SELAINPOHJAISEN REKISTERIN RAKENTAMINEN PIENELLE YHDISTYKSELLE

Lappeenrannan–Lahden teknillinen yliopisto LUT

Tietotekniikan kandidaatintyö

2022

Nelli Kemi

Tarkastaja: TkT Erno Vanhala

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT

LUT Teknis-luonnontieteellinen

Tietotekniikan koulutusohjelma

Nelli Kemi

Selainpohjaisen rekisterin rakentaminen pienelle yhdistykselle

Tietotekniikan kandidaatintyö

33 sivua, 4 kuvaa ja 1 taulukko

Tarkastaja: TkT Erno Vanhala

Avainsanat: rekisteri, tietokannat, sisällönhallintajärjestelmät, pieni yhdistys, ohjelmistotuotanto, case-tutkimus, tapaustutkimus

Nykyään tietoja kerätään paljon erilaisiin tarkoituksiin. Kerääminen vaatii myös järjestelmiä, joihin tiedot voidaan tallentaa niiden myöhempää hyödyntämistä varten. Tällaisia järjestelmiä on monenlaisia, joista yksi yleisesti käytössä oleva järjestelmätyyppi on rekisterit.

Tämän kandidaatintyön tavoitteena on tutkia vaihtoehtoja verkkopohjaisen rekisterin toteuttamiseksi pienelle yhdistykselle. Tutkimusmenetelmänä käytetään case-tutkimusta. Työssä perehdytään erilaisiin työvälineisiin ja olemassa oleviin järjestelmiin rekisterin toteutusta varten. Tutkittavana rekisterinä on Suomen Siiliyhdistyksen siilirekisteri.

Siilirekisterille laadittiin ensin vaatimusmäärittely, minkä jälkeen rekisterin toteutustavaksi valittiin WordPress-sisällönhallintajärjestelmä, jota hyödyntäen rekisteri toteutettiin yhdistyksen verkkosivujen yhteyteen. Toteutuksella pystyttiin vastaamaan kaikkiin Suomen Siiliyhdistyksen projektin alussa esittämiin vaatimuksiin.

Tutkimuksen perusteella pienen rekisterin toteuttaminen sisällönhallintajärjestelmän avulla on mahdollista, mutta lisätutkimusta kaivataan esimerkiksi tietoturvaan ja omien lisäosien ylläpidettävyyteen liittyen. Vaikka sisällönhallintajärjestelmät todettiin hyväksi vaihtoehdoksi, tulisi tutkimustyötä tehdä myös muihin toteutusvaihtoehtoihin liittyen.

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

School of Engineering Science

Software Engineering

Nelli Kemi

Development of a web-based registry for a small association

Bachelor's thesis

2022

33 pages, 4 figures and 1 table

Examiner: D.Sc. (Tech.) Erno Vanhala

Keywords: registry, databases, content management systems, small association, software development, case study

Collecting information for various purposes is nowadays a widespread practice. This practice requires systems where the information can be stored. There is a multitude of systems available, and one category of such systems is different types of registries.

The purpose of this bachelor's thesis is to investigate the different possibilities for developing a web-based registry for a small organization. The research is conducted as a case study. Different tools and existing systems are evaluated for the development process of a registry. The case registry of this thesis is the hedgehog registry of Suomen Siiliyhdistys (the Finnish Hedgehog association).

A software requirements specification was conducted for the case association's new registry, after which the content management system WordPress was chosen as the development tool. With WordPress the registry could be included within the association's pre-existing website. The final product fulfils the requirements and functionalities specified by the association at the beginning of the process.

Based on this study developing a small registry using a content management system is possible but further research would be needed, for example, regarding the cyber security of the system and the maintenance of private plugins. Although content management systems are found to be a valid option, research should be done on other development tools and techniques as well.

KIITOKSET

Haluan kiittää Suomen Siiliyhdistystä mahdollisuudesta tämän aiheen tutkimiseen sekä yhteistyöstä ja luottamuksesta uuden rekisterin kehitystyön aikana.

Suuret kiitokset myös ohjaajalleni Erno Vanhalalle avusta, tuesta ja ohjauksesta työn edetessä.

LYHENNELUETTELO

AWS	Amazon Web Services
CSV	Comma-separated values. Pilkulla erotetut arvot.
HTML	Hypertext Markup Language. Hypertekstin merkintäkieli.
MySQL	MySQL-relaatiotietokantaohjelmisto.
NoSQL	Not only SQL. Ei-SQL-kyselykieli.
OLAP	Online Analytical Processing. Reaaliaikainen analyyttinen prosessi.
PHP	PHP: Hypertext Preprocessor. PHP: Hypertekstin esikääntäjä.
SQL	Structured Query Language. Standardoitu kyselykieli.
WHS	Wobbly Hedgehog Syndrome. Siileillä esiintyvä sairaus.

Sisällysluettelo

Tiivistelmä

Abstract

Kiitokset

Lyhenneluettelo

1 Johdanto ... 7
1.1 Tausta ... 7
1.2 Tavoitteet ja rajaukset ... 8
1.3 Työn rakenne ... 9
2 Tutkimusmenetelmä ... 10
3 Verkkopohjaisen rekisterin toteutusmahdollisuuksia ... 11
3.1 Pilvipalvelut ... 11
3.2 Rekisterin toteuttamisessa avustavat teknologiat ... 12
3.2.1 Tietokannat rekisterin taustalla ... 12
3.2.2 Avoimen lähdekoodin ohjelmistot ... 13
3.2.3 Reaaliaikainen analyyttinen prosessointi ... 14
3.2.4 Sisällönhallintajärjestelmät ... 14
4 Case: Suomen Siiliyhdistyksen siilirekisteri ... 16
4.1 Siilirekisterin vaatimusmäärittely ... 16
4.2 Olemassa olevien järjestelmien kartoitus ... 17
4.3 Toteutustavan valinta ... 18
4.4 Siilirekisterin toteutus ... 19
4.4.1 Tietokantataulujen suunnittelu ... 19
4.4.2 Siilirekisterin tekninen toteutus ... 21
4.4.3 Rekisterin käyttöönoton valmisteleminen ... 25
5 Keskustelu ... 26
6 Yhteenveto ... 29
Lähteet ... 31

1 Johdanto

Tietojen kerääminen erilaisiin tarkoituksiin on nykyään erittäin yleistä. Esimerkiksi yliopistot voivat kerätä opiskelijoidensa henkilötietoja opiskelijoiden opiskelun, opintojen sekä valmistumisen mahdollistamiseksi (LUT 2021). Pelkkä tietojen kerääminen ei kuitenkaan ole hyödyllistä, vaan tarvitaan myös tapoja säilyttää tiedot, jotta niiden käyttäminen myöhemmin on mahdollista. Tätä varten on kehitetty useita yleisesti käytössä olevia järjestelmiä kuten erilaisia rekistereitä. Rekisterien avulla rekisterinpitäjät pystyvät hallitsemaan tarvitsemiaan tietoja sekä hyödyntämään niitä toiminnassaan (Euroopan komissio n.d.). Yliopistojen tapauksessa opiskelijoiden henkilötietoja voidaan säilyttää rekistereissä.

Rekisterien kehittämiseen ja ylläpitoon on olemassa erilaisia valmiita ja puolivalmiita järjestelmiä, mutta riittävällä tietämyksellä rekisterin rakentaminen on mahdollista myös täysin itsenäisesti. Tässä työssä suunnitellaan ja toteutetaan uusi rekisteri Suomen Siiliyhdistykselle.

1.1 Tausta

Suomen Siiliyhdistys on vuonna 2002 perustettu voittoa tavoittelematon yhdistys. Yhdistys on tarkoitettu lemmikkisiiliharrastajille, ja sen tavoitteena on jakaa oikeaa tietoa lemmikkisiileistä sekä tarjota vertaistukea virallisille kasvattajille. Myös siilien jalostuksen valvominen on yksi yhdistyksen tärkeimmistä tehtävistä, ja tässä auttaa heidän ylläpitämänsä siilirekisteri. (Suomen Siiliyhdistys ry n.d.-a.) Rekisteriin on kerätty tietoa muun muassa siilien sukupuolesta, terveydestä sekä eliniästä. Kukaan Siiliyhdistyksen nykyisistä aktiiveista ei kuitenkaan tiedä, miten kyseinen rekisteri on toteutettu, mistä johtuen sen ylläpidosta on tullut mahdotonta. Rekisteri on myös niin vaikeakäyttöinen, että se ei ole enää moneen vuoteen ollut käytössä. Tämän sijaan on hyödynnetty Google Sheets -ohjelmaa, johon on yhteen dokumenttiin kerätty tiedot siileistä.

Nykyisen toteutuksen ongelma on sen rajallinen toiminnallisuus. Dokumentista on vaikea löytää haluttua informaatiota, eikä se toimi kuin siililistan selaamiseen. Usean vuoden jälkeen tiedostoon on kertynyt niin paljon dataa, että sen koko on kasvanut todella suureksi ja

selaamisesta on tullut vaikeaa. Lisäksi niin yksittäisten tietojen kuin erilaisten kokonaisuuksien, kuten siilien sukuhistorian, selvittäminen tiedoston avulla on haastavaa. Tästä johtuen Suomen Siiliyhdistys haluaisi uuden, paremman rekisterin, jossa olisi enemmän toiminnallisuutta.

Vaikka erilaisia tiedonhallintajärjestelmiä on ollut olemassa jo kauan ja verkkopohjaisiakin järjestelmiä 1960-luvulta asti (Błażewicz, Kubiak, Morzy & Rusinkiewicz 2003, s. 3), ei rekistereille edelleenkään ole yleisiä toteutusohjeita tai -vaatimuksia (Forrest, Bartek, Rubinstein & Groft 2011). Lääketieteellisiä rekistereitä, niiden toteutusvaihtoehtoja sekä toteuttamiseen liittyviä ongelmia on tutkittu, mutta esimerkiksi eläinrekistereihin liittyen ei tutkimustyötä ole tehty. Rekistereistä tehty tutkimus ei myöskään ole keskittynyt toteutuksen teknisiin ominaisuuksiin.

1.2 Tavoitteet ja rajaukset

Tämän työn tavoitteena on luoda Suomen Siiliyhdistykselle uusi rekisteri, joka vastaa heidän tarpeitansa. Toteutusta varten pyritään kartoittamaan jo olemassa olevia järjestelmiä ja niiden ominaisuuksia, minkä jälkeen näistä valitaan parhaiten tarkoitukseen soveltuva vaihtoehto.

Työn tarkoituksena ei ole selvittää parasta vaihtoehtoa satunnaisen yhdistyksen rekisterin toteuttamiseen tai ylläpitämiseen eikä yleispäteviä ohjeita kaikenlaisten rekistereiden toteutukselle. Tällainen olisi käytännössä mahdoton toteuttaa, koska erilaiset rekisterit voivat vaatia hyvin erilaisia ominaisuuksia ja mahdollisia toteutusvaihtoehtoja on useita. Rekisteriä ei myöskään toteuteta useammalla järjestelmällä, vaan järjestelmien ominaisuuksien ja tarjoamien palveluiden perusteella valitaan parhaiten tämän rekisterin toteutukseen sopivalta vaikuttava vaihtoehto.

Työn tutkimuskysymykset ovat:

1) Millaisia järjestelmiä rekisterin toteuttamiseksi on olemassa?

2) Miten Suomen Siiliyhdistyksen siilirekisteri tulisi toteuttaa?

Kysymyksiin vastaamiseksi käytetään tutkimusmenetelmänä case-tutkimusta.

1.3 Työn rakenne

Toisessa luvussa perehdytään tarkemmin työssä käytettävään tutkimusmenetelmään. Kolmannessa luvussa tutustutaan tekniikoihin, joita voidaan hyödyntää rekisterin rakentamisessa. Tämän jälkeen neljännessä luvussa suoritetaan case-tutkimus: Ensin määritellään siilirekisterin vaatimukset Suomen Siiliyhdistyksen edustajien kanssa pidetyn tapaamisen perusteella ja sen jälkeen kartoitetaan olemassa olevia järjestelmiä sekä niiden tarjoamia mahdollisuuksia. Vaatimusmäärittelyä hyödyntäen valitaan näistä siilirekisterin toteutukseen parhaiten soveltuva vaihtoehto ja toteutetaan järjestelmä. Viidennessä luvussa pohditaan työn onnistumista sekä esitetään jatkotutkimusideoita.

2 Tutkimusmenetelmä

Tässä työssä tutkitaan mahdollisuuksia rakentaa rekisteri pienelle yhdistykselle. Tutkimuksessa hyödynnetään esimerkkinä Suomen Siiliyhdistyksen siilirekisteriä. Näin ollen tutkimusmenetelmänä voidaan käyttää case-tutkimusta eli tapaustutkimusta.

Case-tutkimuksen tarkoituksena on tutkia tapausta, joka voi olla käytännössä millainen tahansa nykyaikainen ilmiö todellisen elämän kontekstissaan (Yin 2018, s. 15). Tämä soveltuu hyvin siilirekisterin toteuttamiseen: Siilirekisteri on käytössä oleva järjestelmä, joka toimii esimerkkinä yleistyneestä tiedon keräämisestä sekä rekistereiden tarpeesta. Myös uusi siilirekisteri tulee olemaan Suomen Siiliyhdistyksen käytössä valmistuttuaan. Lisäksi tietotekniikan alalla case-tapaukset ovat usein ohjelmistojen kehitysprojekteja (Runeson & Höst 2009), kuten on siilirekisterin rakentaminenkin.

Kontrolloiduista tutkimuksista poiketen case-tutkimukset auttavat ymmärtämään tutkittua ilmiötä oikeassa kontekstissaan (Wohlin, Runeson, Höst, Ohlsson, Regnel & Wesslén 2012, s. 55). Erityisen hyvin case-tutkimuksilla pystytään vastaamaan ”miten” ja ”miksi” -alkuisiin kysymyksiin (Yin 2018, s. 9), kuten toiseen tämän työn tutkimuskysymykseen: Miten Suomen Siiliyhdistyksen siilirekisteri tulisi toteuttaa?

Case-tutkimuksille voidaan määrittää useita tarkoituksia. Tietoteknisten tutkimusten yleisin tarkoitus on kehittävä, ne pyrkivät kehittämään ja parantamaan jotain osaa tutkitusta ilmiöstä. Tässä työssä kehitetään rekisteri pienelle yhdistykselle sekä pyritään lisäämään ymmärrystä näiden kehitysmahdollisuuksista. Myös tutkiva tarkoitus toteutuu tässä työssä, sillä työssä kehitetään uusia näkemyksiä aiheeseen ja annetaan jatkotutkimusideoita. (Runeson & Höst 2009, s. 135.)

Case-tutkimukset pystytään toteuttamaan yhden tai useamman tapauksen tutkimuksina (Yin 2018, s. 17). Tässä tutkimuksessa käsitellään vain yhtä tapausta, Suomen Siiliyhdistyksen siilirekisteriä. Tutkimukset sisältävät usein myös piirteitä muista tutkimusmenetelmistä, varsinkin tutkimusta edeltävä kirjallisuuskatsaus on yleinen (Wohlin et al. 2012, s. 57). Seuraavassa luvussa perehdytään tämän työn osalta merkittävään kirjallisuuteen.

3 Verkkopohjaisen rekisterin toteutusmahdollisuuksia

Yleisesti rekisteri voidaan määritellä paikkana, jossa dataa tai laboratorionäytteitä säilytetään ja josta se on saatavilla tutkimustyötä varten (Humphries 2017). Tämä pätee erityisesti lääketieteellisiin rekistereihin, joihin myös aiheesta tehdyt tutkimukset ovat painottuneet, mutta perusidealtaan se soveltuu kaikenlaisiin rekistereihin: tarvitaan paikka, jossa dataa säilytetään ja josta se saadaan myöhemmin hyödynnettäväksi.

Lääketieteellisiin rekistereihin kohdistuneet tutkimukset ovat myös yleensä painottaneet tietoturvaa, rahoitusta tai lääketieteellistä näkökulmaa rekistereiden kehityksessä, eivät niinkään tutkineet toteutuksia tietoteknisten ominaisuuksien kannalta. Tässä luvussa perehdytään enemmän näihin tietoteknisiin ominaisuuksiin sekä selvitetään erilaisia tekniikoita ja työkaluja, joita rekisterien toteuttamisessa voidaan hyödyntää.

3.1 Pilvipalvelut

Yksi mahdollisuus rekisterin toteuttamiseen olisi jonkin valmiin pilvipalvelun hyödyntäminen. Pilvipalvelut perustuvat ohjelmistoihin ja dataan, jotka ovat käyttäjien saatavilla internetin välityksellä ja datankäsittelykeskuksien avulla. Palvelut eivät vaadi ohjelmistojen lataamista tietokoneille, ja ne ovatkin olleet yksi suosituimmista informaatioteknologian suuntauksista viime vuosina. (Safonov 2016, s. 8-9.) Pilvipalvelut ovat hyvä vaihtoehto yrityksille ja henkilöille, jotka eivät osaa tai halua kehittää omia järjestelmiään, vaan mieluummin ulkoistavat tämän työn ulkopuolisille palveluntarjoajille.

Pilvipalvelut ovat pääsääntöisesti maksullisia ja niistä peritään maksu joksikin ajanjaksoksi, esimerkiksi vuodeksi, kerrallaan (Safonov 2016, s. 10). Maksu sitouttaa käyttäjän palveluun tietyksi aikaa, mutta myös tarjoaa paljon etuja itse kehitettävään järjestelmään verrattuna. Kehittämistyön poisjäämisen lisäksi käyttäjät hyötyvät esimerkiksi palvelun ylläpidosta sekä ongelmien korjaamisesta, mikä voi palvelun hinnasta riippuen vähentää järjestelmän kokonaiskustannuksia.

3.2 Rekisterin toteuttamisessa avustavat teknologiat

Valmiiden pilvipalveluiden lisäksi rekistereitä pystytään toteuttamaan myös täysin itsenäisesti ilman valmiita työkaluja ja viitekehyksiä tai vaihtoehtoisesti niitä hyödyntäen. Varmimmin käyttäjien tarpeita ja toiveita vastaava rekisteri on mahdollista toteuttaa rakentamalla palvelu alusta asti itse. Samaan lopputulokseen voidaan kuitenkin päästä myös hyödyntämällä erilaisia kehitystyötä helpottavia tekniikoita ja palveluita, jotka mahdollistavat omien ominaisuuksien kehittämisen riittävissä määrin. Seuraavaksi perehdytään joihinkin tällaisiin ratkaisuihin, jotka voivat helpottaa rekisterin kehitystyötä.

3.2.1 Tietokannat rekisterin taustalla

Rekisterien vaatimuksiin kuuluu tietojen tallennus sekä tallennetun tiedon hyödyntäminen. Tähän tarkoitukseen soveltuvat hyvin tietokannat. Yksinkertaistettuna tietokanta on kokoelma toisiinsa liittyvää tietoa (Beaulieu 2020, s. 1). Tarkemmin sanottuna tietokanta on rakenteinen kokoelma tietoja, josta tiedot ovat jonkin ohjelman saavutettavissa (Berrington 2017).

Yleisimmin käytössä olevat relaatiotietokannat voivat sisältää yhden taulun tai useita tauluja, joihin tiedot tallennetaan. Taulut kuvastavat jonkinlaista datakokoelmaa, esimerkiksi asiakkaita tai siilejä, ja näiden tietoja. Tauluissa jokaisella rivillä on yksittäinen syöte tai data-arvo, jonka ominaisuuksien arvot löytyvät kyseisen rivin sarakkeista. Yksittäinen sarake kuvaa aina yhtä syötteen ominaisuutta. (Berrington 2017.) Jokaisella rivillä on myös oma ainutlaatuinen tunnus, jota kutsutaan pääavaimeksi (Beaulieu 2020, s. 6). Esimerkiksi taulussa, johon tallennetaan siilien nimi, syntymäaika ja paino, olisi yhdellä rivillä aina yhden siilin tiedot: yhdessä sarakkeessa pääavain, toisessa nimi, kolmannessa syntymäaika ja neljännessä paino. Seuraavalla rivillä olisi toisen siilin tiedot, kolmannella kolmannen siilin tiedot ja niin edelleen.

Tietokannoissa säilytettyyn tietoon päästään käsiksi hyödyntämällä siihen tarkoitettuja sovelluksia. Nämä sovellukset hyödyntävät erilaisia kieliä ja näille tyypillisiä komentoja. Yleinen relaatiotietokantojen kanssa käytettävä kieli on Standardoitu kyselykieli SQL (Structured Query Language), jonka ensimmäinen standardi julkaistiin jo yli 35 vuotta sit-

ten, vuonna 1986. SQL-kyselyiden tuloksena tietokannasta saadaan tauluja, joita kutsutaan tulosjoukoksi. Tietokannasta tiedon etsimisen lisäksi SQL:n avulla pystytään tekemään monenlaisia muutoksia tietokannan tauluihin, kuten luomaan tai poistamaan tauluja ja sarakkeita, lisäämään dataa tauluihin ja muokkaamaan tauluissa olevia tietoja. (Beaulieu 2020, s. 8-12.) Viimeisen kymmenen vuoden aikana on kehitetty myös useita ei-SQL-kieliä, joita voidaan hyödyntää relaatiotietokannoista poikkeavien tietokantatyyppien käsittelyssä (Wu 2021).

Myös laskentataulukoita voidaan osittain hyödyntää tietokantojen tapaan. Laskentataulukotkin mahdollistavat useiden taulukoiden luomisen (Zygiaris 2018, s. 11) sekä tietojen tallentamisen rivien ja sarakkeiden avulla samalla logiikalla kuin tietokantoihin. Laskentataulukoilla ei kuitenkaan ole kaikkia samoja ominaisuuksia kuin tietokannoilla, esimerkiksi rivien uniikki identifiointi ja taulukoiden väliset suhteet eivät toteudu laskentataulukoissa (Zygiaris 2018, s. 11).

3.2.2 Avoimen lähdekoodin ohjelmistot

Avoimen lähdekoodin ohjelmistojen tarkoituksena on, että niiden lähdekoodit ovat kaikkien saatavilla luettavassa muodossa. Sitä vastoin perinteisempien suljetun koodin ohjelmistojen lähdekoodit ovat vain ohjelmiston omistajien saatavilla. Avoin lähdekoodi ei kuitenkaan ole uusi ilmiö, sillä se sai alkunsa jo 1980-luvulla. Tuolloin yhdysvaltalainen ohjelmoija Richard Stallman totesi, että ohjelmien digitaalisen luonteen sekä koodin jakamisen matalien kulujen takia käyttäjillä tulisi olla vapaus käyttää, lukea, muokata ja jakaa koodia tarpeelliseksi katsomallaan tavalla. (Schweik & English 2012, s. 5.)

Lähdekoodin vapaasta jakamisesta etuna ohjelmiston käyttäjälle on ainakin hankintakulujen pois jääminen, kun ohjelmiston saa ladattua ilmaiseksi. Tästä on hyötyä erityisesti henkilöille ja yrityksille, joiden budjetti on rajallinen. Lisäksi avoimen lähdekoodin avulla on mahdollista saada kasaan iso joukko ihmisiä, jotka kehittävät, testaavat ja ylläpitävät ohjelmistoa myös tulevaisuudessa. (Schweik & English 2012, s. 5-6.) Ohjelmiston tarjoajalle tämä on hyödyllistä, koska ohjelmiston ylläpito helpottuu ohjelmiston kehittämisestä kiinnostuneiden henkilöiden ansiosta. Ohjelmiston käyttäjälle taas ohjelmiston jatkuva ylläpito on hyödyllistä mahdollisten ohjelmassa ilmenevien ongelmien kannalta sekä tietoturvaa ja

uusia ominaisuuksia ajatellen. Avoin lähdekoodi myös mahdollistaa ohjelmiston muokkaamisen halutulla tavalla lähdekoodia käsittelemällä ilman, että koko ohjelmistoa tarvitsee rakentaa alusta alkaen.

3.2.3 Reaaliaikainen analyyttinen prosessointi

Yksi teknologia, jota rekisterin rakentamisessa voidaan hyödyntää, on reaaliaikainen analyyttinen prosessi (Online Analytical Processing eli OLAP). Reaaliaikainen analyyttinen prosessi on moniulotteinen datan analysointiteknologia, joka mahdollistaa nopeiden analyysien tekemisen suuristakin määristä dataa. Teknologiaa hyödynnettäessä data organisoidaan indikaattoreiden ja analyysiakseleiden avulla. Tämä mahdollistaa erilaisten toimintojen, jotka muun muassa rajoittavat dataa kyselyissä, käyttämisen. Reaaliaikaisesta analyyttisestä prosessista onkin tullut hallitseva teknologia moniulotteisessa data-analyysissa sen mahdollistamien suuren datamäärän analyysien ansiosta. Teknologia on tarkoitettu käytettäväksi tilanteissa, joissa on tarve kerätä tai muokata dataa tai tehdä datasta yhteenvetoja analyyttisen informaation keräämiseksi. Tällaista informaatiota hyödynnetään usein esimerkiksi päätöksenteossa. (Orlando & Carolina 2019.)

Reaaliaikaista analyyttista prosessia hyödyntävät järjestelmät analysoivat dataa monimutkaisten kyselyiden avulla (Robert 2006, s. XI). Järjestelmät hyödyntävät tietokannoista jollain ajanhetkellä otettuja tilannekuvia, jotka sijoitetaan moniulotteiseen malliin. Tämän toimintatavan tarkoituksena on suorittaa kyselyitä datakokoelmille yksittäisten toimintosarjojen sijaan. (Celko 2006, s. 57.) Toimintamalli nopeuttaa kyselyiden toteuttamista, mistä johtuen se on käytännöllinen ratkaisu erityisesti suurien datakokonaisuuksien käsittelyyn. Näin ollen teknologia voi soveltua erityisen hyvin rekistereihin, joissa kerätyn datan määrä on todella suuri. Toisaalta sitä voi hyödyntää myös tilanteissa, joissa rekisteriin tallennettuja tietoja tulee analysoida usein erilaisia tarkoituksia varten.

3.2.4 Sisällönhallintajärjestelmät

Sisällönhallintajärjestelmät mahdollistavat useiden erilaisten verkkosivujen ja -sisältöjen tuottamisen, joten ne soveltuvat myös verkkopohjaisten rekisterien toteuttamiseen. Sisäl-

lönhallintajärjestelmien avulla on mahdollista käsitellä suuria määriä informaatiota ilman, että kaikkea tietoa tarvitsee käsin kirjoittaa jokaisen yksittäisen sivun HTML-koodiin (Hypertext Markup Language) (Seadle 2006). Suurin osa niin kaupallisista kuin ei-kaupallisistakin sisällönhallintajärjestelmistä on myös aiemmin esiteltyjä avoimen lähdekoodin ohjelmistoja (Maass 2012, s. 5).

Jopa puolet nykypäivän verkkosivustoista on luotu sisällönhallintajärjestelmiä hyödyntäen. Syynä voidaan pitää järjestelmien suunnittelemista niin, että niitä voidaan tarjota suurelle yleisölle, jolla ei välttämättä ole ohjelmointikokemusta tai -tietämystä. Lisäksi sisällönhallintajärjestelmät tarjoavat paljon toiminnallisuutta, josta kehittäjät voivat oman tietämyksensä perusteella hyötyä eri tasoilla. Avoimen lähdekoodin sisällönhallintajärjestelmät pohjautuvat yleensä PHP-ohjelmointikieleen (PHP: Hypertext Preprocessing), ja niillä on laajat käyttäjäyhteisöt, jotka auttavat niiden kehittämisessä ja ylläpitämisessä. (Martinez-Caro, Aledo-Hernandez, Guillen-Perez, Sanchez-Iborra & Cano 2018.) Sisällönhallintajärjestelmää hyödyntämällä käyttäjä saa siis osittain samoja hyötyjä, kuin valmista pilvipalvelua käyttämällä. Järjestelmät kuitenkin mahdollistavat vapaamman sivuston tai järjestelmän rakentamisen kuin täysin valmiit pilvipalvelut, jolloin käyttäjä pystyy paremmin toteuttamaan haluamiaan ominaisuuksia sekä rakentamaan sivustostaan myös ulkoisesti haluamansa näköisen.

4 Case: Suomen Siiliyhdistyksen siilirekisteri

Tässä luvussa perehdytään tarkemmin Suomen Siiliyhdistyksen tarpeisiin ja toiveisiin uudelle siilirekisterille sekä kartoitetaan olemassa olevia järjestelmiä. Kartoituksen jälkeen valitaan siilirekisterin toteutustapa ja rakennetaan uusi rekisteri.

4.1 Siilirekisterin vaatimusmäärittely

Projektin alussa järjestettiin tapaaminen yhdessä Suomen Siiliyhdistyksen edustajan kanssa rekisterin nykytilan sekä uudelta rekisteriltä vaadittujen ja toivottujen ominaisuuksien selvittämiseksi. Tapaamisen perusteella koottiin rekisterille vaatimusmäärittely, joka on esitetty Taulukossa 1. Vaatimukset asetettiin tärkeysjärjestykseen niin, että tärkeimmille ominaisuuksille annettiin arvo ”välttämätön”, seuraavaksi tärkeimmille arvo ”tärkeä” ja vähiten tärkeille arvo ”idea”.

Taulukko 1: Siilirekisterin vaatimusmäärittely

ID	Vaatimus	Kuvaus	Tärkeys
1	Tietokanta	Järjestelmällä on tietokanta, johon tiedot tallennetaan.	Välttämätön
2	Graafinen käyttöliittymä	Graafinen käyttöliittymä toimintojen suorittamista varten.	Välttämätön
3	Lisäystoiminto	Järjestelmään pystyy lisäämään uusien siilien tiedot.	Välttämätön
4	Hakutoiminto	Järjestelmästä pystyy etsimään siilejä.	Välttämätön
5	Sukupuu	Mahdollisuus etsiä sukupuu poikasen nimellä.	Välttämätön
6	Rekisteri ei ole julkinen	Järjestelmä on yksityinen tai suojattu.	Välttämätön
7	Rekisteröinti edelliselle vuodelle	Rekisteriin voi rekisteröidä myös siilejä, jotka ovat syntyneet edellisenä vuonna.	Välttämätön
8	Tiedot voi tallentaa tiedostoon	Tiedot tulee pystyä siirtämään rekisteristä tiedostoon.	Välttämätön
9	Ilmainen	Palvelun, jolle toteutus pohjautuu, tulee olla ilmainen.	Välttämätön
10	Kuolinsyyn lisäys	Mahdollisuus lisätä siilin kuolinsyy rekisteriin.	Tärkeä
11	Automaattinen ID	Järjestelmä luo poikaselle yksilöllisen tunnuksen automaattisesti syntymäajan ja sukupuolen perusteella.	Tärkeä
12	Valikko kasvattajista	Poikasen lisäysvaiheessa mahdollisuus valita rekisteröityneistä kasvattajista.	Tärkeä
13	Hälytys WHS-merkinnästä	Järjestelmä hälyttää, jos siilin suvussa on ollut WHS-tapaus.	Tärkeä
14	Varoitus naaraiden lepoajasta	Järjestelmä hälyttää, jos naaraan lepoaika poikueiden välillä on liian lyhyt.	Idea
15	Hälytys sukulaisuudesta	Järjestelmä hälyttää, jos siilit ovat sukua toisilleen ja saaneet poikasia.	Idea

Vaatimusmäärittelystä huomataan, että suurin osa ominaisuuksista on joko välttämättömiä tai tärkeitä rekisterin toiminnan ja helppokäyttöisyyden kannalta. Erityisesti vaatimukset 5, sukupuun etsiminen poikasen nimellä, 7, rekisteröintimahdollisuus edelliselle vuodelle, sekä 11, yksilöllisen tunnuksen automaattinen luominen, nousivat esiin Suomen Siiliyhdistyksen toiveissa helppokäyttöisyyden lisäämiseksi. Myös vaatimukset 13, 14 ja 15 erilaisiin varoituksiin liittyen olivat tärkeitä toiveita järjestelmän toiminnallisuuksiksi, mutta eivät yhtä kriittisiä ominaisuuksia kuin esimerkiksi edellä mainitut ominaisuudet. Näiden vaatimusten pohjalta pystyttiin seuraavassa vaiheessa selvittämään olemassa olevia järjestelmiä juuri Suomen Siiliyhdistyksen tarpeet huomioon ottaen.

4.2 Olemassa olevien järjestelmien kartoitus

Koska Suomen Siiliyhdistyksen käyttämä rekisteri on ollut Google Sheets-dokumentti, ensimmäisenä perehdyttiin mahdollisuuksiin hyödyntää Google Sheetsiä tai muita Googlen palveluja rekisterin toteutukseen. Voittoa tavoittelemattomille yhdistyksille ilmainen Google Workspace oli yksi mahdollisuus, mutta se ei tarjonnut Google Sheetsin lisäksi muita rekisterin rakentamista helpottavia ominaisuuksia (Google n.d.-a), joten sen hyödyntäminen suljettiin pois vaihtoehdoista jo aikaisessa vaiheessa.

Myös Googlen Cloud -pilvipalveluun tutustuttiin. Palvelu on monipuolinen työkalu, jolla onnistuu niin datan tallentaminen tietokantaan, datan analysointi kuin omien sovellusten kehittäminen ja jakelu pilvipalvelun kautta (Google n.d.-c). Kyseinen palvelu ei kuitenkaan kuuluu Googlen ilmaiseksi voittoa tavoittelemattomille yhdistyksille tarjoamiin palveluihin, vaan se on aina maksullinen palvelu (Google n.d.-b), joten järjestelmän ilmaisvaatimuksen takia jouduttiin myös tästä palvelusta luopumaan nopeasti.

Googlen lisäksi myös Amazonilla on tarjonnassaan pilvipalveluita. Google Cloud -pilvipalvelun tavoin myös Amazon Web Services (AWS) tarjoaa useita toimintoja käyttäjilleen, muun muassa sovellusten kehittämistä ja jakelemista, tietokantojen käyttämistä ja datan analysointimahdollisuuksia (Amazon Web Services 2021a). Vaikka täyden palvelun käyttäminen on maksullista, on tarjolla myös useita aina ilmaisia ominaisuuksia, kuten 25 gigatavun tietokanta sekä 100 minuuttia koodin kääntämistä ja testaamista pilvessä kuu-

kaudessa (Amazon Web Services 2021b). Palvelu voisi siis olla yksi järjestelmistä, jolla siilirekisteri pystyttäisiin toteuttamaan.

Myös reaaliaikaista analyyttista prosessointia hyödyntävistä järjestelmistä löytyy useita vaihtoehtoja. Näistä esimerkiksi ClickHouse on ilmainen avoimenlähdekoodin tietokannan hallintajärjestelmä, jossa hyödynnetään OLAP-tekniikkaa (ClickHouse, Inc. 2021). Toinen tekniikkaan perustuva järjestelmä on Apache Kylin, jonka merkittävimpänä tarkoituksena on suuren määrän dataa analysointi nopeasti ja tehokkaasti (Apache Kylin 2015). Nämä järjestelmät kuitenkin painottuvat suurien datamäärien analysointiin, mikä ei kohtalaisen pienen siilirekisterin kannalta ole tärkeä ominaisuus.

Toteutusmahdollisuutena tutkittiin myös kahta sisällönhallintajärjestelmää, Joomla!:aa ja WordPressiä. Järjestelmät ovat pohjiltaan samankaltaisia ilmaisia PHP ja MySQL -pohjaisia avoimen lähdekoodin järjestelmiä (Open Source Matters, Inc. 2021; WordPress.org 2018). Näistä kahdesta WordPress on käytetympi, mikä tarkoittaa todennäköisesti myös useampia lisäosia ja kattavampaa tukea, sekä yleisesti tunnettu myös helppokäyttöisempänä vaihtoehtona (Mafereka & Winberg 2017). Myös Suomen Siiliyhdistyksen nykyiset verkkosivut on luotu WordPressiä hyödyntäen.

4.3 Toteutustavan valinta

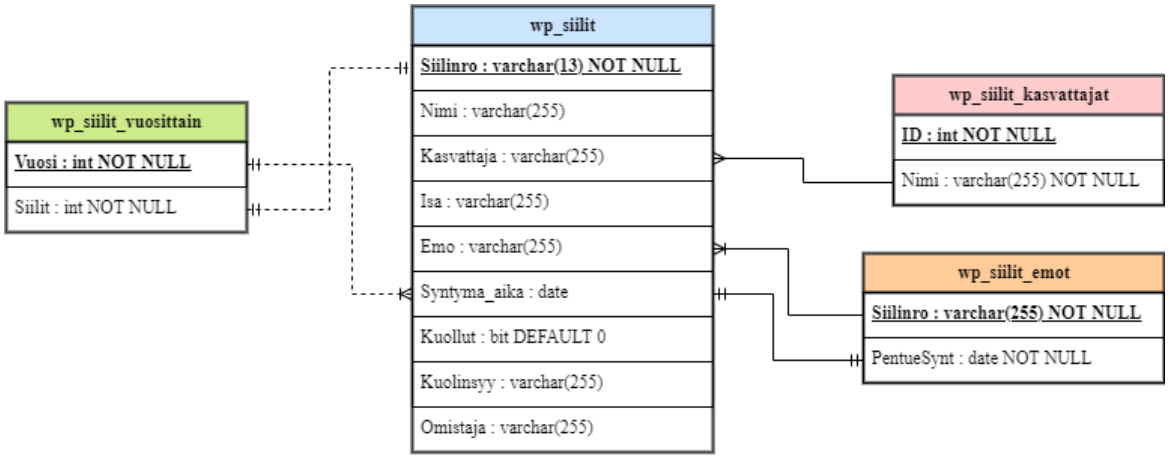
Siilirekisteriin haluttiin useita ominaisuuksia, joita ei löydy valmiista palveluista, kuten siilin tunnuksen muodostaminen automaattisesti syötettyjen tietojen perusteella sekä naaraiden lepoajan laskeminen ja sen laiminlyönnistä varoittaminen. Tällaisia toimintoja varten valitun toteutustavan tulisi antaa mahdollisuus oman toiminnallisuuden lisäämiseen jo mahdollisesti olemassa olevan toiminnallisuuden rinnalle. Tähän parhaiten edellä tutkituista järjestelmistä soveltuvat sisällönhallintajärjestelmät. Koska Suomen Siiliyhdistys oli jo luonut verkkosivunsa WordPressiä hyödyntäen, päätettiin rekisteri rakentaa sivujen yhteyteen WordPressin lisäosien avulla. Lisäksi mahdollisuus hyödyntää myös valmiita lisäosia takaa paremmin rekisterin ylläpidon jatkuvuuden.

4.4 Siilirekisterin toteutus

Ensimmäisenä toteutustavan valitsemisen jälkeen perehdyttiin WordPressissä jo tarjolla oleviin lisäosiin. Tietokannan käsittelyyn, tietojen hakemiseen ja tietojen näyttämiseen löytyi useita lisäosia. Myös jonkinlaisia sukupuu- ja rekisterilisäosia oli saatavilla, mutta niiden toiminnallisuus ei riittänyt kattamaan Suomen Siiliyhdistyksen tarpeita. Niinpä päädyttiin hyödyntämään WP Data Access -lisäosan ilmaista versiota apuna tietokantojen käsittelyssä verkkosivuston kautta. Lisäosa mahdollisti kehitysvaiheessa uusien taulujen luomisen tietokantaan ilman ohjelmointityötä sekä lopullista tuotetta varten tietokannasta löytyvien taulujen näyttämisen, muokkaamisen, tauluista tietojen etsimisen sekä taulujen tietojen tallentamisen tiedostoon. Lisäosan avulla pystyttiin WordPressiin ylläpitäjän sivuille lisäämään Siilirekisteri-valikko, missä näytetään kaikki rekisteriin lisätyt tiedot. Lisäksi lisäosa mahdollisti myös tietojen siirtämisen vanhasta Siilirekisteri-dokumentista uuteen tietokantatauluun tuomalla tiedot yhtenä tiedostona, jossa kaikki arvot on eroteltu toisistaan pilkulla (CSV-tiedostona).

4.4.1 Tietokantataulujen suunnittelu

Kun valmiit lisäosat oli tutkittu ja niiden tarjoamat ominaisuudet hyödynnetty, aloitettiin omien toiminnallisuuksien lisääminen kehittämällä oma WordPress-lisäosa. Ensimmäisenä kehitettiin suunnitelma lisäosaa varten tarvittavista tietokantatauluista. Tauluja päätettiin tehdä neljä: yksi siilien tietojen tallentamiseen, toinen kasvattajien tallentamiseen, kolmas emoja ja näiden poikueita varten ja neljäs vuosittain syntyneiden siilien määrän tallentamiseen. Taulut, niiden sisältämät sarakkeet sekä taulujen väliset suhteet näkyvät Kuvassa 1.



Kuva 1. Tietokantakaavio siilirekisteriin tarvittavista tietokantatauluista

Kaikkien taulujen nimiin lisättiin WordPressin oma WP-etuliite sekä siilit-etuliite. Siilit-etuliitteellä ja suomenkielisellä taulujen nimeämisellä pyrittiin varmistamaan, ettei tietokannasta löydy jonkin toisen lisäosan käyttämää samannimistä taulua.

Siilit-taulun luominen oli selkeä valinta rekisterin kehitystä ajatellen. Jotta rekisteri pystyttäisiin toteuttamaan, tarvitaan tietysti jokaisen siilin tiedot. Käytännössä tämä taulu vastaa Suomen Siiliyhdistyksen tämänhetkistä rekisterin toteutusta, ja siihen lisättiin pääosin samat tiedot. Ainoastaan siilinumerosta tehtiin pakollinen tieto, koska siilirekisteriin aiemmin lisätyillä siileillä ei välttämättä ole kaikkia muita tietoja. Siilinumero oli myös syntymäajan ja siilin kuoleman lisäksi ainoa sarake, jonka sisällön muodosta ja pituudesta voidaan olla varmoja. Tästä syystä muiden tietojen tyypiksi valittiin maksimissaan 255 merkkiä pitkä merkkijono.

Muut kolme taulua toimivat rekisteriä avustavina tauluina. Kasvattajat-taulu lisättiin, jotta rekisteröityneiden kasvattajien tiedot pystytään keräämään yhteen paikkaan ja myöhemmin tietoja hyödyntämään kasvattajan valinnassa siilin lisäämisen yhteydessä. Taulu emoista puolestaan valittiin toteutukseen emojen lepoajan laskemisen työkaluksi. Siilit vuosittain -tauluun pystytään lisäämään vuosiluvut, jolloin poikasia on syntynyt, ja jokaiselle vuodelle sinä vuonna syntyneiden poikasten lukumäärä. Lukumäärän avulla luodaan jokaiselle siilille yksilöllinen tunnus. Erilliset rivit vuosilukujen mukaan mahdollistavat poikasten rekisteröinnin myös aiemmille vuosille.

Kaikki edellä mainitut kolme taulua päätettiin toteuttaa omina tauluinaan sen sijaan, että tiedot poimittaisiin Siilit-taulusta tarpeen mukaan, koska haluttiin mahdollistaa tietojen selaaminen WP Data Access -lisäosan avulla ja pitää rekisteri mahdollisimman selkeänä käyttäjälleen. Pelkkää Siilit-taulua hyödyntäen rekisteri sisältäisi vain yhden taulun, joka pitäisi sisällään kaikki tiedot, jolloin esimerkiksi vuosittain syntyneiden siilien lukumäärä olisi vaikea selvittää. Kun tiedot löytyvät omista tauluistaan, voidaan rekisteriin lisätä näkymät kasvattajien, vuosittain syntyneiden siilien sekä emojen ja näiden viimeisimpien poikueiden selaamiseen.

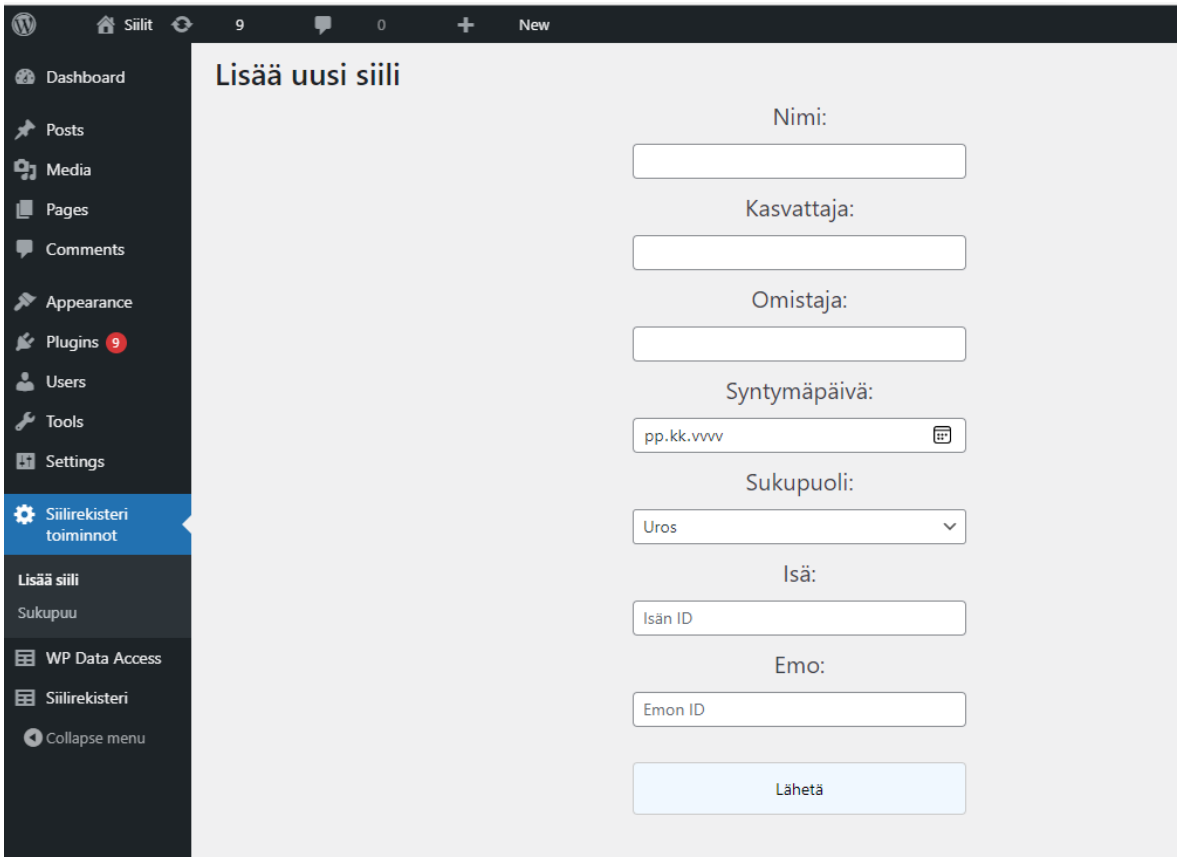
Kaikki neljä taulua myös liittyvät toisiinsa. Siilit-taulusta löytyvää siilin syntymäaikaa sekä emon siilinumeroa hyödynnetään Emot-taulussa emon tunnuksena ja tämän viimeisimmän poikueen syntymäaikana. Jokainen emo voi löytyä taulusta vain kerran samoin kuin yksittäisen siilin syntymäaika, mutta sama emo voi olla useammalla siilillä kuten myös sama syntymäaika.

Jokaiselle siilille voidaan myös merkitä sen kasvattaja, joka löytyy Kasvattajat-taulusta. Kasvattaja voi luonnollisesti rekisteröidä useita siilejä, mutta jokaisella siilillä voi olla vain yksi kasvattaja.

Vaikka Siilit vuosittain -taulu on erittäin tärkeä siilinumeroiden luomisen kannalta, on sillä epäsuorin yhteys varsinaiseen Siilit-tauluun. Jokaisen siilin syntymävuosi tulee löytyä Siilit vuosittain -taulusta, ja tätä vuotta vastaava siilien lukumäärä puolestaan on osa siilin siilinumeroa. Sama syntymävuosi voi siis kohdistua useaan siiliin, mutta siilien lukumäärä tietyllä ajanhetkellä on osa vain yhden siilin siilinumeroa. Kun siilejä lisätään rekisteriin, kasvaa taulusta löytyvä siilien lukumäärä ja näin jokaiselle siilille saadaan yksilöllinen tunnus.

4.4.2 Siilirekisterin tekninen toteutus

Siilirekisterin rakentaminen aloitettiin luomalla luvussa 4.4.1 esitetyt tietokantataulut, jotta rekisterin toiminnallisuutta pystyttiin kehittämään. Tämä aloitettiin luomalla sivu, jonka kautta rekisteriin pystytään lisäämään uusi siili. Sivu näkyy Kuvassa 2.

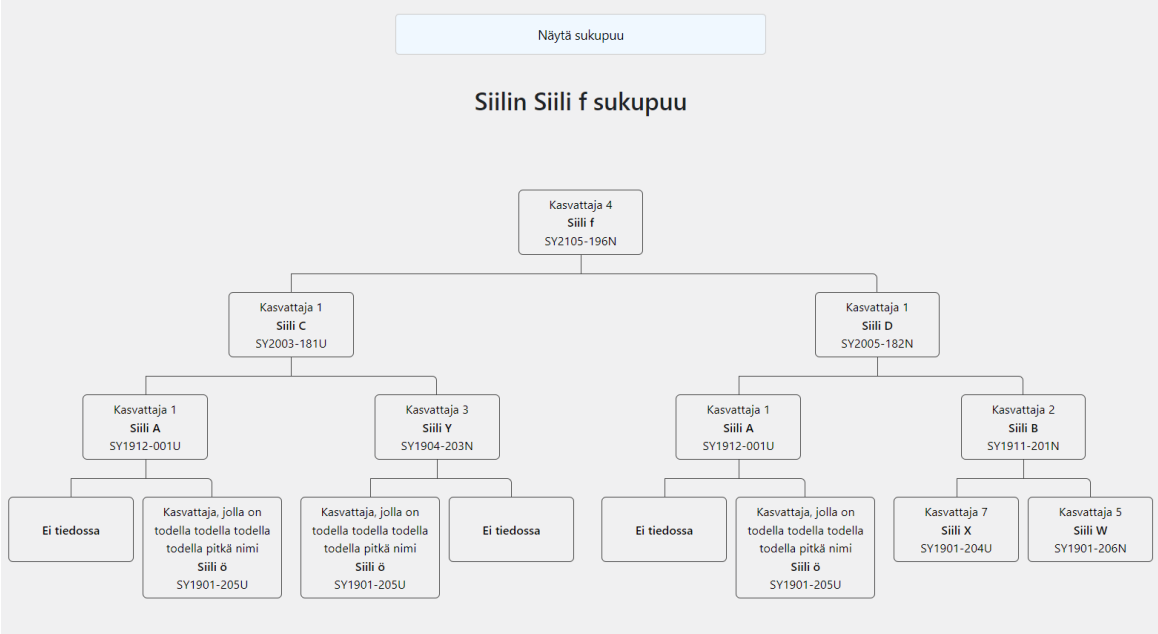


Kuva 2. Siilirekisterin sivu uuden siilin lisäämistä varten

Siilien kasvattajista, siilin isästä sekä siilin emosta tehtiin älykkäät kentät, jolloin tiedossa olevat kasvattajat ja siilit haetaan tietokannasta ja niitä ehdotetaan samalla, kun kenttään kirjoitetaan. Tämä helpottaa samoja kasvattajia useita kertoja lisättäessä sekä erityisesti siilin vanhempien lisäämisessä.

Siilin nimi, syntymäpäivä ja sukupuoli lisättiin pakollisiksi tiedoiksi. Syntymäpäivää ja sukupuolta tarvitaan myös seuraavassa vaiheessa, jossa niiden avulla siilille rakennetaan yksilöllinen tunnus. Ennen tätä kaikki syötetyt tiedot kuitenkin tarkistetaan ja niistä poistetaan ylimääräiset välilyönnit alusta ja lopusta. Kaikkien siilien tunnukset ovat muotoa SY[syntymävuosi][syntymäkuukausi]-[monesko kyseisenä vuonna syntynyt siili tämä siili on][sukupuoli], esimerkiksi SY2112-012N, jos kyseessä on naaras, tai SY2112-012U, jos kyseessä on uros. Tunnuksen luomisessa hyödynnetään Siilit vuosittain -taulua, josta etsitään siilin syntymäajan perusteella oikea vuosiluku ja kyseisenä vuonna syntyneiden siilien lukumäärä. Mikäli vuosilukua ei vielä löydy taulusta, lisätään se sinne. Jokaisen rekisteriin tehdyn lisäyksen yhteydessä siilien lukumäärä kyseisenä vuonna myös päivitetään. Loput tunnuksen luomiseen tarvitut tiedot parsitaan käyttäjän syöttämistä tiedoista.

Seuraava kehitettävä toiminto oli sukupuun etsiminen. Syöttämällä hakukenttään siilin nimi tehdään tietokantahaku kolme sukupolvea ylöspäin ja lisätään löydetyt vanhemmat sukupuuhun. Huomioon otetaan ainoastaan vanhemmat, ei sisaruksia, jotta sukupuusta ei tulisi liian suuri. Kuvassa 3 näkyy siilille Siili f etsitty sukupuu.



Kuva 3. Siilin sukupuu

Kuten kuvasta nähdään, jokaisen siilin kohdalla näytetään kasvattajan nimi, siilin nimi sekä siilin siilinumero. Jos jokin tiedoista ei ole saatavilla, jätetään kyseinen kohta tyhjäksi. Sukupuuhun lisättiin myös hieman toiminnallisuutta: jos käyttäjä napsauttaa hiirellään jotain sukupuussa olevista siileistä, etsitään uusi sukupuu kyseisen siilin nimellä. Esimerkiksi napsauttamalla Siili C:tä saadaan se tutkittavaksi siiliksi, ja näin sen sukua pystytään tutkimaan pidemmälle. Mikäli jonkin siilin vanhempia ei ole merkitty rekisteriin, näytetään näiden kohdalla teksti ”Ei tiedossa”.

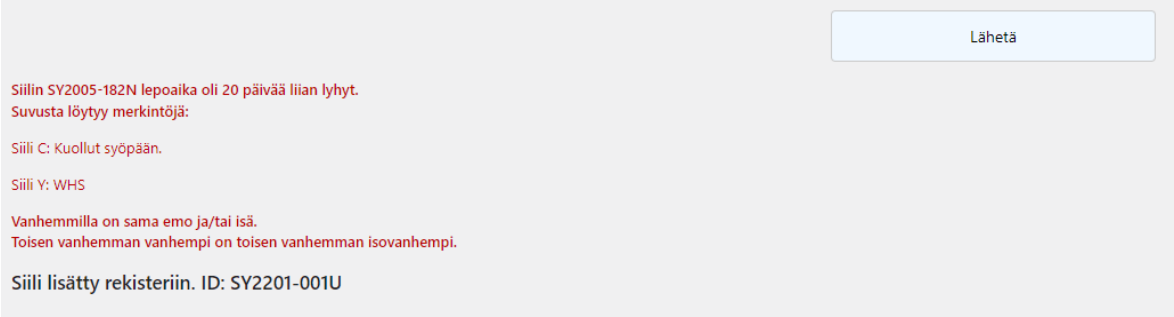
Sukupuun etsimistoiminnon jälkeen olivat kaikki välttämättömäksi määritetyt rekisterin toiminnallisuudet toteutettu. Näin ollen pystyttiin seuraavaksi siirtymään toteuttamaan tärkeät ominaisuudet.

Ensimmäisenä lisättiin siilin lisäyksen yhteyteen varoitus emon laiminlyödystä lepoajasta. Emon lepoajan tulisi kestää vähintään 60 päivää, ja sen laskeminen alkaa kahdeksan viikkoa poikueen syntymän jälkeen (Suomen Siiliyhdistys ry n.d.-b). Lepoajan laskemiseen hyödynnettiin Emot-taulua. Siilin lisäyksen yhteydessä tarkistetaan emon viimeisin poikue

ja päivitetään sen syntymäaika lisätyn siilin syntymäajan mukaan. Mikäli uusi lisätty poikue on kuitenkin syntynyt aikaisemmin kuin viimeisin tiedossa oleva poikue, ei lepoaikaa lasketa eikä poikueen syntymäaikaa päivitetä.

Lepoaikavaroituksen jälkeen lisättiin varoitus suvusta löytyvistä merkinnöistä. Tässä vaiheessa haetaan neljän edeltävän sukupolven tiedot, kuitenkin sukupuun tavoin sisaruksia ei oteta huomioon. Kaikkien siilien tiedot tarkistetaan, ja jos jollain siilillä on jokin merkintä Kuolinsyy-sarakkeessa, kuten merkintä siileillä esiintyvästä WHS-sairaudesta (Wobbly Hedgehog Syndrome), varoitetaan siitä lisäyksen yhteydessä.

Viimeinen lisätty varoitus oli varoitus sukulaisuudesta. Ohjelma osaa varoittaa, mikäli lisätyn siilin vanhemmat ovat sisaruksia, eli niillä on samat vanhemmat, tai puolisisaruksia, eli niillä on yksi yhteinen vanhempi. Myös vanhempien välisistä isä-tytär- ja emo-poika-suhteista varoitetaan sekä siitä, jos toisen siilin vanhempi on toisen siilin isovanhempi. Esimerkkejä varoituksista on nähtävissä Kuvassa 4.



Kuva 4. Lisäyksen yhteydessä näytettäviä varoituksia

Kuvassa näkyvät kaikki siilin lisäämisen yhteydessä mahdollisesti näkyvät varoitukset: varoitus liian lyhyestä lepoajasta, varoitus suvusta löytyvistä merkinnöistä sekä varoitus vanhempien lähisukulaisuudesta. Varoituksista näytetään aina vain ne, jotka toteutuvat lisätyn siilin kohdalla. Mikäli siilin lisääminen rekisteriin ei onnistunut, varoitetaan vain siitä, eikä muiden varoitusten tarpeellisuutta tarkisteta. Samat varoitukset sukulaisuudesta ja suvusta löytyvistä merkinnöistä lisättiin myös sukupuun yhteyteen.

Lopuksi lisättiin toiminnallisuutta lisäosan aktivointivaiheeseen sekä tehtiin parannuksia tietoturvaan. Aktivoinnin yhteydessä luodaan tarvittavat tietokantataulut, jos niitä ei ole olemassa. Ohjelmakoodiin lisättiin lähettäjän tarkistaminen lomakkeiden vastaanottamisen yhteyteen, jolloin pystytään varmistamaan komennon tulevan WordPressistä eikä ulkopuo-

lisilta sivustoilta. Lisäksi kehitetyt toiminnallisuudet muutettiin näytettäväksi sivustolla vain, jos käyttäjä on kirjautunut sisään, sekä lisättiin jokaisen kooditiedoston alkuun tarkistus siitä, että niitä käsitellään WordPressin kautta.

4.4.3 Rekisterin käyttöönoton valmisteleminen

Kun rekisterin toteutukseen oltiin tyytyväisiä, aloitettiin valmistelu rekisterin käyttöönottoa varten. Tähän kuului siilirekisterin vanhan datan parsiminen uuteen rekisteriin sopivaan muotoon. Kaikki vanhasta rekisteristä löytyvä data siirrettiin Microsoft Excel -tiedostoon, minkä jälkeen tietojen muotoilut tarkastettiin. Tiedostosta myös poimittiin kasvattajien nimet sekä vuosittain syntyneiden siilien määrät erillisiin tiedostoihin, jotta tiedot välittyisivät uuteen rekisteriin. Kaikki edellä mainitut tiedostot tallennettiin CSV-tiedostoina sujuvaa siirtoa varten.

5 Keskustelu

Siilirekisteri saatiin toteutettua onnistuneesti kaikki sille asetetut vaatimukset täyttäen, ja myös Suomen Siiliyhdistyksen edustajat olivat lopputulokseen tyytyväisiä. Avoimenlähdekoodin sisällönhallintajärjestelmän ja erityisesti WordPressin käyttäminen oli hyvä päätös, koska valmiit lisäosat kevensivät toteutettavien ominaisuuksien työmäärää ja laajat kehittäjäverkostot takasivat dokumentaation sekä tuen löytymisen. Koska rekisteri saadaan suoraan Suomen Siiliyhdistyksen verkkosivujen yhteyteen, ei sen käyttöönotosta tai käytöstä myöskään synny yhdistykselle ylimääräisiä kustannuksia. Rekisteri on helposti kaikkien niiden saatavilla, joilla on tunnukset yhdistyksen verkkosivuille, mutta ei ulkopuolisten henkilöiden nähtävissä. Lisäksi hyödyntämällä WordPressin tarjoamia tietoturvaominaisuuksia sekä lisäosia pystytään rekisterin ylläpito takaamaan paremmin kuin esimerkiksi täysin itse kehitettynä.

Vaikka toteutus oli onnistunut, jätti työ kuitenkin jotain kysymyksiä jälkeensä. Kehitysvaiheessa järjestelmää ei testattu Suomen Siiliyhdistyksen verkkosivujen yhteydessä vaan kehitys tapahtui erillisessä kehitysympäristössä. Uusi rekisteri saatiin onnistuneesti lisättyä yhdistyksen järjestelmään rikkomatta mitään sivuilla olevaa toiminnallisuutta, mutta pidemmän aikavälin toimivuudesta ei ole vielä tietoa.

Toisaalta myöskään tietoturvaan ei keskitytty yhtä paljon kuin esimerkiksi kaupallisissa projekteissa olisi tehty. Tietokantayhteyksissä käytettiin kuitenkin WordPressin suosittelemia tapoja ja esimerkiksi kaikki kyselyt, joissa hyödynnettiin käyttäjän antamaa dataa, käytettiin valmistelutoiminnon kautta. Tämä valmistelutoiminto tarkastaa syötetyt tiedot, ja näin estetään käyttäjien syöttämän haitallisen SQL-koodin suorittaminen ja vahinkojen aiheuttaminen. Lisäksi jokaisen syötetyn tiedon vastaanotossa tarkistettiin, että tiedot tulivat halutulta sivulta eikä esimerkiksi ulkopuolisen toiminnon kautta.

Myös itse kooditiedostoja olisi voinut edelleen siistiä ja jaotella. Paikoitellen samaa tai samankaltaista koodia kirjoitettiin useampaan kuin yhteen kohtaan. Nämä olivat kuitenkin tilanteita, joissa olisi vaadittu koodin siirtämistä toiseen tiedostoon. Tiedostojen yhdistäminen oli haastavaa, joten siihen ei juurikaan käytetty ylimääräistä aikaa. Lisäksi tiedostojen

koot olivat tässä projektissa vielä kohtuullisia, joten jako useampaan tiedostoon eri toiminnallisuuksien tai aihealueiden perusteella ei ollut välttämätöntä.

Kehitettyyn lisäosaan ei myöskään lisätty mitään toiminnallisuutta sen käytöstä poistamisen yhteyteen, koska käytännössä tämä tarkoittaisi vain tietokantataulujen poistamista ja välimuistin tyhjentämistä. Toiminnallisuuden lisäämättä jättämällä pyrittiin varmistamaan se, ettei tauluja poisteta vahingossa ja näin menetetä tietoja kaikista siileistä. Tämä kuitenkin tarkoittaa, että mahdollisen lisäosan poistamisen yhteydessä tulisi tietokantataulut poistaa manuaalisesti.

Työssä painotettiin ominaisuuksien ja toimintojen olemassaoloa ja toimivuutta, jolloin sovelluksen käyttöliittymä ja ulkonäkö jäivät vähemmälle huomiolle. Käyttöliittymästä pyrittiin tekemään mahdollisimman selkeä ja helppokäyttöinen, mutta sitä voisi vielä kehittää käyttäjää miellyttävämmäksi ja modernimmaksi. Toteutuksessa ei myöskään hyödynnetty ulkopuolisia kirjastoja, jotka olisivat mahdollisesti auttaneet käyttöliittymän luomisessa, koska niiden yhteensopivuudesta lisäosien ja WordPressin päivitysten kanssa ei ollut varmuutta.

Mahdollisessa siilireksiterin jatkokehittämisessä voitaisiin panostaa enemmän käyttöliittymään ja tutkia mahdollisuuksia sen kehittämiseen ulkopuolisten kirjastojen avulla. Lisäksi rekisteriin voitaisiin kehittää toiminnallisuutta siilien sisarusten visualisointiin ja etsimiseen, sillä tämänhetkisessä toteutuksessa sisaruksia ei oteta huomioon lainkaan.

Koska siilirekisteri on pienikokoinen rekisteri, ei vaihtoehtoisten ratkaisujen suorituskykyjä tutkittu. Yksi jatkotutkimuksen aihe voisi siis olla toiminnallisuuksien optimointi ja vaihtoehtojen suorituskykyjen vertailu. Myös tietokantataulujen määrän ja rakenteiden sekä niihin tehtävien kyselyiden tehokkuutta voitaisiin tutkia ja mahdollisesti parantaa, jos datan määrä merkittävästi kasvaa tulevaisuudessa. Esimerkiksi OLAP-tekniikan hyötyä voitaisiin tutkia ja arvioida tarkemmin. Muita mahdollisia jatkotutkimusaiheita ovat rekisterin toteuttaminen muilla tutkituilla järjestelmillä ja tekniikoilla sekä eri sisällönhallintajärjestelmien vertailu rekisterin toteuttamisen kannalta.

Tutkimus toteutettiin yhden tapauksen case-tutkimuksena, jolloin tulosten perusteella ei pystytä yleispätevästi päättelemään rekisterin toteutusmenetelmän toimivuutta. Kuitenkin vaikuttaisi siltä, että sisällönhallintajärjestelmät voivat toimia hyvin pienempien yhdistysten verkkopohjaisten rekisterien toteutukseen. Myös mahdollisuus rekisterin lisäämisestä

yhdistyksen verkkosivujen yhteyteen todettiin toimivaksi vaihtoehdoksi ainakin Suomen Siiliyhdistyksen tapauksessa.

6 Yhteenveto

Tässä työssä tutkittiin erilaisia rekisterin toteutusmahdollisuuksia pienelle yhdistykselle. Tutkimus toteutettiin case-tutkimuksena ja tutkittavana rekisterinä oli Suomen Siiliyhdistyksen siilirekisteri. Yhdessä Suomen Siiliyhdistyksen kanssa keskusteltiin uuden rekisterin vaatimuksista, minkä jälkeen tutustuttiin olemassa oleviin järjestelmiin, joilla rekisteri mahdollisesti voitaisiin toteuttaa. Siilirekisterin vaatimusten perusteella valittiin sopiva toteutustapa ja toteutettiin rekisteri.

Suomen Siiliyhdistyksen vanha rekisteri oli todella vaikeakäyttöinen ja sen toiminnallisuudet olivat heidän tarpeisiinsa nähden puutteelliset. Tästä johtuen heillä oli tarve saada käyttöönsä uusi rekisteri. Koska vastaavia rekistereitä tarjoavia palveluita ei löydy valmiina, eikä tutkimustyötä niihin liittyen ole aikaisemmin juurikaan tehty, päätettiin toteuttaa tämä tutkimus erilaisista toteutusvaihtoehdoista.

Siilirekisteri toteutettiin WordPress-sisällönhallintajärjestelmää hyödyntäen Suomen Siiliyhdistyksen omien verkkosivujen yhteyteen. Rekisteriin pystyttiin toteuttamaan kaikki yhdistyksen projektin alussa toivomat ominaisuudet, ja myös tiedot siirrettiin vanhasta rekisteristä uuteen onnistuneesti. Suomen Siiliyhdistys oli tyytyväinen lopputulokseen ja rekisteri otettiin käyttöön korvaamaan vanha rekisteri.

Tutkimuksen myötä selvisi, että rekisterin toteuttamiselle on useita erilaisia vaihtoehtoja. Rekisterin voi rakentaa itsenäisesti tai hyödyntää valmiita järjestelmiä, kuten pilvipalveluita ja sisällönhallintajärjestelmiä. Onnistuneen toteutuksen perusteella voidaan olettaa, että sisällönhallintajärjestelmät soveltuvat tällaiseen kehitykseen hyvin ja myös mahdollistavat rekisterin ylläpidon tulevaisuudessa. Erityisesti Suomen Siiliyhdistyksen tapauksessa WordPressin käyttäminen oli järkevä valinta sekä näistä syistä että verkkosivujen ja rekisterin yhdistämismahdollisuuden takia.

Tutkimustuloksia voidaan jatkossa hyödyntää vastaavien pienien yhdistysten rekisterien toteuttamisessa. Kuitenkin myös jatkotutkimusta kaivataan, esimerkiksi rekisterien tietoturvaan voitaisiin perehtyä enemmän sekä rekisterien toteuttamiseen muilla järjestelmillä. Myös eri sisällönhallintajärjestelmien hyötyjä juuri rekisterien toteutuksessa voitaisiin ver-

tailla sekä erilaisten ulkopuolisten kirjastojen hyödyntämismahdollisuuksia rekisterin kehittämisessä tutkia.

Lähteet

Amazon Web Services. 2021a. Cloud Services - Amazon Web Services (AWS). [Verkkosivu]. [Viitattu 21.11.2021.] Saatavissa: https://aws.amazon.com/.

Amazon Web Services. 2021b. Free Cloud Computing Services - AWS Free Tier. [Verkkosivu]. [Viitattu 21.11.2021]. Saatavissa: https://aws.amazon.com/free/.

Apache Kylin. 2015. Overview of Apache Kylin4.x. [Verkkosivu]. [Viitattu 21.11.2021]. Saatavissa: http://kylin.apache.org/docs/.

Beaulieu, A. 2020. Learning SQL: Generate, Manipulate, and Retrieve Data. O'Reilly Media, Incorporated. Sebastopol, Yhdysvallat. [E-kirja]. [Viitattu 25.10.2021] Saatavissa: http://ebookcentral.proquest.com/lib/lut/detail.action?docID=6128234. ISBN 978-1-4920-5756-7.

Berrington, J. 2017. Databases. Anaesthesia & Intensive Care Medicine, 18, 3, s. 155–157. [Verkkolehti]. [Viitattu 25.10.2021]. Saatavissa: https://doi.org/10.1016/j.mpaic.2016.11.016.

Błażewicz, J., Kubiak, W., Morzy, T. & Rusinkiewicz, M. 2003. Handbook on Data Management in Information Systems. Springer, Berlin, Heidelberg. [E-kirja]. [Viitattu 23.10.2021]. Saatavissa: https://doi.org/10.1007/978-3-540-24742-5.

Celko, J. 2006. Joe Celko's Analytics and OLAP in SQL. Morgan Kaufmann. San Francisco, Kalifornia, Yhdysvallat. [E-kirja]. [Viitattu 26.10.2021]. Saatavissa: https://search.ebscohost.com/ login.aspx?direct=true&db=e000xww&AN=195632&site=ehost-live. ISBN 978-0-12-369512-3.

Charles M. Schweik & Robert C. English 2012. Internet Success : A Study of Open-Source Software Commons. The MIT Press. Cambridge, Mass. [E-kirja]. [Viitattu 24.10.2021]. Saatavissa: https://search.ebscohost.com/ login.aspx?direct=true&db=e000xww&AN=462604&site=ehost-live. ISBN 978-0-262-01725-1.

ClickHouse, Inc. 2021. Overview | ClickHouse Documentation. [Verkkosivu]. [Viitattu 21.11.2021]. Saatavissa: https://clickhouse.com/docs/en/.

Euroopan komissio n.d. Mikä on rekisterinpitäjä tai tietojen käsittelijä? [Verkkosivu]. [Viitattu 7.10.2021.]. Saatavissa: https://ec.europa.eu/info/law/law-topic/data-protection/ reform/rules-business-and-organisations/obligations/controller-processor/ what-data-controller-or-data-processor_fi.

Forrest, C.B., Bartek, R.J., Rubinstein, Y. & Groft, S.C. 2011. The case for a global rare-diseases registry. The Lancet, 377, 9771, s. 1057–1059. [Verkkolehti]. [Viitattu 22.10.2021]. Saatavissa: https://doi.org/10.1016/S0140-6736(10)60680-0.

Google. n.d.-a. Google Workspace järjestöille: Yhteiskäyttötyökalut - Google järjestöille. [Verkkosivu]. [Viitattu 21.11.2021]. Saatavissa: https://www.google.com/intl/fi/nonprofits/offerings/workspace/.

Google. n.d.-b. Googlen järjestötuen kelpoisuusehdot - Google järjestöille. [Verkkosivu]. [Viitattu 21.11.2021]. Saatavissa: https://www.google.com/intl/fi/nonprofits/eligibility/.

Google. n.d.-c. Products and Services | Google Cloud. [Verkkosivu]. [Viitattu 21.11.2021]. Saatavissa: https://cloud.google.com/products?hl=fi.

Humphries, M.D. 2017. Creating a Registry for Patients with Thoracic Outlet Syndrome. Diagnostics, 7, 2, s. 36. [Verkkolehti]. [Viitattu 22.10.21]. Saatavissa: https://doi.org/10.3390/diagnostics7020036.

LUT. 2021. Tietosuojailmoitus. [Verkkodokumentti]. [Viitattu 28.12.2021]. Saatavissa: https://www.lut.fi/documents/10633/498988/LUT+tietosuojailmoitus+Opiskelijat/ f1b5fee3-dcb7-414c-be6d-030d258f0f07.

Maass, W. 2012. On the Changing Market for Content Management Systems: Status and Outlook. Teoksessa: Maass, W. & Kowatsch, T. Semantic Technologies in Content Management Systems: Trends, Applications and Evaluations. Springer. Berlin, Heidelberg, s. 3-8. [E-kirja]. [Viitattu 25.10.2021]. Saatavissa: https://doi.org/10.1007/978-3-642-24960-0_1. ISBN 978-3-642-24960-0.

Mafereka, M. & Winberg, S. 2017. Analysis and development of an online knowledge management support system for a Community of Practice: Comparing Joomla, Wordpress and Drupal with regard to development of Community of Practice Website. Association for Computing Machinery. New York, NY, USA. [Verkkojulkaisu]. [Viitattu 21.11.2021]. Saatavissa: https://doi.org/10.1145/3077584.3077604. ISBN 978-1-4503-4833-1.

Martinez-Caro, J.-M., Aledo-Hernandez, A.-J., Guillen-Perez, A., Sanchez-Iborra, R. & Cano, M.-D. 2018. A Comparative Study of Web Content Management Systems. Information, 9, 2, s. 27. [Verkkolehti]. [Viitattu 25.10.2021]. Saatavissa: https://doi.org/10.3390/info9020027.

Open Source Matters, Inc. 2021. Joomla! - Content Management System to build websites & apps. [Verkkosivu]. [Viitattu 21.11.2021]. Saatavissa: https://www.joomla.org/ about-joomla.html.

Orlando, Q.-S. & Carolina, S. 2019. A Review on OLAP Technologies Applied to Information Networks. ACM Transactions on Knowledge Discovery from Data, 14, 1, s. 1-25. [Verkkolehti]. [Viitattu 25.10.2021]. Saatavissa: https://doi.org/10.1145/3370912.

Robert, W. 2006. Data Warehouses and OLAP: Concepts, Architectures and Solutions: Concepts, Architectures and Solutions. Idea Group Inc. ISBN 978-1-59904-366-1.

Runeson, P. & Höst, M. 2009. Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, 14, 2, s. 131–164. [Verkkolehti]. [Viitattu 29.12.2021]. Saatavissa: http://dx.doi.org/10.1007/s10664-008-9102-8.

Safonov, V.O. 2016. Trustworthy cloud computing. Wiley-Blackwell. New Jersey, Yhdysvallat. 317 s. ISBN 978-1-119-11351-5.

Seadle, M. 2006. Content management systems. Library Hi Tech, 24, 1, s. 5–7. [Verkkolehti]. [Viitattu 26.10.2021]. Saatavissa: http://dx.doi.org/10.1108/07378830610652068.

Suomen Siiliyhdistys. ry n.d.-a. Etusivu. Suomen Siiliyhdistys ry. [Verkkosivu]. [Viitattu 6.10.2021]. Saatavissa: http://www.suomensiiliyhdistys.fi/.

Suomen Siiliyhdistys ry. n.d.-b. Kasvattajasäännöt. Suomen Siiliyhdistys ry. [Verkkosivu]. [Viitattu 21.11.2021]. Saatavissa: https://www.suomensiiliyhdistys.fi/kasvatus/kasvattajasaannot/.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B. & Wesslén, A. 2012. Experimentation in Software Engineering. Springer. Berlin, Heidelberg. [E-kirja]. [Viitattu 28.12.2021]. Saatavissa: https://doi.org/10.1007/978-3-642-29044-2_5.

WordPress.org 2018. Democratize Publishing. [Verkkosivu]. [Viitattu 21.11.2021]. Saatavissa: https://wordpress.org/about/.

Wu, W. 2021. SQL2X: Learning SQL, NoSQL, and MapReduce via Translation. Association for Computing Machinery. New York, NY, USA. [Verkkojulkaisu]. [Viitattu: 25.10.2021]. Saatavissa: https://doi.org/10.1145/3408877.3432541.

Yin, R. 2018. Case Study Research and Applications: Design and Methods. 6. painos. SAGE. Los Angeles, Yhdysvallat. 319 s. ISBN 978-1-5063-3616-9.

Zygiaris, S. 2018. Database Management Systems : A Business-Oriented Approach Using ORACLE, MySQL and MS Access. Emerald Publishing Limited. Bingley, UK. [E-kirja]. [Viitattu 23.10.2021]. Saatavissa: https://search.ebscohost.com/login.aspx?direct=true&db=e000xww&AN=1813751&site=ehost-live.