



## **Ohjelmistokehittäjän ammatillisten valmiuksien kehittyminen**

Anni Vanhanen

Haaga-Helia ammattikorkeakoulu

Tradenomin tutkinto

Amk-opinnäytetyö

2022

## Tiivistelmä

<b>Tekijä(t)</b> Anni Vanhanen
<b>Tutkinto</b> Tradenomi
<b>Raportin/Opinnäytetyön nimi</b> Ohjelmistokehittäjän ammatillisten valmiuksien kehittyminen
<b>Sivu- ja liitesivumäärä</b> 49
<p>Opinnäytetyö on päiväkirjamallinen työ, jossa esitellään tekijän ammatillisten valmiuksien kehittymistä työuran alussa. Opinnäytetyössä selvitettiin tekijän ammatillisen osaamisen lähtötilanne sekä toimintaympäristö, jossa työtä tehtiin. Tässä opinnäytetyössä esitettävä ohjelmistokehittäjän työ sisältää sovelluskehityksen lisäksi käyttöliittymän suunnittelua. Työtä tehdään asiakasprojekteissa ketteriä menetelmiä, kuten Scrumia, hyödyntäen.</p> <p>Tekijän työtä seurattiin kahdeksan viikon ajan päivittäisinä merkintöinä. Seurantaviikot sijoituivat aikavälille 21.2.2022-22.4.2022. Jokaisen viikon päätteeksi viikkoanalyysissa otettiin viikon aikana esiin noussut aihe tarkempaan käsittelyyn. Seurantajakso alkaa Microsoft Azure -pilvipalvelun opiskelulla ja sen esittelyllä. Kirjoittaja opiskeli työsuhteen alun aikana tiiviisti kyseisen palvelun sertifiointikokeeseen, ja opiskelua seurataan useiden viikkojen ajan. Tästä johtuen opinnäytetyössä esitellään myös erilaisia oppimistyyliä ja pohditaan kirjoittajaa oppijana. Opinnäytetyössä käsitellään myös työyhteisöviestintää ja sen merkitystä työyhteisölle.</p> <p>Työssä esitellään myös Vue.js-sovelluskehystä sekä React.js-sovelluskehystä sekä tutkitaan näiden yhteneväisyyksiä ja eroavaisuuksia. Ohjelmistokehityksen osalta kirjoittaja tutustuu ASP.NET Coreen sekä MVC-malliin. MVC-mallilla tarkoitetaan sellaista sovellusarkkitehtuuria, jossa käyttöliittymässä näytettävä tieto on erotettu sovelluksen sisemmästä logiikasta ja toiminnasta. Näiden kahden välillä toimii kolmas elementti, kontrolleri, joka viestii näiden kahden sovelluksen osan välillä.</p> <p>Kirjoittaja perehtyy lyhyesti myös datan visuaaliseen esittämiseen sekä käyttöliittymäsuunnitteluun, sillä hänen asiakasprojektinsa liittyvät näihin aiheisiin. Datan raportointi ja visuaalinen esittäminen tehdään Qlik Sensellä, joka on datan analysoinnin työkalu. Käyttöliittymäsuunnittelu puolestaan tapahtuu prototyypityökalu Figmalla.</p> <p>Seurantajakso ja sen teemat keskittyvät laajasti ammatillisen osaamisen vahvistamiseen, sillä opinnäytetyön kirjoittaja työllistyi asiakasprojektiin vasta seurantajakson loppupuolella. Työn päättää pohdinta, jossa tekijä reflektoi omaa kehitystään työntekijänä ja sitä, miten omat ammatilliset valmiudet kehittyivät seurantajakson aikana.</p>
<b>Asiasanat</b> Ohjelmistokehitys, ohjelmistokehitys, pilvipalvelut, työyhteisöviestintä, käyttöliittymäsuunnittelu

## Sisällys

1	Johdanto.....	1
1.1	Keskeiset ammattikäsitteet ja lähteet.....	2
2	Lähtötilanteen kuvaus.....	4
2.1	Oman nykyisen työn analysointi.....	4
2.2	Sidosryhmien esittely.....	5
2.3	Työpaikan vuorovaikutustilanteet.....	6
3	Seurantajakson raportointi viikkoanalyseineen.....	8
3.1	Seurantaviikko 1: Microsoft Azure -pilvipalvelu.....	8
3.2	Seurantaviikko 2: ASP.NET Core MVC-malli ohjelmistokehityksessä.....	11
3.3	Seurantaviikko 3: Työyhteisöviestintä Teamsissa.....	17
3.4	Seurantaviikko 4: Minä oppijana.....	21
3.5	Seurantaviikko 5: Datan esittäminen visuaalisesti.....	24
3.6	Seurantaviikko 6: Vue.js-sovelluskehys.....	29
3.7	Seurantaviikko 7: React.js- ja Vue.js-sovelluskehysten yhtäläisyydet ja erot.....	34
3.8	Seurantaviikot 8 ja 9: Käyttöliittymäsuunnittelu.....	39
4	Pohdinta.....	46
	Lähteet.....	48

# 1 Johdanto

Tämä opinnäytetyö seuraa aloittelevan ohjelmistokehittäjän (junior software developer) ammatillisten valmiuksien kehittymistä työsuhteen ensimmäisten kuukausien ajan. Opinnäytetyö toteutetaan päiväkirjana ja työssä kasvua seurataan kahdeksan viikon ajan aikavälillä 21.2.2022-22.4.2022. Viikoittainen raportointi koostuu työpäivien seurannasta ja viikon lopussa kirjoitettavasta viikkoanalyysistä.

Aiemmin olen työskennellyt asiakaspalvelutehtävissä usean vuoden ajan. Runsas kokemus asiakaspalvelusta on valmistanut minua tekemään itsenäisiä päätöksiä työssä sekä toimimaan haastavissa sosiaalisissa tilanteissa. Aiemmassa työsuhteessani kehitin yrityksen digitaalisia palveluita. Osallistuin ERP-projektiin sekä yrityksen uusien WordPress-pohjaisten verkkosivujen julkaisemiseen. Lisäksi koulutin kollegoitani verkkosivujen sisällönhallintaan WordPressissä.

Tällä hetkellä olen töissä suuressa pohjoismaisessa IT-alan yrityksessä, joka työllistää 24 000 henkeä maailmanlaajuisesti. Suomessa työntekijöitä on yli 3000. Yritys on erikoistunut ohjelmistoihin sekä pilviteknologiaan.

Aloittelevan ohjelmistokehittäjän toimenkuvaan kuuluu ohjelmistokehitys asiakasprojekteissa. Projekteissa työskennellään usein ketteriä menetelmiä hyödyntäen.

Työssä tarvitaan projektista riippuen vaihtelevasti eri ohjelmointikieliä. Koska yritys on asiantuntija pilvipalveluissa, myös pilvipalveluiden tunteminen on tärkeä taito. Teknisen osaamisen lisäksi työssä tarvitaan myös projektityöskentelyyn liittyvää osaamista, halua mukautua alalla tapahtuviin muutoksiin ja kykyä työskennellä ryhmässä. Työn tukena olen käyttänyt Stackoverflow-palvelua teknisten ongelmatilanteiden ratkaisemiseen ja Microsoft Learnia Azuren opiskeluun.

Ammatillisen kehittymisen tueksi olen lukenut Robert Martinin Clean Code: A Handbook of Agile Software Craftmanship -kirjaa (2009). Kirja esittelee siistin koodin kirjoittamisen tärkeyttä ja sitä, miten koodiin panostaminen helpottaa työn tekoa myöhemmin projektissa. Kirjasta on ollut hyötyä työhön asennoitumisessa ja opitun työskentelytavan siirtämisessä käytäntöön.

Käyttöliittymäsuunnittelussa ja saavutettavuudessa olen tukeutunut Chris Jacksonin ja Nancy Ciolekin teokseen Digital Design in Action: Creative Solutions for Designers (2017). Kirjassa esitellään hyvän visuaalisen suunnittelun periaatteita ja teoksesta on ollut itselleni apua opinnoissa sekä kevään aikana asiakasprojektissa, jossa olen ensimmäistä kertaa itse vastuussa käyttöliittymän suunnittelusta.

Tavoitteenani on vahvistaa osaamistani front end-ohjelmoijana. Full stack -ohjelmointi on myös minua mahdollisesti kiinnostava urapolku, minkä vuoksi haluan opiskella uuden kielen. C# on valikoitunut kieleksi useissa projekteissa ja tästä syystä aion myös itse opiskella kyseistä kieltä. Haluan myös käytännön kokemusta ketterillä menetelmillä, kuten Scrumilla työskentelystä. Ammatillisten valmiuksien kasvattamiseksi haluan myös opiskella itselleni Azure Fundamentals -sertifikaatin. Tarkoituksena ei ole oppia vähän useasta aiheesta, vaan haluan aluksi vahvistaa osaamistani muutamista eri teknologioistani, jotta voin myöhemmin kasvattaa osaamistani jollain uudella osa-alueella.

## 1.1 Keskeiset ammattikäsitteet ja lähteet

### ASP.NET Core

ASP.NET Core on Microsoftin sovelluskehys, jolla voidaan kehittää erilaisia ohjelmia web-kehityksestä mobiilikehitykseen.

### Back end kehitys

Palvelinympäristössä tapahtuvaa ohjelmistokehitystä, joka tapahtuu näkyvän käyttöliittymän takana olevien toimintojen mahdollistamiseksi.

### Front end kehitys

Selainpään tekniikoilla tapahtuvaa web-kehitystä, jossa rakennetaan palvelulle toimivaa, graafista käyttöliittymää loppukäyttäjää varten.

### Full stack

Full stack kehittäjät hallitsevat sekä front end- että back end -ohjelmoinnin.

### Käyttöliittymäsuunnittelu

Käyttöliittymäsuunnittelulla tarkoitetaan sitä prosessia, jossa suunnittelija luo suunnitelman graafisesta käyttöliittymästä projektissa tehtyjen määrittelyjen pohjalta. Käyttöliittymäsuunnittelussa huomioidaan tarpeellisten toiminnallisten elementtien lisäksi haluttu visuaalinen ilme. Tavoitteena on ohjata käyttäjä prosessin läpi sujuvasti.

### Microsoft Azure

Microsoft Azure on pilvipalvelu, joka tarjoaa useita kymmeniä erilaisia palveluita verkkopalveluiden julkaisemisesta koneoppimiseen. Palveluista voi koota organisaation tarpeisiin sopivan paketin.

Azuren toiminta perustuu kestävyYTEEN ja korkeaan saatavuuteen, joilla pystytään takaamaan yrityksille aina saavutettava palvelu. (Microsoft 2022a.)

### Microsoft Teams

Microsoftin viestintäalusta, jossa käyttäjät ja yhteisöt voivat viestiä videon, äänen ja tekstin välityksellä. Palvelussa voi jakaa ja tallentaa tiedostoja sekä käyttää yhteensopivia sovelluksia työnteon sujuvoittamiseksi.

### MVC-malli

MVC-mallin nimi tulee sanoista Model, View ja Controller. Tällä termillä tarkoitetaan sovellusarkkitehtuurin mallia, jossa sovellus on jaettu kolmeen osaan, jotka toimivat yhdessä. MVC-mallissa sovelluksen toiminta on kirjoitettu malli-tiedostoihin (model-tiedostot). Selainpäässä näkyvä graafinen käyttöliittymä ohjelmoidaan näkymä-tiedostoihin (view-tiedostot). Näiden kahden välillä viestinviejänä toimii käsittelijä, controller. Käsittelijä vie malli-tiedostosta tietoa näkymään, ja näkymältä käyttäjän tekemiä toimintoja takaisin malli-tiedoston käsiteltäväksi.

### Osaamisen sertifikaatti

Sertifioinnilla tarkoitetaan osaamisen virallista todentamista. Kyseessä on todistus, jonka mukaan sen saanut henkilö on hyväksytysti todistanut osaamisensa sertifioinnin kohteen saralla. Osaamisen sertifikaatit saavutetaan valvotussa kokeessa. Sertifikaatti voi olla voimassa toistaiseksi, tai se voi vanhentua ennalta määritellyn ajan kuluttua. Tällöin sertifikaatti pitää uusia sen antajan ilmoittaman ajan kuluessa, tai sertifiointi raukeaa.

### React.js

React on JavaScript-kirjasto, jota käytetään käyttöliittymän ohjelmointiin.

### Scrum

Scrum on ketteriin menetelmiin kuuluva projektinhallinnan malli. Se perustuu lyhyisiin iteraatioihin, joista jokainen pohjautuu edellisessä syklissä, sprintissä, tehtyyn kehitystyöhön. Jokaisessa sprintissä tehtävä työ suunnitellaan suunnittelupalaverissa ja sprintin päätteeksi tehdään katselmus, jossa tarkastellaan tehtyä työtä. Tämä mahdollistaa joustavan kehittämisen siihen suuntaan, jota projektin onnistunut toteuttaminen vaatii.

### Vue.js

Vue.js on JavaScript-sovelluskehys, jolla ohjelmoidaan käyttöliittymiä.

## 2 Lähtötilanteen kuvaus

Tässä luvussa selvitetään seuranta edeltävä lähtötilanne. Mitä erilaisia tehtäviä työ sisältää, mitä sidosryhmiä työlläni on ja millaisia vuorovaikutustilanteita työssä on.

### 2.1 Oman nykyisen työn analysointi

Työtehtäviin kuuluu ensisijaisesti ohjelmistokehitystyö asiakasprojektissa. Työtehtävät vaihtelevat projektista riippuen. Tämän lisäksi työsuhteen alussa työtehtäviin kuuluu omien ammatillisten taitojen kehittäminen. Esihenkilöni ja minulle osoitetun mentorin kanssa käymieni keskustelujen perusteella opiskelen C#-kieltä sekä Microsoft Azure Fundamentals -sertifikaattia varten. Tarkoituksena on lisätä sekä omaa osaamistani sekä saada CV:ni näyttämään paremmalta potentiaalisia asiakasprojekteja varten.

Työtehtävistä suoriutumiseen tarvitaan osaamista eri koodikielistä sekä versionhallinnasta. Tiimimme on erikoistunut Microsoft-pohjaisiin teknologioihin ja osaamista on vahvasti C#:sta. C# onkin yksi ensimmäisistä asioista, jonka opiskeluun panostan. Itse työ tehdään projekteissa, joissa työskennellään ketterin menetelmin esimerkiksi Scrumia hyödyntäen. Scrumin toimintatapojen, kuten päivittäisten daily-tapaamisten, sprint demojen ja retrospektiivien taitaminen ja sprintin aikataulussa työn suorittaminen on oleellinen osa työnteon rytmiä. Lisäksi kyky työskennellä ryhmässä erilaisten ihmisten kanssa ja paineen alla on erittäin tärkeää. Näkisin tärkeäksi taidoksi myös kyvyn ymmärtää asiakasta ja puhua asiakkaalle hänen ymmärtämin termein.

Minulla on useiden vuosien kokemus asiakaspalvelusta ja koen olevani taitava asiakastilanteissa. Tämän lisäksi olen kartuttanut teknistä osaamistani opinnoissani ammattikorkeakoulussa. Tällä hetkellä osaan seuraavia kieliä: Java, React.js, React Native, HTML5 ja CSS3. Opintojeni aikana olen saanut harjoitella projektityöskentelyä ketterin menetelmin, prototyypin rakentamista ja oppia lisää käyttäjäkokemuksesta.

Tällä hetkellä haluan profiloitua front end-ohjelmointiin, mutta voin nähdä itseni myös full stack -kehittäjänä. Olen kiinnostunut myös käyttäjäkokemuksesta sekä käyttöliittymäsuunnittelusta. Minulla on vielä paljon opittavaa, mutta näkisin minulla olevan jo hyvät valmiudet alalla työskentelemiseen. Kaikkein tärkein asia on kuitenkin halu oppia uutta ja kyky mukautua muuttuviin tilanteisiin.

Haluan luoda vahvan pohjan mahdolliselle full stack-kehittämiselle ja C#-ohjelmointikielen opiskelu onkin ensimmäinen askel siihen suuntaan. Näin työuran alussa minun tulee panostaa paljon uuden oppimiseen ja uusien työtapojen omaksumiseen. Näkisin oman osaamiseni olevan hyvällä

aloitustasolla aloittelevaksi ohjelmistokehittäjäksi, vaikka vielä onkin paljon opittavaa. En myöskään koe vielä omaavani kokeneemman ammattilaisen itsevarmuutta.

## 2.2 Sidosryhmien esittely

Sidosryhmät kytkeytyvät vahvasti projektityöskentelyyn. Sidosryhmällä tarkoitetaan tahoja, jolla on jonkunlainen intressi projektin lopputulokseen. Kiinnostuneet tahot voidaan jakaa sisäisiin ja ulkoisiin sidosryhmiin. Sisäisiin sidosryhmiin kuuluvat yleensä organisaation työntekijät ja johto. Ulkoiset sidosryhmät puolestaan muodostuvat esimerkiksi yhteistyökumppaneista, sisällöntuottajista ja esimerkiksi tavarantoimittajista. (Mäntyneva 2016, 124.)

Työpaikan sisäisiin sidosryhmiin kuuluvat minulle osoitetut tukihenkilöt, mentori ja buddy. Mentorin tarkoitus on ohjata ja tukea ammatillisessa kasvussa. Buddyn tehtävä puolestaan on tutustuttaa uusi työntekijä yritykseen ja sen tapoihin. Heidän lisäksi sidosryhmiin kuuluvat samassa projektissa työskentelevät kollegat, esihenkilöt ja laajemmin ajateltuna kaikki sellaiset yrityksen sisäiset tahot, joilla on esimerkiksi taloudellisia intressejä asiakasprojekteja koskien (esimerkiksi yrityksen johto). Tekemäni työ vaikuttaa projektin etenemiseen ja lopputuloksen onnistumiseen, mikä taas vaikuttaa asiakkaan tyytyväisyyteen ja mahdollisesti tuleviin asiakkuuksiin yrityksemme kanssa.

Ulkoisia sidosryhmiä ovat asiakkaat, niiden edustajat ja asiakkaan ratkaisua käyttävät loppukäyttäjät. Asiakasta kiinnostaa ajallaan valmistuva projekti ja se, että projektin lopputulema on siihen panostettujen resurssien, kuten pääoman ja ajan, arvoinen. Loppukäyttäjää taas voisi kiinnostaa toimiva, helppokäyttöinen ratkaisu, joka tuo hänelle jotakin lisäarvoa.

Oman työni kannalta keskeisiä sidosryhmien intressejä on projektin aikataulussa pysyminen sekä toimivan tuotteen luominen. Kehittäjänä sitoudun projektiorganisaatiossa sovittuihin aikarajoihin, työtapoihin ja työmääriin. Tunnistan, että omalla osaamisellani voin vaikuttaa lopputulokseen. Kuvassa 1 esitetään työni kannalta keskeiset sidosryhmät.





Kuva 1. Sidosryhmät.

### 2.3 Työpaikan vuorovaikutustilanteet

Työssä tapahtuvat vuorovaikutustilanteet tapahtuvat niin kasvotusten kuin myös virtuaalisesti.

Työtovereiden kanssa keskustellaan Teamsissa tai toimistolla esimerkiksi kahvitilassa. Suurin osa kollegoistani tekee töitä etänä, joten kohtaamiset toimistolla ovat hyvin harvassa. Tiimillämme on oma Teams-ryhmä, jossa lähinnä tiedotetaan tiimiä koskevista ajankohtaisista asioista. Tiimimme tapaa kerran kuussa Teamsissa yksikköinfon merkeissä, jossa käydään läpi ajankohtaisia asioita. Muutoin itse tiimin sisäiset vuorovaikutustilanteet ovat vähissä.

Yrityksen sisäisiä vuorovaikutustilanteita on myös asiakasprojektien projektiryhmissä. Projektiryhmä tapaa Teamsissa (tai muussa vastaavassa kanavassa) esimerkiksi daily-palaverissa. Projektiryhmän jäsenet viestivät myös muuten keskenään esimerkiksi mahdollisia ongelmia tai epäselvyyksiä selvitettäessä.

Asiakkaiden kanssa kommunikointi tapahtuu projektissa määritellyn kanavan kautta, esimerkiksi virtuaalisesti tavaten Teamsissa. Jotkut projektit voivat vaatia asiakkaan tiloissa työskentelyä.

Etänä tapahtuvissa vuorovaikutustilanteissa on suurempi mahdollisuus väärinkäsityksiin, etenkin jos viestintä tapahtuu ainoastaan viesteillä. Kirjoitetuista viesteistä voi esimerkiksi käsittää väärin kirjoittajan äänensävyyn, minkä takia onkin tärkeää kiinnittää huomiota siihen, miten asioita ilmaisee.

Koen omaavani hyvät vuorovaikutustaidot. Voisin kehittää omaa viestintääni kuitenkin siinä, miten ilmaisen asiani ongelmallisessa tilanteessa. Poikkean herkästi asiasta ja asian punainen lanka hukkuu. Toinen syy epäselvään viestintään voi olla se, että en osaa selittää tai en täysin ymmärrä ongelman ydintä.

### 3 Seurantajakson raportointi viikkoanalyysiin

Seuraan ja reflektoin omaa tekemistäni töissä seuraavan kahdeksan viikon ajan. Joka viikon päätteeksi otan esittelyyn aiheen, joka on noussut esiin viikon aikana.

#### 3.1 Seurantaviikko 1: Microsoft Azure -pilvipalvelu

Maanantai 21.2.2022

Minulla ei ole vielä asiakasprojektia, joten aion käyttää työpäiväni opiskellen ja kasvattaen omia ammatillisia vahvuuksiani. Olen opiskellut kaksi viikkoa Microsoft Azure Fundamentals -sertifikaattia varten ja jatkoin opiskelua tänään. Luen kurssin teoriaa läpi toista kertaa ja keskityin niihin osa-alueisiin, jotka ovat olleet minulle hankalimpia.

Päivän aikana puhuin mentorini kanssa ammatillisesta kasvusta, viestinnästä työympäristössä sekä epäonnistumisesta. Etenkin työuran alussa epävarmuus omasta osaamisesta ja pelko virheen tekemisestä on melko suuri. Useimmiten virhe ei aiheuta kenellekään hengenvaaraa, ja virheet ovat yksi parhaita tapoja oppia. Uutena työntekijänä haluaa tehdä vaikutuksen muihin organisaation jäseniin ja armollisuus omaa itseä sekä oppimista ja tekemistä kohtaan voi olla hankalaa.

Saavutin tavoitteeni tänään. Uuden oppiminen on pitkä ja monisyinen prosessi ja siksi on vaikea arvioida, kuinka paljon on oppinut yhden päivän aikana. Koen onnistuneeni selkeyttämään itselleni tänään läpikäymiäni Azure-palvelun konsepteja.

Tiistai 22.2.2022

Päivän tavoite oli jatkaa Azure Fundamentals -materiaalin kertausta. Tänään keskityin Azuren turvallisuusominaisuuksiin. Kaikista haastavinta palvelun eri osa-alueiden opiskelussa on se, että monet palvelut ja sovellukset tekevät osittain samoja asioita tai eroavat toisistaan hyvin vähän. Sertifiointikokeen kysymykset ovat yksityiskohtaisia ja suurpiirteinen ymmärrys ei riitä kokeen läpäisemiseen.

Keskiviikko 23.2.2022

Päivän tavoitteet ovat tiimimme uuden viestintäkanavan suunnittelu, Azuren opiskelu sekä mahdolliset muut ylimääräiset tehtävät, joita uuden työntekijän täytyy suorittaa. Tarkoitin tällä esimerkiksi yrityksen toimintaperiaatteisiin tutustumista intrassa.

Sovin aiemmin esihenkilöni kanssa suunnittelevani tiimillemme uutta viestintäkanavaa Microsoft Teamsiin. Olen ehdottanut suunnittelua visuaalisen yhteistyöalusta Miron avulla. Aloitin työpäivän pohjustamalla perjantaisen suunnittelukokouksen materiaaleja Miroon ja kokoamalla omia ajatuksiani siitä, miten uusi kanava olisi järkevintä organisoida.

Päivän aikana jatkoin pilvipalveluiden opiskelua. Aikomuksenani on tehdä ensi viikolla harjoituskoe ja sen perusteella joko varata aika sertifiointikokeeseen tai jatkaa opintoja niiltä osa-alueilta, joissa minulla on vielä lisää opittavaa.

Torstai 24.2.2022

Haluan omistaa ainakin yhden päivän viikosta C#-opintoihini. Harjoitustyöni on elokuvakirjasto, jonka toteutan ASP.NET Core MVC-arkkitehtuurilla. Minulla ei ole aiempaa kokemusta C#-ohjelmointikielestä, joten harjoitustyö etenee melko hitaasti. En halua kiirehtiä seuraamani tutoriaalin läpi, vaan aikomuksenani on syvälinen oppiminen ja osaamisen soveltaminen. Harjoitustyöhön on yhdistetty myös SQL-tietokanta sekä projektinhallinta Azure DevOpsissa. Projektin versionhallintaan käytän Gitiä.

Odotuksistani huolimatta harjoitustyö eteni päivän aikana paljon ja pian minulla on sovelluksen runko ja toiminnallisuudet valmiina.

Perjantai 25.2.2022

Päivän tavoite oli luoda tiimillemme uusi ryhmä Teamsiin, sillä tällä hetkellä käytössä oleva ryhmä ei palvele tiimimme viestintää toivotulla tavalla. Ryhmä on jaettu useaan eri kanavaan ja käyttäjä ei välttämättä tiedä, mille kanavalle oma viesti kannattaisi laittaa. Tiimin jäsenet eivät myöskään pidä yhteistä Teamsia tiimin ensisijaisena viestintäkanavana, vaan viestivät apua tarvitessaan esimerkiksi sähköpostitse.

Aloitin työpäivän jatkamalla eilistä C#-harjoitustani noin tunnin verran, kunnes aloitin uuden Teams-ryhmän hahmottelun esihenkilöni kanssa. Olin suunnitellut tapaamista jo hieman etukäteen jäsentelemällä omaa ehdotustani uuden ryhmän kanavista. Löysin myös ideointia varten "reverse brainstorming"-mallin, mutta en perehtynyt malliin sen syvämmässä etukäteen. Reverse brainstorming ei kehnosta valmistautumisestani johtuen juurikaan toiminut. Keskustelimme myös pitkään uuden Teams-ryhmän kielestä. Tiimimme on tällä hetkellä täysin suomenkielinen, mutta tulevaisuudessa tulemme todennäköisesti saamaan myös englanninkielistä vahvistusta. Teams-ryhmän täytyy pystyä mukautumaan tiimissämme tapahtuviin muutoksiin, joten Teams-ryhmän kieleksi valikoitui englanti. Ryhmän keskustelut pitäisi pystyä kääntämään englanniksi Teamsin Translate-toiminnon avulla. Sitä ei meidän ohjelmassamme ole asennettuna, joten selvitän asiaa

lisää IT-osastoltamme. Päivän päätteeksi sain luotua ryhmän, ja ensi viikolla otan tehtäväkseni pyytää vapaaehtoista henkilöä ryhmän ylimääräiseksi ylläpitäjäksi.

Tämän lisäksi kirjoitin tänään yrityksen sisäistä CV:tä, jonka avulla minua voidaan ehdottaa asiakasprojekteihin.

### Viikkoanalyysi

Tämän viikon olen suurimmaksi osaksi käyttänyt Microsoft Azure-pilvipalvelun opiskeluun. Microsoft Azure on Microsoftin pilvipalvelu. Azure kattaa suuren joukon palveluita virtuaalikoneista verkkopalveluiden julkaisemiseen ja analytiikkaan. Microsoft mukautuu alan kehitykseen sekä yritysten muuttuviin tarpeisiin julkaisemalla uusia tuotteita Azureen. (Microsoft 2022a.)

Microsoft Azure tarjoaa Infrastructure as a Service (IaaS), Platform as a Service (PaaS) sekä Software as a Service (SaaS) -ratkaisuja. IaaS-palvelulla tarkoitetaan sitä, että pilvipalvelun tarjoaja hoitaa kuntoon infrastruktuurin (palvelimet, palvelinsalit, tallennustila ja verkkoyhteydet) ja pitää sitä yllä omissa tiloissaan. Asiakas voi käyttää pilvipalveluntarjoajan resursseja internetin välityksellä oman tarpeensa mukaisesti ja maksaa palvelusta käyttönsä mukaan. PaaS tarjoaa asiakkaalle vapauden keskittyä ohjelmistokehitykseen sekä ohjelmistojen julkaisuun ja hallintaan, sillä pilvipalveluntarjoaja huolehtii laitteiston, palvelinten ja verkkoyhteyksien lisäksi käyttöjärjestelmästä ja sen ylläpidosta. SaaS-mallissa pilvipalveluntarjoaja hoitaa kaiken infrastruktuurista käyttöjärjestelmään sekä käytettävään ohjelmistoon. Asiakas tekee ainoastaan tunnukset ohjelmistoon ja maksaa käytöstään. (IBM 2021.)

Pilvipalveluiden ehdoton etu on käyttötarpeen ja asiakasliikenteen mukaan skaalautuvat palvelut. Azurella on palvelinsaleja ympäri maailman, mikä mahdollistaa nopean palvelun kaikkialla maailmassa. Azuren palvelujen korkea saatavuus pohjautuu useisiin alueisiin (regions) ja alueiden sisällä oleviin saatavuusalueisiin (availability zones). Availability zonella tarkoitetaan yhtä tai useampaa konesalia, jotka ovat fyysisesti erillään toisistaan. Tällä varmistetaan palvelun saatavuus vikatilanteissa – jos yksi datakeskus esimerkiksi menettää sähköt, toinen jatkaa vielä toimintaansa. (Microsoft 2022b.)

Microsoft Azure tarjoaa laajasti palveluita sovelluksen julkaisemisesta sen ylläpitoon, päivittämiseen, ja turvaamiseen. Azurella käyttäjä voi hallinnoida sitä, kuka saa tehdä mitäkin asioita organisaation palvelulle. Tämän lisäksi Azure monitoroi sovelluksen toimintaa ja ehdottaa kannattavampia toimintatapoja niin liiketoiminnan kuin turvallisuudenkin kannalta. Viikon aikana opiskelin näitä asioita Microsoft Learnin materiaaleista ja pääsin testaamaan joitakin toiminnallisuuksia kuten resurssien lukitsemista Azure Sandboxilla. Resurssien lukitsemisella estetään resurssien muuttaminen tai poistaminen.

Ensi viikolla haluan kokeilla uusia opiskelutapoja, sillä hidaskäyttötahti pilvipalveluiden suhteen hieman huolestuttaa minua. Tavoitteenani on saada sertifikaatti mahdollisimman pian, esimerkiksi seuraavan kahden viikon kuluessa.

Vaikeimpia osa-alueita minulle ovat olleet infrastruktuuriin ja sen turvallisuuteen liittyvät asiat sekä tunnistamiseen ja ylläpitoon liittyvät konseptit. Näkisin tämän johtuvan siitä, että minulla ei ole aiempaa tietoa tai osaamista kyseisiltä alueilta ja siksi perusasioidenkin ymmärtäminen ja oppiminen on hidasta. Helpointa tähän mennessä on ollut yleisesti Azuren palvelukonseptien hahmottaminen ja sisäistäminen. Esimerkki tällaisesta on juuri aiemmin mainitsemani saatavuusalueet, joihin Azuren toiminta hyvin vahvasti pohjautuu.

Pohtiessani oppimaani tunnistan oppivani helposti monimutkaisiakin konsepteja, mutta lähinnä suurpiirteisesti. Oppimismateriaalit menevät hyvin pikkutarkkoihin asioihin ja kuvittelisin juuri tällaisten asioiden olevan minulle haastavia sertifiointia ajatellen. Kokeeseen valmistautuminen on hyvin pitkälti asioiden ulkoa opettelua ja harjoituskokeen perusteella kokeen kysymykset voivat mennä yksityiskohtiin. Tilanne on kovin erilainen oikeaan elämään verrattuna, sillä tosi tilanteessa minun tarvitsee tietää asiasta sen verran, että tiedän millä hakusanoilla lähdän etsimään lisätietoa, mikäli se on tarpeen.

### **3.2 Seurantaviikko 2: ASP.NET Core MVC-malli ohjelmistokehityksessä**

Maanantai 28.2.2022

Päivän tavoite on edetä C#-harjoitusprojektissa. Sovellus on oikeastaan rungoltaan valmis, ja tämän päivän agendalla onkin projektin julkaiseminen Azureen. Harjoittelen julkaisua ensin valmiilla ASP.NET Core-demoprojektilla ja sen jälkeen julkaisen oman työni. Teen näin, jotta saan toistoa ja toivottavasti paremman muistijäljen prosessista.

Ajauduin ongelmiin jo demoprojektin ajamisessa. Sovellus herjasi IIS-sertifikaattia, mutta sertifikaatin asennus ei auttanut. Pyysin apua mentoriltani, sillä en osannut ratkaista tätä ongelmaa. Lopulta selvisi, että sovellus kaatui yrittäessään hakea selaimessa näytettävää tietoa tietokannasta ennen kuin lokaalia tietokantaa oli edes olemassa.

Azuressa portalissa perustin ensin Azure App Servicen web-sovelluksen hostaamista varten. Tämän jälkeen loin tietokannan Azure SQL Databaseen. Yritin julkaista .NET-sovellukseni ja yhdistää sovelluksen pilveen perustettuun tietokantaan. Tämä ei kuitenkaan onnistunut, sillä minulla on kaksi Azure-tilausta. Olin käyttänyt Azure Portalissa Azuren ilmaista kokeilutilausta ja Visual Studiossa julkaistessani sovellusta ja yhdistäessäni sitä Azureen jouduin käyttämään Visual Studion lataamisen mukana tullutta Azure-tilausta. En voinut valita Visual Studiossa mitään muuta

kuin Visual Studio -tilauksen, joten menin vikaan valitessani Azuren kokeilu-tilauksen Azure Portalissa. Tätä ongelmaa ei olisi syntynyt, jos olisin yrittänyt julkaista sovelluksen kokonaan Visual Studion julkaisu-toiminnon kautta Azureen.

Peruin Azuresta tekemäni muutokset ja poistin demoprojektin. Latasin demoprojektin uudelleen voidakseni julkaista sovelluksen uudestaan eri Azure tilaukselle, mutta en saanut projektia taaskaan ajettua. Tällä kertaa sovellus ilmoitti, että se ei löydä projektin View-tiedostoja. En löytänyt Stackoverflow-sivustolta sopivaa ratkaisua siihen, miksi .cshtml-tiedostot View-kansiossa eivät yhtäkkiä löytyneet. En ollut tehnyt itse uudelleenladattuun projektiin mitään muutoksia. Cshtml-tiedosto on C# HTML-tiedosto, joka näytetään selaimen verkkosivuna.

Tänään pääsin tavoitteeseeni siinä mielessä, että pääsin kokeilemaan sovelluksen julkaisemista Azureen. Valitettavasti kuitenkin en osannut ratkaista itse omia ongelmiani julkaisemisen suhteen.

Tiistai 1.3.2022

Tämän päivän tavoite oli saada demoprojekti taas toimimaan, jotta voisin kokeilla uudelleen sen julkaisemista Azureen. Tämän lisäksi ajatuksena oli tehdä Azure Fundamentalsin harjoituskoe, jotta voisin nähdä mitä osa-alueita pitää vielä kerrata lisää.

Harjoituskoe meni huomattavasti paremmin kuin aiempina kertoina, mutta olin edelleen kaukana läpipääsystä. Sain alhaisimmat pisteet turvallisuus- sekä hallinto-osuudesta ja kertosin näitä osioita päivän aikana Microsoft Learn -portaalista. Microsoft Learn tarjoaa ilmaiset materiaalit Azure Fundamentals -sertifikaatin opiskeluun. Materiaalit selittävät kattavasti Azuren eri palveluita. Mukana on myös videoita, vaikkakin itse koen lukemisen ja muistiinpanojen tekemisen itselleni hyödyllisemmäksi opiskelutavaksi.

Päivän lopussa avasin vielä demoprojektin ja yritin selvittää, miksi ohjelma ei löydä .cshtml-tiedostoa ohjelman käynnistyessä. Selvittäminen oli haastavaa, sillä virheviestiä googlettamalla syitä ja ratkaisuja tilanteeseen löytyi useita erilaisia. Uuden kielen kanssa toimiessa ratkominen on hidasta ja raskastakin, kun ei täysin ymmärrä sovelluksen logiikkaa tai syntaksia. Jatkan tämän ratkomista huomenna työkaverini avustuksella

Pääsin päivän aikana osittain tavoitteisiini: tein Azure Fundamentals-harjoituskokeen kahdesti melko samanlaisin tuloksin, kertosin ja etsin lisää tietoa osa-alueista, jotka menivät huonosti kokeessa. Toisaalta kuitenkin epäonnistuin tavoitteessani ratkaista C#-demoprojektin ongelmat. Minun olisi hyödyllistä lukea lisää MVC-mallista ja sen toiminnasta.

Keskiviikko 2.3.2022

Päivän tavoite oli yrittää ratkaista MVC-demoprojektin käynnistämiseen liittyvät ongelmat, ja saada ohjelma julkaistua Azureen. Tämän lisäksi tarkoituksena on kerrata Azure Fundamentalsia.

Yritin ladata demoprojektin uudelleen useita kertoja, mutta en saanut projektia käynnistettyä onnistuneesti. Ohjelma väitti, että view-kansiossa olevat tiedostot eivät löydy. Kokeilin asettaa projektin Startup Projectiksi siltä varalta, että Visual Studio ei ymmärrä, mikä projekti kuuluu ajaa. Startup Projectiksi asettaminen ei muuttanut mitään. Uskon, että aiemmin tekemäni projektin poistaminen jätti jotain tiedostoja piiloon Visual Studioon, minkä takia uudelleen ladattu projekti ei toimi.

Heikosti menneen ongelmanmäärittelyn johdosta päädyin vielä opiskelemaan Azure Fundamentalsin yksityisyys ja hallinto -osuutta. En päässyt tänään päivän tavoitteeseeni, enkä koe olevani yhtään lähempänä ongelman ratkeamista.

Torstai 3.3.2022

Mentorini löysi eilen erilaisen ASP.NET-demoprojektin ja tutoriaalini Azureen julkaisemisesta. Päivän tavoite olikin kokeilla uudelleen demoprojektin julkaisua Azureen. Latasin Microsoftin sivuilta demoprojektin ja tutoriaalini ohjeiden mukaisesti julkaisin sovelluksen Azureen Visual Studion julkaisu-toiminnon kautta. Ensin perustetaan resurssiryhmä (resource group), johon sovelluksen eri resurssit (web-sovellus, tietokanta) lisätään. Tämän jälkeen luodaan App Service plan, jossa määritellään sovellusta hostaavan web palvelimen ominaisuudet, kuten palvelimen sijainti. Kun raamit sovelluksen julkaisemiselle ovat valmiit, lisätään Azureen tietokanta. Tälle harjoitussovellukselle halusin käyttää SQL-tietokantaa. Tietokannan luomisen jälkeen yhdistetään Azuressa oleva tietokanta koneella olevaan, paikalliseen tietokantaan. Tämä harjoitus sujui erittäin hyvin!

Päivän aikana tein myös Azure Fundamentals AZ-900 -harjoituskokeen. Teen harjoituskokeita muutaman päivän välein, jotta voisin arvioida, koska minun kannattaa varata aika varsinaiseen sertifiointikokeeseen. Tämän päivän tuloksen perusteella kerrattavaa on vielä jonkun verran, vaikka kehitynkin tasaiseen tahtiin.

Pääsin päivälle asettamaani tavoitteeseen. Päivän aikana onnistuin julkaisemaan ASP.NET-sovelluksen Azureen ja ehdin vielä sen lisäksi opiskella Azurea. Tämän päivän onnistuminen tuntui hyvältä viime päivien vaikeuksien jälkeen.

Perjantai 4.3.2022



Tavoitteeni tälle päivälle on jatkaa Azuren opiskelua. Olen lukenut Microsoft Learningin tarjoaman kurssimateriaalin läpi kahdesti, ja tänään ajattelin etsiä vaihtoehtoisen kurssin esimerkiksi Youtubesta.

Päivän aikana aloitin katsomaan kattavaa videota Azure Fundamentalsista ja muokkasin aiemmin tekemiäni muistiinpanoja videolta oppimani mukaan. Microsoftin materiaalit ovat hyvin tekstipainotteisia, mikä on sopinut minulle hyvin. Yleisesti opin paremmin lukemalla ja tekemällä omia muistiinpanoja ymmärtämäni perusteella. Yleensä videolla asiat etenevät nopeasti ja usein joudun kelaamaan videota taaksepäin voidakseni kuunnella asian uudestaan. Microsoftin materiaalit esittävät eri Azuren palvelut yksi kerrallaan ja eri toimintojen yhteensopivuus on jäänyt itselleni hieman epäselväksi, vaikka tekstissä saatetaankin mainita eri palveluiden tukevan toisiaan. Videolla palveluita ja yleisesti niiden logiikkaa oli esitetty eri järjestyksessä kurssin aihealueisiin verrattuna ja Azuren palveluiden kokonaisuus alkoi hahmottua yhtenäisemmin.

Päivän aikana yritin vielä viimeisen kerran selvittää, miksi harjoitusprojektini on hajonnut. Olen etsinyt ratkaisua googlaamalla saamani error-ilmoituksen, mutta Stackoverflow on tarjonnut useita eri ratkaisuja ongelmaan. Yksikään ei kuitenkaan ratkaissut, miksi ohjelma ei löydä View-tiedostoja käynnistyessään. Kysyin apua pidempään koodanneelta kehittäjältä ja hän epäili joidenkin asetusten muuttuneen Visual Studiossa, mutta ei itsekään osannut suoraan paikallistaa ongelman juurta. Huomasimme, että toimivassa projektissa oleva .vs-tiedostokansio oli näkyvässä, mutta rikkinäisessä projektissa .vs-kansio oli piilotettu. Yritin liittää piilotetun kansion takaisin näkyväksi projektille, mutta tämä ei auttanut. Tässä tilanteessa ajankäytön kannalta minun on kannattavampaa tehdä uusi projekti, kun rikkinäisen projektin korjaaminen vaatisi todennäköisesti liikaa aikaa, eikä takeita korjautumisesta ole.

Olen tyytyväinen päivän aikana tekemääni edistykseen Azuren kanssa. Videon katsominen tässä vaiheessa tuntui olevan hyvä ratkaisu, sillä se auttoi minua hahmottamaan Azuren osa-alueita uudella tavalla. Pääsin tänäkin päivänä aamulla asettamaani tavoitteeseen, vaikkakin hajonneet projektit jäivätkin mysteeriksi.

### Viikkoanalyysi

MVC on ohjelmistokehityksen arkkitehtuurimalli, joka erottaa ohjelman kolmeen eri osaan: model (malli), view (näkyvä) ja controller (käsittelijä). Käytin viikon aikana paljon aikaa MVC-projektini tekemiseen ja sen ongelmien ratkaisemiseen.

Malli-tiedostoissa piilee ohjelman tarkoitus ja kaikki ne tehtävät, jotka malli suorittaa tarkoituksensa täyttämiseksi. Malleihin kirjoitettu koodi käsittelee tietokannassa säilytettävää dataa. (Microsoft 2022c.) Projektini Models-kansiossa ohjelman mallina toimii Movie.cs-tiedosto. Tiedosto sisältää

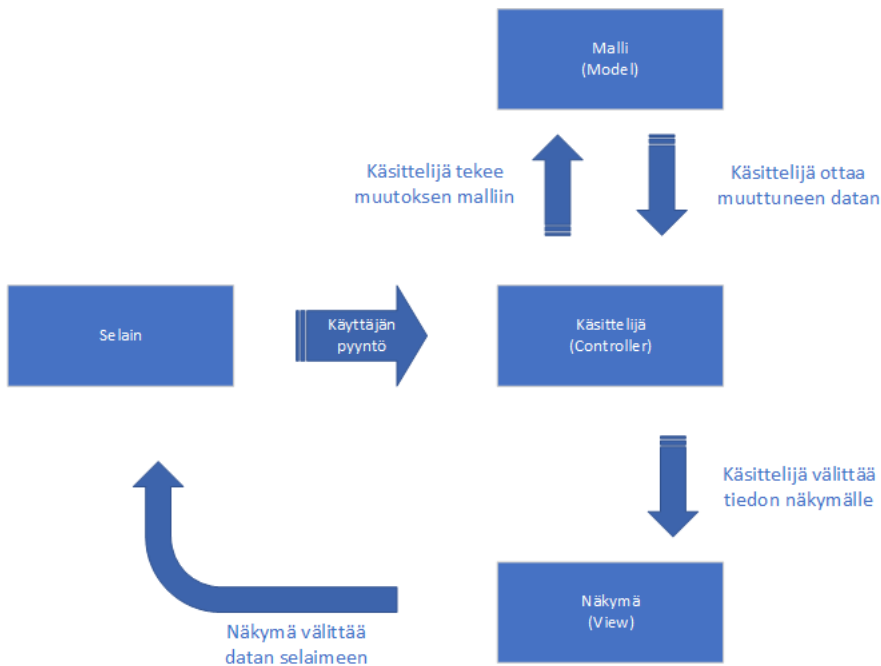
Movie-nimisen luokan, jossa määritellään elokuvasta näytettävät tiedot (nimi, julkaisupäivämäärä, genre, hinta ja luokitus). Luokassa myös annetaan määriytyksiä näille arvoille, esimerkiksi hinnan datatyyppi on määritelty valuutta ja julkaisupäivämäärän datatyyppi on päivämäärä.

Käsittelijä nimensä mukaisesti käsittelee käyttäjän tekemän toiminnon. Tämä tarkoittaa sitä, että käsittelijä tekee malliin muutoksia käyttäjän syöttämän tiedon mukaisesti. Käsittelijän tehtävä on myös valita asiakkaan päätteelle näytettävä malli. (Galloway, Wilson, Allen & Matson 2014, alaluku The controller's role; Microsoft 2022c.) Ohjelmassani on MoviesController, josta löytyy GET- ja POST-toiminnot kaikille näkymien toiminnallisuuksille. GET hakee tietoja tietokannasta näkyviin ja POST lähettää muuttunutta tietoa takaisin tietokantaan. Käsittelijä välittää näitä pyyntöjä ohjelmassa mallin ja näkymän välillä.

Näkymä-tiedostot määrittelevät sen, mitä malli-tiedostossa olevaa tietoa käyttäjälle näytetään. Tämän jälkeen näkymä muuttaa tiedot HTML-muotoon ja näyttää sen käyttäjälle selaimessa. Näkymä-tiedostoissa yhdistetään C#-syntaksia HTML-syntaksin kanssa. Tällaisia sivuja kutsutaan Razor-sivuiksi. (Microsoft 2022c.) Ohjelmassa nämä ovat .cshtml-tiedostoja ja ne löytyvät View-kansiosta. Painin tällä viikolla paljon sovellukseni ongelmien kanssa ja yritin selvittää, miksi cshtml-sivut eivät näy selaimessani, kun ajan ohjelman.

Elokuvaprojektini Views-kansiossa asuvat näkymät seuraaville sivuille: Index (etusivu), Create, Delete, Details ja Edit. Etusivulle haetaan kaikki tiedot tietokannasta foreach-toistorakenteen avulla. Jokaisen elokuvan rivillä on linkit rivin lisätietojen näyttämiseen, muokkaamiseen tai poistamiseen. Linkkiä klikkaamalla aukeaa uusi sivu (view), jossa toiminnon voi suorittaa.

Yksinkertaisesti kerrottuna käyttäjä suorittaa selaimessa toiminnon. Käsittelijä vastaanottaa pyynnön ja tarkistaa, vaatiiko kyseinen toiminto muutoksen tekemistä malli-tiedostoon. Mikäli näin on, käsittelijä tekee muutoksen. Tämän jälkeen käsittelijä palauttaa muuttuneen malli-tiedoston näkymälle, joka kääntää tiedot HTML-muotoon käyttäjälle heijastettavaksi.



Kuva 2. MVC-malli havainnollistettuna

ASP.NET Core on avointa lähdekoodia hyödyntävä kehys, jolla rakennetaan internetiin yhteydessä olevia sovelluksia. ASP.NET Coren on mahdollista hyödyntää pilvipalveluita, mikä tekee siitä moderniin ohjelmistokehitykseen sopivan kehyksen. ASP.NET Core mahdollistaa myös Razor-sivujen sekä Blazorin käyttämisen ohjelmoinnissa. Razor-sivuilla tarkoitetaan sellaisia HTML-sivuja, joihin on lisätty C#-koodia. Blazorilla puolestaan tarkoitetaan C#-koodin käyttämistä yhdessä JavaScriptin kanssa. (Microsoft 2022d.)

ASP.NET Coren MVC-mallin hyödyntäminen ohjelmistokehityksessä on kannattavaa, sillä kehys on ketterämpi edeltäjiinsä verrattuna ja vaatii vähemmän muistia. Tämän vuoksi ohjelma käynnistyy nopeasti. ASP.NET Core MVC on cross-platform kehys, ja sitä voi käyttää Windowsilla, Linuxilla ja MacOS:lla. Kehittäminen tapahtuu yhdellä alustalla riippumatta siitä, onko kyseessä pilvi- vai asiakkaan fyysiselle palvelimelle rakennettu ratkaisu. (Ciliberti 2014, alaluku 1.4.) ASP.NET Core MVC sopii erinomaisesti moderneja, skaalautuvia ohjelmia kirjoittaville ohjelmistokehittäjille.

MVC-arkkitehtuuri on minulle aiemmin tuttu Java-ohjelmoinnista ja sovelluksen tekeminen on tästä syystä tuntunut konseptina helpolta. Vaikeuksia harjoitusprojektissa ei ole aiheuttanut välttämättä itse MVC-malli, vaan muut asiat sen ympärillä, kuten tällä viikolla selvittelemäni virhetilanne. Minulla ei ole koskaan ollut vastaavanlaisia ongelmia ohjelman käynnistämisen kanssa. MVC-mallin tarkempi tutkiminen on selkeyttänyt arkkitehtuurin eri osien toimintaa. Esimerkiksi käsittelijän rooli on avautunut itselleni aivan uudella tavalla ja koko mallin idea on kirkastunut.

### 3.3 Seurantaviikko 3: Työyhteisöviestintä Teamsissa

Maanantai 7.3.2022

Päivän tavoite on katsoa loppuun perjantaina aloittamani Azure Fundamentals -luento Youtubesta ja muokata jo tekemiäni muistiinpanoja videon perusteella.

Videolta opiskeleminen on itselleni hidasta, sillä keskeytän ja kelaan videota taaksepäin ehtiäkseni kirjoittamaan muistiinpanoja. Kuten aiemmin kerroin, minulle luontevampaa lukea tietoa. Halusin kuitenkin kokeilla videon katsomista siitä syystä, että yleensä suullisesti esitettynä teorian voi kertoa eri tavoin ja sellaisilla sanallisilla esimerkeillä, joihin virallisempi tekstimuoto ei välttämättä taivu.

Opiskelu meni odotetunlaisesti, ei tullut yllätyksiä tai vastoinkäymisiä. Yritän vielä etsiä Pluralsight-alustalta lisää minulle sopivia materiaaleja opiskelun tueksi. Pluralsight on virtuaalisen opiskelun alusta IT-alan toimijoille. Palvelussa on kattavasti videoluentoja alan teknologioista.

Tiistai 8.3.2022

Aikomukseni tänään on katsoa luento Clean Codesta ja sen periaatteista ja kerrata Azuren preview-toimintoja.

Katsoin ensimmäisen osan Robert Martinin luentosarjasta Youtubesta. Hän luennoi siitä, miksi koodaaminen on hidasta ja millä tavoilla se voisi olla nopeampaa. Kaikki perimmillään pohjautuu siihen, että projekteissa pitää saada kiireessä valmiiksi ratkaisuja. Kun kehittäjä saa ratkaisun toimimaan, hän ei enää palaa siistimään tekemäänsä ratkaisua, vaan siirtyy seuraavaan tehtävään. Tuloksena useimmiten on sekavaa ja vaikealukuista koodia, jonka lukemiseen ja selvittämiseen menee aikaa. Ideaalitulanteessa kaikki koodattavat asiat suunniteltaisiin ja ohjelmoitaisiin huolellisesti ja lopuksi vielä koodi siistittäisiin helposti luettavaksi seuraavaa kehittäjää varten. Tähän ei varmaan ikinä päästä liikemaailman ja projektien aikatauluista johtuen. (UnityCoin 9.8.2019, 24:30–1:05:23 min.) Luento herätti paljon ajatuksia siitä, kuinka omalla panoksellaan voi helpottaa tai vaikeuttaa seuraavan kehittäjän työtä projektissa.

Luennosta inspiroituneena ajattelin kokeilla aloittaa uuden ASP.NET Core -projektin ja testata, suostuuko Visual Studio ajamaan projektin. En saanut ajettua juuri luomaani projektia. Sain taas samat herjat kuin viime viikolla. Tästä päättelin vian olevan Visual Studiossa ennemminkin kuin projekteissa itsessään, joten poistin Visual Studion asennuksen ja asensin sen uudelleen. Uudelleenasennus ja sen jälkeen yllättäen toimivat projektit entisestään vahvistivat käsitystäni siitä,

että olin viime viikolla onnistunut vahingossa muuttamaan tai hajottamaan jotain Visual Studiossa niin, että se ei onnistunut löytämään .cshtml-tiedostoja View-kansiosta, kun projektin laittoi päälle.

Kuukausi sitten aloittamani elokuvaprojekti siis toimii taas ja julkaisin sen Azureen. Azureen julkaisun aikana olin epähuomiossa tehnyt väärä valintoja SQL-tietokannan luomisessa Azureen ja aluksi sovellus ei sen takia auennut Azuressa. Korjasin Azuressa yhteysasetukset (connection strings) Azuren ja tietokannan välillä ja sovellus alkoi pyöriä Azuressa onnistuneesti. Tarkoituksena on seuraavaksi tehdä sovellukseen lisää toimintoja ja lisätä CI/CL pipeline Azure DevOpsissa.

Koen tämän päivän menneen todella hyvin siitä huolimatta, että päädyin tekemään aivan eri asiaa kuin olin aamulla suunnitellut ja koin pieniä vastoinkäymisiä Azuren kanssa.

Keskiviikko 9.3.2022

Päivän tavoite on tehdä Azure Fundamentals AZ-900 -harjoituskoe ja kerrata vielä viimeisiä asioita. Harjoituskokeen perusteella saatan varata ajan sertifiointikokeeseen.

Tein aamun alkajaisiksi harjoituskokeen ja yllätyksekseni pääsin kokeesta läpi. Kokeessa kysyttiin tällä kertaa asioita, joita en ollut opiskellut oikeastaan lainkaan. Huomasin myös, että minulla yleensä hyvin mennyt osa-alue (Describe Azure concepts) meni tällä kertaa heikoiten, kun taas yleensä heikoiten menneestä osa-alueesta (Describe security concepts) sain parhaan tulokseni koskaan. Olen asettanut harjoituskokeen kertomaan oikean vastauksen välittömästi, kun olen vastannut väärin ja olen huomannut ensimmäisen, intuitiivisen vastauksen olevan oikein monivalintakysymyksissä. Taidan kuitenkin ajatella liikaa, sillä vaihdan intuitiivisen vastauksen lähes aina järkeilyyn vastaukseen, joka useimmiten on väärin.

Kerrattuani vielä lisää tein kokeen uudelleen ja pääsin uudelleen läpi harjoituskokeesta. Tällä kertaa yritin simuloida oikeaa koetilannetta ja valmistautua näin oikeaan koetilanteeseen. Päivän päätteeksi varasin ajan ensi viikolle viralliseen sertifiointikokeeseen.

Pääsin tänään tavoitteeseeni usemmalla kuin yhdellä tasolla – opiskelin Azurea kuten olin suunnitellut, mutta usean viikon opiskelun tuloksena läpäisin harjoituskokeen kahdesti. Aikomuksenani on harjoitella koetta ja koetilannetta vielä useasti seuraavien työpäivien aikana, jotta oikea koetilanne ei tunnu niin jännittävältä.

Torstai 10.3.2022

Päivän tavoite oli käydä aiemmissa Azure Fundamentals -harjoituskokeissa väärinmenneitä kysymyksiä läpi ja harjoitella niitä osa-alueita, joita kysymykset käsittelivät.

Olin eilen pyytänyt IT-osastolta kääntäjä-toiminnon aktivoimista Microsoft Teams-sovellukseeni, jotta voisin testata sen toimintaa. Tämä liittyy muutama viikko sitten luomaani uuteen tiimin Teams-ryhmään ja esihenkilöni kanssa käytyyn keskusteluun siitä, millä kielellä Teamsissa kannattaisi viestiä. Tiimimme on tällä hetkellä täysin suomenkielinen, mutta tulevaisuudessa tiimiimme liittyyne myös englanninkielisiä työntekijöitä ja on tärkeää, että myös he saavat tarvitsemansa tiedon materiaaleistamme. IT-osastolta kerrottiin, että yrityksemme linjauksen mukaan käännös-ominaisuutta ei myönnetä. Keskustelin tästä esihenkilöni kanssa ja tulimme siihen tulokseen, että englanniksi viestiminen täysin suomenkielisessä yhteisössä – vaikkakin mahdollisesti myöhemmin englanninkielisten työntekijöiden hyödyksi – tuntuu väkinäiseltä. Lisäksi on epätodennäköistä, että uudet työntekijät tarvitsisivat chatissa käymiämme tietoja myöhemmin. Virallisemmat materiaalit ja ohjeet ovat englanniksi.

Tein tänään sen, mitä olin suunnitellut tekeväni. Yritän huomenna etsiä jonkin toisen harjoituskokeen, sillä minusta tuntuu, että osaan Microsoftin harjoituskokeen kysymykset osittain ulkoa. Tämä harjoituskoe ei enää realistisesti mittaa osaamistani, jos muistan oikeat vastaukset jo kysymystä lukiessani.

Perjantai 11.3.2022

Päivän tavoite on tutkia, löydäkö muita hyviä harjoituskokeita Azure Fundamentals -kurssin loppumetreille.

Päivän alussa selvittelin Teamsiin lisätyissä kansioissa olevaa hahmoikonin merkitystä. Microsoftin foorumeilla oli paljon kyselyitä OneDrivessä näkyvästä hahmoikonista, mutta ei juurikaan asiaa Teamsissa esiintyvistä ikonista. OneDrivessä hahmo merkitsee sitä, että tiedosto on jaettu jonkun kanssa ja jaon voi lopettaa klikkaamalla ikonia hiiren kakkospainikkeella ja muuttamalla tiedoston asetuksia. Useimmiten ratkaisuna on poistaa tiedoston jakamista varten luotu linkki. Tämä ei kuitenkaan auttanut Teamsin kanssa. Yritin siirtyä OneDriveen ja Sharepointiin muokkaamaan kansioden asetuksia, mutta mistään ei löytynyt jakamisen lopettamiseen vaihtoehtoa. Tiedostosta löytyi tieto, että tiedostoon pääsee käsiksi Teams-ryhmämme omistajat ja jäsenet. Kokeilin itse luoda testaukseen tarkoitettulle Sandbox-kanavalle kansioita ja siirtää niihin tiedostoja ja onnistuin toistamaan minulle raportoidun ongelman kyseisestä ikonista. Loin useampia kansioita ja ikoni poistui kansioista aina, kun loin uuden kansion. Epäilen, että ikoni vain ilmoittaa tiedostojen olevan julkisia jaettuina kaikille ryhmän jäsenille. Löysin ainoastaan yhden ketjun Microsoftin foorumilta, jossa käsiteltiin kyseisiä ikoneita Teamsissa. Siinä käyttäjä sanoi ikonin merkitsevän vain sitä, että kansio on jaettu muiden ryhmänjäsenten kanssa. En ymmärrä, miksi Teams ilmoittaisi ryhmässä jaetun tiedoston olevan ”jaettu” muiden käyttäjien kanssa? Eikö se ole juuri tiedoston lisäämisen tarkoitus?

Name	Modified	Modified By
Shared folder test	About an hour ago	Anni Vanhanen
Shared folder test 1	About a minute ago	Anni Vanhanen
Shared folder test 2	A few seconds ago	Anni Vanhanen

Kuva 3. Kuvaikoni Teamsissa olevassa kansiossa.

Päivän aikana tein myös Azure Fundamentalsin harjoituskokeita muilta sivustoilta, mutta kokeet eivät olleet yhtä laadukkaanoloisia, kuin Microsoft Learnin harjoituskoe. Tavoitteeni tälle päivälle oli melko yksinkertainen, mutta en koe löytäneeni yhtä pätevänoloista harjoituskoetta verkosta. Luulen, että palaan Microsoft Learnin pariin ensi viikolla. Aamulla tekemäni tutkimusmatka Teamsin tiedostonjako-ominaisuuksiin oli mukava tauko opiskelusta.

### Viikkoanalyysi

Viikon suurimman onnistumisen tunteen minulle tuotti Visual Studion aiheuttamien ongelmien ratkeaminen. Näin jälkikäteen ajateltuna en osaa selittää, miksi en yrittänyt poistaa ohjelman asennusta aiemmin. Ratkaisu tuntui ehkä liian radikaalilta ja ainoa asia mikä minua tässä jäi vaivaamaan, oli se, että en tiedä mitä asetuksia olen muuttanut ja millä tavalla.

Vaikka vietinkin suurimman osan tästä viikosta Azure-opintojeni kanssa, pohdin paljon oman työyhteisömme viestintää ja kuinka voisimme saada sitä parannettua. Miten voisimme viestinnällämme varmistaa, että uudet tiimiläiset tuntevat olonsa tervetulleiksi ja kuuluvansa tiimiin? Ja miten ylipäättään voisimme rakentaa yhteisömme viestintää sellaisella tavalla, että kokisimme olevamme tiimi?

Keisalan (2012, 51) mukaan monikulttuurisen työyhteisöviestinnän edistämisen kulmakiviä ovat muun muassa säännölliset tiimipalaverit, dialogisuuden edistäminen ja henkilökohtaiset keskustelut tiimin jäsenten kanssa. Kortetjärvi-Nurmi & Murtola (2016, alaluku Työyhteisöviestintä on vuoropuhelua) puolestaan esittää, että hyvä työyhteisöviestintä perustuu dialogin lisäksi avoimuuteen. Vuoropuhelussa eri näkemykset tulevat kuulluksi ja osallistujat voivat oppia ja kehittyä. Kun kaikilla on mahdollisuus osallistua ja tulla kuulluksi, työhyvinvointi kasvaa. Tämän lisäksi yhdessä tekemisen tunne voimistuu ja yhteisön jäsenet voivat tuntea itsensä motivoituneemmaksi.

Tiimimme on jakautunut ympäri Suomea ja yhteinen palaveri järjestetään jäsenten toiveesta joka toinen kuukausi. Eri paikkakunnille jakautunutta tiimiä yhdistää yhteinen Teams-ryhmä. Ryhmä on epäaktiivinen ja toimii lähinnä ajankohtaisten asioiden tiedotuskanavana. Tiimi ei oikein tunnu tiimiltä, vaan joukolta työntekijöitä, jotka ovat niputettu yhteen työnkuvan perusteella. Loin Teams-ryhmäämme taukuhuone-kanavan, jossa voisimme esimerkiksi pitää virtuaalisia kahvitaukoja yhdessä. Tarkoituksena on saada Teamsista matalan kynnyksen kommunikointikanava: paikka, jossa voi jutella työasioiden lisäksi myös muista, kuin työasioista. Ryhmän aktivoituminen on kuitenkin kiinni jäsenten halusta osallistua – yksinään tähän tavoitteeseen ei voi päästä. Millaisin keinoin hajautunutta ryhmää voisi siis yrittää yhdistää?

Mikäli tiimi muuttuu jossakin vaiheessa kansainväliseksi, edessä on uusi mukautumisen vaihe. Kielen vaihtuminen suomesta englanniksi saattaa olla haastavaa, etenkin jos on totuttu viestimään suomeksi. Uudet tulokkaat voivat elävöittää tiimiä oman kulttuuritaustansa mukaisilla viestintätavoilla. Uusilta ihmisiltä voimme oppia uusia toimintatapoja, saada uutta tietoa sekä haastaa käsityksiämme: voimme siis kehittyä tiiminä. (Lahti, 2014. Luku 13.)

### **3.4 Seurantaviikko 4: Minä oppijana**

Maanantai 14.3.2022

Päivän tavoitteeni on viimeistellä Azure Fundamentalsin opintoja ennen koetta. Tänään aion tutkia omaa oppimistani ja katsoa, löytäisinkö vielä uuden tavan oppia.

Tutkin eri tapoja jäsenellä ajatuksia ja oppimistyökaluja ja päätin tehdä käsitekartan oppimastani. Näin voin itse nähdä, miten jäsentelen asioita ja millaisia suhteita eri asioiden välille olen mielessäni luonut. Käsitekartan tekeminen olisi nopeampaa digitaalisesti, mutta halusin tehdä sen paperille. Ehkä käsin kirjoitettu ja piirretty käsitekartta jää paremmin mieleen. Mauri Åhlbergin (2018, alaluku Mitä käsitekartat ovat?) mukaan käsitekartat paljastavat paljon tekijänsä ajattelusta ja oppimisesta. Koska visuaaliset esitystavat, kuten käsitekartat vaativat paljon ajattelua, ne myös kehittävät sitä. Arvelen voivani tehdä uusia oivalluksia käsitekartan avulla.

En saanut päivän aikana tehtyä täydellistä karttaa. Paljon aikaa kului pohdintaan ja sommitteluun, jotta saisin eri palveluiden väliset suhteet esitettyä mahdollisimman järkevästi. Samaan aikaan täytyy kuitenkin muistaa, että en ole tekemässä virallista opetusmateriaalia tai muuta vastaavaa, vaan kaikki tuottamani materiaali on vain minua itseäni varten ja riittää, jos itse ymmärrän mistä on kyse.

Tiistai 15.3.2022



Päivän tavoite on jatkaa suureksi muodostunutta käsitekarttaa ja saada se valmiiksi.

Kaikista järkevintä minulle oli jakaa asiakokonaisuus pienempiin osiin, ja tehdä pienemmistä kokonaisuuksista käsitekartat. Jaoin alueet seuraavasti: core Azure concepts ja core Azure services, core solutions and management tools ja general security and network security sekä identity and governance ja cost management and service level agreements.

Päivitin myös Teamsissa olevia mallipohjia uuden brändin mukaiseksi. Hain intrasta uudistetut kalvopohjat ja muokkasin niistä tarvittavanlaiset esitykset. Käänsin materiaalin myös englanniksi kaiken varalta.

Tänään pääsin asettamaani tavoitteeseen, ja ehdin tehdä vähän ylimääräistäkin materiaalin päivittämisen merkeissä.

Keskiviikko 16.3.2022

Tänään aion tehdä viimeisen Azure Fundamentals -harjoituskokeen ja tehdä viimeisen kertauksen ennen huomista sertifiikaattikoetta.

Aloitin aamun katsomalla Youtubesta erään yrityksen videon heidän omasta AZ-900-harjoituskokeesta. Jokaisen kysymyksen kohdalla keskeytin videon, vastasin kysymykseen itse ja katsoin oikean vastauksen videolta. Kysymykset olivat kovin erilaisia kuin Microsoftin tarjoamassa harjoituskokeessa. Videon harjoituskokeessa kysyttiin paljon hinnoitteluasioita, joita en totta puhuakseni ole itse opiskellut lainkaan. Tämän jälkeen tein Microsoft Learnista löytyvän harjoituskokeen. En läpäissyt harjoituskoeetta (sain 68.8 % oikein, läpäisyyn vaaditaan 70 %). Tarkistin väärin menneet vastaukset ja huomasin tehneeni paljon huolimattomuusvirheitä, en esimerkiksi ollut lukenut kysymystä kunnolla.

Päivän loppuksi tein harjoituskokeen uudelleen, ja pääsin kirkkaasti läpi. Päivällä tekemäni harjoituskokeen kysymykset tuntuivat vaikeammilta kuin iltapäivällä tekemäni uusinnan kysymykset. Tällä hetkellä minusta tuntuu siltä, että läpikäisy voi olla myös täysin kiinni siitä, mitkä kysymykset kokeeseen arvotaan.

Tänään pääsin kyllä tavoitteeseeni. Päivän aikana sain työyhteisön toimintaan liittyvän tehtävän, mutta aion hoitaa sen vasta huomisen koetilaisuuden jälkeen.

Torstai 17.3.2022

Päivän tavoite on suorittaa Azure Fundamentals -sertifiikaatti ja suunnitella tiimimme workshopia Miroon. Miro on yhteistyöalusta, jossa useat käyttäjät voivat työskennellä yhtä aikaa.

Aloitin aamun ensimmäisen työprojektini perehdytyspalaverilla. Kokouksessa esiteltiin lyhyesti projektin tavoite ja sisältö. Minun työtehtäväni on visualisoida tietokannasta saatava data Qlik Sense -nimisen ohjelman avulla. Työ ei siis ole ohjelmointia, vaan enemmänkin datan käsittelyä helposti ja nopeasti ymmärrettävään muotoon. Uskon aiempien opintojeni käyttäjäkokemuksesta ja työkokemukseni olevan hyödyksi tässä projektissa. Seuraavien työpäivien aikana tehtäväni onkin tutustua tarkemmin Qlik Senseen työkaluna ja selvittää, millaisia visuaalisia esityksiä sillä voi tehdä.

Palaverin jälkeen suoritin Microsoft Azure Fundamentals sertifiointikokeen onnistuneesti. Varsinaisessa kokeessa tuli edelleen vastaan uusia termejä ja Azuren palveluita, joita en ollut tavannut opiskellessani pilvipalvelua. Azure Fundamentals-sertifiointia kohti opiskelevalle henkilölle suosittelisin käyttämään useita lähteitä tiedon hankkimiseen. En usko, että olisin päässyt kokeesta läpi pelkästään lukemalla Microsoft Learnista löytämiäni materiaaleja, vaikka ne olivatkin kattavat.

Iltapäivällä suunnittelin Mirossa workshopia tiimillemme. Workshopin tarkoituksena on tunnistaa mahdollisia ongelmia tiimin toiminnassa ja löytää sellaisia keinoja, joilla voisimme toimia paremmin. Miro on kätevä työkalu, sillä kaikki tiimin jäsenet voivat osallistua yhtä aikaa ideointiin. Äänestämme workshopissa muutaman sellaisen idean, joita haluamme jatkotyöstää. Äänestysjärjestys myös kätevästi alustalla ja äänestyksen tulos on saatavilla välittömästi äänestysajan päätyttyä.

Koen ylittäneeni itseni tänään. Päivään mahtui paljon erilaisia tehtäviä ja sertifiointin saaminen oli selkeä päivän – ja viime viikkojen – kohokohta.

Perjantain 18.3.2022 olin sairaana.

### Viikkoanalyysi

Tällä viikolla kiinnitin paljon huomiota erilaisiin oppimisen työkaluihin ja tekniikoihin valmistautuessani sertifiointikokeeseen.

Mietin alkuvuokosta paljon eri oppimistyyliä ja siihen usein liitettyä VAK-jakoa (visuaalinen, auditiivinen, kinesteettinen). Olen itse kokenut aina olevani visuaalinen ja myös kinesteettinen uuden oppija ja olinkin kovin tyytyväinen keksittyäni käsitekartan avulla opiskelun vielä viime tingassa. Perehdyin oppimisen teoriaan enemmän tällä viikolla, ja yllätyksekseni huomasin paljon kritiikkiä kyseistä mallia kohtaan.

Esimerkiksi Yalen Poorvu Centerin (2021a) sivujen mukaan yhteen ”oppijakategoriaan” rajoittuminen enemmänkin vaikeuttaa oppimista. Tutkimusten mukaan opiskelija ei välttämättä opi

parhaiten omaksumallaan opiskelutyyliä. Omaa oppimista pitäisikin tutkia perusteellisemmin – ajattelemisen ajattelemisen voi auttaa tunnistamaan useita sopivia tapoja. Tätä kutsutaan metakognitioksi (Yale Poorvu Center 2021b.)

Tässä kohdassa, kun uuden omien ammatillisten valmiuksien kasvattamisen ensimmäinen askel on otettu, onkin hyvä pohtia mitä olisin voinut tehdä toisin. Käytin noin kolme viikkoa pelkästään lukemiseen ja muistiinpanojen tekemiseen. Aluksi tein muistiinpanoja käsin paperille sen vuoksi, että käsin kirjoittamisesta voisi jäädä parempi muistijälki. Siirryin kuitenkin pian OneNoteen ja siitä vielä Wordiin, jotta muistiinpanojen tekeminen olisi nopeampaa ja kätevämpää. Neljännen viikon kohdalla aloin katsomaan videoita ja viimeisellä, viidennellä viikolla keksin piirtää käsitekartan. Käsitekartta olisi palvellut minua paremmin, jos olisin tehnyt sen paljon aiemmin. Kartan avulla hahmotin oppimaani eri tavalla ja näin Azuren eri palvelujen väliset yhteydet selkeämmin.

Nyt ajatellessani oppimisprosessiani olisin mahdollisesti hyötynyt enemmän siitä, että olisin edennyt hitaasti kirjoittaen muistiinpanoja käsin ja piirtäen käsitekarttoja. Tämä olisi antanut minulle aikaa käsitellä uutta tietoa. Minulla oli kuitenkin koko ajan kova tarve edetä nopeasti uuteen aiheeseen ja saada opinnot (ja sertifikaatti) suoritettua nopeasti. Tämän takia siirryinkin esimerkiksi kirjoittamaan muistiinpanoni digitaaliseen muotoon.

Miten voisin hyödyntää tätä oivallusta jatkossa? Minulla ei ole tiedossa varsinaisia kokeita lähitulevaisuudessa, vaikka paljon uutta täytyykin opetella. Seuraavaksi opiskelen Qlik Senseä ja sitä, kuinka voin työskennellä sitä käyttäen. Tämä työtehtävä on puhtaasti kokeilun kautta oppimista. Jos tätä ajatusta jatkaa esimerkiksi ohjelmointiin, en voi suoranaisesti hyödyntää oivalluksiani itsestäni oppijana. Ohjelmoinnissa vastaantulevat ongelmat tulee ennemminkin ratkaistua Googlea käyttäen, eikä ulkoa tarvitse muistaa kaikkea. Mikäli päädyn suorittamaan lisää sertifikaatteja, on nyt oppimastani hyötyä.

### **3.5 Seurantaviikko 5: Datat esittäminen visuaalisesti**

Maanantai 21.3.2022

Päivän tavoitteeni on tutustua Qlik Senseen ja luoda alustalle harjoitusprojekti datan visualisoimista varten.

Projektin perehdytyspalaverissa minulle suositeltiin Qlik Senseen tutustumista Youtubesta löytyvien tutoriaalien avulla. Avattuani ensimmäisen tutoriaalin, minulle kävi selväksi, että ohjelmaan tutustuminen ja tutoriaalin seuraaminen ei tule ainakaan aluksi olemaan kovin helppoa. Ohjelman käyttöliittymä on muuttunut jonkin verran ja videolla esitetyt toiminnot löytyivät osittain eri paikoista. Käytin todella paljon aikaa vain ohjelman selailuun ja tarvitsemiä toimintoja

etsimiseen. Qlik Senseen help-sivuilla olevat videot olivat myös vanhoja. Uskoisin pystyväni seuraamaan aloittamaani tutoriaalia, mutta se tulee olemaan hitaampaa kuin luulin. Loin Qlik Senseen itse testisovelluksen, mutta en päässyt vielä tästä pidemmälle. En ehtinyt lisätä sovellukseen dataa, jota olisin voinut visualisoida.

Iltapäivällä minulla oli palaveri tulevasta työpajasta. Palaverissa katsoimme läpi jo tekemiäni materiaaleja Mirossa ja muokkasimme niitä ymmärrettävämpään ja järkevämpään muotoon. Keskustelimme pitkään workshopin järjestämiseen liittyvistä asioista, kuten esimerkiksi osallistujien ideoiden kirjoittamisesta etukäteen ja millä tavalla äänestys kannattaa järjestää jatkokehittettäviä ideoita varten. Ideoiden lisääminen etukäteen työalustalle on suotavaa, sillä kaikki eivät välttämättä pääse paikalle ja muutenkin on hyvä idea antaa osallistujille aikaa pohtia asiaa ja lisätä ehdotuksia. Uskon, että saamme näin eniten hyötyä irti työpajasta. Miro on alustana todennäköisesti tuntematon suurimmalle osalle tiimistämme, joten kirjoitin alustalle näkyviin lyhyet ohjeet työpajassa tarvittaviin työkaluihin. Pidän kovasti perehdyttämisestä ja ohjeiden kirjoittaminen on minulle mieluisaa tekemistä.

En varsinaisesti päässyt tänään aamulla asettamaani tavoitteeseen, sillä oletin pystyväni tekemään kokonaisen tutoriaalipäivän aikana. Olen kuitenkin tyytyväinen iltapäivällä tekemääni edistykseen työpajan suunnittelun kanssa.

Tiistai 22.3.2022

Tänään asetin tavoitteekseni tehdä Qlik Senseen harjoitussovelluksen ja harjoitella datan visualisointia. Tehtävälisällani tänään on myös koko tiimin kutsuminen Miron workshop-laudalle.

Hylkäsin eilen tekemäni testisovelluksen ja latsin tutoriaalipäivän kylkiäisenä tulevan valmiin tiedostopakettin. Tutoriaalia Youtubesta seuraten sain tehtyä liiketoiminnan operaatioita visualisoivan sovelluksen Qlik Sensellä. Koen saaneeni palvelun toimintalogiikasta kiinni tänään, mutta en vielä pystyisi itse suoriutumaan visualisoimisesta kovinkaan hyvin – eikä sellaista vielä yhden harjoituksen perusteella varmasti voi edes odottaakaan. Vaikeinta oli selvittää mistä käyttöliittymän uudistusten myötä eri paikkoihin siirtyneet toiminnot löytyivät. Mielestäni toiminnallisuuksia oli siirretty kovin erikoisiin paikkoihin käyttöliittymässä, ne eivät aina löytyneet aloittelijan silmiin loogisimmanolaisesta paikasta. Yhden harjoituksen perusteella näkisin kuitenkin kehittyväni palvelun käytössä sekä datan käsittelyssä nopeastikin, mikäli saan tarpeeksi toistoja työhön.

Päivän päätteeksi löysin Pluralsightista tunnin mittaisen tutoriaalipäivän samasta aiheesta. Ehdin katsoa tutoriaalipäivän puoleksi ja jatkan huomenna. Pluralsight on IT-alalle videoituja koulutuksia tuottava yritys.

Saavutin tälle päivälle asettamani tavoitteen.

Keskiviikko 23.3.2022

Aikomukseni on tehdä uusia sovelluksia Qlik Senseen ja tulla tutummaksi käyttöliittymän kanssa. Iltapäivällä osallistun projektin palaveriin ja saan toivottavasti lisätietoa projektin aikataulusta ja muusta.

Päivän aikana loin kaksi sovellusta, joista toisen taisin rikkoa ennen kuin pääsin edes luomaan graafeja. Jotain meni pieleen datatiedoston lisäämisessä, enkä saanut enää poistettua väärin lisättyä tiedostoa palvelusta. Sovelluksen voi poistaa, mutta ladattu datatiedosto jää asumaan palveluun. Näin jälkepäin tilannetta miettiessäni huomasin, että en yrittänyt etsiä ratkaisua ongelmaan esimerkiksi Googlesta vaan siirryin toisen sovellusharjoituksen ja datatiedoston pariin. Voisin huomenna vielä selvittää missä menin vikaan ja enkö todella voi poistaa lataamaani tiedostoa palvelusta.

Projektipalaverissa selvitettiin kehitysympäristön rakentamisen kestävän kolmesta neljään viikkoa. Oma työni projektissa alkaa vasta tämän jälkeen, mikä tarkoittanee sitä, että aloitan työt projektissa vasta huhti-toukokuun vaihteessa.

Torstai 24.4.2022

Aioin viettää tämän päivän vielä harjoitellen Qlik Sensen visualisaatio toimintoja.

Erilaisten kaavioiden tekeminen on todella helppoa, mutta tällä hetkellä minulle haastavaa on tehdä niistä kauniita. On ollut haastavaa löytää netistä ohjeita esimerkiksi taustavärien muuttamiseen. Yleisesti Qlik Sensen kanssa toimiminen on ollut melko yksinkertaista alun hapuilun jälkeen, mutta huomaan tarvitsevani jonkun verran toistoa työskentelyn sujuvoittamiseksi.

Tavoitteeni tälle päivälle ei ollut mitenkään erikoinen ja en itseasiassa ole tyytyväinen tämän päivän suoritukseeni. Oikeastaan sitä tämä päivä on ollut – suorittamista. Eikä mitenkään hyvässä mielessä, niin että päivän päätteeksi olisin hoitanut pitkän listan eri asioita, vaan päivä pikemminkin tuntui laahustavan eteenpäin. On tärkeää muistaa myös olla itselleen armollinen uuden oppimisessa ja ylipäättään muutenkin jotain tehdessä. Kenenkään ei ole mahdollista olla tehokas kaiken aikaa ja huonotkin päivät ovat sallittuja. Olen kuitenkin innoissani huomaisesta, sillä tajusin projektityön alun viivästymisen tarkoittavan minulle sitä, että voin käyttää seuraavat viikot jollain omaa itseäni hyödyttävällä tavalla. Ajattelin palata front endin pariin.

Perjantai 25.3.2022

Tarkoitukseni tänään on tutkia Vue.js-sovelluskehityksen (framework) perusteita. Olen pitkään halunnut opiskella tätä JavaScriptin kirjastoa ja tästä voisi olla myös hyötyä itselleni tässä työssä. Tiimissämme ei ymmärtääkseni ole kovin montaa Vue.js-osaajaa, minkä vuoksi uuden taidon osaaminen olisi hyödyllistä oman itseni lisäksi myös tiimillemme.

Minulla meni lähes koko aamupäivä asiakasprojektiin liittyviin valmistelutöihin ja pääsin Vue.js pariin vasta iltapäivällä. Kollegani vinkkasi minulle hyvän tutoriaalisarjan Youtubesta ja vietin iltapäivän asennellen tarvittavia ohjelmia.

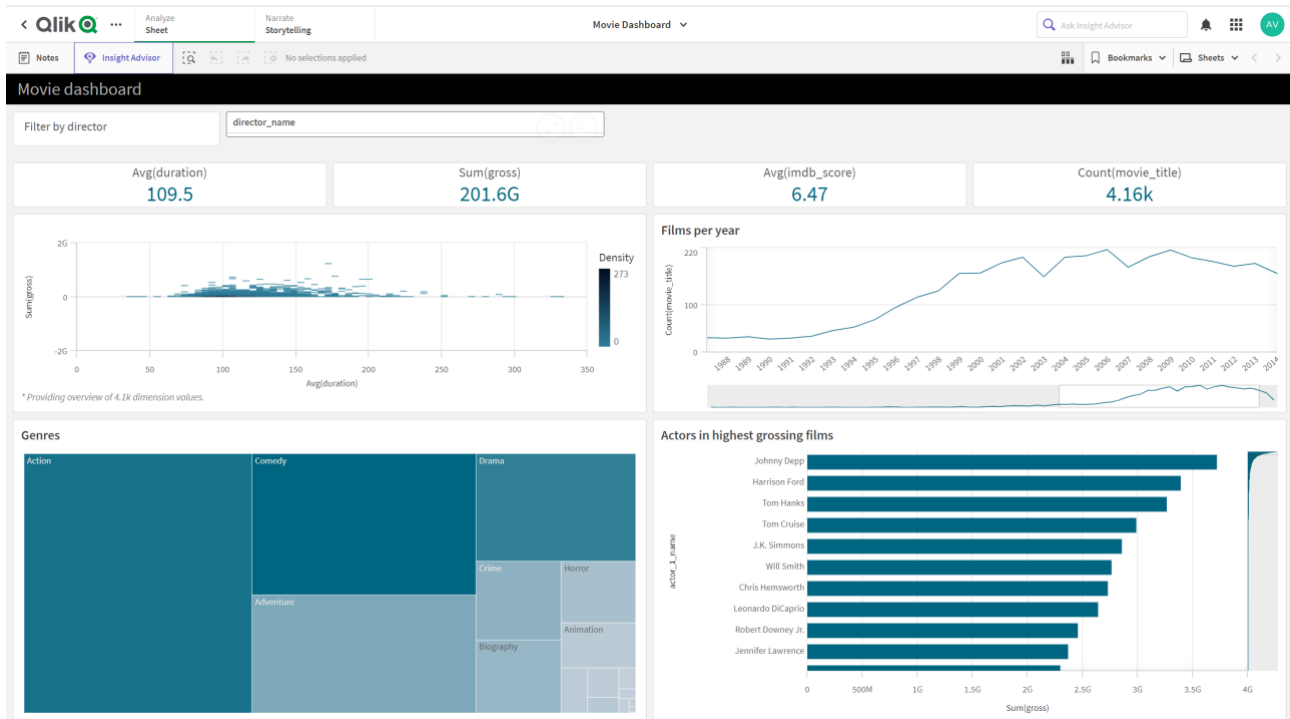
Odotin pääseväni tänään pidemmälle Vue.js kanssa, mutta toisaalta kollegani kanssa käymät keskustelut Vue.js-sovelluskehityksestä sekä React.js-kehityksestä olivat tärkeitä. Olen tähän asti keskittynyt eri materiaaleista oppimiseen ja tajusin tänään, että voin oppia myös keskustelemalla asioista. Tämän päivän keskustelu koski Reactin ominaisuuksia ja samankaltaisuuksia Vuen kanssa.

Etenin tänään hyvin vähän, mutta olen silti iloinen tämän päivän kulusta ja omasta oivalluksestani.

#### Viikkoanalyysi

Viides seurantaviikko pyöri hyvin voimakkaasti datan visualisoinnin ja Qlik Sensen ympärillä.

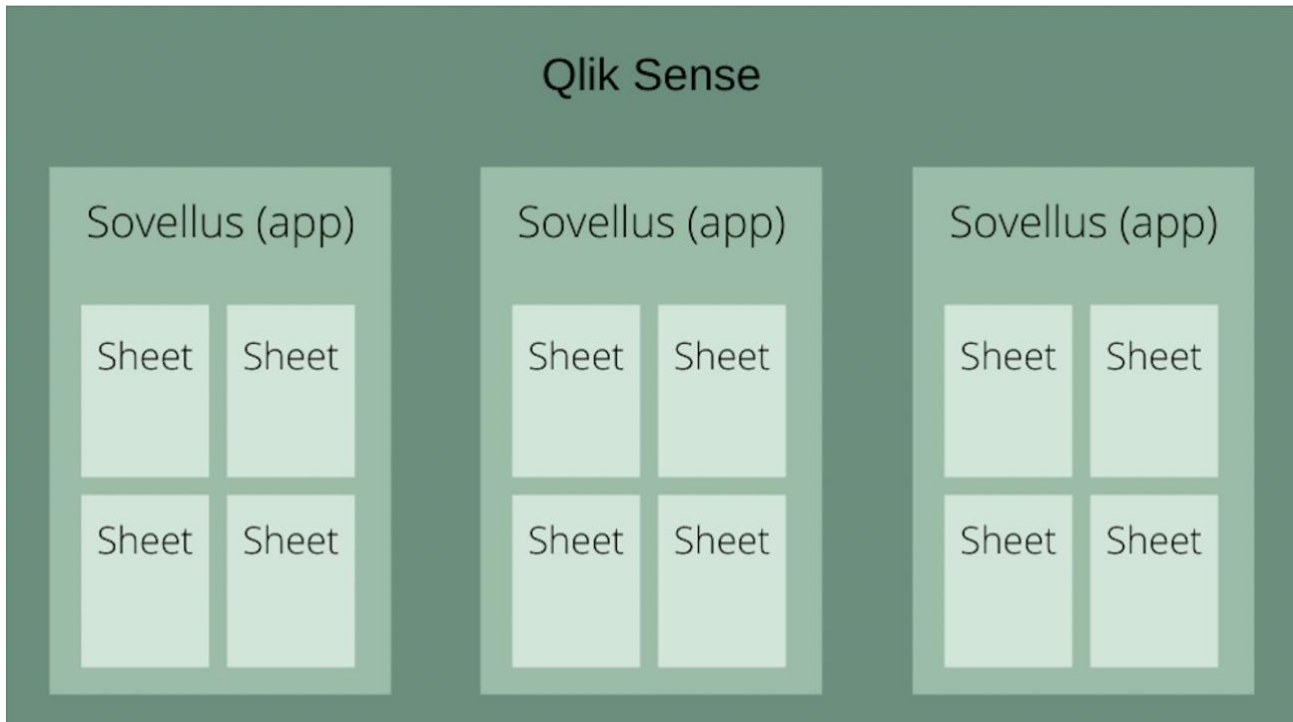
Datan visualisoinnilla tarkoitetaan tiedon muuttamista graafiseen muotoon. Yksittäisiä tietoja yhdistetään loogisiin ja systemaattisiin keinoihin muotoon, joka on helpommin hahmotettava. Visualisointiin käytetään erilaisia kaavioita ja usein ne kootaan yhdelle koontinäytölle (dashboard). Visualisointia tehdään organisaation toiminnan ohjaamisen tueksi. Esimerkiksi liiketoimintaa harjoittavilla yrityksillä visualisoinnissa on mukana keskeisiä liiketoiminnan indikaattoreita, kuten KPI, Key Performance Indicator. (Wilke 2019, luku 2; Fortino 2020, 16.) Alla olevassa kuvassa 4 on esimerkki siitä, millainen koontinäyttö voi olla.



Kuva 4. Projektin koontinäyttö Qlik Sensessä.

Tiedon visualisoinnissa on tärkeää ymmärtää muutamia asioita, jotta se olisi tarkoituksenmukaista ja hyödyllistä. Tämä tarkoittaa sitä, että visualisoidussa esityksessä näytetään vain sellaisia asioita, joilla on merkitystä asiakkaalle ja heidän toiminnalleen. Täytyykin siis ymmärtää, kuka on esityksen yleisö ja mitä heidän pitää saada tietää voidakseen tehdä informoituja päätöksiä. Esityksen ei ole tarkoitus olla kaunis ja viihteellinen, vaan sen tarkoitus on ensisijaisesti tarjota tietoa, jolla voidaan ohjata esimerkiksi päätöksentekoa organisaatiossa. (Fortino 2020, 44.) Tässä jäinkin heti kiinni itselleni halusta luoda kauniita esityksiä: kirjoitinkin siitä torstaina. Toki on ehkä eri asia luoda informatiivisia esityksiä, jotka ovat samalla visuaalisesti näyttäviä kuin luoda kauniita esityksiä pelkästään sen takia, että elementit ovat kauniita. Asiakasprojektini kannalta mietin, että mahdollisimman riisuttu ja dynaaminen ulkonäkö lienee kaikista toimivin, sillä asiakkaan on voitava käyttää esityksiä toimintansa johtamisen päätöksenteossa.

Viikon aikana harjoittelin erilaisten kaavioiden tekemistä Qlik Sensessä. Qlik Sense on siis web-pohjainen BI-työkalu, jonka avulla voi visualisoida organisaation dataa. Palveluun voi luoda useita eri sovelluksia (apps). Sovellukset koostuvat sheeteistä, joissa haluttu data esitetään visuaalisessa muodossa erilaisin kaavioin. Sovelluksessa voi olla yksi tai useampi sheet.



Kuva 5. Qlik Sensen rakenne.

Koska olin ensimmäistä kertaa tekemisissä kyseisen palvelun kanssa tällä viikolla, jouduin selvittämään itselleni kaiken aivan perusteista lähtien. Aluksi palvelun rakenne (kuvattu edellä) vaikutti mielestäni monimutkaiselta: tuntui, että minun piti navigoida todella pitkälle palveluun, jotta sain esiin haluamani näkymän, sheetin. Suurimmat ongelmat tuntuivat liittyvän lähinnä siihen, että käyttöliittymä on uudistunut vuosien saatossa, enkä löytänyt aina heti etsimääni. Minun täytyy vielä myöhemmin perehtyä lisää sheetin ulkonäön muokkaamiseen, sillä en onnistunut siinä vielä tällä viikolla.

Yleisesti arvioisin oppineeni paljon uutta tällä viikolla sekä saaneeni kiinni Qlik Sensen ideasta. Opiskelutapani, videotutoriaalin seuraaminen, on tuntunut kaikista parhaalta oppimisen keinolta. Itse tekemällä ja kokeilemalla olen päässyt oppimaan virheiden kautta. Viime viikolla pohdin itseäni oppijana ja totesin, että opin tietosisältöä parhaiten esimerkiksi kirjoittamalla muistiinpanoja käsin. Vaikka tämän viikon oppimiskokemusta ei voikaan verrata suoraan Azure-opintoihin, voin ehkä löytää joitain yhtymäkohtia sen ja hapuilevan trial and error-tyylisen matkani Qlik Sensen syövereihin kanssa: molempiin oppimiskokemuksiin liittyy tekeminen. Ja vaikka tutustuinkin viime viikolla paljon VAK-oppimistyyliä kritisoivaan kirjallisuuteen, voisin silti lukea itseni kinesteettiseksi oppijatyyppiksi.

### 3.6 Seurantaviikko 6: Vue.js-sovelluskehys

Maanantai 28.3.2022



Tavoitteeni on käyttää koko päivä Vuen harjoitteluun.

Aloitin päivän seuraamalla tutoriaalia ja kopioimalla harjoituksia samanaikaisesti omalla koneellani Visual Studio Codella. En päässyt pitkälle harjoituksissa, sillä työ keskeytyi jatkuvasti. Keskeytyksen jälkeen orientoituminen tehtävään oli hidasta, vaikka sisältö olikin yksinkertainen ja helppo.

Päivän aikana autoin esihenkilöäni Teams-ryhmän ylläpidossa. En alun perin luonut uutisille erillistä kanavaa, vaan olin ehdottanut General-kanavan sisältävän uutisia ja muita yleisiä keskustelunaiheita. Ideanani oli, että General olisi ryhmän ”pääkanava”, johon kaikki tärkeä tieto olisi koottu yhteen. Esihenkilöni kuitenkin toi ilmi, että tärkeät ajankohtaiset uutiset kuitenkin hukkuvat muiden viestien sekaan ja hän ehdotti News-kanavan lisäämistä. Kun hän oli luonut kanavan, ryhdyimme pohtimaan kuinka jo julkaistut uutiset voisi siirtää News-kanavalle. Teamsissa ei pysty suoraan siirtämään jo julkaistua viestiä toiselle kanavalle. Olin aiemmin löytänyt uuden viestin lähettämisen yhteydessä mahdollisuuden lähettää viesti samaan aikaan usealle eri kanavalle ja toin tämän esiin, sillä kuvittelin, että viestin siirtämisen toiminto olisi yhtä piilossa kuin useaan kanavaan julkaiseminen. Hän keksi muokata jo julkaistuja viestejä ja lisätä niihin usealle kanavalle julkaisun, jolloin viestit oli mahdollista siirtää haluamallemme kanavalle.

Tämä oli sellainen tilanne, jossa yhteistyö todella johti ongelman ratkeamiseen, vaikkakaan ratkaisu ei ollutkaan kovin järkevä. Tämä vaikutti olevan mahdollisesti ainoa keino siirtää viesti kanavalta toiselle ja ehdottomasti hyvä esimerkki huonosta käyttöliittymästä.

Oletin ehtiväni tänään pidemmälle Vuen kanssa, joten en sanoisi päässeeni tavoitteeseeni tänään.

Tiistai 29.3.2022

Aion käyttää tämänkin päivän Vuen opiskeluun ja edetä kurssilla pidemmälle kuin eilen.

Seurasin Youtubesta samaa tutoriaalia kuin aiemmin ja tein samaan tahtiin harjoituksia Visual Studio Codella. Jäin muutaman kerran jumiin omien kirjoitusvirheiden kanssa. Esimerkiksi puuttuva välilyönti rikkoi koodin useammaksi minuutiksi. Huomasin, että jotkin CSS-elementit eivät näy selaimessa lainkaan ja sivun elementtien tarkastelua katsoessa kyseisen elementin kohdalla selain ilmoittaa elementin olevan virheellinen. Esimerkiksi kehyksen alareuna (border-bottom) ei näy selaimessa lainkaan. Etsin vain hetken ratkaisua Stackoverflow-sivustolta ja muuten googlaamalla error-koodia, mutta en löytänyt ratkaisua.

En ollut asettanut tälle päivälle kovin tarkkaa tavoitetta, mutta etenin opiskelussa hyvin ja olen tyytyväinen tämän päivän tekemiseeni.

Keskiviikko 30.3.2022

Tänään yritän selvittää CSS-elementteihin liittyvän ongelman koodissani.

Jatkoin aamulla Vue.js-kurssia eteenpäin ja tarkkailin samalla sitä, mitkä tyylit näkyvät sovelluksessani ja mitkä eivät. Pääsin tänään melko pitkälle, kunnes tein tyylimuotoilua linkeille ja muotoilut eivät näkyneet selaimessa. Kehykset (borders) ja linkkien värit eivät näkyneet sivulla lainkaan. Useissa paikoissa ehdotettiin style scopen lisäämistä.

Iltapäivällä meillä oli työpaja, jonka tavoitteena oli kehittää työyhteisöämme. Työyhteisömme jäsenet vaikuttavat kaipaavan lisää yhteisön tunnetta tiimiin ja pohdimme konkreettisia keinoja, joilla voisimme parantaa yhteisöämme. Koska tiimi on kooltaan suuri ja jakautunut ympäri Suomea, täytyy yhteisöllisyyden tunteeseen ja sen ylläpitämiseen nähdä vaivaa. Arvioisin itse etätyöskentelyn myös lisäävän haastetta tässä. Moni tuntui kaipaavan arkisia kohtaamisia työkavereiden kanssa. Yksi ehdotus olikin, että voisimme Teamsissa ilmoittaa toimistolle menostamme, jotta muut voivat halutessaan ja mahdollisuuksien mukaan myös tulla samana päivänä toimistolle töihin.

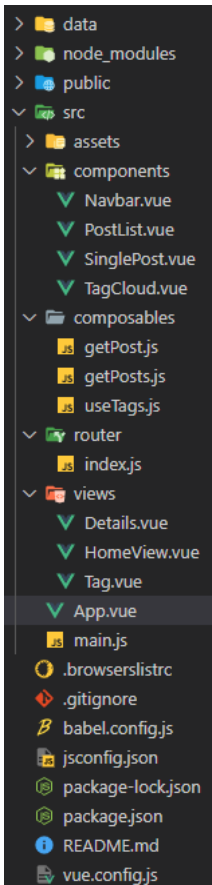
En saanut ratkaistua ongelmaani, joten en päässyt tavoitteeseeni. Aion huomenna kysyä apua kollegaltani.

Torstai 31.3.2022

Päivän tavoitteeni on löytää ratkaisu koodissani oleviin virheisiin tyylimäärittelyissä.

Kävin tyylit läpi tarkasti rivi kerrallaan, mutta en löytänyt mitään vikaa määrittelyistä. Kirjoitin määrittelyn uudelleen ja linkkien värit ja kehykset tulivat näkyviin. En osaa sanoa, mitä tein oikein, mutta koodi toimii ja olen tyytyväinen. Toisen tiedoston tyylimäärittelystä olin unohtanut alakehyksen (bottom border) määrittelystä solidin ja värin, mikä aiheutti kehyksen katoamisen täysin ruudulta.

Jatkoin jo aloittamaani tutoriaalia eteenpäin ja harjoittelin modaalisten ikkunoiden tekemistä. Modaalinen ikkuna aukeaa sivulla olevaa painiketta painamalla. Harjoituksessa tehtiin ensin yksi ikkuna yhdessä, ja sen jälkeen piti luoda toinen modaalinen ikkuna (ja painike sen avaamiseen) itse. Modaalinen piti asettaa sulkeutumaan, kun käyttäjä klikkaa modaalisen ikkunan ulkopuolelle. Ohjelma koostui siis App.vue-tiedostosta sekä Modal.vue-tiedostosta. Modal.vue-tiedostoon määritellään modaalisen ikkunan rakenne (template).



Kuva 6. Vue.js-projektin rakenne

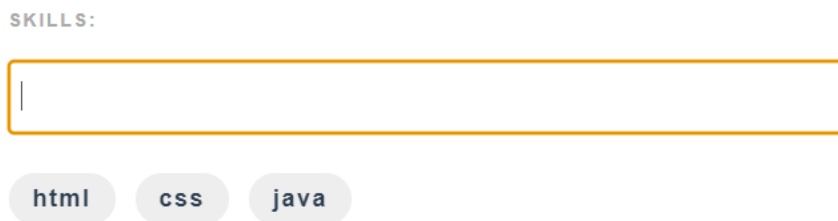
App.vue-tiedostoon puolestaan syötetään tietoa, jonka Modal.vue sitten näyttää html-tiedostossa selaimessa. Yritin luoda Modal.vue-tiedostoon modaalin templatien lisäksi toisen samanlaisen, sillä logiikkani mukaan tässä tilanteessa kaksi templatea on yhtä suuri kuin kaksi modaalista ikkunaa selaimessa. Ja näin se olikin, mutta lopputulos ei silti ollut toivotunlainen. Molemmat modaaliset ikkunat aukesivat päällekkäin sivulle, ja ikkunan alapuolelle tuli toinen, yhden rivin korkuinen tyhjä modaali.

Ratkaisin tilanteen poistamalla toisen templatien Modal.vue-tiedostosta, sillä yksi template riittää. Lisäksi lisäsin App.vue-tiedostoon rivin, joka asettaa toisenkin modaalin näkyväksi klikatessa. En ole koodannut pitkään aikaan, ja huomaan tekeväni paljon ajatus- ja huolimattomuusvirheitä kirjoittaessani. Toisaalta myös omalle tekemiselleen sokeutuu siinä määrin, että yhden css-rivin korjaamiseen menee kaksi ja puoli päivää.

Pääsin tänään tavoitteeseeni, olen tyytyväinen etenemisestäni ja siitä, että löysin viimein ratkaisun CSS-määrittelyssä olleeseen virheeseen.

Perjantai 1.4.2022

Tänään harjoittelin lomakkeen luomista ja lomakkeelle täytetyn tiedon tallentamista muuttujiin. Lomakkeelle pystyi lisäämään taitoja, esimerkiksi html, css ja java. Lomakkeelta pystyi poistamaan lisättyjä taitoja klikkaamalla aluttua taitoa. Yritin ensin tehdä tätä javascriptin splice()-funktion avulla, mutta päädyin lopulta tekemään sen filter()-toiminnallisuuden kanssa. Lomakkeelle lisätyt taidot lisättiin ohjelmassa taulukkoon ja kun yhden taidon klikkaaminen laukaisee delete-funktion, jossa kyseisen taulukon sisältö käydään läpi filter()-funktiolla. Ne solut, joita listasta ei löydy, filteeröidään pois näkyvistä, eli poistetaan näkyvästä listasta.



Kuva 7. Lomakkeelle lisättävät taidot

Aloitin myös Vue routerin opiskelua. Jatkan tästä maanantaina. Pääsin tämän päivän osalta tavoitteisiini ja koen oppineeni uutta ja saaneeni muistinvirkistystä jo opittuun.

#### Viikkoanalyysi

Tämän seurantaviikon aikana syvennyin kunnolla Vue.js-kirjastoon ja sillä luotaviin web-sovelluksiin. Viikon aikana kohtasin useampia ongelmallisia tilanteita opiskellessani, jotka sain ratkaistua pääasiallisesti omin neuvoin.

Vue.js on JavaScriptin sovelluskehys-kirjasto, joka hyödyntää JavaScriptin lisäksi HTML:ää ja CSS:ää. Vuella rakennettavat ohjelmat perustuvat komponentteihin, joista voidaan koota yksittäisiä toimintoja verkkosivulle, kokonaisia verkkosivuja tai -sivustoja. (Hanchett, E. & Listwon, B., luku 1; Vue.js 2022a.) Vue vaikuttaakin olevan hyvin ketterä kirjasto, joka sopii monenlaisiin tarkoituksiin.

Kirjoitin viikolla kohtaamistani ongelmista CSS-muotoilujen kanssa. Yhtenä päivänä totesin koodin yllättäen toimivan ja hyväksyin sen sellaisenaan. En jäänyt selvittämään, miksi koodi yhtäkkiä toimi. Huomaan toimineeni tässä sellaisella tavalla, joka voisi aiheuttaa ongelmia tulevaisuudessa, mikäli olisin tässä tilanteessa rakentamassa oikeaa sovellusta asiakkaalle. Palaan viikolla kolme tutustumaani Clean Code -periaatteeseen. Robert Martin kuvailee teoksessaan "Clean Code: A Handbook of Agile Craftmanship" (2009, 3-4) kiireen ja muun paineen aiheuttamaa nopeasti kirjoitettua koodia ja sotkuisen koodin kasautumista aina vain suuremmaksi taakaksi, jonka selvittäminen kestää kauan. Tässä tilanteessa olisin voinut tutkia kirjoittamaani muotoilua ja

selvittää, mikä aiheutti koodin yhtäkkisen toimimisen. Vaikka se nyt jäikin tekemättä, uskon tämän tiedostamisen auttavan jatkossa vastaavissa tilanteissa.

Vue.js komponentit koostuvat templatesta, johon sivulla näytettävä koodi kirjoitetaan. Template siis määrittää komponentin näkyvän rakenteen, ja HTML-koodi kirjoitetaan templatien sisään. Modaalaisia ikkunoita harjoitellessani tällä viikolla menin vikaan siinä, että en tajunnut templatien muokkautuvan sinne lähetettävän datan mukaisesti. Yritin siis luoda yhteen komponenttiin kahta templatea, ja sain aikaan melko erikoisen näköisen tilanteen näkyviin sivulle. Reactissa vastaavasti sivulla näytettävä sisältö kääritään `<div>` elementin sisään. Ehkä tästä syystä onkin näin jälkikäteen ajateltuna vaikea käsittää, että en ymmärtänyt tätä samaa logiikkaa Vuen kanssa.

Opin tällä viikolla paljon uutta ja löysin uusia kehityskohteita. Tehdessäni Vue-harjoituksia, minun piti käyttää useita kertoja JavaScriptin metodeita, kuten esimerkiksi `filter()` suodattaakseni epähaluttu data pois taulukosta. JavaScript on tullut ajoittain vastaan opinnoissani esimerkiksi juuri Reactin ja nyt Vuen yhteydessä, mutta se on jäänyt aina vähemmälle huomiolle. Esimerkiksi perjantaina jäin jumiin lomake-harjoituksessa, sillä lähdin ratkaisemaan ongelmaa väärän metodin kanssa. Tulevaisuuden kannalta minulle olisi hyödyllistä syventää JavaScript-osaamistani.

### 3.7 Seurantaviikko 7: React.js- ja Vue.js-sovelluskehysten yhtäläisyydet ja erot

Maanantai 4.4.2022

Tavoitteeni tälle päivälle on harjoitella tiedon hakemista Vue.js-projektiin.

Etenin kurssilla routerin lisäämiseen. Router yhdistää selaimen osoitteet Vuen komponentteihin ja näyttää halutut näkymät käyttäjälle selaimessa. Router on minulle tuttu Reactista ja sen ymmärtäminen ja ohjelmointi tuntui minulle tutulta ja helpolta.

Routeriin tutustumisen jälkeen vuorossa oli datan hakeminen sovellukseen. Suorittamalla kurssilla dataa haetaan tietokannan sijasta JSON-tiedostosta. Loin json-datalle oman tiedoston `db.json` ja laitoin sen data-nimiseen kansioon projektin juureen. Latasin json serverin, jotta voisin tehdä hakuja json-tiedostosta. Lokaali asennus onnistui ja json-server tuli näkyviin projektin `package.json`-tiedostoon. En kuitenkaan saanut json-serveriä käyntiin projektin terminaalista komennolla `json-server --watch data/db.json`, sillä json-server ei ollut ohjelman tuntema komento. Selvittelin asiaa ja useimmiten Asian sanottiin ratkenneen sillä, että json-server asennettiin globaalisti lokaalin asennuksen sijaan. Poistin lokaalin asennuksen ja asensin json-serverin globaalisti koneelle, mutta tämäkään ei auttanut. Globaalin asennuksen jälkeen `json-server --watch data/db.json` tuotti errorin, jonka mukaan tiedosto ei voi ladata. Päädyin tutkimaan Microsoftin sivuja error-viestiä tulkitessani, mutta en viisastunut tästä. Json-serverin Githubista issues-sivulta

löytyi kuitenkin samanlainen ongelma kuin itselläni oli ja vaikka useimmilla oli tarjota ongelmaan samoja, jo yrittämiäni ratkaisuja, löysin myös sellaisen komennon, jota en vielä ollut kokeillut ja joka lopulta sai json-serverin käyntiin Visual Studio Codessa. `Npx json-server - -watch data/db.json` käynnisti serverin ja pääsin käsiksi fetch()-funktiolla haettavaan dataan.

Fetch() onkin minulle tuttu Reactista ja datan hakeminen vaikutti yksinkertaiselta. Suuri osa päivästä tuhraantui ongelman ratkaisemiseen, mutta onneksi pääsin eteenpäin omin neuvoin harjoituksessa.

Tiistai 5.4.2022

Tänään aloitin Composition API -harjoituksen Vuella. Päivän aikana on tarkoitus harjoitella sovelluksen watch- ja watchEffect-funktioiden toimintaa.

Olen edennyt tutoriaalissa viimeiseen osaan, jossa harjoitellaan Composition APIa. Tällä tarkoitetaan sitä, että ohjelmaan luodaan komponentti-tiedostoja (jokainen komponentti sisältää yhden ohjelman toiminnallisuuden) ja näitä komponentteja voidaan hakea (import) koodiin sen sijaan, että koko koodi kirjoitettaisiin yhteen tiedostoon. Tämä auttaa jaottelemaan koodin helposti luettavammaksi, sillä jokainen toiminnallisuus on omassa tiedostossaan. API:t pitävät sisällään muun muassa Reactivity API:n, jonka avulla luodaan reaktiivisia elementtejä sovellukseen ja Lifecycle Hooksit, joita käyttäen voidaan suorittaa tapahtumia koodin ajamisen eri vaiheissa. (Vue 2022b.)

Harjoituksessa tehdään blogi, jonka etusivulla on listattu viimeisimmät blogipostaukset, ja otsikkoa klikkaamalla saa koko blogitekstin auki. Päivän harjoituksessa tutkittiin watch- ja watchEffect-kuuntelijoiden toimintaa sekä vertailtiin ref()- ja reactive()-attribuuttien eroa koodissa. Koen ymmärtäneeni näiden attribuuttien eron, mutta koska tämä tulee minulle ensimmäistä kertaa vastaan, en vielä täysin osaa selittää niiden eroa tai yhtäläisyyksiä tai sitä, miten soveltaisin näitä koodissa.

Etenin tänään paljon viimeisessä harjoituksessa, mutta minun täytyy kerrata mainitsemiani asioita, jotta oppisin ne.

Keskiviikko 6.4.2022

Tänään aion tehdä loppuun Composition API -harjoituksen, joka minulla jäi eilen kesken.

Aamu alkoi asiakasprojektin kokouksella. Kokouksessa kävimme läpi asiakkaalle tehtävää tarjousta projektista sekä alustavaa aikataulua projektin ensimmäiseen sprinttiin. On

mielenkiintoista olla mukana asiakasprojektissa aivan sen alusta alkaen ja nähdä sen kaikki alkuvaiheet.

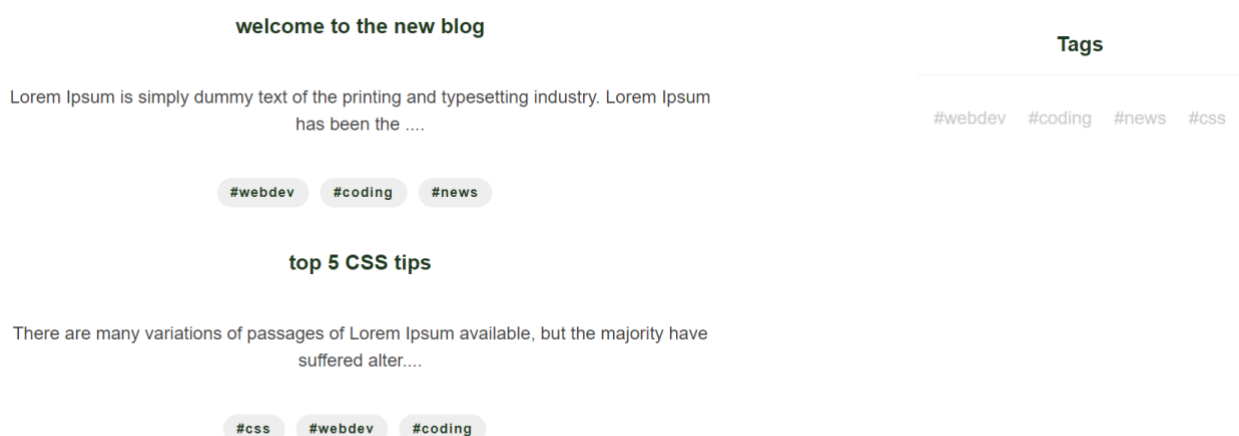
Tutkin harjoituksessa Vue lifecycle hooksien toimintaa. Harjoituksessa ensin tulostettiin viestejä selaimen konsoliin console.log()-kutsulla. Tämä auttoi näkemään, missä vaiheessa tapahtuu mitään. Tämän jälkeen tehtävässä esiteltiin uudelleenkäytettäviä funktioita (composable functions). Funktiot kirjoitetaan JavaScript-tiedostoihin. Näitä funktioita voi käyttää useissa eri komponenteissa, mikä edelleen nopeuttaa koodin kirjoittamista (samaa funktiota ei tarvitse kirjoittaa useita kertoja). Ideana tämä vaikuttaa selkeältä, mutta koin tämän käytännössä haastavaksi.

Sain Vue.js-harjoituksen valmiiksi, mutta aion vielä jatkaa viimeisintä harjoitusta ja soveltaa oppimaani. Olen ylpeä siitä, että sain viimein kurssin loppuun, mutta omaa oppimistani reflektoidessani huomaan oppineeni tämän hyvin pintapuolisesti ja tarvitsen soveltavia harjoituksia syvän oppimisen takaamiseksi.

Torstai 7.4.2022

Tavoitteeni tälle päivälle on soveltaa jo oppimaani ja jatkaa alkuvuokosta aloittamaani Composition API-harjoitusta.

Composition API-harjoituksessa on tähän mennessä luotu yksinkertainen blogi, jonka etusivulla näytetään listaus/esikatselu blogin postauksista sekä jokaiseen postaukseen lisätyt tunnisteet (tagit). Alla olevassa kuvassa näkyy blogin etusivun rakenne.



Kuva 8. Blogin etusivun rakenne

Otsikkoa klikkaamalla saa avattua koko blogipostauksen luettavaksi. Haluan lisätä sivulle sellaisen toiminnon, että tagia klikkaamalla sivulle suodatuu kaikki kyseisen tagin omaavat blogipostaukset. Tämä sujui kehnosti, sain luotua tagiin linkin ja linkin aukeamaan uudelle sivulle, mutta sivun sisältö jäi vielä puuttumaan. Aluksi minulla oli ongelmia Vue routerin kanssa (väärä muuttuja routerin ohjauksessa) ja vielä pitää kirjoittaa sellainen funktio, jolla saan filteröityä näkyviin sellaiset postaukset, jotka on tagattu valitulla tunnistesanalla.

Koodaus ei tänään sujunut lainkaan, enkä kokenut suuria onnistumisia – tai onnistumisia ylipäätään.

Perjantai 8.4.2022

Yritän tänään vielä saada tagien suodatuksen toimimaan Vue.js-harjoituksessani.

Päätin, että en halua suodatettavaksi valitun tagin aukeavan uudelle sivulle (kaikki valitulla tunnisteella tagatyt postaukset listattaisiin uudelle sivulle), vaan että tämä tapahtuu blogin etusivulla. Tämä tarkoittaisi sitä, että minun pitäisi luoda sellainen Tags-komponentti, jonka lisäksi kaikki postaukset listaavaan HomeView.vue-komponenttiin.

Sain tagit näkymään sivulla, mutta en saanut suodatusta toimimaan tagia klikkaamalla. En löytänyt netistä ohjeita tämän tekemiseen, joten tutkin, löytyisikö netistä sellaisia tutoriaaleja, joissa jotain vastaavaa rakennettaisiin sovellukseen. Löysin ohjeen hakutulosten suodattamiseen Youtubesta, mutta en kyennyt sen perusteella soveltamaan ideaa omaan harjoitukseeni.

Viikkoanalyysi

Viikon aikana jatkoin Vueen perehtymistä ja huomasin lisää yhteneväisyyksiä Reactin kanssa. React on JavaScript-kirjasto, jonka avulla rakennetaan käyttöliittymiä sovelluksille. React on ketterä tapa kehittää ohjelmia, sillä tiedon päivittyessä se päivittää ainoastaan muuttuneet komponentit koodista. Ohjelmat koostuvat komponenteista, pienemmistä itsenäisistä koodiosioista, jotka muodostavat ohjelman rakenteen. Reactilla voi koodata niin yksinkertaisia kuin monimutkaisempiakin ohjelmia. (React.js 2022).

React voidaan jakaa kahteen API-kokonaisuuteen. Reactin komponentti API sisältää varsinaisen datan ja ohjelmakoodin. Toinen tärkeä osa ohjelmaa on myös React DOM (Document Object Model), joka piirtää ohjelman sisällön selainikkunaan halutulla tavalla. (Boduch 2017, 11.)

DOM siis määrittelee mitä verkkosivulla näytetään. Se sisältää ohjeet siihen, mitä sivulla näytetään ja millä tavalla. VDOM tarkoittaa virtuaalista DOMia. Kun virtuaalisessa DOMissa tapahtuu muutos, VDOM ottaa itsestään pikakuvan, snapshotin, ja päivittää koko VDOMin. Tämän jälkeen



päivittynyttä VDOMia ja aiempaa versiota (josta pikakuva otettiin) verrataan keskenään ja ainoastaan päivityksessä muuttuneet elementit päivitetään todelliseen DOMiin. Nämä muutokset näkyvät käyttäjälle ruudulla. (Vue.js 2022c, Codecademy 2022a, Codecademy b). Sekä React että Vue hyödyntävät DOMia ja VDOMia.

Reactin tavoin myös Vue hyödyntää uudelleenkäytettäviä komponentteja. Vuessa kaikki data säilötään data-funktioon. Funktio on reaktiivinen, mikä tarkoittaa sitä, että kun funktiossa oleva tieto muuttuu, ohjelma päivittää visuaalisessa käyttöliittymässä näkyvän tiedon automaattisesti. Data-funktiossa säilytettävä tieto on komponenttikohtainen. (Franklin ym. 2019, alaluku How to Create Components in Vue; Jacques 2019, alaluku Reactive Data.)

Vuessa käyttöliittymä koodataan malleihin (template). Malliin kirjoitetaan koko komponentin rakenne HTML-kielellä. Vuen sivujen (Vue.js 2022c) mukaan tämä tekee koodin lukemisen helpoksi niin aloittelijoille kuin taitavammillekin HTML-osaajille, mikä tekee Vueen siirtymisen omalta osaltaan helpommaksi. Tekemässäni harjoituksessa koodin lukeminen oli helpompaa juuri tästä syystä. Malli sisältää kuitenkin myös erilaisia toimintaohjeita (directive), jotka eroavat HTML:stä. Näillä toimintaohjeilla voidaan esimerkiksi näyttää tai piilottaa sivulta tietoa valitun säännön täytyessä v-if -komennolla tai näyttää listana haluttua tietoa v-for -komennon avulla. Näitä koodin toimintaohjeita on useita erilaisia ja niillä voidaan manipuloida sivulla näytettävää tietoa. (Vue.js 2022d.)

Reactissa ohjelmia kirjoitetaan joko luokkakomponenttien tai React Hooks-funktioiden avulla. React Hooks esiteltiin vuonna 2018 ja se pohjautuu komponenttien pilkkomiseen pienemmiksi funktioiksi. Luokkakomponentteja ja Hookseja voi sekoittaa keskenään, mutta käsittelen tässä ainoastaan React Hooksin toimintoja. Varsinainen käyttöliittymä kirjoitetaan funktion return-toimintoon JSX-syntaksilla. Syntaksi saa nimensä JavaScript XML:stä. JSX lisää HTML-kieleen JavaScriptin toimintoja tehden siitä monipuolisemman. (React.js 2022b; Boduch 2017, 19-22.)

Itselleni Vue on ollut samaan aikaan sekä haastavaa että helppoa. Aiempi osaaminen Reactista on todennäköisesti nopeuttanut oppimista, mutta myös samalla vaikeuttanut sitä. Pohtiessani omaa toimintaani esimerkiksi blogiharjoituksen kanssa, yritin luoda "reactmaisii" ratkaisuja koodiini. Koska nämä kaksi ovat keskenään niin samanlaisia teknologioita, on eroja välillä vaikeaa pitää mielessä. Vuessa käyttöliittymän koodaaminen malliin oli alun hämmennyksen jälkeen intuitiivista, sillä olen opintojen aikana kirjoittanut useampia HTML-sivuja ja logiikka tuntui tutulta. Näiden kahden väliltä on kuitenkin mahdoton sanoa kumpi on parempi ja vastaus riippunee kehittäjän omista mieltymyksistä.

### 3.8 Seurantaviikot 8 ja 9: Käyttöliittymäsuunnittelu

Maanantai 11.4.2022

Olin sairaslomalla.

Tiistai 12.4.2022

Olin sairaslomalla.

Keskiviikko 13.4.2022

Olin sairaslomalla.

Torstai 14.4.2022

Päätin ottaa tänään hieman kevyemmin sairasloman jälkeen. Katson koodiani ja yritän selvittää, missä kohtaa minulla on virhe ja miksi tunnisteet eivät suodatu niitä klikatessa harjoituksessani.

Sain tänään tunnisteet toimimaan harjoituksessani. Tunniste ei toiminut sivulla, sillä olin epähuomiossa jättänyt ilmoittamatta propsien siirtämisen default-funktiossa (kuva 9).

```
<template>
  <div class="tag-cloud">
    <h3>Tags</h3>
    <div v-for="tag in tags" :key="tag">
      <router-link :to="{ name: 'Tag', params: { tag }}">
        #{{tag}}
      </router-link>
    </div>
  </div>
</template>

<script>
import useTags from '../composables/useTags'

export default {
  setup(props) {
    const { tags } = useTags(props.posts)
    return { tags }
  },
  props: ['posts']
}
</script>
```

Kuva 9. Kuvakaappaus tunnisteita hallitsevasta koodista.

Loput päivästä harjoittelin css-muotoilua. Pohdin pitäisikö sivu asetella flex- vai grid-rakenteen avulla ja katsoin muistinvirkistykseksi videon näiden eroista. Päätin toteuttaa sivun gridillä, sillä se on minulle tutumpi ja tuntui sopivan paremmin tilanteeseen. Tunnistetta klikkaamalla sivulla näkyy kaikki tekstit, jotka on merkitty kyseisellä tunnisteella.

Tämä päivä meni hyvin ja olin oikein tyytyväinen saatuani harjoitukseni toimimaan halutulla tavalla useiden päivien yrittämisen jälkeen. Palkitsin itseni tekemällä jotain kivaa ja iltapäivä menikin sujuvasti tyylimuotoiluja muistellessa.

Perjantai 15.4.2022

Pitkäperjantai.

Maanantai 18.4.2022

Toinen pääsiäispäivä.

Tiistai 19.4.2022

Minut on valittu käyttöliittymäsuunnittelijaksi asiakasprojektiin. Perehdytykseni uuteen projektiin alkaa pian, joten tämän päivän tavoitteeni on tehdä muutama harjoitus Figmaassa valmisteluna perehdytykseen.

Projektin esittelyn perusteella minun täytyy osata Figman perustoiminnot. Tällaisia perustoimintoja ovat esimerkiksi kehysten luominen, tekstikenttien ja painikkeiden luominen. Kehyksellä (frame) tarkoitetaan prototyypin perustaa, jonka päälle eri elementit lisätään. Päivän aikana katsoin ohjevideoita sekä loin yksinkertaisia prototyyppejä verkkosivuista. Tarkoitukseni oli täysin harjoitella eri komentojen ja elementtien toimintaa, enkä tähdännyt vielä kovinkaan huoliteltuun lopputulokseen.

Arvioisin onnistuneeni tämän päivän muistinvirkistysharjoituksissa hyvin. Jo opittu muistui nopeasti mieleen ja Figman tietynlainen intuitiivisuus käyttäjälle auttoi uuden oppimisessa.

Keskiviikko 20.4.2022

Päivän tavoitteeni on tutustua edelleen Figman toiminnallisuuksiin, jotta siirtymiseni uuteen projektiin olisi mahdollisimman saumatonta.

Tein aamupäivän erilaisia harjoituksia, joissa loin sivulle kehyksen ja lisäsin siihen elementtejä. Tarkoituksena oli luoda verkkosivun prototyyppi. Figma on alustana yksinkertainen ja siihen on helppo rakentaa toiminnallisuksia. Onnistuin luonnostelevaan verkkosivun muutamassa tunnissa.

Iltapäivällä osallistuin uuden projektin ensimmäiseen perehdytykseen. Projektin jättävä käyttöliittymäsuunnittelija esitteli projektia Figmaassa ja sain ensimmäisen silmäyksen siitä, miltä suurempi projekti useine kehyksineen näyttää alustalla. Projektiin on luotu uudelleenkäytettäviä komponentteja, sillä projektissa kehitettävässä sivustossa on usein toistuvia lomakkeita ja lomakekenttiä. Komponentit pitävät kehityksen ketteränä, sillä usein toistuvia asioita ei tarvitse suunnitella eikä ohjelmoida useaan kertaan.

Perehdytyksen jälkeen sain katsottavakseni nauhoitteen edellisen viikon sprint demosta. Demossa ohjelmistokehittäjät esittelivät viimeisimmän sprintin tuotoksen, jonka jälkeen käytiin läpi sprintin aikana esiin tulleita kysymyksiä ja muita huomionarvoisia asioita. Nauhoitteen katsominen oli hyödyllinen oppimiskokemus itselleni, sillä en ole aiemmin päässyt osallistumaan sprint demoön työelämässä.

Tämä päivä sisälsi paljon uutta asiaa niin projektin hallinnan kuin scruminkin suhteen. Olen tyytyväinen päivän aikana tapahtuneeseen kehitykseen.

Torstai 21.4.2022

Perehdytykseni uuteen projektiin jatkuu tänään. Sain eilen tunnukset projektin Confluenceen ja Jiraan, ja aion tänään tutustua projektinhallintaan enemmän.

Confluence ja Jira ovat Atlassianin tuotteita ja ne toimivat yhdessä sujuvasti projektinhallinnassa. Tässä projektissa Confluencessa on esitetty projektin käyttäjätarinat määrittelyineen ja Jirassa tarinat on jaettu tehtäviksi ja osoitettu toteutettaviksi projektin eri jäsenille. Jirassa myös seurataan tehtävien elinkaarta suunnitteluvaiheesta valmiiksi. Vietin aamupäivän käyden läpi projektin tunnistettuja käyttäjätarinoita ja niiden kuvauksia. Tarkoitukseni oli päästä syvällisemmin kiinni projektiin ja ymmärtää, miten käytötapaus muutetaan ulkoasun suunnitelmaksi Figmassa.

Perehdytyksessä pyysin käyttöliittymäsuunnittelijaa näyttämään minulle, millä tavalla hän lähtisi luomaan suunnitelmaa uudesta käyttäjätarinasta. Pyysin tätä sen vuoksi, että en ole aiemmin lähtenyt luomaan suunnitelmaa määrittelyn pohjalta. Lisäksi näin voisin ottaa esimerkkiä hänen hyväksi katsomistaan työtavoista. Koin tämän hyväksi ideaksi ja onnistuin ottamaan muistiinpanoja hänen työskentelytavoistaan ja ongelmanratkaisutilanteista, joita nousi määrittelystä. Määrittelyn pohjalta tehty suunnitelma poikkesi välillä itse määrittelystä sen vuoksi, että suunnittelija koki toisenlaisen ratkaisun toimivan paremmin sivulla. Hän kuitenkin teki valmiiksi myös määrittelmän mukaisen suunnitelman ja aikoo kysyä asiakkaalta mielipidettä siitä, kumpi suunnitelma on sopivampi vaihtoehto.

Huomasin tämän työtehtävän sisältävän paljon vuorovaikutusta asiakkaan kanssa, mikä on tärkeää halutun ja sopivan lopputuloksen varmistamiseksi. Suunnittelija lähettää valmiiksi tehdystä suunnitelmasta ensimmäisen vedoksen asiakkaalle kommentoitavaksi, ja tekee suunnitelmaan muutokset asiakkaan kommenttien mukaisesti.

Päivän päätteeksi minulla oli projektipäällikön kanssa perehdytys Confluencen ja Jiran käyttöön tässä projektissa. En ole käyttänyt kumpaakaan aiemmin, ja opin tänään paljon uutta näiden palveluiden lisäksi käyttöliittymäsuunnittelusta oikeassa asiakasprojektissa.

Perjantai 22.4.2022

Sovin eilen projektipäällikön kanssa ottavani Confluencesta oikean käytötapauksen ja harjoittelevani Figmaa ja käytötapauksen kuvauksen toteuttamista.

Valitsin käytötapauksista sellaisen, jota ei ole vielä aloitettu ja joka vaikutti yksinkertaiselta. Harjoitukseksi valikoitui koontinäyttö, josta voidaan lisätä dataa tai vaihtoehtoisesti katsoa nykytilanteen lisäksi historiatietoja kyseisestä toiminnasta. Aloitettuani harjoituksen tajusin valinneeni huonon esimerkin, sillä kyseistä määrittelyä ei ollut tehty vielä loppuun asti enkä näin pystynyt suunnittelemaan näyttöä kunnolla.

Sovin perehdyttäjäni kanssa, että otan harjoitukseksi sellaisen käyttäjätarinan, jonka hän on jo tehnyt valmiiksi. Näin voisin varmistua siitä, että käyttäjätarinan määrittely on tehty loppuun asti ja että minulla olisi saatavilla malliesimerkki tarpeen tullessa.

### Viikkoanalyysi

Tällä viikolla olen perehtynyt käyttöliittymäsuunnitteluun ja valmistautunut liittymään jo kesken olevaan projektiin. Tehtäväni projektissa on suunnitella käyttöliittymää ja tehdä yhteistyötä sekä asiakkaan että myös ohjelmistokehittäjän kanssa. Suunnittelutyö tehdään Figmaa.

Koska projekti on jo alkanut, ei minun käyttöliittymäsuunnittelijana tarvitse tehdä projektin alustukseen liittyvää määrittelytyötä. Ennen projektin aloittamista suunnittelijan täytyy tietää, minkälaista sisältöä ollaan tuottamassa ja kenelle. Suunnittelijalla on valmiina kysymyspatteristo, joka auttaa selvittämään tarvittavaa tietoa työstä: mitä, miksi, milloin, missä, kenelle ja mihin tarkoitukseen sovellus on tulossa. Kysymykset asettavat työn kontekstiin, jonka perusteella suunnittelu on määrätietoista ja tukee aidosti asiakkaan tarpeita. (Jackson & Ciolek 2017, 2–3.) Tässä projektissa kehitetään tietokantajärjestelmää asiakkaalle. Asiakas pitää rekisteriä organisaatiossa toimivista henkilöistä ja heidän tehtävistään. Asiakkaan täytyy pystyä tekemään hakuja järjestelmään ja tarvittaessa keräämään rekisteröityjen henkilöiden historiatietoja. Ohjelmaa käyttää pääasiallisesti muutama pääkäyttäjä. Sitä käyttää satunnaisesti myös suuri joukko organisaatiossa toimivia henkilöitä, joiden tehtävänä on lähinnä lisätä ja päivittää järjestelmään omia tietojaan.

Jackson & Ciolek (2017, 3) ovat määrittäneet neljä osa-aluetta, jotka määrittelevät käyttäjän ja auttavat suunnittelijaa ymmärtämään käyttäjää paremmin. Suunnittelijan täytyy tuntea palvelun yleisö ja ymmärtää, millainen ihminen palvelua käyttää: minkä ikäinen hän on, mitä sukupuolta käyttäjä on, missä päin maailmaa hän on ja mikä hänen koulutustaustansa on. Käyttäjää kartoittavia piirteitä voi olla useita erilaisia. Tämän lisäksi tarvitaan tietoa käyttöympäristöstä. Käytetäänkö palvelua kotona, töissä tai sisällä vai ulkona? Käyttäjää tutkiessa selvitetään myös miten paljon käyttäjä käyttää aikaa palvelussa: onko kyse nopeasta vierailusta, vai vietetäänkö palvelussa pitkiä ajanjaksoja kerrallaan? Suunnittelijaa kiinnostaa myös laite, jolla asiakas käyttää palvelua.

Näiden perusteella käyttäjästä luodaan käyttäjäpersoonaa, palvelun tyypillinen käyttäjä. Kyseinen henkilö on fiktiivinen henkilö, joka auttaa ymmärtämään millaiselle henkilölle palvelua suunnitellaan. Käyttäjäpersoonan selvittämiseksi voidaan toteuttaa kyselyitä, haastatteluita tai muita havainnointikeinoja. (Jackson & Ciolek 2017, 6.)



Kuva 10. Käyttäjäpersoonan osa-alueet.

Hyvä suunnittelu vastaa asiakkaan tarpeisiin ja tarjoaa ratkaisun asiakkaan ongelmaan. Käyttöliittymän visuaalisessa suunnittelussa tulee ottaa huomioon useita eri asioita hyvän käyttäjäkokemuksen varmistamiseksi. Jackson & Ciolekin (2017, 47) esittävät onnistuneen käyttöliittymäsuunnittelun osallistavan myös tuotteen loppukäyttäjän suunnitteluprosessissa. Loppukäyttäjän mielipiteet toimivista ja toimimattomista suunnitelman osista auttavat kehittämään palvelusta toimivan kokonaisuuden. Käyttöliittymäsuunnitteluprojektissa työskentelemme hyvin läheisesti palvelun loppukäyttäjän kanssa. Heidän osaamisensa ja näkemyksensä siitä, miten asioiden pitäisi toimia auttaa huomaamaan sellaisiakin asioita, joita ei itse välttämättä osaisi heti ajatella. Koen heidän osallistumisensa projektiin tuovan paljon lisäarvoa palvelun kehittämiseen.

Sivun asettelulla voidaan vaikuttaa käyttäjään niin hyvässä kuin pahassakin. Usability.gov-sivuston (2022) mukaan parhaat käyttöliittymät ovat yksinkertaisia ja vaikuttavat näkymättömiltä käyttäjälleen. Tällä tarkoitetaan sitä, että käyttöliittymä on selkeä visuaalisesti ja toiminnallisesti ja käyttäjä etenee sivulla juuri sillä tavalla, kuin palvelun kannalta on tärkeää. Jackson & Ciolek (2017, 47) nostavat tärkeäksi myös käyttäjän kokemuksen palvelun miellyttävästä käyttökokemuksesta. Mikäli käyttöliittymä olisi epäselkeä ja käyttäjä ei osaisi esimerkiksi täyttää sivulla olevaa lomaketta, käyttöliittymäsuunnittelu ei ole onnistunut.

Tämän vuoksi käyttäjää voidaan ohjailta visuaalisin vihjein. Sivun osat asetellaan hierarkisesti niin, että käyttäjän huomio kiinnittyy tärkeään sisältöön luonnollisesti. Yksi tällainen keino voi olla esimerkiksi tärkeän asian näyttäminen suurempana muuhun sisältöön verrattuna. Sivun eri osien

täytyy kuitenkin olla tasapainossa toisiinsa nähden. Elementtien välinen etäisyys on yksi asia, joka ohjaa käyttäytymistä. Toisiaan lähellä olevat asiat mielletään yhteenkuuluviksi, kun taas etäisyys erottaa elementit toisistaan. (Jackson & Ciolek 2017, 10-11.)

Käyttäjälle viestitään visuaalisesti myös värein ja kontrastein. Vihreä väri tarkoittaa hyväksymistä ja punainen varoittaa käyttäjää esimerkiksi poistamisesta. Värejä ei kuitenkaan kannata käyttää liikaa, sillä se vähentää niiden merkittävyyttä. Myös eri fonttityylejä, kokoja ja fontin painoja tulisi käyttää harkiten. Käyttäjän on löydettävä sivulta huomionarvoiset kohdat nopeasti silmäilemällä. (Usability.gov 2022.)

Käyttöliittymäprojektissa suunnittelemani palvelu on ilmeeltään pelkistetty ja asiakkaan toiveesta myös väreiltään hillitty. Käyttöliittymä koostuu lähinnä erilaisista lomakesivuista sekä sellaisista sivuista, joille on listattu tietoa. Sivulla olevia tietoja on pyritty ryhmittelemään sivua jakavin väliviivoin, sillä asiakas on toivonut sivujen olevan mahdollisimman kompakteja. Näin esimerkiksi etäisyydellä ei ole aina mahdollista erotella elementtejä toisistaan. Palvelussa ohjataan käyttäjää kuitenkin yleismaailmallisin värein: vihreät painikkeet tarkoittavat tallentamista tai lähettämistä ja punaiset painikkeet poistamista.



## 4 Pohdinta

Tässä luvussa pohdin omaa kehitystäni työntekijänä päiväkirjamuotoisen opinnäytetyön seurantajakson aikana. Seurantajakson aloittamisen jälkeen olen kasvattanut ammatillista osaamistani useilla eri osa-alueilla.

Päiväkirjamuotoisen opinnäytetyön jälkeen kuvailisin itseäni paljon tietoisempuna työntekijänä. Aloittaessani päiväkirjaa ajatukseni oli seurata ammatillisten vahvuuksien kehittymistä uusien taitojen oppimisen muodossa, mutta näin lopuksi koen suurimman muutoksen tapahtuneen minussa itsessäni. Olen oppinut paljon uusia taitoja näiden kahdeksan viikon aikana, mutta merkittävin kehitys on mielestäni tapahtunut omassa toiminnassani ja siinä, miten tietoisesti seuran omaa toimintaani työssä.

Opinnäytetyön seurantajakso alkoi vaativalla ja pitkäkestoisella opiskelulla. Olin jo työhaastattelussa kertonut tavoittelevani sertifiointia jostakin pilvipalvelusta. Työnantajani haluaa mahdollisimman monen työntekijän sertifoivan itsensä, sillä se tuottaa arvoa työntekijän itsensä lisäksi myös yritykselle. Määrätietoinen usean viikon opiskelu palkittiin Microsoft Azure Fundamentals -sertifikaatilla. Opiskelu oli haastavaa ei vain siksi, että kokeessa kysytään pikkutarkkoja kysymyksiä ja kokeen suorittajan täytyy osata Azuren palvelut yksityiskohtineen ulkoa vaan myös siksi, että koulutusohjelmani opinnot ovat erittäin käytännönläheisiä ja perinteinen suurten konseptien ulkoa opiskelu on jäänyt vähemmälle. Sertifikaatin saaminen on myös hyvä pohja seuraavien sertifiointien hankinnalle. Tämä ei vielä tällä hetkellä ole itselleni ajankohtaista, mutta on tulevaisuuden urasuunnitelmissa. Sertifioinnit ovat lisäksi arvokkaita työmarkkinoilla ja asiakasprojektien saamisessa.

Sertifikaatin saatua olen kasvattanut ammatillista osaamistani opiskelemalla C#:a ja Vueta. Löysin molemmista yhtymäkohtia jo aiemmin opittuihin Javaan ja Reactiin, minkä vuoksi alkeiden oppiminen tapahtui luontevasti ja nopeasti. Uutta taitoa täytyy kuitenkin pitää yllä, jotta osaaminen vahvistuu ja monipuolistuu. Tunnistan tämän itselleni sellaiseksi asiaksi, jota täytyy pitää yllä jatkossa joko omien tai työprojektien kautta. Näiden kahden kanssa koen olevani edelleen aloittelija, enkä ole ehtinyt syventymään kumpaankaan teknologiaan vielä tarpeeksi.

Seurantajakson loppupuolella työni painottui visuaalista osaamista kehittävään työhön: valmistelin itseäni kahteen uuteen asiakasprojektiin Qlik Sensen sekä Figman kanssa. Qlik Sense oli minulle uusi työkalu ja Figma jo ennalta tuttu. Visuaaliset tehtävät ovat itselleni mieluisia, sillä näen suoraan oman käden jäljen. Tiedon visualisoiminen ja sen harjoittelu Qlik Sensessä jäi melko vähäiseksi, mutta odotan perehtymisen jatkuvan lähempänä projektin alkamista.

Päiväkirjamuotoisen oppinäytetyön aikana tutkin paljon itseäni oppijana ensinnäkin sen vuoksi, että sain opetella useita uusia asioita ja ohjelmia lyhyen ajanjakson sisällä. Lisäksi minun täytyi kyetä oppimaan lähes ulkoa suuria pilvipalvelukonsepteja tarkkoja yksityiskohtia myöten sertifiointikokeen vuoksi.

Päiväkirjan kirjoittaminen on pakottanut minut refleктоimaan omaa työntekojani ja olemaan tietoisempi siitä, mitä teen päivän aikana. Joka-aamuinen tavoitteiden asettaminen on tuonut määrätietoisuutta ja rakennetta päivään ja iltapäivisin tekemäni reflektointi tavoitteiden täyttymisestä on puolestaan kirkastanut itselleni sen, mitä on tehty ja mitä on vielä tekemättä. Tämä on auttanut seuraavaan työpäivään asennoitumisessa ja työtehtävien jäsentelyssä. Katsoisin tämän myös olevan minulle itselleni hyödyksi itsenäisessä työskentelyssä ja oman työnteon ohjaamisessa. Aiemmin olen enimmäkseen tehnyt työtä, jossa minulla on ennalta päätetty työtehtävä koko päivälle, ja työ on ajoittain tuntunut suorittamiselta. Edellisen työnantajan palveluksessa tekemäni digitaalisten palveluiden suunnittelemisen oli itsenäistä ja itseohjautuvaa työtä, mikä antoi minulle hyvän pohjan tähän työhön. Aion jatkossakin hyödyntää lyhyitä, työpäivää valmistelevia pohdintatuokioita, jolloin mietin oman työlistani päivälle.

## Lähteet

Boduch, A. 2017. React and React Native. Packt Publishing. Birmingham.

Ciliberti, J. 2017. ASP.NET Core Recipes: A Problem-Solution Approach. Luku 1. Apress. Sparta.

Codecademy 2022a. What is DOM? Luettavissa:

<https://www.codecademy.com/resources/blog/what-is-dom/>. Luettu 17.4.2022.

Codecademy 2022b. React: The Virtual DOM. Luettavissa:

<https://www.codecademy.com/article/react-virtual-dom>. Luettu 17.4.2022.

Fortino, A. 2020. Data Visualization for Business Decisions: A Laboratory Notebook. Mercury Learning and Information. Dulles.

Franklin J., Wanyoike, M., Bouchefra A., Silas K., Campbell C., Jacques N., Omole O. & Mulders, M. 2019. Working with Vue.js. Luku 4. SitePoint Pty. Ltd. Collingwood. E-kirja. Luettu: 8.5.2022.

Galloway, J., Wilson, B., Allen, K. S. & Matson, D. 2014. Professional ASP.NET MVC 5. Luku 1 ja 2. John Wiley & Sons, Inc. Indianapolis.

Hanchett, E. & Listwon, B. 2018. Vue.js in Action. Luku 1. Manning Publications. Shelter Island.

IBM 2021. IaaS vs. PaaS vs. SaaS. Luettavissa: <https://www.ibm.com/cloud/learn/iaas-paas-saas>. Luettu: 6.3.2021.

Jackson, C. & Ciolek, N. 2017. Digital Design in Action. CRC Press. Boca Raton.

Jacques, N. 2019. Jump Start Vue.js. Luku 1. SitePoint Pty. Ltd. Collingwood. E-kirja. Luettu: 8.5.2022.

Keisala, K. 2012. Monikulttuurisen työyhteisön viestintä. Tampereen Yliopistopaino Oy. Tampere.

Kortetjärvi-Nurmi, S. & Murtola, K. 2016. Areena: yritysviestinnän käsikirja. Luku 2. Edita. Helsinki.

Lahti, L. 2014. Monikulttuurinen työelämä: hyväksi ihmiselle, hyväksi bisnekselle. Talentum Media. Helsinki.

Martin, R. 2009. Clean Code: A Handbook of Agile Software Craftmanship. Pearson Education Inc. Boston.

Microsoft 2022a. What is Azure? Luettavissa: <https://docs.microsoft.com/fi-fi/learn/modules/intro-to-azure-fundamentals/what-is-microsoft-azure>. Luettu: 6.3.2022.

- Microsoft 2022b. Azure regions, availability zones, and region pairs. Luettavissa: <https://docs.microsoft.com/en-us/learn/modules/azure-architecture-fundamentals/regions-availability-zones?ns-enrollment-type=learningpath&ns-enrollment-id=learn.az-900-describe-cloud-concepts>. Luettu: 1.4.2022.
- Microsoft 2022c. Overview of ASP.NET Core MVC. Luettavissa: [https://docs.microsoft.com/en-gb/aspnet/core/mvc/overview?WT.mc\\_id=dotnet-35129-website&view=aspnetcore-6.0](https://docs.microsoft.com/en-gb/aspnet/core/mvc/overview?WT.mc_id=dotnet-35129-website&view=aspnetcore-6.0). Luettu: 7.3.2022.
- Microsoft 2022d. Overview to ASP.NET Core. Luettavissa: <https://docs.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-6.0>. Luettu: 7.3.2022.
- Mäntyneva, M. 2016. Hallittu projekti - Jäntevästä suunnittelusta menestykselliseen toteutukseen. Helsingin Kamari Oy
- React.js 2022. React. Luettavissa: <https://reactjs.org/>. Luettu: 19.4.2022.
- React.js 2022b. Introducing Hooks. Luettavissa: <https://reactjs.org/docs/hooks-intro.html>. Luettu: 9.5.2022.
- UnityCoin 9.8.2019. Clean Code – Uncle Bob / Lesson 1. Video. Katsottavissa: [https://www.youtube.com/watch?v=7EmboKQH8IM&ab\\_channel=UnityCoin](https://www.youtube.com/watch?v=7EmboKQH8IM&ab_channel=UnityCoin). Katsottu: 8.3.2022.
- Usability.gov 2022. User Interface Design Basics. Luettavissa: <https://www.usability.gov/what-and-why/user-interface-design.html>. Luettu: 26.4.2022.
- Vue.js 2022a. Introduction. Luettavissa: <https://vuejs.org/guide/introduction.html>. Luettu: 5.4.2022.
- Vue.js 2022b. Composition API FAQ. Luettavissa: <https://vuejs.org/guide/extras/composition-api-faq.html>. Luettu 5.4.2022.
- Vue.js 2022c. Comparison with other frameworks. Luettavissa: <https://v2.vuejs.org/v2/guide/comparison.html?redirect=true>. Luettu: 17.4.2022.
- Vue.js 2022d. Template Syntax. Luettavissa: <https://vuejs.org/guide/essentials/template-syntax.html>. Luettu: 9.5.2022.
- Yale Poorvu Center for Teaching and Learning 2021a. Learning Styles as a Myth. Luettavissa: <https://poorvucenter.yale.edu/LearningStylesMyth>. Luettu: 19.3.2022.

Yale Poorvu Center for Teaching and Learning 2021b. Encouraging Metacognition in the Classroom. Luettavissa: <https://poorvucenter.yale.edu/MetacognitioninClassrooms>. Luettu: 19.3.2022.

Wilke, C. O. 2019. Fundamentals of Data Visualization. O'Reilly Media Inc.

Åhlberg, M. 2018. Käsitekartat tutkimusmenetelmänä. Teoksessa Valli, R. & Aarnos, E. (toim.) Ikkunoita tutkimusmetodeihin: 1, Metodien valinta ja aineistonkeruu: virikkeitä aloittelevalla tutkijalle. PS-kustannus. Jyväskylä. E-kirja. Luettu: 20.3.2022.