukee HTML-sivulta sille määritetyn tagi-attribuutin ja sitoo sen omaan malliinsa, jonka jälkeen niitä voidaan dynaamisesti käsitellä JSON-resursseina. AngularJS sopii parhaiten ”yhden sivun” www-sovelluksille. Tagglen-ohjelmassa merkittävä osa toiminnallisuudesta on ohjelmoitu AngularJS:llä. (AngularJS by Google.)

## 2.5 Amazon AWS

Amazon Web Services on pilvipalvelinjärjestelmä, jossa voidaan julkaista verkkosovelluksia palvelimille. AWS mahdollistaa verkkosovellusten hostaamisen pilvipalvelimelta. AWS:n keskeisimpinä palveluina ovat Amazon EC2 ja Amazon S3. EC2 mahdollistaa käyttäjien vuokrata virtuaalipalvelimen ja suorittaa siinä omia verkkosovelluksiaan ja S3 mahdollistaa tietojen tallentamisen pilvipalveluun. Tagglen-ohjelma hyödyntää Amazonin pilvipalvelinjärjestelmää, jossa se pyörii. (Kuva 3.)



Kuva 3. Amazon Web Services -palvelut (Amazon Web Services.)

## 2.6 Java

Java on olio-ohjelmointikieli, jonka on kehittänyt Sun Microsystems vuonna 1995. Java on yksi maailman eniten käytetyistä ohjelmointikielistä. Java on myös itsessään ohjelma, joka sisältää JRE-ympäristön (engl. Java Runtime Environment). JRE koostuu Javan virtuaalikoneesta JVM sekä Javan luokista ja kirjastoista. JVM mahdollistaa Java-sovellusten ajamisen eri käyttöjärjestelmillä, koska kaikki Java-ohjelmat suoritetaan sen avulla. (What is Java technology and why do I need it?)

Tagglem-ohjelma on kehitetty suurimmaksi osaksi Javalla, sen Enterprise Edition-version avulla. Java Enterprise Edition tarjoaa ohjelmointirajapinnan, jota käytetään verkkopalveluiden kehittämisessä.

## 2.7 Scrum

Scrum on projektinhallintamenetelmä, jota käytetään ketterässä ohjelmistokehityksessä. Scrumissa sovelluskehitys tapahtuu osissa, joita kutsutaan sprinteiksi. Sovelluksen kehittäminen pala kerrallaan lisää tuotettavuutta ja mahdollistaa mukautumisen muutoksiin palautteen mukaan. Voi siis ajatella, että Scrum on kokoelma sääntöjä, joiden avulla projekti voidaan toteuttaa ketterästi. (What is Scrum? 2014.)

Scrum pitää sisällään erilaisia rooleja, jotka ovat tärkeässä osassa menetelmää. Tämän opinnäytetyön kehitysprojektissa toimeksiantajan rooli oli ”Product Owner” ja opinnäytetyön tekijä oli jaetussa ”Scrum Masterin” ja ainoan kehitysryhmän jäsenen roolissa. Scrumissa ”Product Owner” määrittää mitä tehtäviä sprintit sisältävät. Kehitysryhmän jäsenet toteuttavat nuo tehtävät ja ”Scrum Master” huolehtii, että projekti etenee suunnitelman mukaisesti ja auttaa kehitysryhmää projektin aikana. (What is Scrum? 2014.)

## 2.8 Twitter Bootstrap

Bootstrap on käyttöliittymäraja-alue, joka nopeuttaa ja helpottaa verkkosovellusten ulkoasujen toteuttamista. Bootstrapin tärkeimpänä etuna on se, että se tukee automaattisesti responsiivisuutta eli sivustojen skaalautuvuutta riippuen siitä, millä laitteella sovellusta käytetään.

Bootstrap koostuu kokoelmasta CSS-tyylitiedostoja ja Javascript-komponentteja. Bootstrap tarjoaa yhdenmukaiset ja standardoidut apuvälineet nettisivujen ja web-aplikaatioiden luomiseen. Bootstrapin on kehittänyt Mark Otto ja Jacob Thorton vuonna 2011. (Bootstrap. Getting started.)

Tagglem-ohjelmassa Bootstrapilla on keskeinen osa käyttöliittymässä ja sitä käytetään muun muassa sen responsiivisten ominaisuuksien vuoksi, jotta ohjelmaa voisi käyttää paremmin myös älypuhelimilla ja tablettitietokoneilla. Bootstrap-rajapintaa käyttämällä responsiivisuutta ei tarvitse siis erikseen ohjelmoida sovellukseen, joten se nopeuttaa ja helpottaa käyttöliittymäohjelmointia huomattavasti.

Bootstrapin suurimpana ongelmana on, että sitä käyttävät sivut päätyvät helposti hyvin samankaltaisiksi, jos käyttöliittymäelementtejä ei muokata persoonallisiksi ja muista erottuviksi.

## 3 Tulokset

Tässä luvussa kerrotaan yksityiskohtaisemmin opinnäytetyön lopputuloksista ja Tagglem-ohjelman jatkokehitysmahdollisuuksista. Tarkoituksena on esitellä kehitysoisuuden saavutuksia ja pohtia Tagglem-ohjelman jatkokehitysmahdollisuuksia.

### 3.1 Lopputulos

Opinnäytetyön kehitysoisuuden lopputuloksena syntyi ohjelman konekielinen versio ja lähdekoodi, joka palautettiin toimeksiantajan ylläpitämään Git-versionhallintajärjestelmään. Kehitysoisuuden jaksojen aikana työtehtävät sovittiin toimeksiantajan kanssa jaksokohtaisesti ja ne merkittiin tuotteen kehitysjonoon (engl. Product Backlog) Scrum-projektinhallintamenetelmän käytäntöjen mukaisesti. (Liite 1.)

Aluksi keskityttiin parantamaan ohjelman ulkoasua, joka toteutettiin Bootstrapilla ja CSS:llä. Ensimmäinen tehtävä oli korjata ohjelman päävalikkoa siten, että entisen ratkaisun tilalle tehtiin ”alaszeto”-valikko. Tarkoituksena oli yksinkertaistaa päävalikon näkymää käyttäjille.

Toiseksi korjattiin sovelluksen raportointi-näkymää. Tarkoitus oli jälleen yksinkertaistaa näkymää ja poistaa turhia tietoja. Ajatuksena oli miettiä käyttäjälle selkeämpiä tapoja esittää informaatio - muun muassa kampanjan nimi siirrettiin käytettävyyden kannalta parempaan kohtaan.

Tehtävänä oli myös suunnitella uudelleenohjaustoimintaa Tagglem-ohjelman ja Amazon Web Services -palveluiden välillä. Suunnittelu toteutettiin prosessikaaviona. Suunnitelmasta tuli toimeksiantajan mielestä riittävän tarkoituksenmukainen kuvaus halutusta toiminnasta. (Liite 2.)

Tämän jälkeen korjattiin käyttöliittymässä olleita ulkoasuun liittyviä epäkohtia parantaen raportointi-näkymän ulkoasua, käytettävyyttä ja ominaisuuksia. Sovelluskehitysoisuuden loppuaika käytettiin ohjelman konfigurointiin Amazonin pilvipalvelimelle sekä bugien eli virheiden korjauksiin. Korjausten jälkeen käyttäjälle avautui mahdollisuus päästä suoraan päävalikosta itse muokkaamaan omia tietojaan.

### 3.2 Jatkokehitys

NFC-teknologia mahdollistaisi käytännössä sen, että käyttäjät voisivat automaattisesti kirjautua Tagglen-ohjelmaan ja hallita sitä asettamalla mobiililaitte lähietäisyydelle esimerkiksi Tagglen-ohjelman sisältävää kannettavaa tietokonetta. Täten laitteet tunnistaisivat toinen toisensa, eli niillä pitäisi olla jokin yhteinen tunniste.

Tagglen-ohjelmassa nykyisiä ominaisuuksia voisi kehittää pidemmälle. Myös lisäominaisuuksia voisi harkita kehitettävän. Esimerkiksi asiakkaan tilaushistoria olisi ominaisuus, jonka voisi lisätä ohjelmaan ja se olisi hyödyllinen asiakkaan näkökulmasta.

Käyttäjien kuunteleminen tämänkaltaisessa ohjelmassa ja sen kehittämisessä on erityisen tärkeää. Kehittyjä ominaisuuksia voisi testata tietyllä testiryhmällä ja pyytää testaaajien palautetta, niin ohjelman kokonaislaatu saattaisi parantua merkittävästi. On kuitenkin otettava huomioon, että sovelluksen tämänhetkinen versio 0.5.2 ei ole vielä täysin valmis eli versiossa 1.0.0, joten kehitystyö vielä jatkuu ja lopullinen ohjelma voikin näyttää aivan toisenlaiselta.

Tagglen-ohjelma sisältää paljon potentiaalisia kehittämismahdollisuuksia ja NFC-teknologiaa hyödyntämälle se mahdollistaisi saavuttaa aivan uudentyyppisiä käyttötarkoituksia, koska NFC on vielä kehittyvä teknologia. Esimerkiksi mobiililaitteilla maksaminen tulee yleistymään tulevaisuudessa ja tämän ominaisuuden integrointi Tagglen-ohjelmaan toisi uusia mahdollisuuksia sekä yrityksille että käyttäjille.

## 4 Johtopäätökset

Tämä luku käsittelee opinnäytetyössä syntyneitä johtopäätöksiä. Tarkoituksena on pohtia opinnäytetyötä kokonaisuutena sekä saavutettuja hyötyjä opinnäytetyön tekijän ja toimeksiantajan näkökulmista.

### 4.1 Hyödyt toimeksiantajalle

Merkittävimpänä hyötynä opinnäytetyöstä toimeksiantajalle oli se, että he saivat selkeämmän käyttöliittymän Tagglem-ohjelmaan. Tämä opinnäytetyö myös tarjoaa paljon tietoa käytettävyydestä, pilvipalveluista ja NFC-teknologiasta, jota toimeksiantaja voi mahdollisesti hyödyntää omassa liiketoiminnassaan.

Kokonaisuutena sovelluskehitys onnistui tyydyttävällä tavalla. Loppuvaiheessa ilmaantui vaikeuksia käyttöönotto-palvelimen konfiguroinnin kanssa, jotka jäivät ratkaisematta. Kuitenkin käytettävyyttä saatiin hieman paranneltua ja myös toimeksiantaja oli tyytyväinen paranneltuun käytettävyyteen ja ulkoasuun.

### 4.2 Tekijän omat kokemukset ja oppiminen

Kokemuksena opinnäytetyön sovelluskehitysprojekti oli haastava ja opettavainen. Alussa oli ongelmia saada kehitysympäristö toimimaan, jonka vuoksi tehtäviä ei päässyt heti aloittamaan. Lopulta kuitenkin kehitysympäristö saatiin toimimaan ja tämän jälkeen työskentely sujuikin hyvin.

Suurimpina haasteina oli se, ettei AngularJS ei ollut ennestään tuttu rajapinta, joten sen toimintalogiikan ymmärtäminen vei runsaasti aikaa. Ongelmana oli, että Javascript-tiedostoon tehtyjä muutoksia ei saatu toimimaan. Tämän johdosta raportointinäkyvä ei täysin vastannut haluttua muotoa.

Tagglem-ohjelman jatkokehittäminen oli haastavaa, koska se on monimutkainen ohjelma, joka oli alun perin suunniteltu toiseen tarkoitukseen ja ohjelmaan on jäänyt paljon vanhaa koodia, jota ei käytetty ohjelmassa. Tagglem-ohjelma on rakennettu monista tekniikoista, jotka eivät olleet opinnäytetyön tekijälle tuttuja ja siitä johtuen koodin ja ohjelman logiikan ymmärtäminen oli haasteellista. Oppimiskokemuksena kehitystyöprojekti oli kuitenkin mielenkiintoinen ja

opettavainen. Toimeksiantajalta sai aina hyvää palautetta ja tukea työn edetessä ja kommunikointi toimi hyvin molempiin suuntiin.

Oppimiskokemus opinnäytetyöstä oli, että koulutus ei täysin vastannut kaikkiin haasteisiin, koska kaikki työssä tarvittavat tekniikat eivät olleet etukäteen tuttuja. Tietotekninen ala on luonteeltaan nopeasti kehittyvää ja jatkuvasti uudistuva ala, joka vaatii tekijältään alan seuraamista ja itseopiskelua.

## Lähteet

About Near Field Communication. Luettavissa:

<http://www.nearfieldcommunication.org/>. Luettu 22.10.2014

AngularJS by Google. Luettavissa:

<https://angularjs.org/>. Luettu 31.10.2014

Amazon Web Services. Luettavissa:

<https://aws.amazon.com/>. Luettu 31.10.2014

Bootstrap. Getting started. Luettavissa:

<http://getbootstrap.com/getting-started/>. Luettu 22.10.2014

Global Tag. 2014. Luettavissa:

[http://www.global-tag.com/wp-content/uploads/2013/01/nfc\\_technology-1024x715.png](http://www.global-tag.com/wp-content/uploads/2013/01/nfc_technology-1024x715.png).

Luettu 31.10.2014

JavaTpoint. 2014. Luettavissa:

<http://www.javatpoint.com/cloudpages/images/cloud.jpg>. Luettu 31.10.2014

Nielsen, J. 2000. WWW-suunnittelu. Gummerus. Jyväskylä.

Parkkinen, J. 2002. Hyvään verkkopalveluun! Tammer-Paino Oy. Tampere.

SFS-käsikirja 301-1. RFID. Osa 1: Opas. Johdatus tekniikkaan, 2010. SFS. Helsinki.

Usability, Learnability, Memorability. 2009. Luettavissa:

<http://ed-informatics.org/2009/12/28/medical-computing-1/>. Luettu 31.10.2014

What is Cloud Computing? Luettavissa:

[https://aws.amazon.com/what-is-cloud-computing/?nc1=f\\_cc](https://aws.amazon.com/what-is-cloud-computing/?nc1=f_cc) . Luettu 22.10.2014

What is Java technology and why do I need it? Luettavissa:

[https://www.java.com/en/download/faq/whatis\\_java.xml](https://www.java.com/en/download/faq/whatis_java.xml). Luettu 31.10.2014



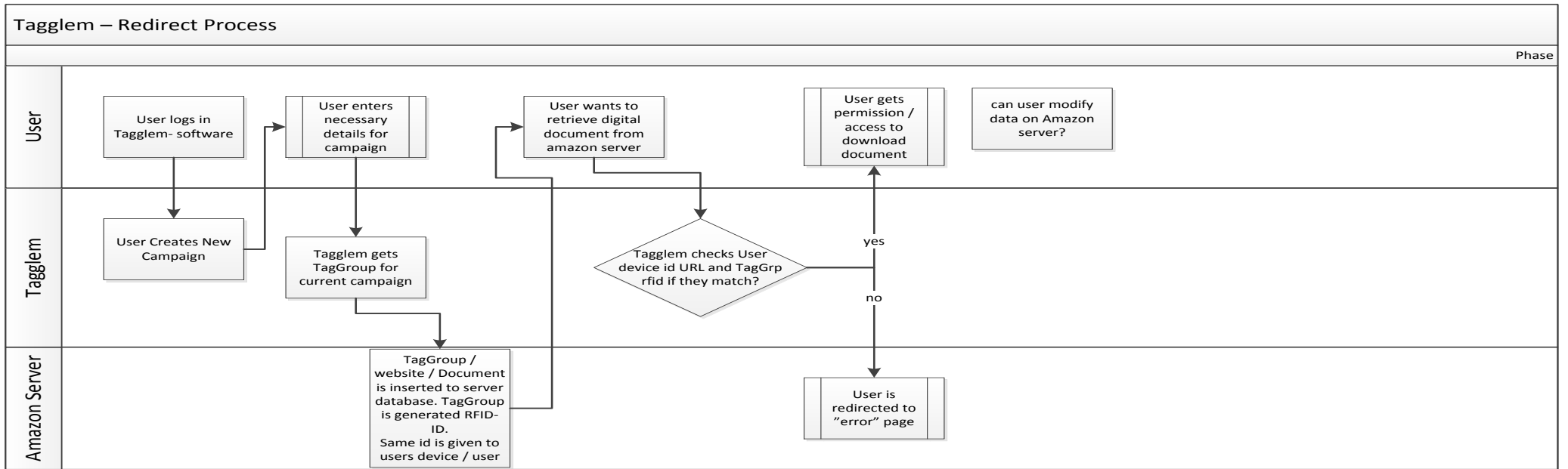
What is Scrum? Luettavissa:

<https://www.scrum.org/Resources/What-is-Scrum>. Luettu 31.10.2014

## Liitteet

ID	TASK	SPRINT	STATUS	ESTIMATED TIME (h)	TIME SPENT (h)
1	<b>Siisti ulkoasu</b>	1	done	12	13
1.1..	Poista turhat headerit	1	done	2	2
1.2..	Korjaa fonttien koot	1	done	3	2
1.3..	Korjaa napit campaign assets -kohdassa	1	done	3	5
1.4..	Poista turha laatikko campaign details -kohdasta	1	done	1	1
1.5..	Poista turha sivuvalikko	1	done	3	3
2	<b>Korjaa Reporting -näkyvä</b>	3	ongoing	56	46
2.1..	Report from -kohtaankampanjan aloituspvm	3	done	4	6
2.2..	Report to -kohtaankampanjan lopetuspvm	3	done	4	6
2.3..	Tutustu saako custom aikaan by time of day, haluaako jakaa päivän 3 vai neljään osaan ja sitten kellonajat	3	ongoing	8	3
2.4..	Valitse defaulttina joku (uusin) raportti kampanjasta	3	ongoing	5	6
2.5..	tarkista /korjaa raportoinnin terminologiaa matchaa asset ja locationin kanssa	3	ongoing	2	2
2.6..	raportoinnin options-sivuvalikko siirrettävä vaakatasoon raportoinnin päälle	3	done	4	3
2.7..	tarviiko refresh-nappulaa?	3	done	1	1
2.8..	Raportin nimi pitää saada borderheaderiin	3	done	3	3
2.9..	Poistettava by taggroup -raportinäkymä	3	done	4	3
2.10..	Fix Grouping, By timeof day: asset name, asset location, asset type	3	ongoing	3	3
2.11..	Piirakkakaavioon by type: asset type grouping	3	ongoing	3	2
2.12..	By timeunit. day = 30 palkkia, Week: 30 palkkia ja month = 12 palkkia	3	ongoing	6	2
2.13..	Stacked nappula ei kovin kuvaava, ehkä collapse ja se voisi olla sivumassa,eikä ikonia tarvitse	3	ongoing	3	3
2.14..	checkkaa että muutokset toimivat excel-raportissa.	3	done	2	2
2.15..	Mieti piirakkakaavion sijaintia paremmaksi	3	ongoing	4	1
2.16..					
3	<b>UI</b>	2	ongoing	22	22
3.1..	Campaign history ja new campaign sublinkeiksi menuun	2	done	7	8
3.2..	Kirjautunut käyttäjä Yläbar-menuun	2	done	7	4
3.3..	Editoimaan kirjautuneen käyttäjän tietotoja menu linkistä	2	done	4	5
3.4..	user management sivuvalikko poistettava, jos turha	2	done	4	5
4	<b>NGinx ja tomcat pystyyn amazonin palvelimelle</b>	4 - 6	ongoing	91	15
4.1..	staattinen html log in sivu () josta kirjautuminen tomcatiin, tagglemin kirjautumissivulle	4 - 6	done	15	2
4.2..	RDS tietokantapalvelu käyttöön, siirrä mysql-data RDS-kantaan	4 - 6	Rejected	15	
4.3..	konffaa NGinx että osaisi mennä tomcat oikeaan instanssiin	4 - 6	ongoing	15	12
4.4..	Tee minified versio tagglem.js, katso toimiiko muutokset	4 - 6	ongoing	10	
4.5..	Tee public key ja lähetä lsmolle emaililla, ja tee private-key,jotta saa otettua yhdeyden ec-2:n puttylla	4 - 6	done	1	1
4.6..	Mieti miten saisi rds kannan palvelemaan useita instansseja /asiakkaita	4 - 6	ongoing	15	
4.7..	rds tietokanta voisi lukea tageja ERP (oogway resurssihallintajärjestelmästä)	4 - 6	Rejected	20	

## Liite 1. Product Backlog



**Java Classes needed:**

- Class bean for user device
- Class bean for RFID tag
- Class service for Redirecting
- Class for adding permission to TagGrp and user device
- Class for reading data from Amazon server and displaying it in Tagglem

**DB Tables needed:**

- Table for devices and users
- Table for device id and tagGrp rfid

**REST HTTP methods for URL:**

GET, PUT, POST, DELETE

**What database?**

Amazon uses: DynamoDB, Amazon relational database Service(RDS), Amazon simpleDB

**Which type of Amazon server.**

Windows, linux, 32 or 64 bit. Do I need to register account?

Virtual instance for development?

Liite 2. Tagglem-ohjelman uudelleenohjaussuunnitelma

Liite 3. Tagglen-ohjelman lähdekoodi (salassa pidettävä)