



Ohjelmistokehittäjän työ alanvaihdon jälkeen

Jan Lohva

Haaga-Helia ammattikorkeakoulu

ICT-Tradenomi

Amk-opinnäytetyö

2024

Tiivistelmä

Tekijä(t) Jan Lohva
Tutkinto ICT-Tradenomi
Raportin/Opinnäytetyön nimi Ohjelmistokehittäjän työ alanvaihdon jälkeen
Sivu- ja liitesivumäärä 53
<p>Tässä päiväkirjamuotoisessa opinnäytetyössä keskityttiin seuraamaan alanvaihtajan työpäiviä ja tehtäviä CGI:llä, sekä osaamisen kehittymistä Amazonin pilvipalveluihin pohjautuvassa projektissa. Aluksi käytiin läpi työympäristö, työtehtävissä vaadittava osaaminen, sekä kehittymisen tavoitteet. Tämän jälkeen on kuvattuna keskeisimmät ammattikäsitteet, jotka auttavat lukijaa ymmärtämään työssä käytettyä sanastoa. Seuraavaksi tuli lähtötilanteen kuvaus, jossa käytiin läpi kirjoittajan taustaa osaamiselle. Näiden jälkeen oli nykyisen työn analysointi, jossa käsiteltiin osaamisen tasoa ja kehittämiskohteita, sekä sidosryhmien esittely, jossa käytiin läpi projektin, työnantajan ja asiakkaan välisiä yhteyksiä. Ennen seurantajakson alkua, käytiin vielä läpi työpäivän vuorovaikutustilanteita ja kokouksia, sekä niissä tarvittavia välineitä, taitoja ja haasteita.</p> <p>Seurantajakso koostui kahdeksasta seurantaviikosta, joissa seurantaa tehtiin viitenä päivänä viikossa. Seurantaviikot ajoittuvat aikavälille 22.04.2024 – 14.06.2024. Jokaisena seurantapäivänä käytiin ensin läpi päivän tehtävät ja tavoitteet. Tämän jälkeen tarkasteltiin tehtävien ja tavoitteiden toteutumista, sekä osaamisen kehittymistä. Jokaisen seurantaviikon jälkeen tarkasteltiin viikkoa yleisesti ottaen, sekä analysoitiin vastaan tulleita ongelmatilanteita ja miten niiltä olisi voinut välttyä.</p> <p>Lopuksi seurantajakson jälkeisessä pohdintaosiossa käytiin läpi kehittymistä ammattilaisena ja mitä oppi päiväkirjamuotoisen opinnäytteen kirjoittamisesta. Lisäksi käytiin läpi, miksei tässä työssä käytetty kirjallisuudesta löytyviä ratkaisumalleja/menetelmiä, sekä miten kirjoittaja aikoo tulevaisuudessa kehittää omaa osaamistaan.</p>
Asiasanat Ohjelmistokehitys, Amazon web services, asiantuntijana kehittyminen

Sisällys

1	Johdanto	1
1.1	Työnantajan, toimenkuvan ja työympäristön kuvaus.....	2
1.1.1	Työnantaja	2
1.1.2	Toimenkuva	2
1.1.3	Työympäristön kuvaus	3
1.2	Työtehtävissä vaadittava osaaminen.....	3
1.3	Ammatillisen kehittymisen tavoitteet.....	3
1.4	Keskeisimmät ammattikäsitteet	4
2	Lähtötilanteen kuvaus.....	5
2.1	Oman nykyisen työn analysointi	5
2.2	Sidosryhmien esittely	6
2.3	Työpaikan vuorovaikutustilanteet	7
3	Seurantajakson raportointi viikkoanalyyseineen	9
3.1	Seurantaviikko 1.....	9
3.2	Seurantaviikko 2.....	15
3.3	Seurantaviikko 3.....	20
3.4	Seurantaviikko 4.....	24
3.5	Seurantaviikko 5.....	29
3.6	Seurantaviikko 6.....	34
3.7	Seurantaviikko 7.....	38
3.8	Seurantaviikko 8.....	44
4	Pohdinta.....	49
	Lähteet.....	52

1 Johdanto

Tämä opinnäytetyö on toteutettu päiväkirjatyyppisesti ja sisältönä on ohjelmistokehittäjän työtehtävät ja ammatillinen kehittyminen AWS (Amazon web services) pilvipalveluihin pohjautuvassa projektissa. Kokemusta ohjelmistokehittäjän työtehtävistä on kertynyt puolitoista vuotta ja nykyisen projektin parissa olen ollut puoli vuotta. Vaihdoin IT-alalle oltuani neljä vuotta töissä kemiantekniikan parissa. Sauma alanvaihdolle avautui, kun pitkän hakuprosessin kautta pääsin Academic Work:n Academyn järjestämään kolmen kuukauden intensiivikoulutukseen. Tämän Academyn kurssin tavoitteina oli suorittaa AWS Certified Developer associate tason sertifikaatti ja työllistyä tiettyyn kohdeyritykseen ohjelmistokehittäjäksi. Opinnäytetyön tyyppi oli helppo valita, sillä päiväkirjatyyppistä opinnäytetyötä pystyy tekemään helposti töiden ohella ja samalla kehittää työtehtävissä tarvittavaa ammatillista osaamistaan. Päiväkirjaa kirjoitetaan kahdeksan viikon ajan ja viitenä päivänä viikossa. Kalenterissa seurantaviikot ajoittuvat aikavälille 22.04.2024 – 14.06.2024. Alla lueteltuna seurantaperiodin työtehtävät tiivistetysti.

1. viikko

- Tarkastukseen asettamisen rajapinnan selvittelyä ja suunnittelua, sekä alustavia toimenpiteitä, kuten tietovaraston perustamista.
- Rajapinta on REST tyyppinen, johon asiakas voi lähettää POST kyselyllä JSON muotoista dataa ja rajapinta muodostaa datan perusteella tarkastuksia.

2. viikko

- AWS resurssien pystyttämistä ja toimintalogiikan rakentamista rajapinnalle TypeScriptin avulla.
- Parityöskentelyä ja tyypitysongelmien ratkaisua.

3. viikko

- Rajapinnan kehitystyötä ja viimeistelyä, jotta kaikki asiakkaan vaatimat hyväksymiskriteerit täyttyvät.
- AWS API-Gateway ja SQS jonojen kanssa työskentelyä.

4. viikko

- Rajapinnan testausta erilaisilla datapaketeilla, asiakkaan yhteysavauksia ja yhteystestauksia, sekä dokumentaation kasaamista.

5. viikko

- Pienten ongelmien korjaamista, sekä paljon kokouksia, suunnittelua, koulutusta ja koodikatselmoitteja.

6. viikko

- Tarkastusraportin prosessin suunnittelua ja backendin toteuttamista TypeScriptillä.
- Tarkastusraporttipohjien tekemistä.
- Pdf tulostepalvelun opettelua.

7. viikko

- Tarkastusraportin prosessiin tarvittavien palveluiden yhdistämistä toimivaksi kokonaisuudeksi. Kehitysympäristöön piti kerralla viedä monta omaa tietovaraston kehityshaaraa, jotta testaus onnistui.

8. viikko

- Tietovarastojen testiympäristöön vientiä kehitysympäristöstä. Uutena työnä aloitin tarkastusraportin lataamisen ja näyttämisen käyttöliittymässä, joka oli pääasiassa käyttöliittymäkehitystä Reactilla.

1.1 Työnantajan, toimenkuvan ja työympäristön kuvaus

1.1.1 Työnantaja

Työnantajani on CGI Suomi Oy, joka kuuluu kanadalaislähtöiseen kansanväliseen IT-alan palveluyritykseen CGI:hin. CGI tarjoaa IT-alan konsultointia, IT-infrastruktuuriratkaisuja, tietojärjestelmien integraatiopalveluja, sekä IT- ja liiketoimintaprosessien ulkoistamispalveluja. Sen asiakaskuntaan kuuluvat julkisen ja yksityisen sektorin toimijat. CGI:llä on noin 3800 työntekijää, joka tekee siitä yhden suurimmista, ellei suurimman it-alan työnantajan Suomessa. Toimipisteitä löytyy mm. Helsingistä, Turusta, Tampereelta, Oulusta ja monista muista suuremmista kaupungeista. Itselleni lähin toimisto löytyy Helsingin Pitäjänmäestä. Toimistolla on 7 eri kerrosta, joissa voi työskennellä avokonttoreissa, neuvotteluhuoneissa ja äänieristetyissä kopeissa. Toimistolla on oma ruokala, kuntosali, saunatilat, kattoterassi ja paljon erilaisia aktiviteettimahdollisuuksia. Työnantaja tarjoaa kannettavan tietokoneen, työpuhelimien, puhelinliittymän, laajakaistan, virkistys ja lounasedun ja paljon muuta.

1.1.2 Toimenkuva

Työskentelen melko suuressa projektissa, jossa on noin 25 henkeä mukana päivittäin. Oma toimenkuvani on toimia ohjelmistokehittäjänä. Suurin osa projektin ohjelmistokehittäjistä on jo pitkään alalla työskennelleitä ja pieni osa noin 1 - 2 vuoden kokemuksen omaavia. Projektiin kuuluu projektipäällikkö, arkkitehtejä, ohjelmistokehittäjiä, Scrum-mastereita, projektin omistaja ja testaajia. Ohjelmistokehittäjänä työtehtävät saattavat vaihdella melko paljon, riippuen aina projektin vaiheesta.

Pääpiirteittäin työtehtävät liittyvät ohjelmointiin (käyttöliittymä ja backend), tietokantoihin ja AWS:n pilvipalveluihin.

1.1.3 Työympäristön kuvaus

Työtehtävät voi hoitaa täysin etänä tai käydä toimistolla, kun siltä tuntuu, oman mieltymyksen mukaan. Toimistolla käymistä kuitenkin kannustetaan erilaisin pörkkänoin ja olemmekin projektiporukan kesken sopineet epävirallisesti toimistopäiviksi torstait, jolloin jokainen pyrkisi pääsemään toimistolle. Tällainen malli on toiminut ainakin omalla kohdalla todella hyvin. Projektissa työkielenä käytetään Suomea, mutta koodissa saa olla asioita myös englanniksi, selkeyden ja ymmärtävyyden vuoksi. Kommunikointiin ja palavereihin käytetään Microsoftin Teamsiä. Päivät alkavat aina aamupalaverilla ja päivän asioita voi käydä lävitse erilaisilla Teams kanavilla tai järjestää itse tarvittavia palavereja esimerkiksi asiakkaan tai tiimiläisten kanssa. Monesti teknisten ongelmien läpikäyntiin käytetään myös ryhmäkeskusteluja.

1.2 Työtehtävissä vaadittava osaaminen

Työtehtävissä vaaditaan Python ja TypeScript ohjelmointikielten, SQL kyselykielen ja AWS:n pilvipalveluiden osaamista ja ymmärtämistä. Git kommentojen ja AWS:n CodeCommit osaaminen versioinnhallinnassa on elintärkeää. AWS näkyy joka puolella ja on osana lähes jokaisessa tiedostossa mitä tehdään tai muokataan. Kaikissa projekteissa, joissa olen ollut mukana, on ollut käytössä Jira (tehtävienhallintaohjelmisto) ja Confluence (dokumentointiohjelmisto), joiden kautta työntekoa ja dokumentointia hallitaan. Kehittämisessä täytyy hahmottaa miten käyttöliittymä ja backend keskustelevat keskenään ja miten kaikki osat keskustelevat AWS:n kanssa. Lisäksi asiakaspalvelutaidot ja kommunikaatio ovat hyvin tärkeitä, sillä asiakas määrittelee mitä, miksi ja milloin tehdään. Usein on hyvä varmistaa asiakkaalta, että miten toteutuksessa asiat toimivat käytännön näkökulmasta ja monesti vastaan tulee paljon pieniä yksityiskohtia, joita suunnittelussa ei välttämättä ole osattu huomioida.

1.3 Ammatillisen kehittymisen tavoitteet

Tavoitteenani on kehittyä ammattilaisena kaikilla mahdollisilla osa-alueilla. Painopisteenä tällä hetkellä on parantaa teknistä ymmärrystä, hyviä käytänteitä, käsitteitä ja sanastoa yleisesti ottaen. Tavoitteina on myös ongelmanratkaisun kehitys ja käyttöliittymäkehityksen oppiminen. Monesti ohjelmistokehittäjänä saattaa myös liikaa keskittyä tekemään pientä osaa isommasta kokonaisuudesta. Tavoitteena on sen osalta pyrkiä välillä pysähtymään ja katsomaan suurempaa kokonaiskuvaa, jolloin hahmottaa paremmin oman tekemisen tarkoituksen ja käyttötavat. Tästä voi olla paljon apua yksittäisiä ja pienempiä teknisiä valintoja tehdessä.

1.4 Keskeisimmät ammattikäsitteet

Taulukko 2 Keskeisimmät ammattikäsitteet

Käsite	Selite
AWS (Amazon Web Services)	Pilvipalvelualusta, joka tarjoaa erilaisia työvälineitä verkkopalveluiden rakentamiseen ja pilvipalveluympäristöjen luomiseen.
Pilvipalvelut	Tietoteknisten palveluiden, kuten infrastruktuurin tai datakeskusten toimitamista ja vuokraamista internetin välityksellä, jossa palveluista laskutetaan käytön mukaan.
Käyttöliittymä	Käyttöliittymä on se ohjelmiston osa, jonka käyttäjä näkee ruudullaan ja jonka kautta käyttäjä pystyy käyttämään palvelua.
Backend	Sisältää kaiken palvelimella suoritettavan sovelluksen tai verkkosivuston osat, johon kuuluvat esimerkiksi yhteydet tietokantoihin, lomakkeiden ja tiedostojen käsittelyt, sovellukseen tai sivustolle kirjautuminen, sekä salasanojen tarkistaminen.
Git	Git on avoimen lähdekoodin versionhallintajärjestelmä, joka seuraa muutoksia tiedostojoukossa. Käytetään koordinoimaan ohjelmoijien työtä, jotka kehittävät yhdessä lähdekoodia ohjelmistokehityksen aikana.
Codecommit	AWS:n isännöimä versionhallintapalvelu, jonka avulla voi tallentaa ja hallita lähdekoodia pilvessä.
SQL	Strukturoitu kyselykieli, jota käytetään tietojen tallentamiseen ja käsittelyyn relaatiotietokannoissa.
Python	Yleiskäyttöinen ohjelmointikieli, jota käytetään verkkosivustojen ja ohjelmistojen luomiseen, tehtävien automatisointiin ja tietojen analysointiin.
TypeScript (TS)	Microsoftin kehittämä ilmainen ja avoimen lähdekoodin korkean tason ohjelmointikieli, joka lisää JavaScriptiin tyyppityksen.
Lambda	AWS:n laskentapalvelu, jolla voidaan suorittaa koodia ilman palvelimien proviisointia tai hallintaa.
Funktio	Itsenäinen koodimoduuli, joka suorittaa tietyn tehtävän, yleensä ottaa vastaan tietoa, käsittelee sen ja palauttavat tuloksen
Rajapinta / API (Application Programming Interface)	Sovellusohjelmointirajapinta mahdollistaa eri sovellusten tai sovellusten osien kommunikoinnin keskenään.

2 Lähtötilanteen kuvaus

Syksyllä 2022 suoritin Academic Workin 3kk intensiivikoulutuksen, jonka tarkoitus oli työllistyä kohdeyritykseen. Koulutuksessa suoritettiin AWS (Amazon web services) certified developer – Associate sertifikaatti ja opiskeltiin JavaScriptiä, sekä yleisiä ohjelmointikäytäntöjä ja välineitä.

Academic Workin kurssilta valmistumisen jälkeen aloitin työt, toimien alikonsulttina, eli kohdeyritys maksoi työstäni Academic Workille ja Academic Work maksoi palkkani. Työskentelin alikonsulttina hieman reilu vuoden verran, jonka jälkeen tein työsopimuksen kohdeyrityksen kanssa.

Tähän mennessä työkokemusta on kertynyt noin 1,5 vuotta ja olen työskennellyt kolmessa eri asiakasprojektissa.

2.1 Oman nykyisen työn analysointi

Työtehtäväni on tällä hetkellä olla ohjelmistokehittäjä, joka toimii osana projektitiimiä. Olen yksi projektin nuorimmista kehittäjistä, joten vastuulleni ei laiteta vaativimpia tai laajimpia kokonaisuuksia. Muuten työtehtäväni ovat hyvin pitkälti samanlaisia, kuin kokeneemmillakin kehittäjillä.

Työtehtävissä tarvitaan ohjelmointiosaamista TypeScript ja Python ohjelmointikielillä, sekä ymmärrystä ja osaamista SQL:stä ja tietokannoista. Lähes jokaisessa työtehtävässäni täytyy kirjoittaa TS tai Python koodia ja muokata tietokannan taulujen tietoja tai hakea tietoja tietokantatauluista. Lisäksi täytyy osata käyttää AWS:n tarjoamia palveluita hyvin monipuolisesti. On hyvä osata hahmottaa miten käyttöliittymä ja backend keskustelevat keskenään ja miten backend osaa tehdä tarvittavat asiat AWS:n puolella. Lisäksi asiakaspalvelutaidot ja kommunikaatio ovat hyvin tärkeitä, sillä asiakkaan kanssa on tekemisissä päivittäin.

Työssä tarvittava osaaminen on muodostunut vähitellen Haaga-Helian kurssien, AW Academyn ja töiden tekemisen kautta. Ensimmäisessä asiakasprojektissa aloitettuani sain myös hyvin opastusta ja apua senior kehittäjiltä. Suurin osa osaamisesta on kehittynyt konkreettisia työtehtäviä tehdessä.

Ammatillinen kehittymiseni koen olevan vasta alkutaipaleella, eli kaikki tarvittava perusosaaminen löytyy monelta osa-alueelta, joka mahdollistaa itsenäisen työskentelyn. Kehitettävää on kuitenkin paljon ja IT-alalla toimiminen on jatkuvaa kasvamista ammattilaisena.

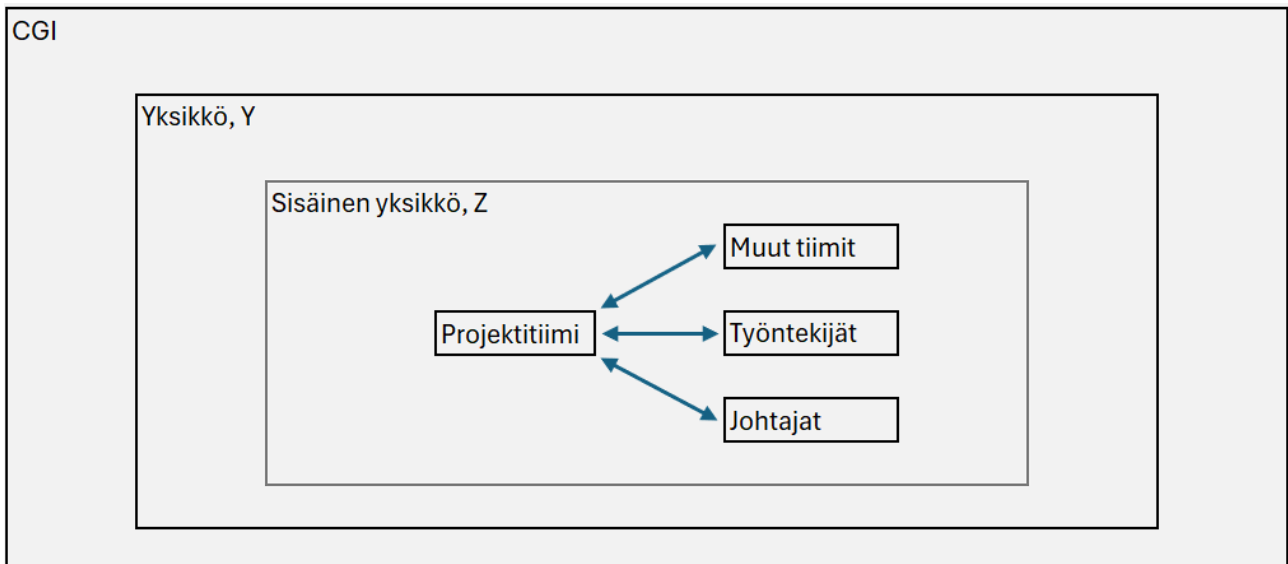
Jatkossa tulen panostamaan eniten tekniseen kehittymiseen, sillä projekteissa käytettäviä teknologioita on todella paljon ja niitä tulee koko ajan vain lisää opittaviksi. Vähiten olen työskennellyt käyttöliittymien parissa, joten niissä riittää tässä kohtaa eniten oppimista. Käyttöliittymäpuolella esim. React olisi hyvä saada haltuun ainakin perusosaamistasolla, sillä sitä käytetään nykyisessä projektissani paljon.

Koen osaamiseni olevan riittävää suhteessa osaamisvaatimuksiin, sillä harvoin joutuu työskentelemään sellaisten teknisten asioiden parissa, joista ei ole mitään kokemusta tai tietoa. Vaikka tällainen tilanne tulisikin vastaan, niin siihen usein saa perehdytystä ja tukea joltain tiimijäseneltä.

Koen osaamiseni olevan jossain taitavan suoriutujan ja aloittelevan toimivan välissä, sillä minulla on usein ymmärrys tekemistäni työtehtävistä, mutta tarvitsen yleensä jotain täydentäviä tietoja niiltä henkilöiltä, jotka ovat työn suunnitelleen ja kuvailleet tiketille. Tämä saattaa myös johtua suunnittelun aikapaineesta, että tärkeitä asioita on unohtunut mainita. Olen päivittäin tekemisissä kokeneiden kehittäjien kanssa, enkä missään tapauksessa ole vielä lähelläkään sitä tasoa. Työpäivän puolelta ei niinkään ole erityisvaatimuksia osaamiselle, vaan tärkeintä on, että työtehtävät edistyvät ja junioreimmat kehittäjät saavat kasvatettua osaamistaan.

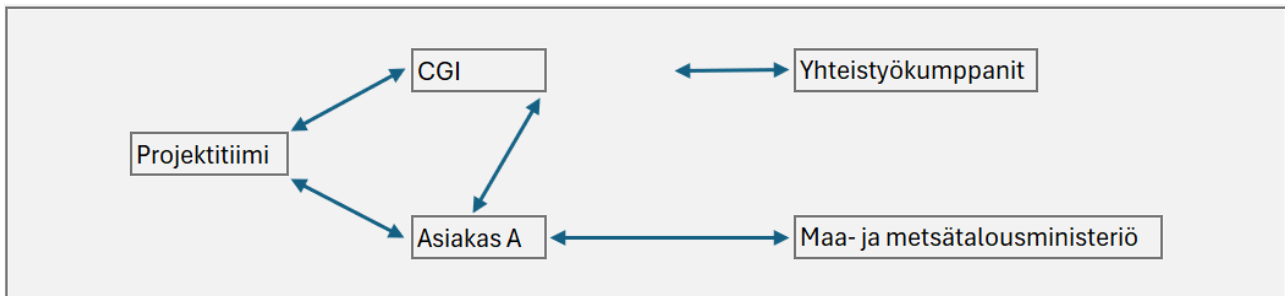
2.2 Sidosryhmien esittely

Sisäisiin sidosryhmiin kuuluvat työnantaja, yksikkö, sisäinen yksikkö, yksiköiden työntekijät ja johtajat, sekä projektitiimi ja muut projektitiimit. Työnantajani on suuri yritys ja toiminta on jakautunut moneen erilliseen yksikköön, joiden sisällä on vielä useita pienempiä sisäisiä yksiköitä. Yksikössä Y työntekijöitä on noin 800 ja sisäisessä yksikössä Z noin 100. Projektitiimiini kuuluu noin 20 henkilöä. Tämänhetkinen asiakasprojektini on riippuvainen myös monesta muusta rinnakkaisesta projektista, joita tehdään samalle asiakkaalle samanaikaisesti. Projektit hyödyntävät toisiaan erilaisten rajapintojen kautta. Kuvassa 1 on kuvattuna keskeisimmät sisäiset sidosryhmät.



Kuva 1 Sisäiset sidosryhmät

Ulkoisia sidosryhmiä ovat työnantajan yhteistyökumppanit, varsinainen asiakas, jolle projektia tehdään ja viranomaispuolella maa- ja metsätalousministeriö. Maa- ja metsätalousministeriö määrittelee asiakkaalle lainsäädännön näkökulmasta tarvittavat asiat ja yksityiskohdat, joita projektissa toteutetaan. Työnantajalla on myös lukuisia yhteistyökumppaneita laitehankinnan puolella, kuten Elisa, Samsung, Apple, Dell ja niin edelleen. Kuvassa 2 on kuvattuna ulkoiset sidosryhmät.



Kuva 2 ulkoiset sidosryhmät

Oman työni kannalta tärkeimpiä ovat asiakkaan intressit, koska ne ovat myös projektitiimin ja työnantajan näkökulmasta kaikkein tärkeimpiä. Olemme tekemässä tärkeää ja isoa palvelua, jonka kautta tulee liikkumaan suuria rahavirtoja ja joka vaikuttaa merkittävästi Suomen metsätalouteen ja luontoon. Asiakkaan intresseihin vaikuttaa merkittävästi Maa- ja metsätalousministeriö, joka määrittelee lakien voimaantulospäivämäärät. Nämä päivämäärät heijastuvat omaan ja projektin työhön niin, että tiettyjä asioita täytyy olla valmiina tiettyihin päivämääriin mennessä ja tämä puolestaan aiheuttaa huomattavaa aikapainetta. Asiakkaan intressit välittyvät minulle yleensä palaverien kautta, mutta myös projektipäällikkö, arkkitehti ja scrum master tuovat hyvin asiakkaan toiveita ja tavoitteita esiin.

2.3 Työpaikan vuorovaikutustilanteet

Aloitin työt koronapandemian jälkeen, jolloin suurin osa työntekijöistä oli tottunut tekemään töitä täysin etänä ja ensimmäisen asiakasprojektini tiimiläiset halusivat tehdä työt täysin etänä. Omakseen nopeasti saman tavan kuin tiimiläisilläni, sillä en toimistolla tuntenut vielä ketään. Onneksi työnantajan puolesta järjestettiin after work -tapahtumia ja pikkujoulut, joissa pääsi paremmin tapaamaan ja juttelemaan oman yksikön väen kanssa. Myös asiakkaan puolelta järjestettiin suunnittelupäivät melko pian aloittamiseni jälkeen, jonne jokainen projektitiimiläinen osallistui. Nuo suunnittelupäivät auttoivat hyvin tutustumaan ja ymmärtämään muiden ammatillisia ja roolillisia taustoja.

Nykyään osa projektitiimiläisistä käy melko säännöllisesti toimistolla ja pyrimme pitämään yhden toimistopäivän viikossa, jolloin suurin osa pyrkisi pääsemään paikalle. Toimistolla vuorovaikutustilanteet eivät ole niin muodollisia ja pääsee tutustumaan ihmisiin paremmin.

Suurin osa keskusteluista ja kaikki palaverit käydään Teamsin välityksellä. Päivittäin on dailyt eli aamupalaverit, joissa käydään päivän suunnitelma läpi yleisesti ja työntekijäkohtaisesti, sekä ilmoitetaan mikäli on jotain tärkeitä uutisia. Aamupalavereissa voidaan käydä myös yleisesti kuulumisia läpi, sekä työnteon esteitä, poissaoloja ja lomasuunnitelmia. Näihin dailyihin niin kuin suurimpaan osaan muistakin palavereista osallistuu myös aina asiakkaan puolelta väkeä.

Viikkotasolla järjestetään viikkopalaveri, testaushavaintojen läpikäynti ja teknisten ongelmien läpikäynti. Joka sprintin päätteeksi asiakkaalle järjestetään sprinttidemo, jossa esitellään 2 viikon aikana valmistuneet toteutukset ja sprinttiretro. Näissä demoissa jokainen kehittäjä esittelee oman työnsä tulokset ja pääsee kuulemaan asiakkaan näkökulman toteutuksesta. Retrossa käydään läpi sprintin onnistuminen tavoitteisiin nähden, hyvät ja huonot asiat, kehityskohteet ja parannusehdotukset.

Kuukausitasolla vuorovaikutustilanteita ovat yksikköinfot ja esimiehen kanssa käytävät 1on1 keskustelut. Yksikköinfoissa käydään läpi yleisesti yksikön kuulumiset ja uutiset, mahdolliset työntekijöiden vuosipäivät ja esimerkiksi opinnäytetyöesitykset. Esimiehen kanssa käytävissä keskusteluissa voi puhua mistä vaan. Yleensä käydään läpi kuulumiset ja ratkotaan jos on tullut vastaan jotain ongelmia.

Etänä työskennellessä en ole huomannut suurempia haasteita kommunikoinnissa, sillä asioista pystyy hyvin pienellä kynnyksellä keskustelemaan niin tiimin kuin asiakkaankin kanssa ja varamaan yhdessä palaveriajan joko kahden kesken tai isommalla porukalla jos se havaitaan tarpeelliseksi. Ja vaikka ei ehtisi kaikkia Teams chatteja ja Teams kanavilla tapahtuvia keskusteluja seuraamaan niin yleensä riittää lukea vain ne viestit, joihin itse on pingattuna.

Pieneksi haasteeksi ja kehityskohteeksi kuitenkin näen asiakkaalle esiintymisen. Olen persoonallisuuspiirteiltäni introvertti, mikä monesti aiheuttaa sen, että jännitän esiintymistä ja luon turhaan itselleni liikaa paineita ja tämä saattaa välillä näkyä myös asiakkaalle. Toisaalta uskon ja tiedän asiakkaalta löytyvän ymmärrystä tähän, sillä olen huomannut, että monilla muilla on myös kehitettävää tämän osalta.

3 Seurantajakson raportointi viikkoanalyysineen

Seurantajakso koostuu kahdeksasta peräkkäisestä viikosta. Joka viikolla kirjataan päivittäistä seurantaan maanantaista perjantaihin. Jokaisen päivän kohdalla kirjataan ennen työpäivän alkua päivän tavoitteet, kuvaus päivän tehtävistä ja miten tehtävät liittyvät osaamisen kehittämisen teemoihin. Työpäivän lopussa käydään läpi, miten tavoitteet ja tehtävät onnistuivat, sekä miten osaaminen kehittyi päivän aikana. Viikon lopuksi tehdään viikkoanalyysi, missä pohditaan viikon aikana tulleita ongelmia ja niiden ratkaisuja, sekä esitellään viikon aikana selvitettyt asiat.

3.1 Seurantaviikko 1

Maanantai 22.04.2024 tehtävät ja tavoitteet

Päivän tehtävinä on:

- Kerrata mihin pisteeseen ehti tekemään uuden rajapintatietin aloitusta ja selvitystyötä
- Selvittää arkkitehdiltä, miten uusi rajapinta halutaan toteuttaa ja minne se rakennetaan
- Tarkistaa, että edellisellä viikolla valmistunut käyttöliittymätoteutus toimii edelleen, eikä vaadi itseltäni toimenpiteitä
- Osallistua seuraaviin palavereihin: daily, viikkopalaveri, yksikköinfo

Yksi päivän tavoitteista on käydä läpi kaikki keskustelut kanavilla ja chateissa, asiakkaan Teams, sähköpostit, jotta pääsee kaikesta ajan tasalle, sillä olin viime viikolla kipeänä loppuviikosta. Päivän päätavoitteena on saada parempi ymmärrys ja hahmotus siitä, minkälaista rajapintaa lähden toteuttamaan. Tavoitteena ja toiveena olisi myös saada jotain konkreettista tehtyä. Jokaisen työpäivän aikana pyrin myös seuraamaan muiden tiimiläisten tekemistä ja auttamaan heitä, mikäli se on mahdollista. Käytännössä tämä näkyy useimmiten katselmointitilaisuuksiin osallistumisena, joissa kehittäjä demoaa, eli esittelee muille kehittäjille valmista toteutusta ja koodia, sekä käy läpi tietin kuvauksen ja vaatimukset. Katselmointitilaisuuksien tarkoituksena on löytää kehityskohteita ja virheitä, jotta ne saadaan korjattua ennen kehityshaaran liittämistä codecommitin päähaaraan. Mikäli kaikki näyttää hyvältä, niin katselmoija (toinen kehittäjä) antaa hyväksynnän kehityshaaran yhdistämispyynnölle (PR) ja merkitsee hyväksymiskriteerit kuitatuiksi.

Päivän tehtävät liittyvät osaamisen kehittämisen teemoissa ongelmanratkaisuun, tiedonhakuun ja teknisiin asioihin. Täytyy löytää vastaukset sellaisiin asioihin, joita tiketille ei ole kirjattu. Ensimmäinen ongelma on, että minne uusi rajapinta toteutetaan. Vaihtoehtoina on tehdä uusi tietovarasto eli säilytyspaikka lähdekoodille tai tehdä kehityshaara jo olemassa olevaan tietovarastoon. Kun se on ratkaistu, niin seuraavana ongelmana on, että miten rajapinta toteutetaan. Toteutustapoja saattaa

olla erilaisia ja niistä täytyy osata valita tähän tapaukseen sopivin. Nämä ongelmat saattavat kulkea myös vähän käsi kädessä. Minulla ei kauheasti ole vielä kokemusta rajapintojen kehityksestä, joten vastaan voi tulla paljon uutta teknistä asiaa opeteltavaksi ja sisäistettäväksi.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Sain käytyä keskustelut ja viime viikon asiat läpi, joten koen olevani taas ajan tasalla. Viime viikolla valmiiksi saamani käyttöliittymätoteutus toimi edelleen ja sen osalta ei tarvinnut tehdä mitään. Pääsin hyvin alkuun rajapinnan toteutuksen kanssa, vaikka arkkitehti ei ollutkaan töissä. Sain näkökulmia toiselta kokeneelta kehittäjältä ja päädyimme, että rajapinnalle tehdään oma tietovarasto. Tämä ratkaisu valikoitui siitä syystä, että asiakas käyttää rajapintakutsuihin FME:tä (paikkatietoaineistojen monipuoliseen muuntamiseen, muokkaamiseen, yhdistämiseen ja jakamiseen tarkoitettu ohjelmisto) omasta konesalistaan, joten tietoliikenteen raja-
aus helpottuu. Toisena syynä oli, ettei yksi olemassa olevista tietovarastoista paisu liian suureksi ja kolmanneksi, ettei saman tietovaraston parissa työskentele liian montaa henkilöä, joka saattaa vaikeuttaa/hidastaa esim. käyttöönottoja ja infrastruktuurin pystytystä. Pääsin aloittamaan tietovaraston rakentamista, jonka kanssa otin mallia toisen rajapinnan rakenteesta. Aamupäivästä tuli myös tieto ongelmasta, joka liittyi tekemääni automaattitarkastus lambdaan ja sen selvittämiseen meni aikaa. Ongelma ratkesi ja liittyi lambdaan käyttämään toisen projektin rajapintaan. Rajapinnasta puuttui yksi olennainen kenttä WFS (Web Feature Service) rajapinnasta.

Pääsin ensimmäistä kertaa pystyttämään AWS CDK sovelluksen ja pääsin tutustumaan miten CDK stackissä määritellään AWS resurssit. Aikaisemmin ei ole tarvinnut kauheasti perehtyä siihen, mitä kaikkia tiedostoja tarvitaan ja miten ne linkittyvät toisiinsa, jotta saadaan tietokantayhteydet ja paikallinen kehityspuoli toimimaan. Sain tänään hyvin monipuolista käytännön kokemusta AWS projektin rakenteista ja kehityksestä, mutta paljon jäi vielä opeteltavaa.

Tiistai 23.04.2024 tehtävät ja tavoitteet

Päivän tehtävät:

- Rakentaa rajapinnan tietovarastoa eteenpäin
- Perehtyä paremmin AWS CDK
- Selvittää rajapinnan logiikkaa
- Tuurata toisessa projektissa, sillä ylläpitäjä on lomilla
- Daily

Tavoitteena olisi päästä käymään arkkitehdin kanssa läpi rajapinnan logiikkaa ja infrastruktuurin pystyttämistä. Mikäli tämä ei ole mahdollista, niin täytyy koittaa selvittää muilta kehittäjiltä, ketkä

ovat olleet suunnitteluprosessissa mukana. Tietovarastossa tarvitaan paljon erilaisia tiedostoja, niin täytyy selvittää mitkä kaikki ovat tarpeellisia tämän rajapinnan kannalta. Tarkoituksena olisi perehtyä AWS CDK ja CDK stackin asioihin. Miten esim. AWS API Gateway pystytetään. Toisessa projektissa on tapahtunut VPN muutoksia, joten täytyy ottaa selvää niistä ja varmistaa että saan yhteydet etäkoneelle etätyöpöydän kautta. Päivän tehtävät nojautuvat vahvasti ongelmanratkaisuun, teknisiin asioihin ja kehittäjän kehittymiseen.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Sain rajapinnan tietovarastoa rakennettua siihen pisteeseen, että sitä voi kohta ajaa paikallisesti. Sain monipuolisesti kokemusta AWS:n arkkitehtuurin rakentumisesta, kun katsoi ja otti mallia toisen rajapinnan rakenteesta. Etenemistä hieman hidasti se, että arkkitehti oli sairaslomalla tänään, samoin kuin niin sanotusti vara-arkkitehtikin. Toisen projektin VPN yhteyden kanssa tuli ongelmia liittyen käyttäjätunnuksiin, mutta sain ne ratkaistua ja tein muutaman asiakkaan toivoman ylläpito-tehtävän. Tässä ongelmanratkaisu tosin vei hieman liikaa työajasta, joka jarrutti rajapinnan tekemistä ja selvitystyötä.

Sain päivän aikana kasvatettua lisää typescript osaamista liittyen luokkiin ja tyyppeihin. Sain myös lisää kokemusta AWS CDK stackin api gatewayn, vpc:n ja securitygroupin määrittämisistä.

Keskiviikko 24.04.2024 tehtävät ja tavoitteet

Tehtävät:

- Arkkitehdin kanssa läpikäynti rajapinnan asioista
- Rajapinnan edistäminen
- Daily

Tavoitteena on käydä arkkitehdin kanssa rajapinnan toteutusta ja siihen liittyviä huomioitavia asioita, sillä tiketille ei aina ole merkittynä kaikkia tarvittavia huomioitavia asioita. Täytyy selvittää miten asiakas käyttää rajapintaa ja haluaa rajapinnan toimivan ja katsoa vaikuttaako se toteutukseen millä tavalla. Tavoitteena olisi myös kokeilla ottaa käyttöön resurssit AWS:n puolelle.

Päivän tehtävät pyörivät hyvin laajasti teknisten asioiden ja ongelmanratkaisun ympärillä. Paljon selvitystyötä ja sen perusteella tehtäviä päätöksiä.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Yleisesti ottaen tavoitteet etenivät hyvin, arkkitehti oli vielä kipeänä, mutta sain häneltä kuitenkin huomioitavia asioita ja vastauksia kysymyksiini. Sain tehtyä manuaalisesti koonnin, eikä se tuottanut virheitä, mikä oli lupaavaa. Sain myös käynnistettyä rajapinnan paikallisesti, joten pääsin aloittamaan rajapinnan toiminnallisuuden rakentamista. Toiminnallisuuden osalta täytyy vielä miettiä, että mitä kaikkia toimenpiteitä rajapinta tekee siihen saapuvalle datalle. Itselläni on tietynlainen näkemys, mutta se on hyvä käydä läpi vielä muiden kesken. Sain ohjeistukseksi, että ennen käyttöönottoa dev eli kehitysympäristöön, kannattaa lukea muutosraportti läpi cdk diff komennolla. Tuota en vielä ehtinyt testaamaan ja perehtymään, että mistä siinä on kyse.

Torstai 25.04.2024 tehtävät ja tavoitteet

Tehtävät:

- Rajapinnan toiminnallisuuden rakentamista
- Cdk stack rakenteen selvittäminen
- Daily

Tavoitteena olisi käydä läpi rajapinnan toiminnallisuutta, minkälaisia tietokantamuutoksia siinä suoritetaan, minkälaista virheenkäsittelyä tarvitaan. Täytyisi selvittää myös sitä, miten rajapinnan käyttö saadaan rajattua tiettyyn konesaliin. Pitäisi tutustua ja selvittää mitä kaikkea cdk diff muutosraportti kertoo. Osaamisen kehittämisen teemojen puolesta päivä nojautuu jälleen teknisten asioiden ja termien selvittelyyn, sekä ongelmanratkaisuun, sillä rajapinnan kehityksessä varmasti tulee vastaan erilaisia virheitä.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Sain työpäivän aikana selvitettyä ja selkeytettyä ajatuksiani siitä, mitä rajapintaan tulevalle datalle halutaan tehdä. Rajapinta tulee vastaanottamaan JSON muotoista dataa POST kutsun rungosta. Olin aikaisemmin jo hahmotellut JSON skeemaa, jonka avulla validoidaan saapuva data, mutta se vaati vielä muokkaamista ja sain implementoitua sen nyt käyttöön. Sain myös varmistuksen siitä, että rajapinta muodostaa datan avulla tietokantaan "tarkastuksia", jotka muodostuvat erillisinä riiveinä ja sisältävät tietyn määrän tietokenttiä. Tämän jälkeen rajapinnan tulisi lähettää tiedot uusista tarkastuksista AWS:n olevaan SQS (Simple Queue Service), josta tarkastukset ohjautuvat tarkastusprosessiin. Tämä SQS jonoon lisääminen vaatii vielä enemmän perehtymistä, jotta hahmottaa miten se kannattaa ja miten sen pystyy tekemään helpoiten.

Päivän aikana tuli myös yllättävä korjauspyyntö asiakkaalta liittyen yhteen tekemistäni automaattitarkastuksista. Tämä vaati ongelman syyn selvittämistä ja paljastui, että spesifikaatioihin oli laitettu

yhteen kohtaan O-kirjain nollan tilalle. Tämä ratkesi pienellä tietokantamuutoksella, jossa tosin tuli vastaan itselle uusi virhe. Löysin tähän vastauksen projektin kanavalta keskusteluja selaamalla, sillä samaa ongelmaa oli tapahtunut muille tiimiläisille aikaisemmin.

Yleisesti osaamisen kehittämisessä pääsin hyvin vauhtiin rajapinnan paikallisessa kehityksessä ja opin miten sitä kannattaa tehdä VScoden Thunder client -laajennuksen avulla. Ehdin myös hieman tutustumaan SQS jonojen toimintaan ja miten niitä on tässä projektissa käytetty. Opin myös käyttämään uuden tyylistä tietokannan sql komentoa, jolla saadaan lisättyä kätevästi dataa useampiin kenttiin samanaikaisesti.

Perjantai 26.04.2024 tehtävät ja tavoitteet

Päivän tehtävät:

- Selvittää miten tarkastusprosessiin lisätään tarkastuksia SQS jono kautta ja miten se toteutetaan rajapinnassa
- Edistää tarkastusten muodostamista rajapinnassa
- Sprinttidemoon valmistautuminen

Tavoitteena on edistää rajapinnan toiminnallisuutta, jotta tarkastuksien lisääminen tietokantaan toimisi ja että muodostetut tarkastukset saataisiin SQS jonon kautta tarkastusprosessiin. Tarkastusprosessissa on useita lambdaa, joita suoritetaan tietyssä järjestyksessä, riippuen tarkastuksen tiedoista. Se käytännössä ohjaa tarkastuksen oikeaan tilaan ja päivittää tarvittavat tiedot tietokantaan. Tässä tosin on vielä todella paljon tehtävää ja selvitettävää ennen kuin kaikki toimii.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Kävimme projektitiimin kanssa läpi rajapinnan toimintaa ja omia suunnitelmia sen suhteen. Sain vahvistusta omaan tekemiseen ja hieman askelmerkkejä, että missä järjestyksessä asiat kannattaa tehdä. Paikallisen kehityksen kautta rajapinta eteni hyvin, mutta tuli vastaan tietokannan taulun päivitysongelmia, jotka johtuivat typescriptin tyyppityksistä, niissä riittää opettelemista, vaikka melkein joka päivä osaaminen kehittyykin. Täytyynee rajapinnan lambdan perusrungon tekemiseen keskittyä ensi viikolla, ennen kuin lähtee toden teolla miettimään SQS jonotusta. Sovimme myös erään kokeneemman AWS kehittäjän kanssa, että ensi viikon alkupuolella voisimme yhdessä käydä läpi cdk stack tiedoston, jonka avulla AWS resurssit rakennetaan. Siihen perehtyminen yksin saattaa olla turhan aikaa vievää, joten kaverin avulla saadaan säästettyä aikaa ja hermoja.

Viikkoanalyysi

Viikko on ollut hyvin poikkeava aikaisemmista viikoistani projektissa, ja sen vuoksi tuottanut paljon ongelmia. Projekti jakaantui vasta vähän aikaa sitten kahteen pienempään ryhmään, jonka seurauksena toiminta ja roolitukset muuttuivat paljon. On täytynyt pohtia täysin erilaisia asioita kuin tavallisesti ja koittaa hahmottaa suurempia kokonaisuuksia kuin normaalisti. Monen ei aikaisemmin ollut tarvinnut esimerkiksi tehdä käyttöliittymäpuolta vielä ollenkaan, joten se osa tuo paljon haasteita. Jokaiselle on tullut täysin uutena asiana iso projektin kokonaisuus, jota ei aikaisemmin olla toteutettu vielä ollenkaan. Tähän liittyy paljon uutta termistöä ja AWS rakenteita. Hahmotus kokonaiskuvasta alkaa vasta pikkuhiljaa rakentumaan, kun saadaan pieniä palasia valmiiksi. Onneksi arkkitehti ja suunnittelussa olevat henkilöt ovat usein helposti ja nopeasti saatavilla, mikäli kysyttävää ilmaantuu. Tämä viikko oli siitäkin poikkeuksellisen haastava, että oli paljon väkeä lomilla tai kipeänä

Suurimmat ongelmat tällä viikolla olivat:

- Rajapinnan sijainti
- Rajapinnan toiminnallisuus
- Miten asiakas käyttää rajapintaa
- CDK projektiin liittyvät asiat
- Typescript ohjelmointikielenä

Alkuvuikosta rajapinnan sijainti ratkesi kohtuu nopeasti kysymällä apua kokeneelta kehittäjältä. Toiminnallisuus hahmottui pitkin viikkoa miettimällä tiketin hyväksymiskriteereitä, kysymällä mielipiteitä ja näkemyksiä muilta kehittäjiltä, sekä lukemalla dokumentaatiota ja tutustumalla aikaisempiin toteutuksiin. Varmistin myös omaa näkemystä kysymällä asiakkaalta, että miten ja mistä kautta he aikovat rajapintaa käyttää. AWS:n cloud development kitistä tuli opittua jonkin verran koodia ja dokumentaatiota lukemalla. Typescriptin kanssa suurimmat ongelmat itsellä tulevat tyypityksistä. Monesti ottaa mallia muiden toteutuksista, joita on ollut tekemässä kehittäjät, keillä kokemusta typescriptillä kodaamisesta on vuosia. TS osaaminen kehittyy pikkuhiljaa ja virhe kerrallaan. Voisi toki olla hyödyllistä käydä ihan joku kurssikin aiheesta.

API Gatewayn REST API kehitykseen löytyy ainakin jonkin verran dokumentaatiota, mikä olisi hyvä käydä läpi. Uskon, että sitä kautta löytyy hyviä toimintamalleja ja tulee termistö tutummaksi. Varmasti tekisi hyvää lukea REST API:STA yleisemminkin, jotta muodostuisi parempi käsitys hyvistä käytänteistä.

Tällä viikolla jäi ratkaisematta se, miten tarkastusprosessiin lisätään tarkastuksia SQS jonon kautta, siihen täytyy perehtyä enemmän seuraavilla viikoilla. Myös AWS projektiin liittyen jäi paljon

selvitettäviä asioita resurssien muodostuksen ja erilaisten komentojen suhteen. Rajapinnan kehitys tulee jatkumaan ainakin ensi viikon ja mahdollisesti pidempäänkin.

Jos voisin kelata takaisin viikon alkuun niin käyttäisin viikon aikana enemmän aikaa netistä löytyvän teknisen dokumentaation lukemiseen. Samoin kun tulee jokin virhe vastaan, niin siihen perehtymiseen voisi käyttää hieman enemmän aikaa, vaikka ratkaisu olisikin löytynyt. Välillä saattaa myös jäädä itsekseen pohtimaan asioita turhan pitkään, kun pitäisi rohkeammin kysellä muiden näkemyksiä ja mielipiteitä.

3.2 Seurantaviikko 2

Maanantai 29.04.2024 tehtävät ja tavoitteet

Päivän tehtävät:

- Sprinttidemoon valmistautuminen
- Rajapinnan kehittäminen
- Palaverit:
- Sprinttidemo
- Sprinttiretro
- Viikkopalaveri
- Sprinttisuunnittelu

Sprintti eli kahden viikon kehitysjakso päättyy, joten päivä on hyvin palaverivoittoinen, eikä kehitykselle jää kauheasti aikaa. Aamulla on tavoitteena valmistautua sprinttidemoon, jotta asiakkaalle jää selkeä kuva tehdystä toteutuksesta esitykseni jälkeen. Minun esiteltävä toteutukseni oli puoliksi käyttöliittymäpuolta ja puoliksi niin sanottua back endiä, eli mitä taustaprosesseissa tapahtuu. Tarkoituksena tästä on esitellä käyttöliittymäpuolella toiminnallisuus. Palaverien väliin jää sieltä täältä aikaa puolesta tunnista pariin tuntiin, jonka aikana tavoitteena on edistää uuden rajapinnan toteutusta.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Ehdin aamusta valmistelemaan vanhan toteutuksen esityksen ja esittely sujui hyvin, eikä asiakkaalta noussut esiin mitään huomautettavaa tai kommentoitavaa. Viikkopalaveri oli peruttu, joten se vapautti kehitykselle tunnin verran enemmän aikaa. Yksi projektin kehittäjistä oli jäämässä lomille, eikä hän viitsinyt aloittaa tässä kohtaa mitään uutta tikettiä, joten ehdimme käydä läpi rajapinnan cdk stackin. Kävimme pala palalta läpi tiedoston rakenteen, joka opetti itselle todella paljon siitä, mitä resursseja tällainen rajapinta vaatii AWS:n puolella.

Päivän aikana opin, että rajapinnalle kannatta cdk stackissä muodostaa oma lokiryhmä ja nimetä se selkeästi, sillä muuten AWS nimeää sen hyvin sekavasti ja jatkossa on vaikea löytää mikä lokiryhmä liittyi tämän rajapinnan lokeihin. Otin mallia toisesta rajapinnasta, miten kannattaa tehdä ip osoitteiden salliminen, eli mistä osoitteista rajapintaa voidaan kutsua. Ymmärrys verkko-osoitteista kasvoi, sillä kävimme läpi, miten AWS:n Route 53 palvelun verkkotunnukset linkittyvät API Gatewayn kautta.

Tiistai 30.04.2024 tehtävät ja tavoitteet

Tehtävät:

- Palaverit: Daily
- Ratkaista tietokantahakuun liittyvä ongelma
- Käydä työkaverin kanssa läpi missä hän voisi olla avuksi

Tavoitteena on käydä nykyinen tilanne läpi työkaverin kanssa ja katsoa, miten saisimme kehitettyä rajapintaa yhdessä mahdollisimman tehokkaasti. Täytyy selvittää, miksi yksi tietokantahaku ei toimi. Selvitystä ehkä hankaloittaa se, ettei tule mitään virheilmoitusta.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Saimme työkaverin kanssa käytyä läpi rajapinnan toiminnan ja päädyimme, että hän suunnittelee, miten haastavampi tietokantapäivitys tehdään, jonka avulla saadaan lisättyä useampia tarkastuksia tietokantaan yhdellä kertaa. Toteutuksessa tullaan käyttämään transaktiota, jolla joko kaikki päivitykset tietokantaan toteutuvat kerralla tai sitten mikään niistä ei mene läpi. Sain myös selvitettyä, että yhden tietokantahaun ongelmat liittyivät `forEach()` -metodilla silmukointiin ja asynkronisiin funktioihin joita olin käyttänyt. Osaaminen tuntui kehittyvän heikosti, sillä tuntui että pyörin samojen ongelmien kanssa ympyrää, vaikka koitin dokumentaation kautta löytää ratkaisuja. Täytyisi paremmin perehtyä asynkronisten funktioiden toimintaan ja siihen, miksi ESLint laajennus VScodessa herjaa erilaisten silmukointien käytöstä liian raskaina toimenpiteinä.

Keskiviikko 01.05.2024

Vapaapäivä kun oli vappupäivä.

Torstai 02.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Palaverit: Daily
- Rajapinnan kehitystä
- Käydä läpi työkaverin tekemä funktio

- Täytyy palata versioon, jossa tietokantapäivitys toimi ja jatkaa siitä

Päivän tavoitteena olisi saada rajapinta siihen pisteeseen, että kun se vastaanottaa listan objekteja, niin niiden tietojen avulla saataisiin muodostettua tietokantaan aina tarkastusrivi per objekti.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Pääsin siihen pisteeseen, että yhden tarkastuksen muodostaminen tietokantaan onnistuu. Sain osittain rakennettua tai valmisteltua useamman tarkastuksen muodostamista. Suuri osa päivästä meni transaktioiden, async await funktioiden, typescriptin tyyppityksien ja interfacejen, sekä map() metodin ja silmukointien (loops) harjoitteluun ja paranteluun. Sain kokeneemmilta kollegoilta paljon vinkkejä liittyen odotustilassa oleviin lupauksiin ja silmukointien välttämiseen map() -metodin kautta.

Perjantai 03.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Daily
- Tyyppitysongelmien ratkaiseminen
- Rajapinnan logiikan muuntaminen usean tarkastuksen muodostamiseksi

Tavoitteena olisi saada edistettyä rajapintaa siihen pisteeseen, että POST kutsun tekemällä muodostuu useita tarkastuksia tietokantaan. Tämä vaatii paljon rakenteellisia muutoksia.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Tavoitteessa onnistuttiin, eli sain rajapinnan muodostamaan useita tarkastuksia. Kävin päivän aikana ratkaisuja etsiessä paljon dokumentaatiota läpi liittyen asynkroniseen kehittämiseen ja miten lupauksien ratkaiseminen / hylkääminen (promise resolve/reject) toimii. Jouduin paljon selvittämään typescriptin map toiminnallisuutta ja funktiorakenteita yleisesti ottaen. Kävin myös läpi ja koekelin varmaan jokaista mahdollista for silmukointia ratkaistessa ongelmia. Koen että typescript osaaminen kehittyi huomattavasti tämän päivän aikana, koska harvoin tulee näin paljon luettua dokumentaatiota ja koodiesimerkkejä.

Viikkoanalyysi

Työviikko oli päivän lyhyempi, mutta tuntui silti raskaammalta kuin normaali viikko. Itselle jäi ehkä hieman tunne, ettei saanut kauheasti aikaiseksi. Toisaalta kun miettii niin varsinaisia päiviä, jolloin kehitystyölle oli kunnolla aikaa, olivat tiistai, torstai ja perjantai, koska maanantaina oli niin paljon

kokouksia. Ja jos tästä valosta katsotaan, niin viikko sujui oikeastaan kohtuu tehokkaasti, sillä sain rajapinnan pohjarakenteet valmiiksi, sekä merkittävän osan toiminnallisuudesta toimimaan.

Tämän viikon ongelmina olivat pääpiirteittäin katsottuna:

- CDK stack rakenne
- Rajapinnan toiminnallisuuden kehittäminen
- Ensin yhden tarkastuksen muodostaminen
- Logiikan muuntaminen niin, että saadaan useita tarkastuksia muodostettua

CDK stackin rakenne saatiin korjattua, kun sain apua toiselta kehittäjältä, joka on ollut paljon tekemisissä erilaisten AWS palveluiden ja niiden pystyttämisen kanssa. Luultavasti olisin saanut tehtyä tuon ilman apuakin, mutta se on välillä ajallisesti viisaampaa käyttää muiden tietotaitoa hyväksi. Varsinkin, kun projektin etenemisellä on hieman kiire ja olemme aikataulusta jäljessä. Rajapinnan olisi voinut ottaa käyttöön AWS:n puolella ja sitä kautta selvittää puuttuvat tai virheelliset resurssit. Tällä hetkellä koitan saada kehitystyötä tehtyä mahdollisimman paljon paikallisesti, jottei muodostu ylimääräisiä kustannuksia projektille AWS:n palveluista.

Rajapinnan toiminnallisuuden kehittämisessä käytin ensin yhtä JSON muotoista objektia, jonka se ottaa vastaan POST kutsun rungossa (body). Tämä oli hyvä tapa saada perusrakenteet muodostettua ja jonka kautta oli helpompi hahmottaa, että miten objektilistan käsittely tulisi toimimaan. Jotta objektilistaa sai käsiteltyä, täytyi tehdä perusrunkoon paljon erilaisia listoja ja niiden silmukointia (looping).

Päivittäisessä tekemisessä kohdatut ongelmat:

- Työkaverin apufunktion toiminnallisuuden ymmärtäminen
- Minkä tyyppistä dataa funktio ottaa vastaan
- Minkä tyyppisen tietokantaoperaation se muodostaa
- Typescript ohjelmointikieleen liittyvät ongelmat
- Tyypitykset
- Objektilistojen silmukointi (looping)
- Asynkroniset tietokantaoperaatiot

Viikon alussa tosiaan sain hieman apuja, ja työkaveri ehti tekemään avukseni yhden funktion, jolla saadaan useampia tarkastuksia muodostettua tietokantaoperaatiossa. Tämän funktion toiminnallisuuden ymmärtäminen oli pieni haaste, sillä en aluksi pystynyt kunnolla kokeilemaan sitä yhdellä objektilla eli ilman objektilistaa. Ensin täytyi tehdä ja muokata koodin alkupään logiikkaa kuntoon ja kun sain sen tehtyä, niin pienellä muokkauksella sain myös tietokantaoperaation kuntoon.

Typescriptin kanssa tuli paljon erilaisia pieniä ongelmia, joita viikon aikana joutui selvittämään. Käsiteltävänä oli erilaisia listoja ja objekteja, joilla kaikilla oli omat tyyppinsä, joten jos jossain kohdassa oli asetettu tyyppi väärin, ei mikään oikeastaan toiminut. Nämä ongelmat kuitenkin korjautuivat, kun vaan luki virheilmoituksen ja kävi huolellisesti kaikki tyypitykset läpi.

Asynkronisissa funktioissa tuli vastaan ongelmia listojen läpikäymisissä ja tietokantaoperaatioissa. ESLint, eli aputyökalu, jolla nähdään nopeasti erilaiset virheet koodissa, antoi erilaisia virheilmoituksia, jotka hieman hämmensivät, sillä olin kirjoittanut toimivaa koodia. Osan ESLintin virheilmoituksista sai onneksi tavallaan piilotettua tai otettua pois käytöstä joissain osissa koodia. Voi olla, että tekemäni ratkaisut koodissa ovat hyvien käytänteiden vastaisia tällä hetkellä ja täytyy niitä käydä vielä läpi ensi viikolla.

Jatkoa ajatellen tulisi perehtyä enemmän asynkronisiin funktioihin ja niiden hyviin käytänteisiin.

Mielestäni (Logrocket) on hyvin kuvattuna perusteet. Omat muistiinpanoni aiheesta:

- Asynkroninen funktio palauttaa aina lupauksen (promise)
- Kun lupaus saadaan täytettyä, seuraava tapahtuma voidaan suorittaa
- Lupaus voidaan ratkaista (resolve) tai hylätä (reject)
- Jos lupausta ei ole ratkaistu tai hylätty, se on odotustilassa (pending)
- Asynkronisia funktioita voidaan suorittaa sarjana käyttämällä `”.then(”` metodia
- `Await` voidaan tehdä vain asynkronisen funktion sisällä

Hyvä esimerkki funktiosta, jonka teoretia itselleni tuli vastaan. Eli lupauksen kanssa täytyy muistaa määritellä myös virheen tyyppi, eikä vain onnistuneessa tapauksessa palautettavan muuttujan tyyppiä. Kuvassa 3 on esimerkki virheenkäsittelystä asynkronisessa funktiossa, jossa on määritely, että lupaus on joko lista tai merkkijono, mikäli tapahtuu virhe.

```
const fetchEmployees = async (): Promise<Array<Employee> | string> => {
  const api = 'http://dummy.restapiexample.com/api/v1/employees'
  try {
    const response = await fetch(api)
    const { data } = await response.json()
    return data
  } catch (error) {
    if (error) {
      return error.message
    }
  }
}
```

Kuva 3 Esimerkki virheenkäsittelystä asynkronisessa funktiossa

`”Promise.all(”` -metodia voidaan käyttää, kun halutaan ajaa useita asynkronisia funktioita rinnakkain. Metodi palauttaa yhden lupauksen ja täyttyy, kun kaikki rinnakkain tapahtuvat lapaukset ovat

täyttyneet ja hylkää lupauksen, mikäli jokin rinnakkain tapahtuvista lupauksista ei täyttynyt (Mozilla). Kuvassa 4 on esimerkki Promise all -metodin käytöstä.

```
const p1 = Promise.resolve(3);
const p2 = 1337;
const p3 = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve("foo");
  }, 100);
});

Promise.all([p1, p2, p3]).then((values) => {
  console.log(values); // [3, 1337, "foo"]
});
```

Kuva 4 Esimerkki "Promise.all()" -metodin käytöstä javascriptissä

3.3 Seurantaviikko 3

Maanantai 06.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Palaverit
- Daily
- Viikkopalaveri
- Kevään toteutusjakson läpikäynti
- Tarkastaa ja mahdollisesti tiivistää rajapinnan koodia
- Aloittaa tarkastuksien vieminen SQS jonoon

Päivän tavoitteena on käydä läpi rajapinnan koodia projektilaisten kanssa ja mahdollisesti parantaa toimivuutta, esimerkiksi nopeuttamalla kyselyn vastauksen saamista. Rajapinnassa muodostuvien tarkastusten viemistä tarkastusprosessin AWS SQS jonoon täytyisi tarkastella ja lähteä toteuttamaan.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Palaverit veivät pienen siivun päivästä, mutta sain poistettua koodista silmukoita ja korvattua niiden toiminnallisuutta kartoituksella eli käyttämällä lista.map() -metodia. Kartoitustaidot kehittyvät pikkuhiljaa, mutta täytyy jatkaa niiden harjoittelemista. Sain myös tutustuttua enemmän SQS jonon toiminnallisuuteen.

Tiistai 07.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Palaverit
- Daily
- Tiimin neljännesvuosisuunnittelupäivät
- Tikettien eli työtehtävien valmistelua
- Jatketaan rajapinnan kehitystä
- Tarkastusten lähettäminen tarkastusprosessin SQS jonoon

Tiimin 1 neljännesvuosisuunnittelupäivät, joten on toimistopäivä pitkästä aikaa (Itse kuulun tiimiin 2). Päivän tavoitteena olisi saada tehtyä tarkastusten lähettäminen tarkastusprosessin SQS jonoon ja testattua sen toimivuus.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Tikettien valmistelua ei ollut. Rajapinnan kehitys eteni hyvin ja kokonaisuus alkaa muodostua. Tarkastuksien lähettäminen tarkastusprosessiin toimii nyt ainakin osittain, mutta tarkastusprosessin koodia täytyy vielä korjata, koska se meni virhetilaan tietyissä testitilanteissa. Päivän aikana tuli hyvin kokemusta AWS:n SQS jonoista ja tilakoneista, sekä niiden virhetilanteiden selvityksestä.

Keskiviikko 08.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Daily
- Käydä läpi rajapintatiketin hyväksymiskriteerit
- Selvittää miten asiakirjojen rajaaminen 100 kpl/kutsu kannattaa tehdä
- Käytetäänkö JSON skeemaa validoinnissa

Päivän tavoitteena olisi käydä läpi rajapintatiketin hyväksymiskriteerit ja varmistaa, että toiminnallisuus on tehty niin kuin tiketissä on kuvattu. Täytyy selvittää mikä on kannattavin tapa tehdä rajapintaan tulevien kutsujen validointia. Tässä täytyy selvittää, miten tehdään rajapinnan rungossa tulevan objektilistan koon rajaaminen sataan objektiin ja miten tarkistetaan, että objekteissa oleva data on oikean muotoista rajapinnan käsiteltäväksi. Kaikki validointi on todennäköisesti tehtävissä JSON skeeman avulla.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Päivän aikana varmistin, että asiakkaan ei tarvitse käyttää rajapinta-avainta rajapintakutsuissa. Alkupäivästä tein yksinkertaisen tarkistuksen objektilistan pituudelle. Jos objektilista on pidempi kuin 100 niin rajapinta palauttaa "413 content too large" -virheviestin käyttäjälle. Tämän kanssa tuli itselle vähän haasteita, sillä sain paikallisessa testauksessa 400 invalid input virhettä aikaiseksi. Tämä ongelma kuitenkin johtui lopulta siitä, että olin jättänyt sql muuttujan määrittelemättömäksi, joten koodin lopussa koodi koitti sulkea määrittelemättömän tietokantayhteyden. Tämä opetti itseäni hyvin ongelmanratkaisutilanteissa, joissa virheviesti tai viestit voivat olla hyvinkin epämääräisiä. Loppupäivästä päädyin validoimaan rajapintaan saapuvan datan JSON skeemalla. Skeeman mukaiseen validointiin käytin typescriptille sopivaa AJV JSON skeema -validaattoria ja tämän käyttö vaati hieman opettelua, mutta sain sen lopulta toimimaan. Päivä kuluikin suurimmaksi osaksi skeemaa tehdessä ja testailla.

Torstai 09.05.2024

Helatorstai, joten vapaapäivä

Perjantai 10.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Daily
- Sprinttidemoon ja retroon valmistautuminen
- Rajapinnan käyttöönotto dev ympäristöön

Päivän tavoitteena on saada rajapintatiketti valmiiksi. Tämä vaatii vielä rajapinnan käyttöönoton AWS ympäristössä, eli täytyy tehdä käyttöönotto AWS:ssä kehitysympäristöön. Tässä käyttöönotossa todennäköisesti tulee joitain virheitä mitkä täytyy ratkaista ja sen jälkeen testata rajapinnan toimivuus kehitysympäristössä. Mikäli testauksessa kaikki vaikuttaa toimivan, niin sen jälkeen täytyy tehdä pyyntö kehityshaarojen yhdistämiselle AWS:n Codecommit versionhallintaan ja laittaa pyyntö muiden tiimiläisten katselmoitavaksi. Jos korjausehdotuksia tai parannettavaa ei löydetä, niin pidetään katselmointitilaisuus, jossa vielä esitellään ratkaisun toiminta.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Rajapinnan käyttöönotossa tuli vastaan vain pieni virhe liittyen domain nimeen, joka oli helppo ratkaista. Testauksessa huomasin, että rajapinta palauttaa sitä kutsuessa "400 invalid input" vastauksen, vaikka missään lokerissa ei ollut virheviestejä. Tähän selvisi, että rajapinnan vastauksen viesti täytyy olla muodoltaan JSON merkkijono, eikä perinteinen merkkijono. Näiden ongelmien ratkettua pääsi varsinaiseen testausteen kiinni. Testauksessa huomasin, että vaikka rajapinta palauttaa vas-

taukseksi ”200 OK” niin tarkastusten mentyä SQS jonon kautta tarkastusprosessiin, tarkastusprosessi epäonnistuu useissa tapauksissa. Syynä tähän oli virheelliset kuvionumeroiden tiedot tietokannan ja rajapintakutsun välillä. Tämä on ongelmallista, sillä käyttäjälle jää mielikuva, että kaikki onnistui hyvin, eikä saa heti tietoa, että jokin meni pieleen. Tähän täytyi kehittää ratkaisu, jossa rajapinta tarkistaa, että tietokannasta varmasti löytyy tietyt kuvionumerot, jotka käyttäjä on rajapintakutsussa antanut. Tavoite tiketin valmistumisesta ei toteutunut, mutta toiminnallisuus on nyt ainakin pääosin valmis.

Viikkoanalyysi

Koko viikko liittyi yhden ja saman rajapintatiketin tekemiseen niin kuin aikaisemminkin. Tällä viikolla keskittyminen oli toiminnallisuuden tekemisessä. Viikko oli päivän normaalia lyhyempi, mutta tiketti on nyt lähes valmis. Maanantaina 13.05.2024 loppuu nykyinen sprintti, joten ensi viikolla on vielä pieni mahdollisuus saada tiketti valmiiksi, hyväksytyksi ja sitä kautta asiakkaan testattavaksi.

Viikon aikana esiin tulleita ongelmia

- API Gatewayn ongelmat
- JSON sanoman muoto POST pyynnössä
- Responsen runko JSON merkkijono
- Tarkastusprosessin virheet
- Käyttäjä ei pysty syöttämään samalle asiakirjalle useampia tarkastuksia rajapinnan kautta

Päivittäin vastaan tuli useita pieniä ohjelmointiongelmia, jotka johtuivat lähinnä omasta kokemattomuudesta Typescriptin kanssa. Suurimpaan osaan näistä löytyi joko googlettamalla vastaukset tai hetken aikaa miettiessä virheen sanomaa. AWS:n API Gateway aiheutti hieman päänvaivaa, sillä siihen liittyvistä asioista minulla ei kauheasti ole kokemusta. Esimerkiksi kun oli tehnyt cdk stackin käyttöönoton AWS ympäristöön ja testasi rajapintaa AWS ympäristön kautta paikallisen kehitysympäristön sijaan, niin oli hankala selvittää virheitä, kun lokeilta ei saanut mitään virheviestiä. Tiimiläisen avustuksella selvitimme, että rajapinnan palauttaman vastauksen runko osio täytyy olla JSON merkkijono, jotta oikeanlainen vastaus muodostetaan. API Gateway hyväksyy vain tietyssä JSON muodossa olevan POST pyynnön sisällön, joka aiheutti ongelmia, kun koitin saada tehtyä JSON skeeman mukaista validointia rajapinnalle. VScodessa käyttämäni Thunder client laajennoksen mukaan pyyntöni oli JSON muotoista, mutta jostain syystä se ei kelvannut API Gatewaylle ja vaati pienen muutoksen. JSON skeemoihin voi ottaa mallia lähteestä (AWS apigateway data models).

Tarkastusprosessin virheet olivat virheitä, jotka tulivat, kun rajapinnan muodostamat tarkastukset lähetettiin SQS jono kautta tarkastusprosessiin, ja siellä tapahtui virhe, mistä johtui ettei tietokan-

taan muodostunut tarkastuksen tarkastettavia kuvioita. Yksi virhe johtui käyttäjän syöttämistä virheellisistä kuvionumeroista, jonka vuoksi täytyi rajapintaan tehdä tietokantaoperaatio, joka tarkisti, että käyttäjä on syöttänyt kuvionumerot oikein. Toinen virhe oli yksinkertaisempi, jossa kuvionumerot oli lähetetty tarkastusprosessiin numeromuotoisina, vaikka tarkastusprosessi käsitteli niitä merkkijonoina. Helppo korjaus, mutta ongelman selvittämiseen meni kuitenkin aikaa, sillä virheviesti ei ollut selkeimmästä päästä.

Perjantaina huomasin myös, että käyttäjällä voi olla käyttötapaus, jossa hän haluaa muodostaa samalle asiakirjalle useamman tarkastuksen, mikä ei tämänhetkisessä toteutuksessa onnistu. Laitoin tästä kysymyksen asiakkaalle, mutta en ehtinyt saamaan vielä vastausta. Vihjasin, että käyttäjä voi kyllä helposti tehdä peräkkäin pyyntöjä rajapintaan, joten ei välttämättä ole tarpeellista saada tehtyä yhdellä pyynnöllä. Tästä voi tulla vielä paljon muutoksia, mikäli asiakkaan mielestä nykyinen toiminnallisuus ei ole riittävä.

Haluan mainita, että tällä viikolla on erityisesti korostunut virheenkäsittelyn tärkeys ja olisi hyvä lisätä koodiin ennemmin liikaa virheenkäsittelyä kuin liian vähän. Testausta nopeuttaa huomattavasti se mitä nopeammin saadaan selvyys virheen syystä. Koodissa voi tulla tilanteita, jotka eivät varsinaisesti aiheuta virhettä suoritukselle, niin tämän kaltaisissa tapauksissa voi olla hyödyllistä itse muodostaa virhe `throw new Error('Jokin meni pieleen')` -tyyppisesti. Lähteestä (Clouddevs error handling) löytyy hyviä ohjeita typescriptin virheiden käsittelyyn, joista täytyy koittaa jatkossa ottaa enemmän mallia omassa koodissa.

3.4 Seurantaviikko 4

Maanantai 13.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Sprinttidemo ja retro
- Viikkopalaveri
- Sprinttisuunnittelu
- Tarkastukseen asettamisen rajapinnan suorituskyvyn testaus

Maanantain tavoitteena on saada tehtyä tarkastukseen asettamisen rajapinnan testausta ja kehitystä palaverien väliin jäävinä aikoina. Tarkoituksena olisi saada testattua rajapinnan nopeutta eri asiakirjojen määriillä. Katsotaan miten kauan 25/50/75/100 asiakirjan prosessoinnilla kestää, sillä AWS API Gateway:ssä tulee aikakatkaus 29 sekunnin kohdalla.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Viikkopalaveriä ei ollut. Rajapinta toimi hyvin sadallakin asiakirjalla, prosessointiin kului 14,5 sekuntia, joten sen osalta ei täytynyt tehdä muutoksia. Päivän aikana tarkentui myös yhden parametrin numero, jonka korjasin oikeaksi. Testauksissa tarkistin, että asiakirjan tila päivittyy oikein tietokannassa. Tein rajapinnan dokumentaation confluenceen, jossa otin mallia muiden rajapintojen dokumentaatioista. Rajapinnan testaukset veivät suurimman osan päivästä, sillä oli työlästä katsoa tietokannasta oikeantyyppiset testitapaukset ja muodostaa niistä POST pyyntöön JSON objekti.

Tiistai 14.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Daily
- Tarkastukseen asettamisen rajapinnan testausta AWS ympäristössä
- Katselmointitilaisuuteen valmistautuminen
- Katselmointitilaisuus

Päivän tavoitteena on saada rajapintatiketti valmiiksi. Tähän sisältyy vielä pientä testausta. Täytyy tehdä kehityshaaran yhdistämispyyntö ja pyytää tiimiä katselmoimaan koodi. Täytyy valmistautua esittelemään toteutus muille kehittäjille ja pitää katselmointitilaisuus.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Sain valmisteltua ja pidettyä katselmoinnin. Katselmoinnissa nousi esiin kysymys, että voidaanko tarkastuksia muodostaa tietokantaan, jos asiakirjalle ei ole tehty päätöstä. Selvitin tämän arkkitehdiltä ja vastaus oli kyllä. Tämä tarkoitti, että piti tehdä vielä pieni viilaus koodiin. Koodin osalta tiketti tuli valmiiksi ja ilmoitin asiakkaalle, että he voivat tehdä yhteystestin omasta konesalistaan. En kuitenkaan saanut asiakkaalta vielä vastausta.

Aloitin loppupäivästä uutta tikettiä, jossa tehdään toiseen rajapintaan, tiedonsiirtorajapintaan maastotarkastaja parametrin lisäys ja asiakirjan käsittelijän päivittäminen rajapintapyynnössä tulleeeksi maastotarkastajaksi. Tämän osalta ehdin vasta hieman tutustumaan koodiin ja selvittämään tikeillä epäselviksi jääneitä asioita.

Keskiviikko 15.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Daily
- Täytyy pyytää asiakasta testaamaan tarkastukseen asettamisen rajapinnan yhteydet koneisiin Pata2 koneesta
- Tarkastukseen asettamisen rajapinnan dokumentaation parantelu

- Tiedonsiirtorajapinnan tiketin edistämistä

Päivän tavoitteena olisi saada asiakas testaamaan yhteydet tarkastukseen asettamiseen rajapintaan ja mahdollisesti ratkoa siihen liittyviä ongelmia, mikäli niitä ilmenee. Täytyy myös käydä tiimiläisen kanssa tiedonsiirtorajapinnan tiketin toteutusta läpi, sillä rajapinta on melko monimutkainen.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Asiakkaan puolelta testattiin yhteys tarkastukseen asettamisen rajapintaan ja se toimi osittain, joten jouduin tekemään ip avauksia rajapintaan. En kuitenkaan saanut vielä vastausta, että auttoivatko avaukset. Kävin myös arkkitehdin kanssa läpi tämän rajapinnan toteutusta ja sitä miten helposti siihen saadaan tulevaisuudessa muokattua lisätoiminnallisuutta. Päivitin dokumentaatiota esiin nousseiden asioiden osalta. Sain aloitettua tiedonsiirtorajapintatiketin koodin kirjoitusta, mutta paikallinen testaus osoittautui hankalaksi, joten täytyy vielä ratkoa siihen liittyviä ongelmia.

Torstai 16.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Daily
- Täytyy mainita asiakkaalle rajapinnan yhteyksien testauksesta
- Täytyy selvittää miten tiedonsiirtorajapinnan paikallisen testauksen saa toimimaan

Tavoitteena on varmistaa asiakkaalta, että tarkastukseen asettamisen rajapinnan yhteydet toimivat niin kuin pitää. Lisäksi täytyisi saada tiedonsiirtorajapinnan paikallinen testaus toimimaan. Mikäli tämä osoittautuu liian hankalaksi, täytyy rajapintaa testata suoraan AWS:n kautta.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Sain asiakkaan testaamaan yhteydet tarkastukseen asettamisen rajapintaan ja huomasin, että olin katsonut ip osoitteet väärästä dokumentaatiosta, joten tein taas pienen korjauksen sen osalta. Nyt yhteyksien pitäisi toimia, mutta en saanut vielä varmistusta asiakkaan puolelta.

Tiedonsiirtorajapinnan paikallinen testaus eteni hyvin, kun huomasin, että muutosten jälkeen täytyi aina tehdä uusi koonti, eli rakentaa sovellus, jotta muutokset astuivat voimaan. Paikallisessa kehityksessä tekemäni muutokset toimivat oikein, joten julkaisin sen AWS puolelle testattavaksi. Huomasin, että AWS:n puolella tuli ongelma koodin tyyppityksien kanssa, joita olin muokannut paikallista kehitystä varten. AWS Cloudwatch lokeista huomasin, että lokitus ei toimi ja siitä tuli virhe, joten täytyi lisätä lokitusoikeuksia Lambda funktiolle.

Päivän aikana tuli kasvatettua osaamista AWS:n lokituksiin liittyen, sillä lokeja muodostetaan monessa eri paikassa ja välillä on haastavaa löytää oikea paikka, mistä lokeja tulisi lukea.

Perjantai 17.05.2024 tehtävät ja tavoitteet

Tehtävät:

- Daily
- Tarkastukseen asettamisen rajapinnan pata2 koneen yhteystestauksen selvitys
- Tiedonsiirtorajapintatiketin katselmointitilaisuuden järjestäminen
- Muiden kehittäjien katselmointitilaisuuksiin osallistumista

Itselläni on tällä hetkellä kaksi tikettiä kesken, joten tämän päivän tavoitteena olisi saada ne valmiiksi. Ensimmäinen tiketti vaatii sen, että asiakas saa testattua ja todettua tarkastukseen asettamisen rajapinnan toimivaksi. Toisen tiketin osalta täytyy pitää katselmointitilaisuus ja tehdä sen osalta korjauksia, mikäli muilta kehittäjiltä tulee jotain huomioita. Täytyy myös osallistua muiden kehittäjien katselmointitilaisuuksiin, tarkistaa heidän koodiaan, hyväksyä kehityshaarojen yhdistämispyyntöjä ja hyväksyä tiketin hyväksymiskriteereitä.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

En tavoittanut asiakkaan testaajaa, joten tarkastukseen asettamisen rajapinnan tiketti jäi vielä kesken. Tiedonsiirtorajapinnan tiketistä pidin katselmointitilaisuuden, jossa löytyi pari pientä korjattavaa asiaa liittyen koodissa suoritettavien tietokantaoperaatioiden järjestykseen. Tein korjaukset, sain hyväksynnät kehityshaaran yhdistämispyyntöön ja tiketin hyväksymiskriteereihin, joten tiketti tuli valmiiksi. Loput päivästä kului koodikatselmoiteihin ja katselmointitilaisuuksiin, jotta saatiin muiden kehittäjien töitä valmiiksi. Loppupäivästä tuli vielä huomio yhdestä automaattitarkastuksen virheestä sovelluksen puolella, jonka selvitystyötä ehdin hieman aloitella.

Viikkoanalyysi

Viikko kului pitkälti kahden eri tiketin tekemisen parissa ja molemmat liittyivät rajapintojen kehitykseen. Alkuviikko kului tarkastukseen asettamisen rajapinnan viimeistelyssä ja loppuviikko tiedonsiirtorajapintatiketin tekemisessä, sekä tarkastukseen asettamisen rajapinnan yhteysongelmien selvityksessä. Perjantaina loppupäivästä tuli vielä yksi automaattitarkastuksen korjaustiketti, jota ehdin vasta pintapuolisesti aloittamaan.

Viikon aikana esiin tulleita ongelmia ja selvitettäviä asioita:

- Kysymys tarkastukseen asettamisen rajapinnan toiminnasta
- Voidaanko tarkastuksia muodostaa tietokantaan, jos asiakirjalle ei ole tehty päätöstä

- Tarkastukseen asettamisen rajapinnan yhteyksien hidas testaus
- Toisen kehittäjän tekemään tiedonsiirtorajapintaan perehtyminen
- Tiedonsiirtorajapinnan paikallinen kehitys ja testaus
- Tiedonsiirtorajapinnan toiminnon lokitus toimintalokiin
- Automaattitarkastuksen korjaus

Tarkastukseen asettamisen rajapinnan katselmointitilaisuudessa nousi esille toisen kehittäjän toimesta, että voidaanko tarkastuksia muodostaa, jos asiakirjalle ei ole tehty päätöstä. En löytänyt tähän vastausta tiketiltä tai muualta dokumentaatiosta, joten varmistin asian arkkitehdiltä. Oletukseni oli, että niin ei voida tehdä, mutta arkkitehdin vastaus oli, että niin voidaan tehdä, jonka vuoksi täytyi tehdä pieni loogisuusmuokkaus koodiin. Jotta tiketti saataisiin valmiiksi, pitäisi hyväksymiskriteerin ”Yhteys rajapintaan toimii asiakkaan konesalista” täyttyä. Tämän testaaminen osoittautui hitaaksi, sillä en saanut asiakkaan testaajalta montaa vastausta viikon aikana ja tiketin valmistuminen siirtyi seuraavalle viikolle. Tämä tiketti tulisi saada valmiiksi tämän sprintin aikana, jotta yksi kokonainen feature tulee valmiiksi ja pääsee testaukseen. Uskon, että seuraavalla viikolla tämä valmistuu ennen sprintin loppua, joka on käytännössä perjantaina.

Tiedonsiirtorajapintaan liittyvä tiketti aiheutti myös paljon ongelmia. Ensin täytyi tutustua kohtuuisen ja monimutkaisen rajapinnan toimintaan. Sain onneksi apua rajapinnan tehneeltä kehittäjältä ja dokumentaatio oli tehty selkeäksi. Tiketissä tehtävä asia oli kohtuuisen pieni ja helppo toteuttaa, mutta koodin testaaminen aiheutti ongelmia. Toinen kehittäjä oli testannut tekemiään asioita suoraan AWS:n kautta, joten paikallisen kehityksen eteen ei ollut tehty juuri mitään. Tässä täytyi hieman pyytää apua ja ottaa mallia yhden toisen rajapinnan paikallisesta kehityksestä. Lopulta parin pienen asian ymmärtäminen ratkaisi ongelmat ja toiminnallisuuden testaus ja toiminta tulivat testattua toimiviksi. Hyväksymiskriteereissä oli mainittu, että asiakirjan käsittelijän päivitys maastotarkastajaksi täytyy lokittaa toimintalokiin. Tämä ei aluksi toiminut ja vaati hieman selvittelyä muiden lokien kautta, että mistä ongelma johtui. Lopulta selvisi, että AWS:n lambda funktiolta puuttui oikeudet päivittää toimintalokia. Olen aika vähän joutunut tekemään töitä oikeuksien antamisien kanssa, mutta siihen onneksi löytyi hyvä esimerkki toisesta tietovarastosta ja homma ratkesi.

Perjantaina ehdin hieman selvittämään erään automaattitarkastus lambda:n virheilmoituksen syytä. Vika johtui siitä, että tietokantaoperaatiossa palautuu tyhjä lista, jonka koodissa oletetaan sisältävän erilaisia attribuutteja. Tähän en ehtinyt miettimään korjausta.

Tällä viikolla ei kauheasti tullut Typescript haasteita vastaan, mikä tuntui ihan positiiviselta. AWS:n palveluihin ja niiden lokituksiin tutustuminen oli mielekästä tekemistä, jota ei niin paljon aikaisemmin ole päässyt tekemään. Itseni on ehkä vähän yllättänyt, että miten vähän työtehtävissä on tarvinnut osata ja selvittää AWS:n pilvipalveluihin liittyviä asioita.

Kertasin tällä viikolla hieman Amazonin pilvipalveluiden käytäntöjä ja käyttöoikeuksia. Käytännöt (policyt) määrittävät käyttöoikeuksia ja niitä voidaan liittää käyttäjille, käyttäjäryhmille, rooleille ja resursseille. Käytännöt ovat muodoltaan JSON objekteja, jotka voivat antaa luvan esimerkiksi tehdä poisto tai päivitysoperaatioita tietokantaan. AWS tukee kuutta käytäntötyyppiä, identiteettiin perustuvat käytännöt, resurssipohjaiset käytännöt, käyttöoikeusrajat, organisaation SCP:t (Service Control Policy) eli palvelun valvontakäytäntö, ACL:t (Access Control List) eli S3 ämpärin kulunvalvontaluettelo ja istuntokäytännöt. Perusteista löytyy lisää lähteestä (Amazon g.).

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "1",
    "Effect": "Allow",
    "Principal": {"AWS": ["arn:aws:iam::account-id:root"]},
    "Action": "s3:*",
    "Resource": [
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
    ]
  }]
}
```

Kuva 5 Esimerkki resurssipohjaisesta käytännöstä

Kuvassa 5. on esimerkki resurssipohjaisesta käytännöstä, joka on JSON objekti. Käytäntö määrittellee, että tietyn AWS-tilin (account-id) jäsenet voivat suorittaa mitä tahansa Amazon S3 -toimintoja ryhmässä, jonka nimi on "mybucket". Se sallii minkä tahansa toiminnon, joka voidaan suorittaa S3 ämpärille tai sen sisällä oleville esineille. Tähti kohdassa s3:* kuvaa kaikkia toimintoja ja mybucket/* kohdan kauttaviiva + tähti kuvaa kaikkia ämpärin sisällä olevia objekteja.

3.5 Seurantaviikko 5

Maanantai 20.05.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Viikkopalaveri

- Automaattitarkastuksen korjaus
- Tarkastukseen asettamisen rajapinnan yhteyksien selvitys
- Katselmointi ja katselmointitilaisuuteen osallistuminen
- Selvitellä uuden tiketin aloittamista

Päivän tavoitteena olisi saada korjattua virheellisesti toimiva automaattitarkastus, tämä jäi hieman kesken viime viikon perjantaina. Täytyy pyytää asiakasta testaamaan konesalistaan yhteys tarkastukseen asettamisen rajapintaan. Aika hajanaista pientä tekemistä pitkin päivää ja täytyisi selvittää minkä tiketin voi ottaa työn alle seuraavaksi.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Sain korjattua automaattitarkastuksen, se veikin yllättävän paljon aikaa päivästä, sillä alkuperäisen toteutustiketin määrittelyt olivat puutteelliset ja täytyi tehdä paljon selvitystyötä. Itse ratkaisu oli yksinkertainen ja vaati koodiin tarkistuksen, että jos tietokantakyselyn palauttama lista oli tyhjä, niin automaattitarkastusta ei tarvinnut suorittaa.

Asiakas sai testattua yhteydet rajapintaan ja sai vastaukseksi "can't connect securely to this page. This might be because the site uses outdated or unsafe TLS security settings". Epäilykseni oli alkuun, että asiakkaan konesalin koneissa ei ole käytössä internet asetuksista TLS 1.2 salausprotokolla, jota AWS:n API Gateway käyttää. Tämä vaatii vielä selvittelyä ja lisää yhteystestauksia tällä viikolla.

Tiistai 21.05.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Kehittäjien kipupisteet palaveri
- SAFe kertaus koulutus
- Rajapinnan TLS ongelman selvitys

Palaverit vievät ison osan päivästä, mutta täytyy pyrkiä selvittämään asiakkaan saama TLS virheviesti ja ratkaista yhteysongelma.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Päivän aikana tuli käytyä läpi vähän sekalaisesti kaikkea liittyen verkkoyhteyksiin ja niiden yleisiin ongelmiin. Selvitin esimerkiksi, miten komentokehötteen curl komennolla saisi tehtyä oikeantyyppisen POST pyynnön rajapintaan, jotta saadaan rajattua mahdolliset selaimen liittyvät ongelmat pois. Curl on tietokoneohjelmistoprojekti, joka tarjoaa komentorivityökalun tietojen siirtämiseen

eri verkkoprotokollien avulla ja nimi tulee sanoista "Client for URL". Curlilla asiakas sai virheviestiksi "curl: (35) Recv failure: Connection was reset", mikä viittaisi siihen, että nyt jokin palomuuuri tai vastaava estää yhteyden. Kun pyysin asiakasta lisäämään curl komentoon -v (verbose) vaihtoehtoa, niin siitä saadun virheen mukaan epäonnistumisen syynä on "schannel: failed to receive handshake, SSL/TLS connection failed". Tämä ei lähtökohtaisesti vielä auttanut kauheasti, mutta on helpompi selvittää ongelmaa, kun saa vähän erilaisia ja selkeämpiä virheviestejä käsiteltäväksi.

Keskiviikko 22.05.2024 tehtävät ja tavoitteet

Tehtävät

- Suunnittelupäivä
- Rajapinnan yhteysongelmien selvitystä

Koko päivä menee toimistolla seuraavan toteutusjakson suunnittelun merkeissä ja tavoitteena on siinä sivussa koittaa selvittää asiakkaan yhteysongelmia tarkastukseen asettamisen rajapintaan liittyen.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Suunnitelma tuli valmiiksi ja vaikutti toteutuskelpoiselta. Yhteysongelmat eivät vielä ratkenneet, sillä asiakas ehti tehdä testauksia vasta aivan päivän lopussa. Sain kuitenkin katsottua ja rajattua AWS:n rajapinnan pääsylokeista (access log), että mistä osoitteista asiakas koittaa yhteyttä muodostaa. Ongelman ratkaisu jatkuu seuraavana päivänä ja täytyy varmaan kysellä apuja muilta kehittäjiltä.

Torstai 23.05.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Story valmistelu
- Yksikköpalaveri
- Rajapinnan yhteysongelmien selvitys
- Selvittää lisää tekemistä

Tavoitteena saada selvitettyä rajapinnan yhteysongelmat ja kysellä scrum masterilta seuraavaa tikkettä.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Päivän aikana selvisi, että tietohallinnon täytyi tehdä yhteysavaus AWS projektin kehitystilin ja asiakkaan konesalin välille. Samalla tehtiin avaukset muihinkin tarvittaviin ympäristöihin. Yhteys siis kulkee ensin asiakkaan konesalista välityspalvelimelle, josta se ohjautuu AWS:n API Gatewayn kautta rajapintaan. Rajapinnan asetuksiin täytyi tehdä avaus välityspalvelimen osoitteelle, jotta asiakkaan testaus onnistui. Yhteysongelman ratkettua loppupäivä kului tikettien valmisteluiden ja katselemointien parissa. Käytiin myös tiimin kanssa läpi kehitysehdotuksia, jotta jatkossa tikettien sisältö olisi laadukkaampaa ja helpommin ymmärrettävää kehittäjille.

Perjantai 24.05.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Tikettien läpikäyntiä
- Uuteen tikettiin tutustumista

Tavoitteena on läpikäydä uusia tulevia tikettejä, jotta ne ovat selkeitä ja sisällöltään riittäviä. Olisi myös hyvä saada selville, minkä tiketin seuraavaksi otan työn alle. Päivä on normaalia lyhyempi, sillä iltapäivästä alkaa työnantajan järjestämät kesäjuhlat.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Aamupalaverissa nousi asiakkaan testaajalta huomio rajapinnan yhteyksiin liittyen. Täytyy rajata pois yhteydet asiakkaan toimistoverkosta rajapintaan. Sain rajapintatiketin viimein valmiiksi. Muuten päivä kului tikettien läpikäynneissä ja hieman seuraaviin tehtäviin tutustuessa, sillä varsinaista kehitystyötä ei saanut aloittaa, ennen kuin asiakas on kuitannut tiketit valmiiksi.

Viikkoanalyysi

Kulunut viikko oli hieman normaalista poikkeava kehitystyön osalta. Oli normaalia enemmän palaverieita ja koulutuksia, sekä suunnittelupäiviä, jotka veivät aikaa varsinaiselta kehitystyöltä. Muutenkin tuntui, että tuli tehtyä paljon kaikkea pientä sekalaista tehtävää ja osaamisen kehittäminen jäi vähäisemmälle tasolle.

Viikon aikana esiin tulleita ongelmia ja selvitettäviä asioita:

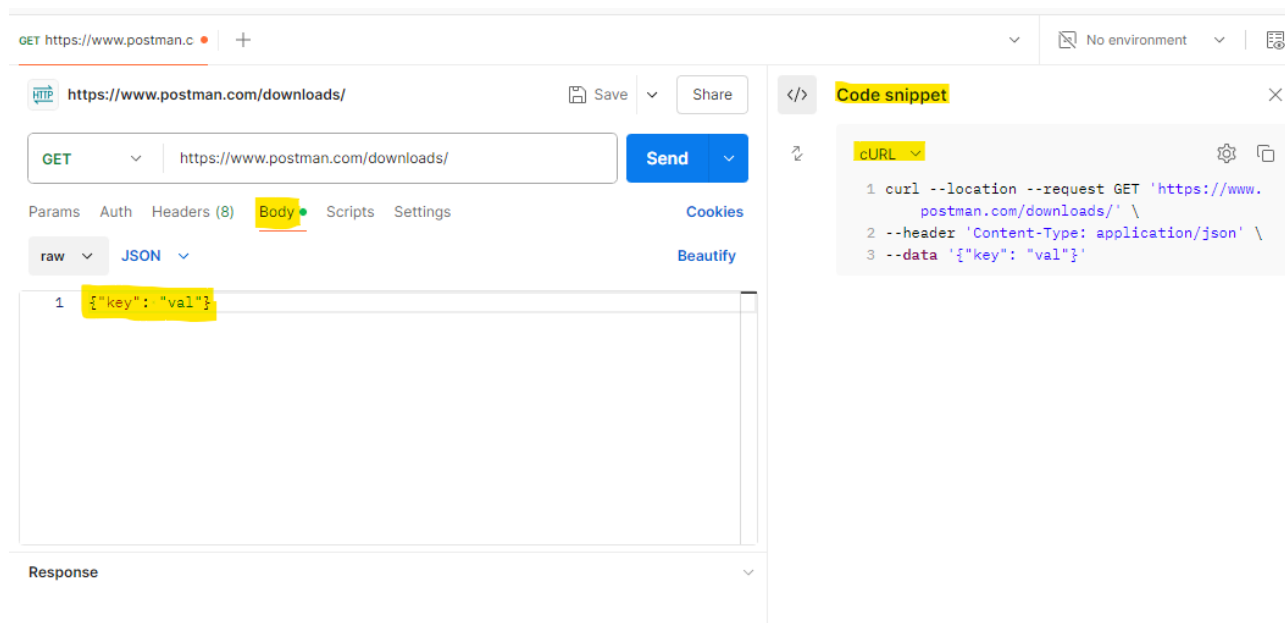
- Automaattitarkastuksen korjaus
- Asiakkaan konesalin yhteydet tarkastukseen asettamisen rajapintaan
- Curl komennot

Automattitarkastuksen korjaus oli ns. bugikorjaus, joka vei puutteellisen määrittelyn vuoksi turhan paljon aikaa. Tältä voisi välttyä sillä, kun alkuperäiselle tiketille merkittäisiin selvästi esimerkiksi

kommentteihin, että tiketille on tehty myös korjaustiketti jossain vaiheessa. Korjaustiketille voisi myös kuvata tarkemmin, että miksi siinä oleva muutos tehdään.

Asiakkaan konesalin yhteydet tarkastukseen asettamisen rajapintaan osoittautuikin yllättävän haastavaksi ongelmaksi. Aluksi ajattelin, että tuo vaatii vain oikean IP-osoitteen löytämisen ja sen mukaisen avauksen. Ongelma ratkesikin puoliksi sillä, mutta ennen sitä tuli vielä tietohallinnon tehdä sellaisia yhteysavauksia, joita kehittäjät eivät edes pääse tekemään. Tässä oli myös haasteena asiakkaan testaajan kiireellisyys, sillä aina yhden korjauksen tai testauksen testaamisessa saattoi kestää päivä tai pari, joten sain vastauksia liian hitaasti. Jatkossa olisi parempi selvittää tämänkaltaiset ongelmat varaamalla selkeä aikaikkuna tietyille kehittäjille ja asiakkaan työntekijöille, jonka aikana kaikki paneutuvat ongelman ratkaisemiseen.

Aikaisemmin olen kuullut muiden paljon puhuvan ”kurlaamisesta” eli curl komentojen käytöstä, mutta itseltä löytyy hyvin vähän kokemusta näistä. Yhteyksien ja niiden ongelmien selvityksessä komennot vaikuttavat olevan hyödyllisiä, joten koitin myös itse vähän tutustua niihin. Olen kehitystyössä tehnyt rajapintapyyntöjä ”Postman” nimisellä ohjelmalla ja löysin sieltä hyvän työkalun, jonka avulla voi millaisen tahansa pyynnön kääntää curl komennoksi.



Kuva 6 Esimerkki miten tehdä curl komento GET pyynnöstä

Sivustolta (PhoenixNAP) voi tutustua lisää erilaisiin curl komentojen lisävaihtoehtoihin, eli miten komentoon saadaan lisättyä erilaisia toimintoja. Itse käytin esimerkiksi -v vaihtoehtoa, joka tarkoittaa ”verbose”, eli monisanainen, joka kertoo komennon vastauksessa enemmän erilaisia vaiheita liittyen yhteyspyyntöön.

3.6 Seurantaviikko 6

Maanantai 27.05.2024 tehtävät ja tavoitteet

Tehtävät

- Sprinttidemoon valmistautuminen
- Sprinttidemo
- Sprinttiretro
- Sprinttisuunnittelu
- Uuden tiketin aloittelua

Palaverit vievät suurimman osan päivästä, mutta tavoitteena olisi päästä konkreettisesti aloittamaan uuden tiketin toteutusta. Tiketissä tehdään erilaisten työlajien tarkastusraporteille pohjat ja toteutetaan taustaprosessi, joka muodostaa tarkastusraportit, kun käyttäjä painaa nappia käyttöliittymässä.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Sprinttidemoon valmistautumisessa huomasin, että AWS:ssä oli väärä versio tiedonsiirtorajapinnasta, joten täytyi julkaista kehityshaarasta uusin versio kehitysympäristöön. Tarkastukseen asettamisen rajapinta ei myöskään toiminut niin kuin piti, sillä tarkastusprosessiin oli tullut muutoksia. Tämä vaati myös pienen korjauksen, jotta sprinttidemossa toiminnallisuus olisi kunnossa. Lopuksi tikettien esittely sujui hyvin ja asiakkaalta ei tullut huomautettavaa.

Uuden tiketin kanssa tuli hieman alkuvaikeuksia, koska selvisi, että tarkastusraporttien taustaprosessin osalta täytyi tehdä lisäselvitystä arkkitehdin puolelta. Esiin nousi, että käytettävä tekniikka olisi ”Pollaus” tai ”WebSocket”. Lähtökohtaisesti itsellä ei ollut mitään käsitystä siitä mitä nämä tarkoittivat, mutta ehdin pintapuolisesti selvittämään ternejä.

API:n pollauksessa kyse on siitä, että asiakasohjelma lähettää toistuvasti pyyntöjä API:n osoitteeseen ja vertailee, muuttuuko vastaus. WebSocket API:lla voi lähettää viestejä palvelimelle ja vastaanottaa tapahtumapohjaisia vastauksia ilman, että täytyy lähettää kyselyä palvelimelle vastausta varten. Siinä avataan kaksisuuntainen interaktiivinen viestintäistunto, jossa on matala viive ja joka voidaan pitää auki tarvittava aika.

Tiistai 28.05.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Tarkastusraporttien muodostamisen taustaprosessin läpikäyntiä/suunnittelua

- Tarkastusraporttipohjien tekeminen
- Tietovarastojen vienti Testi- ja QA ympäristöihin

Tavoitteena olisi viedä aamupäivästä minulle merkityt CodeCommit tietovarastot testi- ja QA-ympäristöihin. En ole tätä aikaisemmin tehnyt, niin täytyy pyytää joltain pikaopastus, jotta kaikki menee varmasti oikein. Loppupäivästä olisi hyvä saada raporttipohjat tehtyä ja toivottavasti jossain kohtaa ehditään käymään taustaprosessia läpi arkkitehdin ja muiden kehittäjien kanssa.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Aamupäivä kului ympäristöjen päivityksissä. Aamupalaverin jälkeen sovittiin, että toinen kehittäjä selvittää, soveltaisiko WebSocket taustaprosessiin. Tein itse käyttöliittymään valmiiksi napin, jotta WebSocket asiaa voisi testata nopeammin. Kävi ilmi, että WebSocket APIA ei pysty paikallisessa kehityksessä testaamaan, joten siitä todennäköisesti luovutaan. Käyttöliittymän napin osalta nousi esiin kysymyksiä, liittyen minne se laitetaan ja millaisena. Laitoin tästä kysymyksen käyttöliittymäsuunnittelijalle.

Keskiviikko 29.05.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Korjauksia tuntikirjanpitoon
- Käydään aamupalaverin jälkeen läpi tarkastusraporttien taustaprosessia
- Käyttöliittymäsuunnittelijalta tulee visuaalinen luonnos napista

Päivän tavoitteena olisi saada taustaprosessia eteenpäin. Täytyy koittaa tehdä lambda funktio ja katsoa mallia asiakirjan puolella tehdystä vastaavanlaisesta toteutuksesta. Toteutus on aika laaja ja monimutkainen, mutta toivottavasti se lähtee aukenemaan.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Taustaprosessi päätettiin toteuttaa "pollauksella", mutta en vielä tiedä miten ja missä se täytyy varsinaisesti toteutuksen osalta huomioida. Pääsin alkuun toteutuksen kanssa, mutta etenee hitaasti, sillä asiakirjapuolen toteutuksen dokumentaatio on aika vähäistä ja hajanaista. Tein myös raporttipohjat valmiiksi ja ne eivät vielä tällä hetkellä sisällä mitään kovin järkevää.

Torstai 30.05.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Tarkastusraporttien muodostamisen jatkaminen

Tavoitteena olisi saada lambda muodostamaan raporttipohjien mukaiset tarkastusraportit S3 ämpäriin. Tämän osalta täytyy ainakin selvittää, miten pdf tiedostot muodostavan palvelun kutsuminen toimii ja vaatii.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Sain annettua lambdalle tarvittavat oikeudet, jotta pdf-palvelun kutsumisen pitäisi toimia. Täytyy vielä selvittää, miten tätä pystyy paikallisesti testaamaan, jos se edes on mahdollista. Kehitystyö eteni kivasti ja ymmärrys taustaprosessin toiminnasta kasvoi.

Perjantai 31.05.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Tarkastusraporttien taustaprosessin kehitystä
- Pdf-muodostamisen palveluun tutustumista/testausta

Tavoitteena tehdä ainakin funktiot tiedoston nimen muodostamiselle, tiedoston kielen valinnalle, käytettävän raporttipohjan nimen valinnalle. Olisi kiva, jos saisi varsinaisen tarkastusraportin muodostettua S3 ämpäriin.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Lisäsin lambdaan tietokantaoperaation, jonka avulla saadaan haettua tarvittavia parametrejä. Tein funktiot raporttipohjan nimen valinnalle, tiedoston nimen muodostukselle, tiedoston kielen valinnalle. Selvitin ja toteutin, miten saadaan nappia painavan käyttäjän sähköposti ja sitä kautta etunimi/sukunimi päivitettyä tarkastusraportin tiedoston nimeen. Lisäsin tarkistuksen, jossa katsotaan riittääkö käyttäjän oikeudet muodostaa tarkastusraportteja ja onko tarkastus oikeassa tilassa. Sain paljon perustoiminnallisuuksia tehtyä ja testattua, mutta en vielä päässyt varsinaiseen tarkastusraportin muodostamisen testaukseen kiinni.

Viikkoanalyysi

Viikko oli melko perinteinen sprintin vaihtumisen osalta, tuli uusi laaja tiketti työn alle, jonka osalta täytyi tehdä paljon selvitystyötä ja asioihin tutustumista, ennen kuin pääsi varsinaisessa kehitystyössä vauhtiin. Tikettiin tutustuessa ajattelin, että minulle on selvää mitä tiketissä pitää saada aikaiseksi, mutta yllätyksenä tuli, että kuinka laajasta kokonaisuudesta loppujen lopuksi oli kyse.

Yhteenvedona, tämän viikon aikana täytyi tutustua ja selvittää seuraavia asioita:

- Minne rakennan raporttien muodostamisen prosessin
- Miten tarkastusprosessissa muodostetaan erilaisia raportteja
- Miten asiakirjoille muodostetaan päätösraportteja
- Mitä koodia voin poimia muista raporttitoiteuksista
- Miten saan lisättyä tarvittavat oikeudet lambdaalle
- Miten html koodi toimii raporttipohjissa

Loppuviikosta kaikki alkoi hiljalleen hahmottumaan ja sain paremman kuvan kokonaisuudesta ja itsevarmuutta tekemiseen. Monesti tuntuu, että on vaikea lähteä liikkeelle, jos ei kunnolla hahmota tai ole varma siitä mitä on tekemässä. Pollaukseen liittyvissä asioissa on vielä selvitettävää ja täytyy ensi viikolla kysyä, oliko mitään samantyyppisiä toteutuksia vielä tehtynä, joista voisi ottaa mallia.

Pollauksessa siis on tässä tapauksessa kyse siitä, että nappia painettaessa käyttöliittymä lähettää kyselyjä backendin puolella olevaan API osoitteeseen ja kun odotetaan oikeaa vastausta, käyttöliittymässä näytetään pyörivä hyrrä. Kyselyjä lähetetään mahdollisesti säännöllisin aikavälein ja tietyn aikaa, jonka jälkeen lähetykset voidaan lopettaa. Selvitettävänä on siis ainakin aikaväli ja kokonaisaika. Käyttöliittymä katsoo aina saamansa vastauksen perusteella, että voidaanko siirtyä prosessissa eteenpäin. Mikäli backend puolella tapahtuu virheitä, täytyy myös ne osata huomioida ja mahdollisesti lopettaa kyselyjen lähetys. Lähteestä (Amazon h) löytyy hyvä esimerkkivideo, jonka tyypeistä ratkaisua tässä haetaan.

Pollauksen hyvät puolet:

- Saatavuus, helppo implementointi API osoitteisiin
- Ei isoa viivettä, mikäli kyselyväli tai taajuus säädetty oikein
- Säädettyvyys, kyselyvälin voi säätää minuutista 24 tuntiin
- Matala verkon kulutus, jatkuvan viestinnän minimointi

Pollauksen huonot puolet:

- Ei reaaliaikaista dataa

- Palvelinkuormitus, toistuvat kyselyt voivat kuormittaa palvelinta turhaan
- Kyselyjä tehdään paljon turhaan, joka on kustannuserä

Usein pollausta vertaillaan webhookkeihin ja WebSocketeihin, mutta tehdään näiden osalta vertailuja, mikäli suunnitelmat tästä vielä muuttuvat.

3.7 Seurantaviikko 7

Maanantai 03.06.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Viikkopalaveri
- Tarkistaa tarkastusraportin lokituksen muodostuminen toimintaloki -lokiryhmään
- Täytyy selvittää, miten tulostuspalvelu toimii, joka muodostaa pdf raportit
- Täytyy katsoa ja selvittää, että käyttäjän tunnistaminen on kunnossa, jotta saadaan päivitettyä tietokantaan tarkastusraportin lisääjä oikein

Tavoitteena on opetella käyttämään tulostuspalvelua, joka siis on oma tietovarastonsa, jossa saadaan muodostettua .pdf muotoisia liitteitä S3 ämpäriin.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Lähdin aluksi liikkeelle tulostuspalvelusta ja kokeilin ajaa siinä olevaa lambda funktiota ensin paikallisesti. Tähän oli jokin verran tehty dokumentaatiota, joka auttoi, mutta lopulta jäin jumiin johonkin Javaan liittyviin ongelmiin. Kaikki Javaan liittyvät toiminnot ja ongelmat ovat minulle vähän tuntemattomampia, joten päädyin viemään kehityshaarani suoraan AWS puolelle. Tämän jälkeen testasin ajaa lambdaa AWS:ssä. Tätä varten oli dokumentaatiossa valmiiksi JSON pohjia, joita pystyi kokeilemaan ja muokkaamaan. Sain lopulta tarkastusraportit muodostettua S3 asti, kun huomasin, että tarkastusraporttipohjissa oli jäänyt HTML koodiin yksi ylimääräinen `</main>` rivi, joka aiheutti virheitä lambdan suorituksessa.

Suurin osa päivästä kului tulostuspalvelun opetteluun, mutta lopulta opin millä logiikalla se toimii. Tämän seurauksena tein myös pieniä korjauksia backend puolen lambdaan, jotta sieltä lähetetään oikean muotoista dataa tulostuspalveluun. Loppupäivästä käytiin vielä toisen kehittäjän kanssa läpi hänen tekemäänsä pollaus-toteutusta, jossa käyttöliittymästä pollataan backendiä. Täytyy myöhemmin palata tuohon ja tutustua tarkemmin.

Tiistai 04.06.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Yksikön tietoisuus karttateknologioihin
- Kokeilla viedä asiakirjapalvelun kehityshaara AWS puolelle ja testata lambdan yhteys tulostuspalveluun
- Selvittää missä tietokantaan päivitetään tarkastusraporttien lisääjä ja lisäyspäivämäärä

Tavoitteena olisi saada asiakirjapalvelun backend lambdan ja tulostuspalvelun lambdan välinen toiminta kuntoon, jotta saadaan muodostettua tarkastusraportteja. Tämän jälkeen voisi siirtyä käyttöliittymätoteutuksen tekemisen pariin.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Aamupalaverissa asiakkaalta tuli toive tehdä pieni tietokantapäivitystehtävä toiseen projektiin, jossa olen ollut mukana aikaisemmin. Tämä onnistui ongelmitta. Aamupäivästä tein backendin puolelle tietokantaoperaation, jossa Liite tauluun lisätään uudelle riville tarkastusraportin muodostukseen liittyvät tiedot, kuten lisääjä ja lisäyspäivämäärä.

Vein kehityshaarani asiakirjapalvelusta ja tulostuspalvelusta AWS:ään ja koitin testata näiden välistä toimintaa. Yritin tehdä testejä AWS puolella ja kutsua rajapintaa POSTMAN ohjelmalla, mutta kummassakin tapauksessa tuli ongelmia auktorisoinnin kanssa. Nämä ongelmat liittyivät siihen, että rajapintaa ei kutsuttu käyttöliittymän puolelta, joten puuttui käyttöliittymän kirjautumiseen liittyvät käyttäjätiedot kutsuista. Tähän ei ehtinyt löytyä ratkaisua.

Keskiviikko 05.06.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Asiakirjapalvelun ja tulostuspalvelun testauksen ongelmien selvittäminen

Tavoitteena on saada palvelujen välinen testaus toimimaan.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Sain testattua palvelujen toiminnan poistamalla koodista auktorisoinnin ja kovakoodaamalla tarkastuksen tunnusteen väliaikaisesti. Tämän jälkeen tuli vielä monimutkainen ja pitkä virheviesti liittyen axiosiin. Axios on lupauspohjainen http kirjasto node.js:lle, jonka avulla voi lähettää http pyyntöjä. Tämän ratkaisemisessa lähdin katsomaan muualta koodista, että missä muodossa tietoa on lähetetty tulostuspalveluun ja huomasi eroavaisuuksia oman ja muiden toteutusten välillä. Kun yhtenäistin tiedonlähetyksen, niin ongelma ratkesi ja sain tarkastusraportteja muodostumaan S3.

Palautin backendiin auktorisoinnin ja kokeilin käynnistää käyttöliittymän paikallisesti, samalla kun kehityshaarani olivat AWS puolella toiminnassa. Olin jo aikaisemmin tehnyt käyttöliittymään napin, jota painamalla pystyi testaamaan backendin toiminnan. Harvinaista kyllä, mutta kaikki toimi käyttöliittymän, backendin ja tulostuspalvelun välillä kertakokeilulla. Tämän jälkeen lähdin katsomaan, miltä käyttöliittymäsuunnittelija oli suunnitellut käyttöliittymätoteutuksen näyttävän. Lisäsin käyttöliittymään haitarityyppisen valikon, jossa on nappi ”Muodosta tarkastusraportti”. Tämän napin painamisen yhteyteen lisäsin vielä ponnahdusikkunan, jossa varmistetaan, että käyttäjä varmasti haluaa muodostaa uuden tarkastusraportin ja korvata jo olemassa olevan tarkastusraportin, mikäli sellainen on tehty aikaisemmin. Käyttöliittymän kehityksestä itsellä on kulunut jo aikaa, joka vaati paljon tutustumista nykyiseen toteutukseen, sillä käyttöliittymäkehitystä on tapahtunut paljon.

Torstai 06.06.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Käyttöliittymätoteutuksen jatkaminen
- Selvittää, miten tarkastusraportin lisääjä ja lisäyspäivämäärä saadaan lisättyä käyttöliittymään
- Selvittää, miten tarkastusraportin saa lisättyä ladattavaksi käyttöliittymästä
- Selvittää, miten edellisen tarkastusraportin korvaaminen toteutetaan
- Selvittää miten pollaus tehdään käytännössä

Tavoitteena on saada käyttöliittymäpuolta eteenpäin, tämä vaatii luultavasti apua muilta kehittäjiltä, sillä olen tehnyt käyttöliittymäpuolta niin vähän ja toteutus on todella laaja.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Aamupalaverissä sain arkkitehdiltä ohjeistusta, että tiketti on saatava valmiiksi tässä sprintissä ja toteutus saa olla tässä vaiheessa hyvin yksinkertainen, eikä tarkastusraportin tarvitse olla ladattavissa käyttöliittymästä. Tämä tarkoitti, että myöskään pollausta ei tarvitse tässä kohtaa toteuttaa.

Käyttöliittymään lisäsin napille ominaisuuden, missä tulee pyörivä hyrrä indikoimaan, että tarkastusraporttia muodostetaan. Tein myös niin, että nappi on harmaana, eikä painettavissa, mikäli käyttäjällä ei ole oikeuksia muodostaa tarkastusraporttia tai tarkastus on väärässä tilassa. Lisäsin taustaprosessiin tarkastuksen, jossa katsotaan, onko aikaisempaa tarkastusraporttia olemassa ja jos on, niin poistetaan se ja muodostetaan uusi. Varmistelin arkkitehdiltä vielä hyväksymiskriteerejä ja kaikki pitäisi toteutuksessa olla kunnossa tällä hetkellä. Kysyin lisäksi asiakkaalta, että mitä nimeä käytetään tarkastusraportille ruotsiksi ja sain vastauksen ja lisäsin sen taustaprosessiin.

Loppupäivästä testasin, että toteutuksessa kaikki komponentit toimivat yhdessä ja hyväksymiskriteerit toteutuvat, siivosin toteutusta ja julkaisin 3 kehityshaaran yhdistämispyyntöä muiden kehittäjien katselmoitavaksi.

Perjantai 07.06.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Sprinttidemoon ja retroon valmistautuminen
- Arkkitehdin tikettien kommentoiminen
- Tarkastusraporttien muodostamisen katselmointitilaisuus
- Katselmointitilaisuudessa havaittujen korjauksien tekeminen, mikäli nousee esille jotain
- Muiden kehittäjien katselmointeihin osallistuminen

Tavoitteena olisi saada oma tiketti katselmoitua ja valmiiksi, sekä osallistua muiden kehittäjien katselmointitilaisuuksiin, jotta saadaan mahdollisimman paljon tikettejä valmiiksi ennen sprintin loppumista.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Aamusta luin arkkitehdin linkittämän tiketit läpi ja arvioin niiden laajuuden, sekä mikäli tiketeille täytyisi vielä täydentää jotain toteutuksen kannalta oleellista. Valmistauduin oman tiketin katselmointitilaisuuteen, mutta ilmaantui erikoisia ongelmia, joista en saanut mitään virheviestejä tulostumaan. Tämä kuitenkin korjaantui ehkä itsestään tai sitten jotain vain oli mennyt pieleen kehityshaarojen viennissä AWS puolelle. Sain pidettyä katselmointitilaisuuden omalle tiketille ja esiin nousi muutamia pieniä huomiota, sekä yksi välttämätön korjaus. Tein korjauksen tarkastusraporttien korvaamiseen liittyen, sekä muita pieniä muutoksia. Tiketti tuli loppupäivästä valmiiksi ja kaikki vaikutti toimivan kehityshaarojen yhdistämisen jälkeenkin.

Viikkoanalyysi

Viikko meni nopeasti ja pääsin tavoitteeseeni, eli sain tiketin valmiiksi ennen sprintin loppua. Perjantai oli kiireinen kaikilla kehittäjillä, joten tikettini katselmointitilaisuus jäi hieman kesken, sillä alkoi pakollinen palaveri juuri katselmoinnin aikaan. Minulla oli kolme kehityshaaran yhdistämispyyntöä katselmoitavana tikettiin liittyen ja kaiken huolelliseen tarkistamiseen olisi mennyt paljon aikaa. On mahdollista, että jotain viilattavaa vielä jäi, mutta onneksi prosessin edetessä asiakas kyllä ilmoittaa, mikäli jotain muutoksia täytyy tehdä. Toivon myös, että kun maanantaina tulee sprinttidemo, että mitään ei ole rikkoutunut toteutuksen suhteen, sillä samojen tietovarastojen kanssa oli niin sanotusti ruuhkaa perjantaina.

Viikon aikana esiin tulleita ongelmia:

- Tulostuspalvelun opettelu
- Backendin testaaminen tulostuspalvelun kanssa
- Backendin auktorisoinnin ongelmat
- Käyttöliittymäkehitys
- Kolmen eri palvelun testaaminen yhdessä
- Muiden kehittäjien kanssa töiden koordinointi

Aikaisempi viikko meni pitkälti tutustuen uuteen tikettiin ja rakennellessa backendiä asiakirjapalveluun. Tällä viikolla tekeminen keskittyi enemmän eri palveluiden yhdistämiseen ja niiden toiminnan testaukseen. Yleensä pyrin testaamaan kaiken mahdollisen paikallisessa kehityksessä, mutta tämän tehtävän kanssa se osoittautui haastavaksi. Asiakirjapalvelun backendiä pystyi ajamaan paikallisesti tiettyyn pisteeseen, mutta siinä piti koodin alkupuolelta kommentoida auktorisointi pois. Yhteyttä tulostuspalveluun ei saanut muodostettua paikallisesti, joten siinä kohtaa piti vaihtaa AWS:n puolelle. Myöskään tulostuspalvelun testaaminen paikallisesti ei onnistunut ja ajansäästösyistä päätin senkin osalta siirtyä lopulta tekemään kehitystä pilven puolella. Ja vaikka siirtyi tekemään kehitystä pilveen, niin silti auktorisoinnin liittäminen testitapauksiin osoittautui liian monimutkaiseksi, eikä muiltakaan kehittäjiltä löytynyt ratkaisuja. Onneksi moniin ongelmiin löytyy kiertoteitä ja tässä tapauksessa päädyin kommentoimaan osan koodista ja kovakoodaamaan tiettyjä arvoja väliaikaisesti, jotta testaus onnistui.

AWS:n puolella koodin testauksessa hankaloitti se, että myös moni muu kehittäjä halusi laittaa kehityshaaransa pilveen ja testata sitä kautta töidensä toimivuutta. Meillä on onneksi tähän liittyen Teamsissä oma kanava, jossa jokainen ilmoittaa, kun käyttöönottaa oman kehityshaaransa kehitysympäristöön. Kanava samalla helpottaa koordinoimisessa, sillä aikaleiman perusteella voi katsoa milloin toinen kehittäjä on käyttöönoton tehnyt ja voi laittaa itsensä niin sanotusti jonoon siksi aikaa, kunnes toinen on valmiina. Koitin tähän liittyen etsiä tietoa, että onko olemassa jotain suositeltavia tapoja, mutta en löytänyt mitään. Paras tapa olisi varmistaa uusia projektin osia tehdessä, että niitä pystyy testaamaan ja kehittämään paikallisesti mahdollisimman hyvin ja yksinkertaisesti.

Oli myös mukavaa päästä tällä viikolla jatkamaan käyttöliittymäkehitystä. Siinä minulla on varmasti eniten opittavaa ja se on myös mielenkiintoista. Käyttöliittymän kehityksessä on myös se hyvä puoli, että siellä on jo lähes kaikkea mahdollista toteutettu, joten kaikkia ratkaisuja ei tarvitse itse keksiä. Käyttöliittymäpuolen töistä on kuitenkin muodostunut vähän pelottava kuva, sillä tietovarasto on valtavan iso ja toteutus on tehty Next.js:llä, enkä ole ikinä sitä opiskellut. Reactiakin olen opiskellut vain vähän, mutta se on osoittautunut silti kohtuu helpoksi ymmärtää ilman suurempaa kokemustakin. Olen huomannut, että myös kokeneemmilla kehittäjillä on ollut enemmän vaikeuksia

käyttöliittymään liittyvien toteutuksien kanssa. En tiedä onko tämä miten yleistä vai vain projekti-kohtaista.

Seuraavalla viikolla jatkan todennäköisesti käyttöliittymäkehityksen parissa, sillä tällä viikolla valmistuneelle tarkastusraporttien luonnille tulee jatkoa. Jatkossa tarkastusraportit täytyy olla ladattavissa käyttöliittymän puolelta ja käyttöliittymään täytyy päivittää myös raporttien muodostaja ja muodostuspäivämäärä. Tämä olisi ainakin itselle mieluista tekemistä ja toivottavasti pääsen vaikuttamaan asiaan.

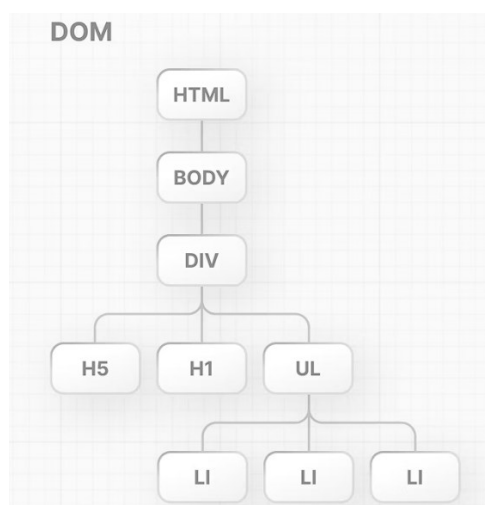
Next.js perusteita tukemaan käyttöliittymäkehitystöitä:

Next.js on joustava viitekehys Reactille, joka tarjoaa rakennuspalikoita nopeille full-stack verkkosovelluksille. Näitä rakennuspalikoita ovat esimerkiksi:

- Käyttöliittymä
- Reititys
- Datat noutaminen
- Staattinen ja dynaaminen renderöinti
- Skaalautuvuus ja suorituskkyky

React tarjoaa kätevästi elementit, jotka käyttäjälle näytetään, mutta next.js hoitaa tarvittavat konfiguroinnit, työkalut ja rakenteen, jotta näistä yhdessä tulee optimoitu sovellus (Next.js a).

DOM on HTML-elementtien objektiesitys. Se toimii ns. siltana koodin ja käyttöliittymän välillä ja siinä on puumainen rakenne. DOM rakenteiden avulla voi muokkaukset kohdistaa tiettyihin elementteihin ja muuttaa niiden tyyliä, sekä sisältöä.



Kuva 7 Objektiesityksen puumainen rakenne (Next.js b)

JSX on JavaScriptin syntaksilaajennus, jossa voi kirjoittaa koodia sekä JavaScriptillä, että HTML:llä. Selaimet eivät kuitenkaan osaa lukea tätä suoraan vaan tarvitaan JavaScript kääntäjä, joka kääntää tiedoston tavalliseksi JavaScriptiksi (Nextjs c). Pintapuoleisella tietojenhaulla nextjs vaikuttaa tarjoavan työkaluja helpottamaan kehitystyötä Reactin kanssa, eikä tästä ole kauheasti hyötyä ilman, että osaa tehdä käyttöliittymäkehitystä Reactilla. Täytyy myöhemmin vielä palata Reactiin ja Next.js:ään syvällisemmin opintojen puolella.

3.8 Seurantaviikko 8

Maanantai 10.06.2024 tehtävät ja tavoitteet

Tehtävät

- Sprinttidemon suunnittelu ja testaus
- Sprinttidemo
- Sprinttiretro
- Sprinttisuunnittelu
- Testiin vienti

Tälle päivälle on paljon kokouksia ja loppupäivästä AWS testiympäristön päivityksiä. Uutta tikettiä tuskin ehtii aloittamaan.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Sprinttidemo sujui hyvin, eikä asiakkaalta noussut esiin huomautettavia asioita. Päivä kului pitkälti palaverien parissa ja osallistuin hieman testiin vienteihin.

Tiistai 11.06.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Uusi tiketti työn alle
- Tarkastusraportin lataaminen ja näyttäminen käyttöliittymässä

Tavoitteena tehdä mahdolliset selvitystyöt uuteen tikettiin liittyen ja päästä aloittamaan toteutusta. Täytyy katsoa, miten käyttöliittymäsuunnittelija on toiminnallisuuden hahmotellut.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Lähdin ensin tekemään <Link> -tyyppistä nappia ja sille funktion, jonka tarkoituksena on käynnistää tarkastusraportin lataaminen käyttäjän tietokoneelle. Nappi näkyy käyttöliittymässä ”Lataa tarkastusraportti” nimisenä. Lisäsin myös lisääjän ja lisäyspäivämäärän kentät. Jäin loppupäivästä hieman jumiin siihen, että tarkastusraportin lataaminen avasi uuden välilehden, joka ilmoitti ”403 – forbidden” viestin. Latauslinkissä on jokin varmaan väärin, täytyy tarkistaa sen toiminta.

Keskiviikko 12.06.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Viikkopalaveri
- Selvittää kumpaa kahdesta eri liitepalvelun lambdaista käytetään muodostamaan liitteelle linkki s3
- Latauslinkin perään tulee liitteen ID, täytyy selvittää, vaatiiko tämä backend muutoksia

Tavoitteena on saada tarkastusraporttien lataaminen toimimaan tämän päivän aikana. Sitä ennen täytyy katsoa mikä ero on kahdella hyvin samalta näyttävällä lambdailla, jotta tietää kumpaa tulisi käyttää muodostamaan s3:seen linkki, jotta lataus tarkastusraportin lataus onnistuu.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Liitepalvelun lambdajen välinen ero selvisi, ne hakevat dataa eri tietokantatauluista. Tein myös backendiin hieman muutoksia, jotta s3 päivittyy tiedoston nimen perään tietokantaan muodostuvan liitteen tunniste. Selvitin arkkitehdiltä, että halutaanko tarkastusraportin latautumisen käynnistyvän heti kun linkkinappia painetaan, vai onko tarkoitus aueta uusi välilehti, jonka kautta tarkastusraportti on ladattavissa. Vastaus oli, että tehdään toteutus niin kuin on aikaisemmin hankkeiden liitteille tehty. Kävin katsomassa käyttöliittymästä, miten hankkeilla liitteiden lataus toimii ja tiedoston lataus alkoi heti kun linkkiä painettiin. Omassa toteutuksessa linkin painaminen avaa uuden välilehden, joten tämän selvittämisessä meni paljon aikaa, että mitä oli tehty eri tavalla. Lopulta laitoin meidän Teams kanavalle kysymyksen tästä, jos joku osaisi auttaa ongelman selvityksessä. Kävi ilmi, että esimerkki hankkeen liitteen latauksesta, jonka olin löytänyt käyttöliittymästä, oli erikoistapaus ja poikkesi normaalista. Tämä tarkoitti käytännössä sitä, että liitteiden kuuluikin avautua ensin uuteen välilehteen, joten oma toteutus oli tehty oikealla tavalla, eikä tarvinnut tehdä korjauksia.

Torstai 13.06.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Tavoite ja kehitysprosessi -koulutus

- Täytyy tarkistaa, että puuttuuko tarkastusraporttien lataamisesta jotain ja testata lisää
- Mahdollisesti tehdä kehityshaaran yhdistämispyynnöt, jos kaikki on kunnossa tiketillä

Päivän tavoitteena on saada testattua tarkastusraporttien lataaminen, tarkistaa hyväksymiskriteerit ja laittaa kehityshaaran yhdistämispyynnöt muiden kehittäjien katselmoitavaksi. Mikäli kaikki sujuu hyvin, tiketti voisi valmistua tänään.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Huomasin testatessa, että kun tarkastusraportti muodostetaan, niin muodostuksen jälkeen käyttöliittymässä ei päivity lisääjä ja lisäyspäivämäärä heti, vaan sivu täytyy erikseen virkistää. Ongelman selvityksessä kesti kauan, eikä oikein muiltakaan tuntunut löytyvän ratkaisua. Lopulta tein backendiin muutoksen, jossa backend lähettää käyttöliittymälle tiedot päivittäjästä ja päivityspäivämäärästä ja muut tarvittavat muuttujat kutsun yhteydessä. Tämän jälkeen, kun käyttöliittymässä muodostetaan tarkastusraportti, niin samassa funktiossa päivitetään tarkastuksen mallin uudet muuttujat set funktioilla. Toiminta on vähän saman tyyppinen kuin Reactin useState, mutta arvot päivitetään malliin (model). Ratkaisu toimi, vaikka en itsekkään täysin ymmärrä miksi, mutta käyttöliittymä jotenkin hakee tietoja eri paikoista eri tavalla, liittyen MobX, jota käyttöliittymässä käytetään. Tämän jälkeen huomasin, että kaikki ei vielä kuitenkaan toiminut oikein. Täytyi vaihtaa muuttujien arvoja riippuen tilanteesta, että onko tarkastusraporttia jo olemassa vai muodostetaanko se vasta. Jos avaa tarkastuksen, jolla on tarkastusraportti, tiedot näytetään tietokannassa olevan datan perusteella ja jos avaa tarkastuksen, jolla ei ole tarkastusraporttia, tiedot näytetään, kun käyttäjä on lähettänyt kutsun backendiin ja käytetään sieltä tulevaa dataa. Huomasin myös, että nappien nimiä täytyi vähän muuttaa aina tilanteesta riippuen, joten korjasin myös sen. Lopulta kaikki vaikutti toimivan niin kuin pitää ja loppupäivästä tein kehityshaaran yhdistämispyynnöt käyttöliittymän tietovarastoon ja asiakirjapalvelun tietovarastoon muiden kehittäjien tarkistettavaksi.

Perjantai 14.06.2024 tehtävät ja tavoitteet

Tehtävät

- Daily
- Tarkastusraporttien latauksen tiketin katselmointitilaisuuteen valmistautuminen
- Katselmointitilaisuuden järjestäminen

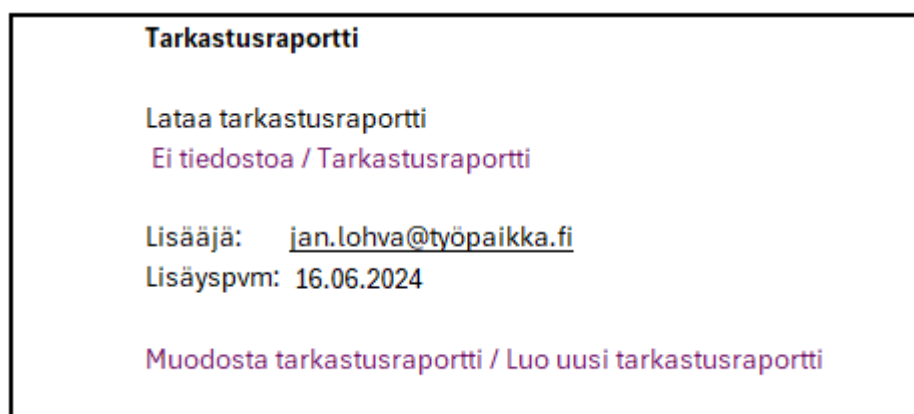
Pystyn tekemään vain puolikkaan työpäivän, joten tavoitteena olisi pitää katselmointitilaisuus ja saada tiketti valmiiksi.

Tavoitteiden ja tehtävien toteutuminen, sekä osaamisen kehittyminen

Tavoitteeseen päästiin ja tiketti tuli valmiiksi. Katselmoinnissa ei onneksi noussut esille kuin pieniä huomioita liittyen koodin järjestykseen, jotta rakenne olisi selkeämmin järjestelty.

Viikkoanalyysi

Tämä viikko keskittyi pääosin käyttöliittymäpuolen kehitykseen, josta itsellä on vähiten kokemusta ja osaamista. Kehitystyö eteni kuitenkin yllättävän hyvin, sillä ei tarvinnut lisätä mitään monimutkaista toiminnallisuutta, vaan suurimpaan osaan koodista pystyi ottamaan mallia muiden tekemistä ratkaisuista.



Kuva 8 Tarkastusraportin lataamisen havainnekuva

Kuvassa 8 on pääpiirteittäin hahmoteltu tämän viikon tiketin sisältö käyttöliittymän näkökulmasta. Lähtötilanteessa, jossa tarkastusraporttia ei vielä ole kertaakaan tehty tarkastukselle, sen voi muodostaa painamalla "Muodosta tarkastusraportti". Jos sellainen on olemassa, niin silloin napissa lukee "Luo uusi tarkastusraportti". Vastaavasti lukee "Ei tiedostoa" jos tarkastusraporttia ei ole ja "Tarkastusraportti" -linkki näkyy kun tarkastusraportti löytyy tietokannasta ja s3 ämpäristä. Lisääjä ja lisäyspäivämäärän kentät ovat tyhjiä kun raporttia ei ole ja täydentyvät kun sellainen tehdään.

Tällä viikolla ensimmäisenä suurempana ongelmana oli saada tarkastusraportin linkki toimimaan. Tähän ratkaisu oli onneksi helppo, sillä aluksi linkki ohjasi väärään osoitteeseen, josta puuttui tiedoston nimen perästä liitteen tunniste. Huomasin tämän, kun katselin tietokannasta ja s3 ämpärin kautta muiden liitteiden nimiä.

Kun sain linkin toimimaan ja tarkastusraportti avautui uudelle välilehdelle näkyviin pdf muotoisena, aloin epäilemään, että haluttiinko tiedoston latauksen alkavan. Kun arkkitehti kertoi, että toiminnallisuus olisi oltava samalla tavalla kuin aikaisemmin toteutetuilla hankkeen liitteillä, tarkistin asian ja löytämäni esimerkitapaus aloitti tiedoston latauksen saman tien, kun nappia painettiin. Tästä kertyikin aikaa vievä selvitystyö, jonka päätteeksi sain kuulla, että löytämäni esimerkitapaus oli normaalista poikkeava. Toteutukseni oli ollut oikein alusta alkaen, eikä muutoksia onneksi

täytynyt tehdä. Asia oli kyllä turhauttava, sillä tein turhaa ongelmanratkaisua puolen päivän verran. Vaikka arkkitehti onkin kiireinen, niin tuossa tapauksessa, kun kerroin hänelle kahdesta mahdollisesta toteutustavasta, olisi hänen pitänyt selvittää asia ja tarjota vastaus.

Seuraava aikaa vievä ongelma tuli vastaan testauksissa, kun huomasin, että tarkastusraportin lisääjä ja lisäyspäivämäärä eivät päivittyneet käyttöliittymälle heti raportin muodostuttua. Myöskään linkki tarkastusraportin katselemista varten ei toiminut ennen kuin sivun oli itse manuaalisesti päivittänyt. Tähän koitin aluksi saada ratkaisua Reactin useState hookilla, mutta se ei toiminut. Selvittelin asiaa ja useStaten ollessa asynkroninen, se ei toimi asynkronisen funktion sisällä oikein. Sain tiimiläiseltä ehdotuksen, että jos malliin lisäisi uudet tarvittavat muuttujat ja päivittäisi ne funktion sisältä käsin, niin homma voisia toimia. Ja toimikin. Tämän jälkeen täytyi enää vaihtaa kenttien päivitykset toimimaan joko tietokannan datalla tai funktiosta saadulla datalla, joka osottautui kohtuullisen helpoksi tehtäväksi.

Tästä käyttöliittymätiketistä jäi pintapuoleisesti mieleen se, että olisi hyvä, jos käyttöliittymän rakennetta olisi käyty kunnolla läpi ja pidetty tarvittava perehdytys ennen kuin meidän tiimimme 2 alkaa tekemään käyttöliittymäkehitystä. Teemme tällä hetkellä suurimmaksi osaksi käyttöliittymätoteutuksia, joten vielä ehtisi perehdytyksen toteuttamaan. Täytyy tähän liittyen saada mainittua asiasta, ennen kuin ns. käyttöliittymän suunnittelusta vastaava kehittäjä vaihtaa toiseen projektiin.

Ylipäättänsä kun tiedetään, että projektissa on kehittäjinä henkilöitä, joilla ei käyttöliittymäkehityksestä tai mistä vaan oleellisesta osaamisalueesta ole kokemusta, pitäisi mahdollisimman aikaisessa vaiheessa järjestää tarvittavat koulutukset. Näin ainakin toimitaan varmasti suurimmalla osalla aloista, mutta en osaa sanoa, että onko IT-ala siinä suhteessa erilainen. Veikkaukseni on, että omassa tapauksessani esimiehellä on liian paljon työntekijöitä, jonka johdosta työhyvinvointi saattaa kärsiä tämän kaltaisilla osa-alueilla, sillä aikaa on vain rajallisesti.

4 Pohdinta

Tavoitteenani oli kehittyä mahdollisimman monella osa-alueella, esimerkiksi käyttöliittymäkehityksen, kokonaisuuksien hahmottamisen, hyvien käytänteiden, käsitteiden ja sanaston, sekä teknisen ymmärryksen saralla. Olen mielestäni kehittynyt reippaasti monella näistä osa-alueista tämän parin kuukauden aikana. Suurimmaksi osaksi osaaminen on kasvanut käyttöliittymäkehityksen puolella, mutta myös TypeScriptin ja AWS:n osalta. Kun aloitin opinnäytetyön kirjoittamista, niin tiimissä oli tapahtunut juuri suuri rakennemuutos, josta johtuen tämän 2 kuukauden aikana myös työnkuvani ja arkityöni muuttui huomasti. Tästä johtuen työtehtävät ovat olleet laajempia ja haastavampia ja sitä kautta myös osaamisen kannalta kehittävämpiä. Olen päässyt paljon monipuolisemmin käyttämään AWS:n palveluja ja joutunut opettelemaan, miten eri järjestelmien osat kommunikoivat keskenään. On täytynyt myös paljon enemmän seurata muiden tikettejä, jotta pystyy hahmottamaan toteutuksen tilaa ja ymmärtää omien tikettien tarkoituksen isossa massassa. Nykyään käytän enemmän aikaa uusien asioiden ihmettelyyn, opetteluun ja ymmärtämiseen, sillä olen huomannut, että samat ongelmat tulevat monesti uudestaan esiin, jolloin on tehokkaampaa sisäistää asiat heti niiden ilmaannuttua. Jokin tekninen uusi asia saattaa monesti olla niin laaja, että sillä hetkellä riittää vain tietyn palasen opetteleminen siitä, jotta saa omia töitään eteenpäin, mutta myöhemmin tulee lisää siihen liittyviä asioita, jolloin täytyy kuitenkin lukea ja opetella käyttöä lisää. Tällöin tiedonhakuun menevää aikaakin saa minimoitua, kun lukee ja opettelee kattavasti jo ensimmäisellä eteen tulevalla kerralla. Sitä mukaa kun osaaminen karttuu, niin myös oma itseluottamus kasvaa, niin tekemisessä, päätöksen teossa, kuin muiden auttamisessakin. Nykyään, kun pyrkii paremmin hahmottamaan kokonaisuuksia, niin huomaa, että osaa myös vastaila paremmin asiakkaan kuin tiimiläistenkin kysymyksiin ja pystyy olemaan paremmin avuksi koko projektin kannalta.

En tämän opinnäytetyön aikana lähtenyt perehtymään uusien ratkaisumallien tai menetelmien kirjallisuuteen, koska tämänhetkisessä projektissani jokainen tiketti voi olla täysin uudenlaista tekemistä verrattuna aikaisempaan. Lisäksi koen, että tällä hetkellä pystyn parhaiten kehittämään itseäni ottamalla mallia kokeneilta kehittäjiltä. Tässä pyrin siis aktiivisesti seuraamaan heidän käytänteitään ja ratkaisumalleja, jotka soveltuvat omaan tekemiseeni. Jos esimerkiksi joudun aloittamaan jo tietyvästi monimutkaista tikettiä tai koen itse, etten kunnolla ymmärrä tiketin tavoitteita, niin pyrin käymään asiat läpi puhelussa sellaisen kokeneen henkilön kanssa, joka on parhaiten aihealueesta perillä. Tavoitteenani on myös nykyään osallistua enemmän katselmointitilaisuuksiin kuin aikaisemmin, sillä niiden kautta pystyy seuraamaan paremmin projektin etenemistä ja uusien toiminnallisuuksien kehittymistä. Katselmoinneissa tulee käytyä myös hyvin läpi muiden tekemää koodia ja on helppo esittää kysymyksiä monimutkaisilta näyttävistä asioista. Alalla pidempään olleet ymmärtävät hyvin mielenkiinnon ja heiltä saa paljon hyviä vastauksia ja selityksiä, että miten ja

miksi jokin asia on esimerkiksi tehty. Katselmoinneissa meillä on ollut myös tapana jutustella ajankohtaisista asioista ja monesti kokeneemmat kehittäjät jakavat hyviä käytänteitä ja vinkkejä esimerkiksi käytettäviin ohjelmistoihin liittyen. Monesti kehittäjät livekoodaavat, jos on huomattu jokin virhe toteutuksessa ja pääsee myös livenä seuraamaan heidän ongelmanratkaisumenetelmiään. Tämän opinnäytetyön ansiosta olen panostanut enemmän itseni kehittämiseen kehittäjänä ja uutena tapana on tullut, että katson Teamsin kalenteria enemmän päivä ja viikkotasolla, jotta hahmotan viikon kulun paremmin.

Opinnäytteen kirjoittamisen aikana opin taas olemaan opiskelija, sillä olen pitkään keskittynyt tekemään töitä ja ehkä pyrkinyt siihen, että saan mahdollisimman paljon valmiiksi, sen sijaan että sisäistäisin ja oppisin asioita kunnolla. Aikaisemmin olen monesti vain pintapuoleisesti selannut tai vilkaissut teorian uusista asioista, mutta nykyään katson ne tarkemmin, vähän samaan tapaan kuin lukisin koulun opetusmateriaaleja, ja pyrin löytämään ja ymmärtämään tärkeimmät asiat. Olen oppinut, että kannattaa tutustua ja selvittää vastaan tulevat tuntemattomat asiat heti, sillä ne tulevat vastaan usein myöhemminkin. Olen myös oppinut ottamaan työpäivästä enemmän aikaa opetteluun ja kokonaisuuksien hahmotteluun, sekä työn suunnitteluun.

Huomasin projektin jakauduttua kahtia, kun työtehtävät monipuolistuivat, että työtehtävien osalta on vaikeaa poistua omalta mukavuusalueelta ja mieluusti tekisi vain niitä tehtäviä, joista tietää osaavansa jo jotain tai ainakin niitä, joiden lopputuloksen pystyy hahmottamaan jo esimerkiksi tikein otsikon perusteella. Kun alkoi tekemään ensimmäisiä monimutkaisempia tehtäviä, niin aikaa meni yllättävän paljon selvittelyyn ja suunnitteluun, kuin myös oman suunnitelman epäilyyn. Välillä tuntui, että käyttää liian paljon aikaa tikettiin tutustumiseen ja että pitäisi päästä nopeammin jo toteutuksessa eteenpäin. Huomasin kuitenkin, että myös monilla kokeneemmilla kehittäjillä kesti aikaa päästä alkuun ja myös he joutuivat tekemään paljon selvitystyötä niin asiakkaan kuin tiiminkin puolelta. Jossain vaiheessa huomasin, että nykyinen työskentely on tavallaan uusi normaali. Aikaisemmin saatoin koittaa valikoida itselle nimettyjä tikettejä enemmän, mutta nykyään otan rohkeammin työtehtäviä vastaan niiden kiireellisyysjärjestyksessä, vaikka asiat tuntuisivatkin itselle uusilta ja monimutkaisilta. Tämä johtuu pitkälti siitä, että olen tiedostanut vaikeiden työtehtävien opettavan ja kasvattavan itseäni ammattilaisena kaikista eniten. Tämän parin kuukauden aikana itseluottamus on kasvanut huomasti, kun kokemusta on karttunut. Tulevaisuudessa tämä varmasti kiihdyttää omaa kehitystä ja pystyn olemaan entistä paremmin avuksi myös muille kehittäjille. On ollut myös mielenkiintoista huomata, että meidän tiimimme on hioutunut paremmin yhteen, mitä vaikeammiksi ja monimutkaisemmiksi toteutukset ovat menneet. Yhteishenki on kasvanut, apua saa entistä nopeammin ja kysymyksiin vastataan vaikkei välttämättä olisikaan tiedossa täysin oikeaa vastausta. Yhdeltä tiimiläiseltä tuli mielestäni osuvat kommentit ”Aikaisemmin työtehtävät ovat olleet lähinnä

koodausharjoituksia, mutta nykyinen tekeminen on sitä oikeaa ohjelmistokehitystä”. Tämä mielestäni kuvaa hyvin nykyisiä työtehtäviä, kun täytyy hahmottaa asioita laajemmin, sekä pystyä selvittämään ja etsimään mahdollisia virheitä AWS:n syövereiden massiivisuudesta.

Tulevaisuudessa tulen varmasti kehittämään osaamistani tekemällä monipuolisesti erilaisia työtehtäviä. Täytyy reippaasti tarttua haasteisiin ja ottaa työn alle niitä itselle vaikeilta ja uusilta tuntuvia aihepiirejä. Aion kuunnella ja tarkkailla erityisesti kokeneita kehittäjiä ja testata sopivatko ja auttavatko heidän vinkkinsä omassa tekemisessäni. Mikäli jossain vaiheessa tulee sellainen tilanne eteen, että työtehtävistä ei enää tunnu kehittyvän tarpeeksi, niin siinä kohtaa täytyy varmasti miettiä esimerkiksi työtehtävien vaihtoa projektin sisällä tai jos mahdollista, niin uuteen projektiin siirtymistä. Nykyinen työnantajani on onneksi niin suuri yritys, että yrityksen sisällä on lähes loputtomasti kehittymismahdollisuuksia. Tässä urani alkuvaiheessa, pyrin varmasti kokeilemaan erilaisia työtehtäviä ja mahdollisuuksia monipuolisesti, mutta toivon, että löydän myös paremmin sen oman mielenkiinnon kohteeni ja osaamisalueeni jossain kohtaa ja pystyn sanomaan olevani todella pätevä siihen osaamiseen liittyen.

Nykyiset työtehtävät ovat olleet hyvin AWS painotteisia ja olen pitänyt siitä. Tulen jatkossa hankkimaan lisää AWS sertifikaatteja, sillä ne ovat ehkä paras ja nopein tapa kasvattaa pilvipalveluosaaamista. Niiden osalta alkuun olisi hyvä suorittaa ratkaisujen arkkitehti, tieto-insinööri, sekä koneoppimisen insinööri työtoveri/kumppani -tason sertifikaatit ja kokemuksen kasvaessa, myöhemmin suorittaa pätevän- ja erikoistumistasen sertifikaatteja. On olemassa varmasti lukuisia muitakin hyödyllisiä sertifikaatteja, joiden kautta pystyy oppimaan ja vahvistamaan osaamistaan, mutta täytyy niitä harkita tilanteen mukaisesti. Osaamista voisi kehittää myös jatkamalla ylempään ammattikorkeakouluun, jonka varmasti teen jossain vaiheessa lähitulevaisuutta.

Opinnäytetyön tekeminen auttoi hyvin hahmottamaan omaa osaamisen tasoa ja mahdollisia kehittymiskohteita, sekä herättämään paloa itsensä kehittämiseen. Alaa vaihdettua saattaa tulla aluksi pieni seinä vastaan, kun vertaa muiden alalla pidempään olleiden osaamista omaan osaamiseen ja tuntee epätoivoa, ettei koskaan tule yltämään samalle tasolle. On niin paljon kiinni otettavaa osaamisessa, että kuilu saattaa tuntua joskus hyvinkin ylitsepääsemättömältä. Ainakin itse tunsin näin aluksi. Tämä tunne kuitenkin onneksi katoaa nopeasti, kun tiedostaa nöyrästi oman osaamisensa tason ja jaksaa vain sinnikkäästi yrittää ja kehittää itseään eteenpäin.

Lähteet

Amazon a. Mikä on AWS. Luettavissa: <https://aws.amazon.com/what-is-aws/>. Luettu: 21.04.2024.

Amazon b. Mikä on AWS Codecommit. Luettavissa: <https://docs.aws.amazon.com/codecommit/latest/userguide/welcome.html>. Luettu: 21.04.2024.

Amazon c. Mitä on SQL. Luettavissa: <https://aws.amazon.com/what-is/sql/>. Luettu: 21.04.2024.

Amazon d. Mikä on AWS Lambda. Luettavissa: <https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>. Luettu: 21.04.2024.

Amazon e. REST API kehitys API Gateway:ssä. Luettavissa: <https://docs.aws.amazon.com/apigateway/latest/developerguide/rest-api-develop.html> Luettu: 28.04.2024.

Amazon f. Rest API:n tietomallit. Luettavissa: <https://docs.aws.amazon.com/apigateway/latest/developerguide/models-mappings-models.html> Luettu: 12.05.2024.

Amazon g. AWS-identiteetin ja käyttöoikeuksien hallinnan käytännöt ja käyttöoikeudet. Luettavissa: https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html Luettu: 21.05.2024.

Amazon h. Amazon Web Services kyselyt. Luettavissa: <https://aws.amazon.com/blogs/compute/from-poll-to-push-transform-apis-using-amazon-api-gateway-rest-apis-and-websockets/> Luettu: 21.05.2024.

Businesscredit. Keitä ovat yrityksen tärkeimmät sidosryhmät. Luettavissa: <https://www.businesscredit.fi/blog/keita-ovat-yrityksen-tarkeimmat-sidosryhmat>. Luettu: 21.04.2024.

Clouddevs. Virheen käsittely TypeScriptissä, parhaita käytäntöjä. Luettavissa: <https://clouddevs.com/typescript/error-handling/> Luettu: 12.05.2024.

Logrocket. Opas async/await TypeScriptissä. Luettavissa: <https://blog.logrocket.com/async-await-typescript/> Luettu: 05.05.2024.

Mozilla: Promise.all() -metodi. Luettavissa: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise/all Luettu: 05.05.2024.

Nextjs a. Mitä ovat react ja nextjs. Luettavissa: <https://nextjs.org/learn/react-foundations/what-is-react-and-nextjs> Luettu: 10.06.2024.

Nextjs b. Käyttöliittymän renderöinti. Luettavissa: <https://nextjs.org/learn/react-foundations/rendering-ui> Luettu: 10.06.2024.

Nextjs c. Reactin käytön aloittaminen. Luettavissa: <https://nextjs.org/learn/react-foundations/getting-started-with-react> Luettu: 10.06.2024.

PhoenixNAP. Linux curl komennot selitettynä esimerkein. Luettavissa: <https://phoenixnap.com/kb/curl-command> Luettu: 26.05.2024.