

Työni ohjelmoijana

Ernojussi Jääskeläinen



Tekijä(t) Ernojussi Jääskeläinen	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Työni ohjelmoijana	Sivu- ja liitesivumäärä 53 + 0
Opinnäytetyön otsikko englanniksi Working as programmer	
<p>Opinnäytetyöni on päiväkirjamuotoinen ja se kertoo ohjelmoijan arjesta kansainvälisessä teknisissä turvallisuuspalveluissa ja –ratkaisuja tuottavassa yrityksessä. Päiväkirjaa kirjoitin 12 viikon ajan 26.08.2019 ja 15.11.2019 välisenä aikana. Työskentelen yrityksessä ohjelmointikehityspuolella.</p> <p>Työympäristössäni on yhdeksän muuta henkilöä ja jokaisella meistä on oma työtoimenkuvamme. Itse olen laajalti vastuussa yrityksen tuottamasta responsiivisesta mobiililaitteilla käytettävästä verkkosovelluksesta. Käsittelen tätä sovellusta useasti päiväkirjamerkinnöissäni sekä viikkoanalyysissäni.</p> <p>Päiväkirjani koostuu päivittäisistä merkinnöistä sekä viikkoanalyysistä, joissa käsitellään tarkemmin esiin nousseita aiheita. Olen opinnäytetyöviikkojeni aikana päässyt tekemään laajan määrän erilaisia työtehtäviä, jotka ovat vaihdelleet aina yksinkertaisista ulkoasumuuksista autentikointiin asti.</p> <p>Päiväkirjaa kirjoittaessani olen käsitellyt useita aiheita merkinnöissäni sekä viikkoanalyysissäni. Koko kyseisen työskentelyn aikana Angular on ollut keskipisteessä, olen työskennellyt sen parissa lähes joka päivä. Työtehtävieni pohjalta olen saanut mahdollisuuden oppia back-end:ä sekä ymmärtänyt millä tavoin se keskusteleesimerkiksi front-end:ssä käytettävän Angular:n kanssa. Back-end puolella olen päässyt tutustumaan erilaisiin autentikointitapahtumiin ja siihen liittyviin aihepiireihin. Pääosassa tällä alueella on ollut SSO (kerta kirjautuminen), SLO (kertauloskirjautuminen), SAML2 xml-protokolla ja salasana sekä niiden salausmenetelmät.</p> <p>Opinnäytetyöni aikana olen hyödyntänyt opiskelujeni aikana oppimiani asioita. Tällaisia on muun muassa tietokannat sekä niiden pyynnöt ja kyselyt. Viimeisillä viikoilla aloitin työskentelemään erään uuden projektin parissa. Olen siellä hyödyntänyt aiemmin oppimiani projektinhallintataitoja sekä laatinut projektin aloituksessa tarvittavia dokumentteja.</p> <p>Päiväkirjan kirjoittamisen aikana olen saanut tilaisuuden tutustua useisiin aiheisiin syvällisemmin. Tämän myötä olen oppinut monesta aiheesta paljon uutta, ja se on antanut minulle mahdollisuuden kehittää itseäni ammatillisesti. Tulen tulevaisuudessa varmasti hyödyntämään sekä tämän opinnäytetyön aikana oppimiani asioita että artikkeleita ja dokumentaatiota.</p>	
Asiasanat ohjelmointi, front-end, back-end, tietokanta, C#	

Sisällys

1	Johdanto	1
2	Lähtötilanteen kuvaus	3
2.1	Oman nykyisen työn analyysi.....	3
2.2	Sidosryhmät työpaikalla	5
2.3	Vuorovaikutustaidot työpaikalla	6
3	Päiväkirjaraportointi.....	7
3.1	Seurantaviikko 1	7
3.2	Seurantaviikko 2	12
3.3	Seurantaviikko 3	16
3.4	Seurantaviikko 4	21
3.5	Seurantaviikko 5	25
3.6	Seurantaviikko 6	29
3.7	Seurantaviikko 7	33
3.8	Seurantaviikko 8	37
3.9	Seurantaviikko 9	41
3.10	Seurantaviikko 10	45
4	Pohdinta ja päätelmät.....	49
	Lähteet	51

1 Johdanto

Työskentelen yrityksessä X ohjelmoijana. Aloitin työskentelyn harjoittelun muodossa 07.01.2019 ja tällä hetkellä minulla on vakituinen työsopimus. Opinnäytetyöni on päiväkirjamuotoinen, tulen kirjoittamaan 12 viikon ajan päiväkirjaa töistäni. Ensimmäinen viikko pitää sisällään yleiskatsauksen tämän hetkisestä tilanteestani. Tämän jälkeen kirjoitan kymmenen viikon ajan päivittäin merkintöjä ja viikon lopuksi viikoittaisen analyysin. Viimeinen päiväkirjaviikko pitää sisällään lopulliset pohdinnat sekä analyysin kehittymisestääni päiväkirjamerkintöjen ajalta. Aloitan päiväkirjan kirjoittamisen 26.08.2019 ja viimeinen päiväkirjamerkintä tulee 15.11.2019.

Yritys on turvallisuuteen keskittyvä yritys ja se on osa globaalisti tunnettua konsernia, jolla on 60 eri maassa toimintaa. Näistä 14 maata sijaitsee Euroopassa. Suomessa on useita toimipisteitä usealla eri paikkakunnalla. Oma toimipisteeni sijaitsee Helsingissä. Yritys tuottaa monia erilaisia tuotteita sekä sovelluksia turvallisuuden eri alueisiin.

Työni on hyvin ohjelmointipainotteista. Tarvitsen työssäni tietoa sekä taitoa useasti eri aihealueesta. Käytössäni tällä hetkellä ohjelmointikielistä on Angular, TypeScript sekä C#. Olen työssäni paljon tekemisissä rajapintojen sekä tietokantojen ja tietokantakyselyiden kanssa. Viime aikoina olen tutustunut erilaisiin autentikointikirjastoihin sekä olen ollut tekemisissä Microsoftin Active Directoryn kanssa.

Työskennellessäni olen hyödyntänyt erilaista ammattikirjallisuutta ja ohjelmointikielten sekä kirjastojen dokumentaatioita. Dokumentaatioista käytössäni on ollut muun muassa C# ja Angularin omat dokumentaatiosivut sekä kirjastojen puolesta SustainSys.Saml2 GitHubista löytyvä dokumentaatio. Kirjallisuuden puolesta Angular:ssa olen hyödyntänyt Josephin (Joseph D. Booth 2019) Angular Succinctly-kirjaa sekä Angular 2+ Notes For Professionals (Goalkicker 2018). C#:ssa käytössäni on ollut Svetlinin (Svetlin Nakov and Team 2013) kirjoittama kirja Fundamentals of Computer Programming with C#.

Keskeisimmät ammattikäsitteet työssäni ovat:

Back-end	Palvelinpuoli. Koodi, jota ajetaan palvelimella
Front-end	Selainpuoli. Koodia, jota käytetään selaimen puolella
C#	Microsoftin kehittämä ohjelmointikieli
HTML5	Ohjelmointikieli, jota käytetään verkkosivuissa
CSS	Tekniikka, jolla määritetään tyyliä HTML-sivuille
TypeScript	Ohjelmointikieli, joka pohjautuu JavaScriptiin
Angular	Googlen ylläpitämä avoimen lähdekoodin JavaScript-ohjelmistokehys
JSON	JavaScript Object Notation. Tiedostomuoto tiedonvälitykseen
Rajapinta	Ohjelmat pystyvät keskustelemaan ja siirtämään tietoja tämän kautta
SQL	Structured Query Language. Tietokantakieli, jolla voidaan mm. tehdä tietokantahakuja ja -päivityksiä.

2 Lähtötilanteen kuvaus

2.1 Oman nykyisen työn analyysi

Työtoimenkuvani koostuu monesta erilaisesta tehtävästä, joista luettelo alta (Taulukko 1).

Taulukko 1. Työtehtävät ja niiden painoarvo

Työtehtävien ajankulutus (100%)	Työtehtävä
75%	Uuden palvelun luominen
5%	Toimintojen lisäys vanhaan palveluun
10%	Vanhan palvelun toimintojen muokkaus
10%	Erilaisien ongelmien tutkiminen ja korjaus

Työtehtävissäni keskeisessä osassa on ohjelmointi. Tällä hetkellä suurin osa ajastani kuluu uuden palvelun luomisessa. Kyseisen palvelun tarkoituksena on toimia niin sanottuna portaalina muille yrityksen tuotteille. Toimintaperiaatteena portaalin kautta käyttäjät suorittavat sisäänkirjautumisen ja pääsevät käyttämään kaikkia heille hankittuja sekä tarkoitettuja tuotteita ilman erillistä kirjautumista.

Muut luettelossa listaamani työtehtävät vievät vain pienen osan nykyisestä työajastani. Viime aikoina ainoa laajempi ominaisuus, minkä olemme lisänneet yrityksen tuotteeseen, on ollut harjoitteluni aikana luomani mobiililounasleimaus-ominaisuus. Kyseiseen ominaisuuteen olen tehnyt muutamia korjauksia sekä muutoksia. Jokainen työtehtävä on kuitenkin ainutlaatuinen ja pitää sisällään erilaisia kokonaisuuksia. Suurin osa työajastani on kuitenkin puhtaasti front-end- sekä back-end-ohjelmoinnin parissa. Front-end:ssä työskenteleminen aina nettisivujen ulkoasusta back-end puolen pyyntöihin asti ja siitä eteenpäin rajapintoihin sekä tietokantakyselyihin.

Työtehtäväni ovat monipuolisia ja ne vaativat paljon monenlaista osaamista eri aihealueista. Front-end käsitellessä tarvitaan tietoa muun muassa nettisivujen rakenteesta, niiden tyyllityistä ja näiden kummankin muodostamasta kokonaisuudesta. Käytössäni front-end puolella on Angular, joka on TypeScript- kielellä tehty avoimen lähdekoodin web-sovelluskehys. Back-end tehtävissä tarvittavaan osaamista sekä ymmärrystä C#-ohjelmointikielstä ja erilaisista kirjastoista kuten SustainSys.Saml2 sekä ohjelmointirajapinnoista. Kirjastot ovat erilaisia ohjelmia tai luokkia, jotka sisältävät funktioita, joita hyödynnetään muissa ohjelmissa. Ohjelmointirajapinnat tarkoittavat tasoja, joiden kautta eri ohjelmat pystyvät kommunikoimaan toistensa kanssa.

Aloittaessani työharjoittelun yrityksessä X, osaamiseni oli pitkälti vain koulun kurssien myötä saamani tietoa. Harjoittelun sekä kesätyön aikana olen päässyt oppimaan laajemmin ohjelmoinnista yleisellä tasolla. Olen saanut mahdollisuuden syventyä moneen eri ohjelmointikieliin sekä ymmärtämään miten rakennetaan sovelluksia loppukäyttäjälle, ja sovelluksen taustan tapahtumiin. Päivittäinen työni vaatii suurelta osin uusien asioiden oppimista, sillä olen kuitenkin vasta nuori ohjelmoija vähäisellä kokemuksella suuressa sekä kehittyvässä ohjelmoinnin maailmassa. Minulta kuten muiltakin aloittelevilta ohjelmoijilta vaaditaan hyvää keskittymiskykyä, ongelmanratkaisutaitoja, oma-aloitteellisuutta sekä vuorovaikutustaitoja.

Mielestäni olen kehittynyt alkuvuoden harjoittelujakson sekä kesätyön myötä merkittävästi. Olen saanut työpaikkani puolesta mahdollisuuden valita itse suurimman osan työtehtävistäni, näin ollen olen saanut mahdollisuuden päästä aloittamaan uuden portaalipalvelun luomista. Tämänkaltaisen työtehtävän antaa mahdollisuuden oppia uusia asioita sekä päästä toteuttamaan itseään mahdollisimman laaja-alaisesti. Mielestäni olen suoriutunut työtehtävistäni tähän mennessä kiitettävästi. Olen muun muassa luonut uuden ominaisuuden, joka on tällä hetkellä yrityksen X suurella asiakkaalla käytössä.

Ammatillisen tasoni pohja on rakentunut usean vuoden opiskelujen myötä. Koulussa olen saanut mahdollisuuden päästä opettelemaan uusia taitoja erilaisten harjoitusten sekä tehtävien avulla. Siirtyessäni yritykseen X töihin pääsin hyödyntämään koulussa oppimiani taitoja sekä oppimaan uusia piirteitä monesta jo osaamastani taidosta. Kehittymistä on kuitenkin eniten tapahtunut töissä ollessani, sillä siellä pääsee työstämään käyttöön tarkoitettuja tuotteita. Tuotteiden kehitys nostaa motivaatiotasoa tehdä kunnollista tulosta, sekä tuottaa käyttäjäystävällisiä toimintoja loppukäyttäjälle. Kehittymisen myötä olen myös päässyt etenemään alkutason töistä huomattavasti monimutkaisimpiin kokonaisuuksiin.

Oletan kehittymisen olleen sen kaltaista, että allekirjoitin vakituisen työsopimuksen. Ammatillinen kehitykseni näkyy toiminnassani muun muassa siinä, että minulla on mahdollisuus kehittää itsenäisesti yrityksen tarjoamia tuotteita sekä luoda uusia palveluita. Oppimista jatkuu päivittäin, sillä jokainen päivä on erilainen ja haasteellinen. Parhain tapa itselleni oppia uusia asioita sekä kehittyä ohjelmoijana on erilaisten erehdysten sekä ongelmatilanteiden ratkaiseminen. Lisäksi pelkällä työnteolla on suuri merkitys kehittymisen kanssa sekä sen myötä itsevarmuus omaan työntekoon kasvaa todella kattavasti. Olen saanut mahdollisuuden päästä hyödyntämään tämän hetkistä taitotasoa uusia harjoittelijoiden kanssa, avustamaan sekä opettamaan heille joitakin käyttämiemme kielten perusteita. Olemme yhdessä päässeet oppimaan paljon uusia piirteitä tämän hetkisistä ohjelmointikielistämme. Kaiken tämän toiminnan kautta olen pystynyt ymmärtämään, että

vaikka kaikki ei aina heti onnistuisikaan niin se ei ole kuitenkaan maailmanloppu, sillä jokaiseen ongelmaan pystytään löytämään yksinkertainen ratkaisu. Lisäksi olen ymmärtänyt, että oman työni taustalla on isompi kokonaisuus eli syntyy halu luoda uusia sekä monipuolisia asioita, jotta näkee oman työnsä kehittymisen.

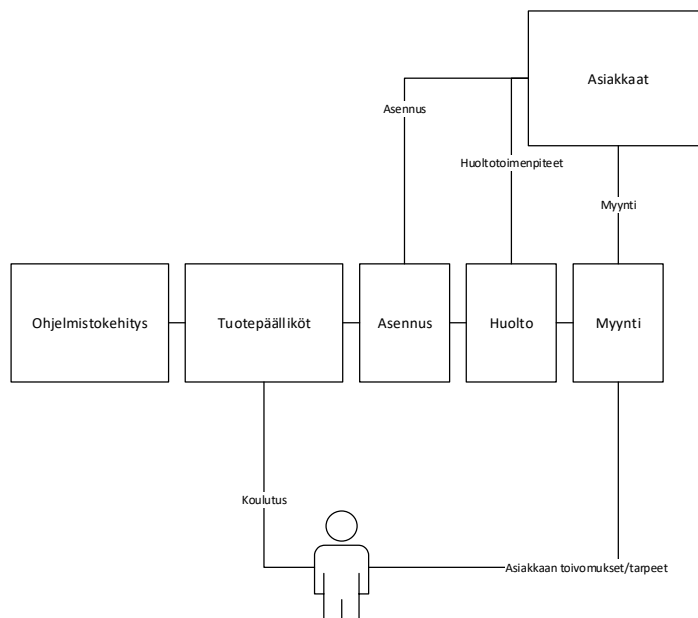
Ohjelmoinnin maailmassa oppiminen ei kuitenkaan koskaan tule loppumaan, sillä ohjelmointi kehittyy koko ajan. Ohjelmoinnissa käytettävät kirjastot sekä ohjelmointikieliet muuttuvat. Syntyy uusia ohjelmointikieliä, joiden rakenne on aivan poikkeavaa muihin kieliin nähden sekä lisäksi monet erilaiset ratkaisutavat saattavat muuttua vuosien varrella. Henkilökohtaisesti minulla on vielä matkaa edessäpäin ohjelmoijana, sillä kuten mainitsinkin, niin ohjelmointi kehittyy koko ajan. Ohjelmoinnin kehityksen myötä minun pitää oppia uusia asioita ja käyttämään uudempia tekniikoita sekä uudenlaisia tapoja.

2.2 Sidosryhmät työpaikalla

Työhöni vaikuttavat sidosryhmät ovat suurimmaksi osin sisäisiä sidosryhmiä. Sisäisten sidosryhmien kautta asiakkaat, jotka kuuluvat ulkoiseen sidosryhmään voivat mahdollisuuksien mukaan vaikuttaa työni suuntaan. Sidosryhmät, jotka ovat vaikutuksissani työhöni ovat:

ohjelmistokehitys, tuotepäälliköt, huolto, asennus sekä myynti.

Sidosryhmäkaavio



Ylläolevasta taulukosta nähdään, että työhöni vaikuttavat useat sidosryhmät. Jokaisella sidosryhmällä on omat vaikutusalueensa. Ohjelmistokehitys vaikuttaa työtoimenkuvaani

eniten, ja tavoitteena on kehittää ominaisuuksia, jotka helpottavat ohjelmiston käyttöönottoa, käyttöä sekä päivittämistä. Tuotepäälliköt vastaavat henkilökunnan koulutuksista, ja ohjelmiston kehitys muuttaa heidän osaamistarpeitaan. Huollon sekä asennuksen toimintaan vaikuttaa suureksi osin kehityksen suunta, sillä sen yhtenä osana on tuotanto-organisaation työn helpottaminen. Lopuksi myynnin sidosryhmä antaa kehitykselle suunnan, sen puolesta mitä toiveita sekä tarpeita tulee asiakkaiden suunnasta. Myynnin palautteet ovat yksi tärkeimmistä asioista kehityksen puolesta.

2.3 Vuorovaikutustaidot työpaikalla

Normaalin työpäivän aikana vuorovaikutustilanteita tulee useasti, yleensä tilanteet ovat kehityksestä keskustelua sekä mahdollisten ongelmatilanteiden ratkaisemista. Muita vuorovaikutustilanteita ovat erilaiset palaverit, joissa pääasiallisena sisältönä on keskustella muiden ohjelmoijien kanssa tämänhetkisestä tilanteesta. Palavereissa käydään läpi edellisen palaverin jälkeen tehtyjä töitä sekä keskustellaan, mitä tulevia tehtäviä olisi edessä seuraavien viikkojen tai kuukausien ajaksi.

Tällä hetkellä en ole paljoa tekemisissä asiakaspalvelun tai erilaisten sidosryhmien kanssa. Suurimmaksi osin sidosryhmät keskustelut sijoittuvat ohjelmistokehityssidosryhmän sisälle, ja muiden sidosryhmien asiat kulkevat vanhempien kollegojen kautta minulle. Ainoina haasteina vuorovaikutustilanteissa on vielä vähäinen kokemukseni ja tietotaitoni monesta eri aiheesta, sillä en välttämättä pysty heti keskustelun ohessa ymmärtämään keskusteltavaa asiaa tai kehitysehdotusta. Tämä kuitenkin tulee kehittymään kokemuksen myötä.

3 Päiväkirjaraportointi

3.1 Seurantaviikko 1

Maanantai 02.09.2019

Työpäivän alussa:

Maanantain työtehtäviin kuuluu suurimmaksi osin mobiililounasleimaus-ominaisuuden muutoksen tekoa. Tällä hetkellä ravintoloiden nimet ovat pakotettuja mahtumaan 16 merkkiin, ja nimet sekä tiedot haetaan rajapinnan kautta toisesta ohjelmasta. Tavoitteena on mahdollistaa nimen muutos pidemmäksi hyödyntämällä ravintolan yksilöivää tunnusta.

Työpäivän lopussa:

Päivälle suunniteltu tehtävä onnistui ja sitä tehdessä tuli kohdattua useita ongelmia. Aloitin tehtävän tekemällä SQL Server -tietokantaan uuden taulun ravintolan tietoja varten. Taulu koostuu rivin yksilöintitunnuksesta sekä ravintolan yksilöintitunnuksesta ja nimikentästä. Taulun luonnin jälkeen loin funktion ravintolataulun tietoja varten, jotta pystyn käyttämään sitä monessa eri kohdassa kuten, kaikkien ravintoloiden haussa sekä kuiteissa. Funktion tarkoituksena on muokata ravintolan nimi vastamaan tietokannassa olevaa nimeä, jos tauluun on syötetty ravintolan yksilöivätunnus siihen kuuluvaan kenttään. Kyseistä funktiota kutsuin jokaisessa lounasleimaus-metodissa, ja oman testauksen jälkeen se näyttäisi toimivan suunnitellusti. Päivän aikana tuli opittua hieman lisää, millä tavoin pystytään hyödyntämään kerran tehtyä funktiota useammassa paikassa.

Tiistai 03.09.2019

Työpäivän alussa:

Tämän päivän työtehtäviin lukeutuu portaalisovelluksen kehittämistä, jonka pohjana käytetään toisen sovelluksen rakennetta. Tavoitteena tälle päivälle on pystyä määrittelemään, mitkä sovellukset näkyvät tietyille henkilölle portaalin valikossa.

Työpäivän lopussa:

Työtehtävä osoittautui yllättävän monimutkaiseksi, sillä en ollut aivan tietoinen, miten tällä hetkellä muodostetaan sovelluksessa käytettävät oikeudet. En saanut tehtävääni tänään loppuun asti, vaan suurin osa ajasta meni koodin tutkimisessa ja ymmärtämisessä. Olen nyt tietoinen, siitä miten sovelluksessa määritellään oikeudet ja millä tavoin niitä hyödynnetään monessa eri osa-alueessa. Tiedot haetaan kirjautumisen aikana tokeniin, josta ne puretaan ja asetetaan selaimen muistiin. Päivän aikana opin nykyisen projektini pohjana

käytettävästä sovelluksesta ja siitä millä tavoin autentikoinnissa haetaan tietyt tiedot ja miten niitä hyödynnetään sisäänkirjautumisen jälkeen.

Keskiviikko 04.09.2019

Työpäivän alussa:

Tänään tarkoituksena on jatkaa, siitä mihin eilen jäin tehtävässäni eli pystyä määrittelemään henkilökohtaisesti, mitkä sovellukset näkyvät käyttäjälle. Tavoitteeksi asetan sovel-lusmäärittelyn, lisäksi koitan tänään aloittaa henkilökohtaisten roolien määrittelyä eri appli-kaatioihin. Tällä hetkellä jokainen sovellus käyttää portaalisovelluksen rooleja, vaikka niille on luotu roolinmäärittely-tauluun omat rivinsä, jotka viiteavaimen avulla viittaavat tiettyyn sovellukseen.

Työpäivän lopussa:

Sain tänään tehtyä tavoitellun tehtävän, henkilölle näkyy nyt vain hänelle kuuluvat appli-kaatiot. En kuitenkaan hyödyntänyt tässä eilen ymmärtämäni tapaa käyttää selaimen muistissa olevaa dataa. Päädyin ratkaisemaan tämän ongelman käyttämällä tietokan-nassa olevaa taulua, jonka avulla määritellään rooleja sovellukseen. Tein henkilön kirjau-duttua sisään tarkastuksen, että löytyykö henkilön yksilöivää tunnusta roolien määrittely-taulusta. Tutkin, löytyykö riveistä määriteltynä rooleja applikaatioon. Jos niitä ei ollut niin, kyseistä applikaatiota ei näytetä sivulla.

Torstai 05.09.2019

Työpäivän alussa:

Tarkoituksena tälle päivälle on aloittaa roolien määrittely eri sovelluksiin portaaliprojek-tissa, eli pystytään yhdessä näkyvässä määrittelemään henkilölle hänelle kuuluvat roolit tietyissä sovelluksessa. Tavoitteena on pystyä listamaan kaikki henkilöt, sekä hakea sovel-lukseen kuuluvat roolit checkboxeiksi, joita klikkaamalla pystytään asettamaan halutut roo-lit.

Työpäivän lopussa:

Sain tehtävän päivän aikana puoleenväliin. Pystyy hakemaan kaikki käyttäjät listaksi si-vulle, sekä tuomaan oikea määrän valintaruutuja eli checkboxseja näkyville riippuen siitä, mihin sovellukseen on tarkoitus muuttaa henkilöiden rooleja. Esimerkkinä, jos sovelluk-sessa on kolme erilaista roolia, niin sivulle tulostuu käyttäjän nimi vasempaan reunaan ja nimen viereen tulostuu klikattavat checkboxit eri sarakkeisiin. Jokaisen sarakkeen otsik-ona on roolin nimi.

Perjantai 06.09.2019

Työpäivän alussa:

Tämäkin päivä tulee jatkumaan sovellusten roolien parissa. Toivon tänään pääseväni siihen pisteeseen, että pystyn klikkaamaan henkilöille rooleja ja kyseiset valinnat tallentuisivat tietokantaan asti. En ole kuitenkaan aivan varma, pääsenkö tähän pisteeseen sillä, eilen loppupäivästä huomasin, että joudun muuttamaan tämän hetkisiä tietokantahakuja sekä html-tiedostoa ja siellä olevia Angularin direktiivejä sekä määreitä.

Työpäivän lopussa:

Päivä meni suunnitellusti roolien parissa. Tietokantaan tallennus jäi vielä keskeneräiseksi. Tallenna-nappia pystytään painamaan, mutta jossain kohtaa koodia data saa arvokseen "null" eli tyhjä ja tästä syystä tietokannan taulusta poistuu henkilölle asetetut roolit. Päivä oli kuitenkin onnistunut, sillä opin uusia puolia Angular:sta. Ymmärsin, miten pystyn hyödyntämään vielä enemmän Angularin omia direktiivejä sekä määreitä. Käytin roolien määrittelyssä esimerkkinä vanhan sovelluksen rakennetta, jossa jokaiselle roolille asetettu html:n puolella omat kohtansa. Hyödynsin rakenteen ideaa ja tein siitä paljon joustavamman hyödyntämällä Angularia. Nykyään käytän Angularin ngFor-direktiiviä, jolla käydään tietokannasta haetut roolit Array:sta läpi ja jokaiselle roolille tehdään samat html-tagit sekä asetellut (Kuva 1). Opin myös päivän aikana, että voin käyttää ngFor:n sisällä toista ngFor-direktiiviä. Toisen ngFor-direktiivin sisällä olevalla attribuutilla, jossa on käytössä ngModel-direktiiviä, voidaan syöttää arvo, joka koostuu kummankin ngFor:n arvoista.

Kuva 1. Esimerkki *NgFor:sta

```
<div class="col">
  <span *ngFor="let x of RolesOfApplication;">
    <span> {{x.roleName}}</span>
  </span>
</div>
```

Viikkoanalyysi:

Viikko oli mielenkiintoinen ja opin uusia asioita Angular:sta sekä siitä, millä tavoin sitä pystytään hyödyntämään monipuolisesti. Opin uudelleen käytettävien funktioiden merkityksen. Tehdessäni lounasleimaukselle ravintolan-nimi muutoksia huomasin, kuinka hyödyllistä, on tehdä funktioita, joita pystytään käyttämään useassa eri paikassa. Uudelleen käytettävät funktiot vähentävät turhaa toistoa koodissa, sekä mahdollisissa muutostilanteissa ei tarvitse muuttaa kuin yhdestä kohtaa koodia ja sen muutokset tulevat voimaan kaikkiin niihin paikkoihin, missä funktiota hyödynnetään.

Viikon aikana tuli kohdattua monia erilaisia ongelmia. Ongelmia oli sekä front-end:ssä sekä back-end:ssä. Front-end puolen ongelmat koostuivat enimmäkseen Angular:sta sekä sen omista toiminnoista. Back-end:n puolella haasteet ja ongelmatilanteet sijoituivat tietokantapyyntöihin ja sieltä tulevien vastausten muotoiluun. Kaiken kaikkiaan viikko oli erittäin monipuolinen ja töiden pohjalta kehittymiseni kasvoi, kun jouduin ratkaisemaan erilaisia ongelmia sekä kehittämään uutta koodia.

Viikon mielenkiintoisin ongelmatilanne oli mielestäni roolien parissa. Asia ratkesi, kun hyödynsin Angular:ssa olevia direktiivejä. En ollut kuitenkaan tietoinen, että direktiivejä pystytään asettamaan toistensa sisään sillä tavoin kuin itse tein sen. Käytin Angular:n direktiiveistä hyödynseni ngForia sekä ngModelia.

Angular on itsessään ohjelmistokehys, joka hyödyntää erilaisia attribuutteja, määreitä sekä direktiivejä toiminnoissaan. Angular käyttää pääasiallisesti ohjelmointikielensä TypeScriptiä, joka tyyliään paljon JavaScriptiin viittaavaa. TypeScript:ssä pystytään asettamaan luokkia, rajapintoja sekä moduuleita kuten muissakin ohjelmointikielissä (Shah 2017). Kuten aiemmin mainitsinkin Angular:ssa on käytössä erilaisia omia direktiivejä, näistä yleisimpiin käytössä oleviin kuuluu ngFor sekä ngModel.

Goalkicker:ssä (Goalkicker 2018) kuvaillaan ngFor:ia direktiiviksi, jonka avulla pystytään jokin tietty asia toistamaan niin monta kertaa kuin se esiintyy ngFor:n ehdossa. NgFor:n yhteydessä pystytään myös käyttämään muita asetuksia, jotka ovat lueteltuna Goalkickeri:ssä luvussa 12. Yksi yleisemmistä asetuksista on index, joka asettaa jokaiselle toistettavalle asialla numeerisen arvon alkaen numerosta 0. Toinen mainitsemani direktiivi oli ngModel, jonka ideana on mahdollistaa datan liikkuminen Angular:n html-tiedoston ja TypeScript tiedoston välillä. NgModel:lla voidaan tehdä niin sanottuja binding-tapahtumia, jotka ovat joko yksinsuuntaisia, jolloin dataa voidaan asettaa vain yhteen suuntaan joko html-tiedossa olevaan attribuuttiin tai sieltä pois. NgModel:n ollessa kaksisuuntainen

voidaan asettaa html:ään dataa tai vastaanottaa sitä sieltä TypeScript-tiedoston puolelle. NgModel:n muoto riippuu siitä kumpaa tapaa käytetään, jos käytössä on yksinsuuntainen silloin ngModel:n muoto on "[ngModel]" ja jos kaksisuuntainen silloin "[(ngModel)]" (Jose Carlos 2016).

Aiemmin mainitsin ongelmastani, jonka ratkaisin edellä kirjoittamieni direktiivien avulla. Aluksi käytin ngForia, jonka avulla tulostin näkyville kaikki applikaation roolit sekä checkboksit. Tämän jälkeen hyödynsin uudestaan samaa direktiiviä ja tein uuden tulostuksen, jonka ideana oli tuoda tieto siitä, mitkä checkboksit tulisi ruksata kenenkin henkilön kohdalla. Yhdistyn näistä kahdesta ngFor:sta tulevia tietoja toisen ngFor:n sisällä olevan attribuutin ngModeliin. Muunneltu esimerkki kyseistä koodista löytyy alapuolella olevasta kuvasta (Kuva 2).

Kuva 2. Esimerkki useasta *NgFor:sta

```
<tr *ngFor="let x of arrayOfPeople;">
  <td>
    <div class="col">
      <label>
        <a [textContent]="x.Name">/a>
      </label>
    </div>

    <div class="col">
      <label *ngFor="let a of ArrayOfApplicationRoles;">
        <input type="checkbox" [(ngModel)]="x[a.RoleName]"/>
      </label>
    </div>
  </td>
</tr>
```

3.2 Seurantaviikko 2

Maanantai 09.09.2019

Työpäivän alussa:

Aikomuksenani on jatkaa roolisivuston kehittämistä. Aion saada tallennuksen toimimaan, sekä mahdollisesti lajiteltua listan roolien mukaan. Määrittelen, että lista muuttuu sen mukaan, mistä roolista painetaan. Henkilöt, joilla on kyseinen rooli aktiivisena, tulevat listalla ylimmäksi.

Työpäivän lopussa:

Työpäivä ei sujunut aivan ongelmitta, sillä huomasin pienen ongelman roolien kanssa. Jos roolinnimitys sisälsi toisen roolin nimityksen sovellus oletti, että henkilöllä on kummatkin roolit käytössä. On olemassa roolit x sekä xy ja henkilöllä on rooli xy, silloin hänellä olisi myös rooli x. Kyseissä ongelmassa meni koko päivä, sillä roolien purkaminen ja tarkastelu tapahtui useassa eri paikassa. Ratkaisin ongelman tekemällä tarkistuksen, jolla katsotaan kokonaisuudessaan annetut roolit henkilölle ja varmistetaan, että jokainen rooli oli oikein kirjain kirjaimelta.

Tiistai 10.09.2019

Työpäivän alussa:

Tarkoituksena on päästä jatkamaan roolien tallennusta, sekä sen myötä mahdollistaa roolien määrittely henkilölle, kun siirrytään toiseen applikaatioon. Päivän aikana täytyy myös hieman lukea ja opiskella, miten token ja selaimen muisti toimivat, kun tehdään siirtymisiä applikaatioiden välillä ja miten niiden tietoja voidaan muuttaa.

Työpäivän lopussa:

Roolien tallennus jäi tänäänkin vielä tekemättä, sillä aloitin heti aamusta token:sta ja selaimen muistista lukemisen. Pienen opiskelu hetken jälkeen testasin hieman niiden toimintoja, sekä tutkin miten kokonaisuudessaan niiden tiedot muodostuvat sekä miten sisäänkirjautumisen yhteydessä tiedetään, minne sisäänkirjautumisen tiedot viedään. Lopulta ymmärsin, että tietojen lähetyksessä hyödynnetään niin kutsuttuja grant-typeja eli metodeja, jonka avulla pystytään antamaan käyttöoikeuksia. Näiden myötä osataan myös ohjata tiedot oikeaan päätepisteeseen eli end-pointtiin.

Keskiviikko 11.09.2019

Työpäivän alussa:

Tänään yritän saada tehtyä applikaatioon siirtymisen, siten että tokenin tiedot muuttuvat ja muuttuneet tiedot puretaan selaimen muistiin käytettäväksi applikaatiolle. Ideana on, että tämän myötä henkilö saa hänelle kuuluvat roolit applikaatioon ja hän pystyy tekemään hänelle kuuluvia toimintoja applikaatiossa.

Työpäivän lopussa:

Päivä menikin suurimmaksi osin palaverissa vanhemman kollegani kanssa, yhdessä tutkimme tämän hetkistä tilannettani portaaliprojektin kanssa. Tutkimme läpi viimeisen kauden aikana tekemiäni uusia koodeja sekä vanhan koodin muutoksia. Keskustelun myötä pääsin myös tiedustelemaan hieman, millä tavoin vanhassa projektissa toteutetaan autentikointi sekä tokenin prosessi kokonaisuudessaan. Palaverin pohjalta sain hyvän yleiskuvan siitä, missä tilanteessa olen tällä hetkellä sekä miten tulen jatkamaan projektin kanssa. Tutkimisen aikana tuli tehtyä myös ideointia, mitä kaikkea lisää voisin toteuttaa portaaliiin. Yksi näistä ominaisuuksista oli käyttöoikeuksien määrä johonkin applikaatioon eli määritellään maksimi käyttäjämäärä applikaatiolle.

Torstai 12.09.2019

Työpäivän alussa:

Tämän päivän tarkoituksena olisi saada valmiiksi eilen aloittamani tehtävä ja hyödyntää tehtävän teossa eilen saamiani tietoja kollegani keskustelun pohjalta. Oletan, että saan tehtävän tänään vihdoinkin valmiiksi ja pääsen sitten huomenna pohtimaan uusia ongelmia.

Työpäivän lopussa:

Sain tänään tehtyä tokenin tietojen muuttamisen. Tehtävä oli yllättävän monimutkainen, sillä vaikka keskustelinkin eilen autentikointipuolesta, silti sen kanssa tuli kohdattua monia erilaisia ongelmatilanteita, joiden pohjalta jouduin kyselemään vielä hieman lisätietoa kollegalta. Päivän aikana löysin uusia puolia autentikoinnista sekä sen toteutustavoista. Mielinkiintoisinta oli ymmärtää, miten sovelluksessa kutsutaan back-end:n autentikointia sekä mitä tapahtuu silloin, kun token vanhenee ja millä tavoin tämä tilanne hallitaan back-end:ssä.

Perjantai 13.09.2019

Työpäivän alussa:

Töihin saavuttuani vanhempi kollegani mainitsi, että tarvitsisi tutkia salasanaikäytäntöjä yleisesti sekä pohtia millä tavoin voitaisiin mahdollisesti asettaa salasanoja tietokantaan kirjautumista varten. Näillä näkymin päiväni tulee kulumaan pitkälti salasanojen sekä salausten parissa, sillä tämä aihe on itselleni todella tuntematonta. Toivon, että päivän aikana saisin hyvän yleiskäsityksen siitä, millaisia salauksia on olemassa ja miten ne eroavat toisistaan sekä mikä niistä on hyödyllisin.

Työpäivän lopussa:

Päivä meni kokonaisuudessaan, miten sen ajattelinkin menevän, opiskelussa ja erilaisissa pienissä testauksissa. Testauksissa pitkälti katsoin, millä tavoin salasanoista muodostetaan hash hyödyntäen hajautusalgoritmeja. Hash on tulos, mikä saavutetaan, kun käytetään jotakin hajautusalgoritmia eli esimerkiksi sanasta "testi" voidaan algoritmia käyttäen muodostaa numero- ja kirjainyhdistelmä kuten "F1E1C6EA766397606475AB41D7F124258DA887B9", joka periaatteessa tarkoittaa samaa asiaa kuin sana "testi". Mainitsemani numero- sekä kirjainyhdistelmä tehtiin SHA-1 hajautusalgoritmia hyödyntäen.

Viikkoanalyysi:

Tämä viikko oli todella moninainen ja tein paljon erilaisia asioita, vaikka viikon aluksi ajattelin, että suurin työstäni olisi vain roolitukseen ja applikaatioon siirtymisen toteuttamista. Viikon aikana tuli kohdattua monenlaisia haasteista, joista eniten mieleeni jäi haasteet tokenin kanssa. Haasteiden ja ongelmantilanteiden pohjalta pääsin sisäistämään paremmin mikä token on, miten sitä käytetään sekä miksi se on käytössä.

Toinen kiintoisa puoli tässä viikossa oli, että sain mahdollisuuden työskennellä salasanojen sekä sen salauksen parissa. Kyseinen aihe oli itselleni aivan tuntematon, enkä tiennyt aluksi yhtään, mistä edes lähteä liikkeelle. Olen kuitenkin nyt ollut useita kuukausia töissä yritys X:llä ja on hienoa huomata, että minulle annetaan uudenlaisia tehtäviä, joiden parissa en ole koskaan ennen päässyt työskentelemään ja näiden tehtävien pohjalta pääsen laajentamaan ammatillista osaamistani uusille tasoille.

Perjantaina minulle annetun tehtävän puolesta, pääsin opiskelemaan sekä tutkimaan kaikkia erilaisia salasanaikäytäntöjä sekä salasanan hajautustekniikoita, joita kutsutaan hajautusalgoritmeiksi. Niiden ideana on muuttaa annettu merkkijono eli esimerkki salasana

salattuun muotoon, joka koostuu erilaisista merkeistä, kuten numeroista, kirjaimista sekä erikoismerkeistä. Tarkempi muoto lopulliselle tulokselle määräytyy sen mukaan mikä algoritmi on käytössä. Tutkiessani salaustapoja löysin Stack Overflow-nimisestä foorumista keskustelun siitä, mikä salaustapa on parasta käyttää, kun halutaan salata salasanoja. Keskustelun pohjalta sain perustiedot, siitä minkälaisia salaustapoja on olemassa ja millä tavoin mitäkin niistä tulee käyttää.

Salaustavoista on olemassa niin sanotusti nopeita salauksia sekä hitaita salauksia. Artikkelien sekä keskustelun pohjalta ymmärsin, että salasanojen kanssa tekemisissä ollessa tulisi aina välttää nopeita salauksia, vaikka nopeat salauksetkin pitäisivät sisällään salt:n eli suolan, joka on periaatteessa lisämerkkijono jo olevan merkkijonon mukaan. Nopeat salaukset ovat yksinkertaisempia merkkijonoa ja niitä on tarkoitus käyttää, kun halutaan salata jokin tieto todella yksinkertaisesti sekä mahdollisimman aikaa säästävästi (Dan Arias 25.04 2018). Eräs nopean salauksen tavoista on SHA-1, joka muodostaa sanasta pelkästään numero- sekä kirjainsarjan. Hitaiden salausmenetelmien ideana on, että niiden käytettäessä merkkijono yleensä lasketaan useaan kertaan sekä laskemisen aikana merkkijonolle annetaan suola. Suolan sekä uudelleenlaskennan myötä salasanasta tulee todella luotettava ja sitä on todella vaikea murtaa hakkerin toimesta. Hitaita salauksia on muuan muassa Argon2, Bcrypt.

Salasanojen salauksena on mahdollista käyttää mitä tahansa äsken mainitsemaani hidasta salausalgoritmia. Näiden kaikkien perusidea on samanlainen, eli niiden avulla merkkijonosta muodostetaan numero-, kirjain- sekä erikoismerkkijohdistelmä, johon lisätty generoitu suola. Algoritmeista kuitenkin löytyy erilaisuuksia otetaan aluksi käsittelyyn argon2. Argon2:sta on mahdollista hyödyntää erilaisia muunnoksia, joiden nimet ovat Argon2i, -d, -id. Kaikkien parametreihin kuuluu määrätieto siitä, miten monta kertaa salaus tehdään sekä kuinka paljon hyödynnetään muistia, tämä kyseinen arvo annetaan kibibiteinä yksi kibibitti on 1024 bittiä (Argon2 2019). Toisena algoritmina on Bcrypt, sen pohjana on käytetty niin kutsuttua Blowfish-tekniikka. Bcryptin ideana on, että Blowfish-tekniikka yhdistettynä salausten määrään tuottaa todella hitaita salauksia, joiden purkamiseen kuluu kanssa huomattava määrä aikaa. Tämän tekniikan suurimpana hyötynä on mahdollisuus määrittellä, kuinka hitaasti salaus tehdään, määrittelyn myötä voidaan tehdä salauksen purkamisesta todella monimutkaista sekä hidasta hyökkääjille (Dan Arias 31.05.2018).

3.3 Seurantaviikko 3

Maanantai 16.09.2019

Työpäivän alussa:

Perjantaina opiskelin aiheesta salasanat ja niiden salausmenetelmät. Tämän pohjalta tulen päivän aikana tekemään sovellukseen kirjautumisen hyödyntäen tietokannassa olevia tunnuksia. Kirjautuminen tulee olemaan vain "proof of concept"-versio. Sen ideana on vain testata sekä esittää millä tavoin kirjautumisen voisi tehdä ja nähdään, onko tämäntapainen kirjautuminen sopivaa sovelluksellemme. Päivän tavoitteena on saada kirjautuminen toimimaan sekä salata salasanat hyödyntämällä opiskelemiani salausalgoritmia käyttäen.

Työpäivän lopussa:

Päivä meni suunnittelemani tehtävän parissa ja sain tehdyksi kirjautumisen ja salasanan salauksen. Salasanan salauksessa hyödynsin Bcryptiä, sillä se oli kaikista nopein ja yksinkertaisin salausalgoritmi lisätä olemassa olevaan projektiin. Lisääminen tapahtui asentamalla Nuget-paketti, jonka jälkeen pystyin käyttämään Bcryptin omia funktioita salaukseen, sen purkamiseen sekä vertailuun. Lisäksi tein lisäominaisuuden, jonka avulla käyttäjä pystyy vaihtamaan oman salasansa tietokannasta. Alapuolella esimerkkejä millä tavoin Bcrypt:ssä käsitellään salasanoja.

Salasanan salaaminen – `Bcrypt.HashPassword("Salasana");`

Salasanan varmistaminen – `Bcrypt.Verify("Annettu salasana", tietokannasta haettu salattu salasana);`

Tiistai 17.09.2019

Työpäivän alussa:

Töihin tullessani huomasin, että en ollut tehnyt eiliseen salasanan muutokseen mitään validointia eli määritellyt minkälainen salasanan tulisi olla. Päivän aikana yritän tehdä validoinnin sekä tehdä tarkistuksia, kun vaihdetaan salasana. Tarkistuksiin lukeutuu uuden salasanan syöttäminen kahdesti ja sen tarkastaminen kummallakin kerralla. Salasanojen tarvitsee olla samanmuotoisia ja salasanan muutos voi tapahtua vasta kun, vanha salasana on syötetty ja sen on tarkistettu oikeaksi tietokannassa olevan salasanan kanssa.

Työpäivän lopussa :

Sain päivän aikana tehtyä salasanalle kaikki ajattelemani tarkistukset. Uutta salasanaa ei voida asettaa, jos vanhaa salasanaa ei olla syötetty tai se ei täsmää

kannassa olevan salasanan kanssa. Uusi salasanan pitää myös syöttää kaksi kertaa, ja niiden pitää olla täysin samassa muodossa. Virhetilanteissa sivulle tulostetaan viesti, jossa kerrotaan mikä asioista ei onnistunut esimerkiksi vanhan salasanan ollessa virheellinen näytölle tulostetaan siihen liittyvä virheilmoitus. Näiden lisäksi tein validoinnin hyödyntämällä Regex-kaavaa, eli tapaa, jonka avulla voidaan määritellä missä muodossa annetun merkkijonon tulee olla, sekä mitä ehtoja merkkijonoon liittyy. Regex:n avulla voidaan esimerkiksi määritellä pakolliset kirjaimet, merkkijonon pituus sekä erikoismerkit, jotka ovat sallittuja. Työpäivän lopuksi esittelin parin päivän aikana tekemäni työni vanhemmalle kollegalleni, jonka mielestä salasanan hallinta sekä tietokantatunnuksien kautta kirjautuminen voitaisiin tulevaisuudessa implementoida käytössä olevaan sovellukseen.

Keskiviikko 18.09.2019

Työpäivän alussa:

Tälle päivälle ei ole suunniteltuna minkäänlaisia työtehtäviä, joten yritän saada roolien tallennuksen tietokantaan toimimaan, sekä muuttaa kyseisen sivun ulkoasua ja koodirakennetta siistimmäksi. Uskoisin, että saan nämä tehtävät tänään tehdyksi, jos päivän aikana ei ilmaannu muita asioita.

Työpäivän lopussa:

Päivän aikana sain viimeisteltyä tallennuksen sekä muokattua sivun ulkoasun käyttäjäystävällisemmäksi. Roolien tallennuksen yhteydessä jouduin muuttamaan olemassa olevaa koodia, sillä kollegani mainitsi, että olisi hyvä käyttää erästä tietokannan taulua, jonka pohjalta haetaan applikaatioiden roolit. Poistin applikaation roolit applikaatiotaulusta ja muutin koodin käyttämään uutta taulua, johon lisäsin vain viittauksen applikaatiotauluun. Pienen testauksen jälkeen koodi näytti toimivan, joten siivosin koodista turhat kommentit sekä tulostukset. Seuraavina päivinä tulen lisäämään roolisivulle vielä rajoituksia, kuten kuinka monelle eri henkilölle voidaan asettaa rooleja applikaatiokohtaisesti.

Torstai 19.09.2019

Työpäivän alussa:

Keskustelin töihin tullessani kollegani kanssa, että tarvitsisi päivän aikana kasata sovellusversio, johon olisi liitettynä osa tekemistäni ominaisuuksista. Päivä luultavammin tulee kulumaan pitkälti tämän tehtävän parissa, sillä tekemäni ominaisuudet ovat portaaliprojektissa ja niitä tarvitsee sieltä saada liitettyä toiseen projektiin.

Työpäivän lopussa:

Päivä meni odotetusti yhdistämisen parissa. Kohtasin muutamia ongelmia tehdessäni tehtävää, suurin osa näistä oli puhtaasti viittausvirheitä sekä backend:n puolella namespace-virheitä. Sain tehtävän tehtyä hieman ennen työpäivän loppumista ja aloitin siirtämään uusinta versiota testausympäristöön, jotta jokainen henkilö, jolla oikeus tähän ympäristöön pääsee tekemään testausta ja näkemään, että versio on mahdollista laittaa jakoon asiakkaille. Sain siirrettyä projektin testausympäristöön, mutta jostain syystä kyseinen versio ei suostunut käynnistymään. Tämän työn parissa tulen jatkamaan huomenna aamulla.

Perjantai 20.09.2019

Työpäivän alussa:

Tänään aikani tulee kulumaan eilen löytämäni ongelman parissa. Kyseisen ongelman korjaamisen jälkeen teen testauksia ja tarkistan, että kaikki lisäämäni toiminnot ovat kunnossa. Päivän loppuun mennessä yritän lisäksi saada annettua kollegoilleni tunnukset käyttöön, jotta he pääsevät testaamaan vielä itse tämän hetkistä versiota ja mahdollisesti etsimään virhetilanteita.

Työpäivän lopussa:

Päivä kului kokonaisuuden mainitsemieni tehtävieni parissa. Version käynnistymisongelmaan kului muutama tunti, sillä sain virheilmoituksen "503 Service unavailable" ja yritin ratkaista kyseistä ongelmaa usealla eri tavalla. Lopulta ongelma ratkesi hyvinkin yksinkertaisella tavalla, sillä jostakin syystä IIS:ssä (Internet Information Services) sovellussarjat eli application poolit olivat sammutettuina jonkun henkilön toimesta. Uudelleenkäynnistin sammutetut sovellussarjat, jonka jälkeen versio käynnistyi. Loppuaika kuluikin testauksessa sekä tunnusten antamisen parissa. Eräs kollegani huomasi muutaman ongelman, jotka minun täytyy korjata ensi viikon alussa saadaksemme version julkaisukuntoon.

Viikkoanalyysi:

Viikko oli vaihteleva sekä viikon tehtävien myötä taitoni kehittyivät tietyissä osa-alueissa. Ongelmatilanteiden myötä opin uusia asioita, koska jouduin itsenäisesti tutustumaan uusiin vieraisiin aihealueisiin. Tämänkin viikon perjantain version käynnistysongelma oli todella opettavainen ja yllättävä, sillä en olisi varmasti osannut tarkistaa IIS:ssä olevia asioita, ellen olisin tutkinut netistä, mitkä asiat voivat aiheuttaa 503-numerolla alkavia ongelmia. Ongelmien ratkaiseminen itsenäisesti on hyvä tapa kasvattaa omaa ongelmanratkaisukykyä ja sen myötä oma ammatillinen silmä kehittyi ymmärtää nopeammin syitä sille, minkä takia jokin virhe tai bugi ilmaantuu jossakin tietyssä tilanteessa.

Olen tällä viikolla aloittanut pitämään dokumenttia, johon kirjaan kollegojeni kanssa käytävien keskustelun pääasioita sekä ajatuksia, miten joitakin asioita voidaan kehittää tai muuttaa. Aloitin kirjaamisen, koska olen saanut enemmän tehtäviä työkseni sekä tehtävien kautta olen päässyt keskustelemaan monipuolisia keskusteluja useamman eri kollegan kanssa. Olen pystynyt tuomaan esille omia näkemyksiäni useastakin asiasta. Dokumentin avulla pystyn seuraamaan mitä kaikkia tehtäviä, minulla olisi ja mitä niistä on keskusteltu. Ilman minkäänlaista tapaa ylläpitää tehtäviäni on erittäin mahdollista, että jossakin vaiheessa ei muista tehdä jotakin tarvittavaa tehtävää.

Viikosta parhaiten jäi mieleen alkuviikon tehtävät, jotka sisälsivät salasanoja, niiden salausta sekä validointia. Aihe itsessään on vaikuttava ja merkittävä monella eri tapaa, sillä kirjautumisen sekä salasanojen avulla pystytään rajamaan henkilöiltä oikeuksia päästä käsi suljettuun sisältöön. Nykypäivänä suurimmassa osassa palvelun salasanoissa on käytetty hyödyksi erilaisia validointisääntöjä. Validointi tarkoittaa jonkin merkkijonon tarkistamista. Tarkistuksen aikana annettu merkkijonoa verrataan, että se vastaa ennalta määritettyjä kriteereitä. Salasanoissa useasti käytetyt kriteerit ovat muun muassa merkkijonon pituus, isojen sekä pienin kirjainten määrä, numerot sekä joissakin tapauksissa erikoismerkit kuten huutomerkki.

Validoinnissa käytetään hyödyksi säännöllisiä lausekkeita, jotka sisältävät merkkijonon kriteerit. Michael (Michael Fitzgerald 2012) kuvaa säännöllisiä lausekkeita koodatuiksi merkkijonoiksi, joita käytetään mallina annettavaa merkkijonoa vasten. Säännölliset lausekkeet ovat olleet jo kauan käytössä, sillä niiden avulla pystytään helposti määrittelemään, mitä jokin tietty merkkijono saa sisältää. Tämän myötä pystytään välttämään monet erilaiset hyökkäykset sovellukseen kuten esimerkiksi sql-injektiot, joiden tarkoituksena on päästä käsiksi tietokantaan suorittamalla tietokantapyyntöjä muun muassa tekstikentän kautta. Ohjelmointikielissä säännölliset lausekkeet ovat tehty eri tavoilla ja näyttävät mahdollisesti erilaisilta, vaikka niiden toiminnallisuus on aivan samanlainen. Esimerkkinä voidaan muun käyttää sähköpostin validointia, jonka lauseke on C#:ssa erilainen kuin Swift:ssä (Kuva 3). Lisäksi alapuolella muutama yksinkertaisempi merkkijonovalidointi C#:ssa (Kuva 4).

Kuva 4. Erilaisia validointeja C#

```
C# Puhelinnumero
(?:\d{3}\d{3})?-? *\d{3}-? *-?\d{4}

0408316187 - Hyväksyy
04213512 - Ei hyväksy
040-651-3123 - Hyväksyy

C# Nimi
^([a-zA-Z]+)([-\s'][a-zA-Z]+)*?$

Matti - Hyväksyy
lsmo - Ei hyväksy
Teppo! - Ei hyväksy
John-Doe - Hyväksyy
```

Kuva 3. Sähköpostin validointi C# ja Swift

```
C#
^(?("[\s\.\+\?<!\@]|(([\0-9a-z]((\.(?!\.))|[-!\#\$\%&'\*\+\/=\?\^\`\{\}\|\~\w])*)?(?<=[\0-9a-z])@))?(\[([\d{1,3}\.]{3}\d{1,3})|(([\0-9a-z]
[-\w]*[\0-9a-z]*\.)+[a-z0-9][\-\a-z0-9]{0,22}[a-z0-9]))$

Swift
[A-Z0-9a-z._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,64}
```

3.4 Seurantaviikko 4

Maanantai 23.09.2019

Työpäivän alussa:

Töihin tullessani kollegani mainitsi huomanneensa erään ongelman tekemässäni versiossa. Ongelma oli uloskirjautumisessa. Sisäänkirjautuminen tapahtuu jonkun ulkopuolisen kirjautumisen kuten Azure AD:n kautta. Painiketta painettaessa henkilö kirjataan ulos sovelluksesta, mutta käyttäjä pysyy sisäänkirjautuneena ulkopuolisessa palvelussa. Päiväni tulee kulumaan tämän tehtävän sekä erään toisen tehtävän parissa. Päivän kuluessa toivon saavani perjantaina huomattavan pienemmän ongelman kokonaan korjatuksi, sekä yritän saada uloskirjautumisongelman korjaamisen etenemään.

Työpäivän lopussa:

Tehtävät etenivät suunnitellusti ja sain korjattua pienemmän ongelman, joka oli sisäänkirjautumisessa. Ongelmaksi osoittautui, että osa henkilöistä ohjautui väärään sisäänkirjautumispaikkaan, vaikka heille oli asetettu eri kirjautumistapa. Väärin ohjaus johtui tekemistäni ehtolausekkeista, jotka eivät ottaneet huomioon kaikkia mahdollisia tilanteita, jolloin henkilö ohjautui väärän paikkaan. Korjaus oli yksinkertainen ja vaati vain hieman muutoksia lausekkeisiin. Sain aloitettua uloskirjautumisen tutkimisen ja ymmärsin, mitä minun täytyy huomenna tehdä, jotta saisin uloskirjautumisen toimimaan halutulla tavalla.

Tiistai 24.09.2019

Työpäivän alussa:

Tullessani töihin keskustelin muutaman henkilön kanssa, millä tavoin lähtisin liikkeelle uloskirjautumisen suhteen. He ehdottivat, että minun kannattaisi aloittaa perustasosta eli opiskella millä tavoin normaalisti kertauloskirjautuminen (SLO, Single Logout) toimii sekä lukea pohjamateriaalia SAML:sta (Security Assertion Markup Language), joka tarkoittaa xml-standardia, jolla voidaan jakaa käyttäjän tietoja turvallisesti. Päiväni sisältää kokonaisuudessaan opiskelua mainitsemistani aiheista, sekä mahdollisesti pieniä testauksia aiheeseen liittyen.

Työpäivän lopussa:

Päiväni meni kokonaan opiskelussa. Opin päivän aikana, mikä on SLO:n idea, ja mitä sen avulla tavoitellaan. Pääasiallisesti sen avulla mahdollistetaan uloskirjautuminen kaikista sessioista, jotka ovat kyseisellä henkilöllä avoinna. SLO:ta käytetään usein hyödyksi silloin, kun sisäänkirjautuminenkin tapahtuu SSO:n kautta eli Single Sign-On kautta, joka

tarkoituksena on yhden kirjautumisen kautta päästään käyttämään kaikkia saatavilla olevia sivuja sekä sovelluksia. Huomenna aloitan SLO-uloskirjautumisen implementoimisen tämänhetkiseen uloskirjautumiseen.

Keskiviikko 25.09.2019

Työpäivän alussa:

Päiväni tulee jatkumaan SLO:n implementoinnin parissa. Aluksi joudun hieman tutkimaan, mitä kaikkea back-end:ssä on tehty autentikoinnin suhteen ja onko siellä jo tehty aikaisemmin joitakin tarvittavia asioita uloskirjautumista varten. Tavoitteena tänään on pystyä saamaan uloskirjautuminen siihen vaiheeseen, että se yrittäisi poimia tarvittavat tiedot back-end:stä käyttöönsä.

Työpäivän lopussa:

Päivä ei mennyt aivan suunnitellusti. Koko päiväni kului koettaessani saada SLO:ta varten tietoja back-endiin. Sisäänkirjautuminen tapahtuu jonkun ulkopuolisen käyttäjähakemiston kautta. Back-end ei vastaanottanut uloskirjautumiseen liittyviä tietoja kuten sessionIndex- tai logoutNameIdentifier-tietoa. Tämä johtui siitä, että käytin testauksessa omassa virtuaalikoneessani olevaa itse konfiguroimaani ADFS:ää eli Active Directory Federation Serviceä, jossa saml2-sisäänkirjautumisen vastauksessa ei ollut mukana lainkaan tarvittavia tietoja. Sain ratkaistua tämän ongelman hyödyntämällä kehitysympäristössämme olevaa Azure AD:ta, jonka kautta kirjaudu palveluumme. Kirjautumisen yhteydessä back-end löysi automaattisesti etsimäni tiedot oikeisiin kohtiin. Huomenna yritän hyödyntää löytämiäni tietoja, kun yritän uloskirjautua sovelluksesta.

Torstai 26.09.2019

Työpäivän alussa:

Päiväni tulee jatkumaan uloskirjautumisen parissa. Päivän aikana aion hyödyntää eilen löytämiäni tietoja ja niiden avulla tekemään uloskirjautumisen käyttäjähakemistosta asti. En ole aivan varma, miten tällä hetkellä käytössämme oleva autentikointikirjasto on mahdollistanut saml2-uloskirjautumispyynnöt.

Työpäivän lopussa:

Sain selville, miten uloskirjautuminen tapahtuu nykyisen kirjaston kautta. Sitä varten kirjastosta löytyy oma funktionsa, jota kutsumalla aloitetaan federated sign-out, jonka tarkoituksena on suorittaa uloskirjautuminen ulkopuolisesta käyttäjähakemistosta. Kohtasin kuitenkin tämän kanssa ongelman, sillä sain aloitettua uloskirjautumisen prosessin, mutta

jonkin syyn takia uloskirjautuminen ei saanut tietoonsa sessionIndexiä, jonka avulla pystytään tarkentamaan mihin sessioon kuuluu tehdä uloskirjautuminen. Päiväni kului lopulta kyseinen tehtävän parissa, enkä saanut sitä ratkaisua ennen kotiinlähtöäni.

Perjantai 27.09.2019

Työpäivän alussa:

Aloitan aamuni jatkamalla eilen löytämäni ongelmaa sessionIndexin käytöstä. Koetan saada haettua indeksin käyttöön sekä sen jälkeen toteuttamaan oikeaoppisen uloskirjautumisen. Päiväni tulee kulumaan näiden tehtävien parissa.

Työpäivän lopussa:

En saanut päivän aikana tehtyä uloskirjautumista, mutta sain kuitenkin haettua talteen tarvittavat tiedot uloskirjautumista varten. En tule luultavammin jatkamaan tätä tehtävää muutamaan viikkoon, sillä keskustelin ennen kotiinlähtöäni kollegani kanssa tulevista tarpeellisimmista tehtävistä. Ensi viikon alussa teen muutaman pienemmän muutoksen uloskirjautumiseen, jonka jälkeen yritämme saada version julkaistuksi.

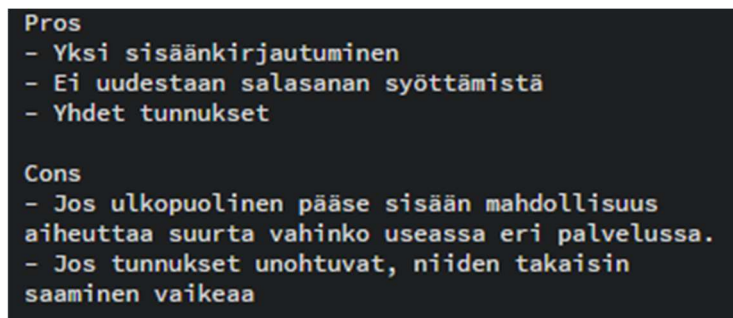
Viikkoanalyysi

Yksi päiväkirjaviikko on jälleen ohitse. Viikon aikana opiskelin uusia aiheita, sekä paransin ymmärrystäni jo aikaisemmin lukemistani aiheista. Uusia aiheita tälle viikolle oli kertakirjautuminen (SSO) sekä kertauloskirjautuminen (SLO). En ollut aikaisemmin tutustunut näihin aiheisiin kuin pintapuolisesti. Oli mielenkiintoista päästä näkemään ja tutkimaan aiheita syvällisemmin sekä päästä ymmärtämään, millä tavoin nämä kyseiset sisään- sekä uloskirjautumistavat toimivat. Opiskelin myös aiheiden pohjalta Saml2:ta, joka on lyhenne termistä Security Assertion Markup Language 2.0. Saml2 on XML-tyylillä tehty protokolla, minkä avulla pystyy siirtämään kirjautumis- sekä henkilödataa eri osapuolien välillä (SAP Community 2013). Saml2 on käytössä, kun tehdään sisäänkirjautumisia käyttäjähakemiston kuten Azure AD:n kautta johonkin toiseen sovellukseen.

Tämän viikon aiheisiin perustuen on mukava nähdä, että uusille nuorille ohjelmoijille kuten itsekin niin annetaan tehtäviksi sekä opiskeltavaksi paljon uusia asioita eikä pidetä heitä vain tutuissa sekä turvallisissa aihepiireissä. Viikon aiheinani oli pitkälti autentikointi ja se on nykypäivänä merkittävä puoli sovelluksissa ja sen on tehtävä hyvin, että kukaan ulkopuolinen henkilö ei voi saada tietoja käsiinsä tai saada mahdollisuutta päästä tekemään muutoksiin mihinkään palveluun tai sovelluksen asetuksiin.

SSO on sessio, johon käyttäjä kirjautuu sisälle käyttäen yksiä ainoita sisäänkirjautumistunnuksiansa. Tämän avulla mahdollistetaan se, että käyttäjä pystyy yhdelle sisäänkirjautumiselle käyttämään kaikkia hänelle kuuluvia sovelluksia sekä palveluita, jotka ovat hänelle tarjolla. Usein SSO:n yhteydessä käytetään hyödyksi erilaisia käyttäjähakemistosta kuten Microsoftin omia hakemistoja. Tunnettuja SSO tyylillä olevia sosiaalisen median kirjautumistapoja on muun muassa Google, Twitter sekä Facebook. SSO:sta kuitenkin löytyy sekä huonoja, että hyviä puolia (Kuva 5) (Margaret Rouse 2019). SSO:n sessioita uloskirjautuminen tapahtuu hyödyntämällä SLO:ta, jonka tarkoituksena on tuhota kaikkia voimassa olevat sessiot uloskirjautumisen yhteydessä. Uloskirjautumisen aikana SLO lähettää uloskirjautumispyynnön sisäänkirjautumisessa käytettyyn palveluun, palvelu lähettää takaisin vastauksen. Hyväksytyin vastauksen myötä SLO tuhoaa sessiot ja ohjaa käyttäjän oikeaan ennalta määritettyyn uloskirjautumisen osoitteeseen.

Kuva 5. Kertasisäänkirjautumisen hyvät ja huonot puolet



3.5 Seurantaviikko 5

Maanantai 30.10.2019

Työpäivän alussa:

Viikkoni alkaa uloskirjautumisen muutoksien teossa, joista mainitsinkin viime viikon lopulla. Ensiksi teen jonkinlaisen pop-up ikkunan, joka avautuu klikatessa Kirjautu ulos -kohtaa. Ikkunassa on esillä teksti, jonka sisältö riippuu siitä, minkä kautta käyttäjä on alun perin kirjautunut sovellukseen. Kirjautuminen voi tapahtua ulkopuolisen hakemiston tai adlds:n (Active Directory Lightweight Directory Services) kautta

Työpäivän lopussa:

Päiväni kului mainitsemani tehtävän parissa ja siinä jouduin hieman tutkimaan back-end:n sisältöä ja sitä, miten on yksinkertaisinta Angular:ssa käyttää pop-up-ikkunoita. Back-endistä jouduin lisäämään käyttäjäkohtaisesti yhden uuden lisätiedon, joka oli sisäänkirjautumistapa. Tämän avulla pystyin asettamaan halutun tekstin näkyville tekemääni pop-uppiin. Pop-up itsessään on tehty hyödyntäen Angularin modalia, joka oli todella yksinkertainen ja hyvin muokattavissa oleva komponentti.

Tiistai 1.10.2019

Työpäivän alussa:

Tälle päivälle tehtäväkseni olen ajatellut tehdä erään IP-tarkastuksen. Tarkastuksessa katsotaan, mistä IP-osoitteesta henkilö kirjautuu nykyiseen palveluun. Osoitteen mukaan määritellään, mitä kaikkea hän voi nähdä, sekä mitkä roolit hänelle voi antaa.

Työpäivän lopussa:

Tehtävässä meni koko päivä, ja sain sen valmiiksi ennen kotiinlähtöäni. Työtehtävä oli yllättävän mielenkiintoinen ja vaikea, sillä en ole itse ennen ollut tekemisissä IP-osoitteiden kanssa enkä ollut tietoinen millä tavoin mask toimii IP:n kanssa. Maskilla tarkoitetaan määrää, kuinka monta eri IP-osoitetta on saatavilla kyseisen verkon alaisuudessa. Hyödynsin tässä .NET:stä löytyvää System.Net.IPAddress- luokkaa, jonka avulla pystyin tarkistamaan IP:n oikeellisuuden sekä vertailemaan, että kuuluuko saatu IP-osoite annettuun verkon osoitteeseen. Vertailun avulla pystyin määrittelemään, saako henkilö sisäänkirjautuessa kaikki hänelle kuuluvat roolit ja oikeudet vai vain osan niistä.

Keskiviikko 02.10.2019

Työpäivän alussa:

Keskustelin äsken kollegani kanssa, ja minun pitää aloittaa uuden ominaisuuden tekeminen vanhaan sovellukseen. Ominaisuus on ollut jo jonkin aikaa haluttu toiminnallisuus asiakkaiden puolesta ja se helpottaisi suuresti heidän toimintaansa tietyissä asioissa. Aloitan tekemisen ominaisuuden suunnittelulla ja toivon saavani tehtyä hieman front-end-puolta valmiiksi.

Työpäivän lopussa:

Sain päivän aikana aloitettua ominaisuuden työstämisen front-end:n ulkoasuun asti. Ulkoasua tehdessä huomasin, että muutamasta hyödyntämästäni komponentista on tullut uusia versioita. Nämä versiot tarvitsisivat Angular-päivityksen toimiakseen, joten keskustelin asiasta vanhemman kollegani kanssa ja hänen mielestään tässä kohtaa ennen uuden toiminnallisuuden tuottamista, olisi hyvä päivittää Angular uusimpaan versioon. Päivitys tulee olemaan huomisen työtehtäväni.

Torstai 03.10.2019

Työpäivän alussa:

Päiväni tulee kulumaan eilen mainitsemani Angular-päivityksen merkeissä. Tulen päivittämään Angularin kuudennesta versiosta kahdeksanteen versioon. Uskon, että tehtävän parissa minulla kuluu ainakin tämä päivä. Projekti, johon tulen päivittämään Angularia on erittäin laaja ja se sisältää huomattavan määrän erilaisia komponentteja, jotka tarvitsevat kaikki myöskin päivityksiä toimiakseen uusimman Angular-version kanssa.

Työpäivän lopussa:

Päiväni meni kuten sen ajattelin eli Angular-päivityksessä. Jouduin iltapäivällä tekemään koko projektin palautuksen alkupisteeseen ennen päivityksen aloittamista. Tämä johtui siitä, että koko projekti meni rikki ja se ei suostunut lainkaan käynnistymään. Käynnistys ei antanut minkäänlaista virheilmoitusta siitä, mistä tämä olisi edes johtunut. Tulen huomenna jatkamaan tästä tilanteesta ja yritän uudestaan saada tehtyä päivityksen, sillä onnistuneen päivitysprosessin myötä voisin päivittää jo käytössä olevat sovellukset myöskin uusimpaan versioon.

Perjantai 04.10.2019

Työpäivän alussa:

Aloitan pian uudelleen päivittämään Angularia. Olen ajatellut suorittaa päivityksen pienemmissä palasissa eli aluksi päivitän Angularia vain yhden tai muutaman version eteenpäin ja yritän tämän jälkeen saada projektin käynnistymään sekä toimimaan normaalin tapaan. Tavoitteenani on saada Angularia päivitettyä vähintään muutaman versio ylöspäin, jotta saisimme käyttöön useiden eri komponenttien uusia ominaisuuksia. Toivon kuitenkin saavani päivitettyä Angularin aina kaikista uusimpaan versioon asti.

Työpäivän lopussa:

Päivitys ei tänäänkään onnistunut niin hyvin kuin olisin sen toivonut onnistuvan. Sain päivitettyä Angularia muutaman version verran sekä lisäksi sain uusia versioita myös käyttämistämme komponenteista. Päivitys päätettiin kollegani kanssa jättää hetkeksi tähän versioon asti, sillä uudesta aloittamastani ominaisuudesta oli tullut useampikin kysely päivän aikana. Ensi viikosta alkaen keskityn pelkästään ominaisuuden työstämään, jotta saisimme mahdollisimman pian valmiiksi esittelyversion, joka toimisi tietokantakyselyn kanssa.

Viikkoanalyysi

Viikko kului todella nopeasti. Sen aikana pääsin oppimaan taas uusia asioita sekä pääsin työstämään uutta ominaisuutta nykyiseen mobiilisovellukseen. Tällä viikolla tuli tehtyä monia erilaisia tehtäviä, joista jokainen oli omalla tavallaan mielenkiintoisia sekä osittain haastaviakin. Pääsin käsittelemään Angularin komponentteja, Angularin päivityksiä sekä opiskelemaan ja tutkimaan itselleni entuudestaan täysin tuntematonta asiaa eli IP-osoitteita ja niiden hyödyntämistä.

Viikon aikana ymmärsin työskentelystäni myös sen, että ei kannata asettaa liian tarkkoja tavoitteita päiviä kohtaan. Koetin saada suoritettua Angularin päivitys yhden päivän aikana ja lopputuloksenahan oli, että en saanut sitä kunnolla tehtyä, sain päivitettyä sitä vain muutaman version verran ylöspäin. Liian tarkkojen tavoitteiden myötä aiheutuu turhaa stressiä ja työn laatu saattaa huonontua huomattavasti yrittäessä nopeasti päästä asettamaansa tavoitteeseen. Opin asettamaan työtehtäville realistia aikarajoja stressaamatta pystyisinkö suorittamaan tehtävän aikarajaan mennessä. Työn, että koodin laatu on huomattavasti laadukkaampaa silloin kun edellä mainitsemani kriteerit täyttyvät.

Tehtävät olivat tällä viikolla kattavia, parhaiten itselleni mieleeni jää IP:iin liittyvät työtehtävät. Niiden pohjalta pääsin oppimaan, miten IP toimii sekä millä tavoin IP kuuluu johonkin verkkoon. IP tarkoittaa osoitetta, joka koostuu numerosarjoista. Osoitteita on olemassa kahta erilaista on IPv4 ja IPv6. Näiden avulla pystytään lähettämään IP-paketteja eri

osoitteiden välillä, osoitteet näkyvät loppukäyttäjille tekstimuotoisena. Esimerkkinä tästä voi olla IPv4-osoite 192.xx.xx.xxx, joka voi tarkoittaa ”www.esimerkki-sivu.com” (WhatIsMyIPAddress 2019). Jokainen IP-osoite on yksilöllinen ja kuuluu osoiteavaruuteen (Address Space). Osoiteavaruudella tarkoitetaan sitä, kuinka suuri määrä osoitteita voi olla käytössä jonkin operaattorin tai sen asiakkaan alaisuudessa.

Tällä viikolla suurimpana yllätyksenäni koin Angularin päivittämisen, sillä olin oletanut, että tämäläinen päivittäminen olisi suhteellisen helppoa ja yksinkertaista. Angular on laajassa käytössä oleva framework, joten voitaisiin ymmärtää, että siihen tulevat versiopäivitykset olisivat hyvin nopeasti tehtävissä. Päivittäminen usean version välillä on todella monimutkaista, sillä väliversioissa on voinut tulla useita rakenteellisia muutoksia, jotka pitäisi muistaa ottaa huomioon, kun tehdään tämänkaltaisia päivityksiä. Tämä työtehtävän myötä sisäistin, että versioiden päivittäminen on erittäin monimutkaista ja sen kanssa kohtaa monentapaisia ongelmia. Vaikka viikon versiopäivitys ei onnistunutkaan kuten olisin sen halunnut. Opin suuren määrän erilaisia asioita tehdessäni päivitystä kuten hyödylliset työtavat.

3.6 Seurantaviikko 6

Maanantai 07.10.2019

Työpäivän alussa:

Viikkoni aluksi työskentelen uuden ominaisuuden parissa. Muokkaan viime viikolla päivittämiäni komponenttien toimintoja sopivimmiksi sovellukseen, sekä lisään päivityksen yhteydessä tulleita lisäominaisuuksia nykyisiin toimintoihin.

Työpäivän lopussa:

Päiväni kului ominaisuuden front-end:n parissa. Sain muokatuksi kaikki nykyiset toiminnot haluamallani tavalla ja nyt kaikki perustoiminnallisuudet ovat käyttökunnossa. Huomenna aloitan komponentin TypeScript-tiedoston tekemisen ja mahdollisesti saan tehtyä erilaisia tietokantakutsuja back-endiin.

Tiistai 08.10.2019

Työpäivän alussa:

Aloitan tänään tekemään uuden ominaisuuden TypeScript-tiedostoon toiminnallisuuksia ja funktioita. Aluksi tulen tekemään funktion, jonka avulla pystytään hakemaan tietokannassa olevasta taulusta tarpeelliset tiedot komponentille.

Työpäivän lopussa:

Sain päivän aikana tehtyä mainitsemani tietokantahaun loppuun ja komponentti vastaanottaa tiedot TypeScript-tiedostossa. Seuraavaksi tarkoitukseni on näyttää tietyt tiedot front-end:n puolella. Nykyisen mobiilisovellus-projektin tietokantahakuprosessin polku on mielenkiintoinen. Aluksi lähetetään pyyntö TypeScript-tiedostosta luodulle Angularin servielle. Servicestä lähetetään uusi pyyntö ennalta määritellyn osoitteeseen, joka vastaanotetaan back-end:n controller-luokassa oikeassa funktiossa. Funktiossa kutsutaan haluttua metodia, josta lähetetään seuraava pyyntö, joka kulkee useamman eri paikan kautta tiettyyn tietokantapalveluun, josta lopulta tehdään tietokantapyyntö ja tästä pyynnöstä tuleva vastaus palautetaan samaa polkua takaisin TypeScript-tiedostoon asti.

Keskiviikko 09.10.2019

Työpäivän alussa:

Eilen sain työpäivän lopussa haettua tietoja komponentin TypeScript-tiedostoon. Tänään

tulen asettamaan saadun datan front-end:n puolelle oikeisiin paikkoihin ja sen jälkeen tulen tekemään Post-metodin, jonka avulla pystyn lisäämään uutta dataa tietokantaan.

Työpäivän lopussa:

Päivän aikana sain tehtyä kaikki suunnittelemani tehtävät. Nyt front-end:ssä näytetään tietoja, jotka tulevat kannasta ja näistä tiedoista osaa hyödynnetään myös Post-metodissa. Post-metodissa on mukana suoraan valittuja sekä käyttäjän itse kirjoittamia tietoja. Päivän kuluessa kohtasin muutaman virhetilanteen post-metodia tehdessä, sekä huomaisin pienen bugin get-metodissa. Kaikki näistä ongelmista olivat back-end:n tietokantapyyntöjen yhteydessä ilmaantuvia ja niiden korjaaminen onnistui pelkästään muuttamalla hieman tietokantapyyntöjen rakennetta.

Torstai 10.10.2019

Työpäivän alussa:

Puolet tästä päivästä tulee kulumaan ominaisuuden työstämisen parissa ja toinen puoli kuluu palaverissa. Palaveri on erään suunnitellun projektin esittelypalaveri, jossa tullaan käymään läpi projektin sisältöä ja aloitusajankohtaa.

Työpäivän lopussa:

Aamupäivän aikana sain tehtyä työstämäni komponentin lähes valmiiksi. Komponenttiin tarvitsee enää tehdä pieniä muutoksia, kuten kenttien validointia sekä sääntöjä siitä, milloin mikäkin kenttä on näkyvässä. Iltapäivä meni palaverin merkeissä. Tulemme aloittamaan tämän projektin jo ensi viikon aikana. Ensimmäiset tehtävät sisältävät teknologioiden tutkimista sekä päätöksentekoa siitä, mitä teknologioita tulemme käyttämään kyseisessä projektissa.

Perjantai 11.10.2019

Työpäivän alussa:

Tänään työskentelen jälleen uuden komponentin parissa. Koetan saada haluamani pienet muutokset tehtyä sekä mahdollisesti aloittamaan erään vanhemman komponentin muuttamisen, jotta sitä voisi hyödyntää uuden ominaisuuden kanssa. Jos minulla on tänään aikaa, niin yritän korjata yhden Visual Studio -kehitysympäristön ongelman, jonka onnistui saamaan itselleni, kun tein Angular-päivityksiä.

Työpäivän lopussa:

Päivän kuluessa sain viimeisteltyä tämän hetkiset ominaisuudet uudessa komponentissa. Suoritin manuaalitestausta ja satuin löytämään muutaman pienen bugin eri osa-alueissa, jotka kaikki korjaantuivat suhteellisen yksinkertaisesti. En tänään aloittanut vielä vanhan komponentin muuttamista, mutta sain korjattua mainitsemani Visual Studio -bugin. Tämä ongelma oli, että Visual Studioni ei tunnistanut tiettyjä komentoja, jonka vuoksi jouduin käynnistämään projektini manuaalisesti komentorivin kautta joka kerta.

Viikkoanalyysi

Jokainen työviikkoni on ollut tähän mennessä mielekäs ja opettavainen. Tälläkin viikolla pääsin tekemään erilaisia asioita kuten uuden ominaisuuden luomista ja osallistumaan uuden projektin esittelypalaveriin. Uutta ominaisuutta tehdessä pääsin käsittelemään sekä front-end:iä että back-end:iä. Kummatkin näistä alkavat olemaan minulle tutumpia kuin esimerkiksi puoli vuotta sitten, jolloin ominaisuuden tai pelkästään uusien kenttien luominen vei minulta huomattavasti enemmän aikaa. Tästä huomaan, että kehitystä on tapahtunut ohjelmointini saralla sekä ylipäättänsä ammatillinen kehitykseni on ollut nousujohteinen.

Ongelmanratkaisutaitonikin on huomattavasti parantuneet. Tämän huomasin tällä viikolla, kun perjantaina ratkoin Visual Studio – ongelmani. Ongelmani oli, että Angular-päivitysten yhteydessä jostakin tietämättömästä syystä Visual Studio kadotti kokonaan polun globaalisti asennettuun Angulariin. Yritin korjailla tätä ongelmaa aluksi poistamalla Angularin globaalisti sekä asentamalla uudestaan hyödyntäen Node.js:n npm-metodia. Tämä ei kuitenkaan korjannut ongelmaa, joten jouduin hieman tutkimaan netistä useallakin eri hakusallalla korjausvaihtoehtoja. Löysin Stack Overflow-nimiseltä foorumilta erään artikkelin vastauksista (Stack Overflow 2018), jossa oli linkki GitHub:ssa löytyvään artikkeliin, missä lopulta eräs käyttäjä oli antanut vastaukseksi tavan, miten minun ongelmani voisi ratkaista. Tässä vastauksessa ehdotettiin, että poistetaan Node.js ohjauspaneelin kautta, jonka jälkeen poistetaan tietokoneelta polusta "kayttaja/App-Data/roaming/" kansiot npm sekä npm cache. Tämän jälkeen piti ladata uusin stabiili version Node.js:stä ja asentaa Angularin versio globaalisti seuraavanlaisessa komennolla "npm install -g @angular/cli". Näiden vaiheiden jälkeen käynnistin vielä koneeni uudestaan ja tämän jälkeen avasin Visual Studio sekä käynnistin projektin, jolloin Visual Studio löysi taas Angularin koneeltani.

Node.js on suuressa suosiossa oleva avoimen lähdekoodin runtime-ympäristö(runtime-environment), jonka pystytään käynnistämään erilaisia JavaScript-pohjaisia sovelluksia ja palveluita. Node.js oli aluksi hyötykäytössä selaimella, mutta nykyään sen avulla pystytään ylläpitämään kokonaisia sovelluksia. Patel (Patel P. 2018) mainitsee kolme tärkeintä

kohtaa miksi Node kannattaisi ottaa käyttöön. Yksi näistä mainituista syistä on että, Node hyödyntää tapahtumapohjaista mallia, joka tekee siitä erittäin kevyen sekä käytännöllisen.

Kaikki, jotka käyttävät Node.js työtehtävissään tuntevat varmasti termin npm(Node Package Manager). Npm on julkinen kokoelma erilaisia avoimen lähdekoodin projekteja, nämä projektit voivat olla erilaisia komponentteja, funktioita tai muita tämän kaltaisia. Näitä kaikkia voidaan asennuksen jälkeen hyödyntää omissa projekteissa haluamillansa tavoilla. Asennus tapahtuu komentorivin kautta hyödyntäen npm-komentoa, jonka formaatti on voi olla esimerkiksi "npm install paketin nimi" tai "npm install -g paketin nimi", tässä tilanteessa -g tarkoittaa, että paketti asennetaan koneelle globaalisti eikä pelkästään siihen projektiin missä komentorivi on käynnistetty. Mutta jos kyseissä -g-optiota löydy lausumetodista silloin paketti ladataan projektin kansioon nimeltä node_modules (Nodejs 2018).

3.7 Seurantaviikko 7

Maanantai 14.10.2019

Työpäivän alussa:

Tullessani töihin keskustelin kollegani kanssa uudesta ominaisuudesta, sekä lisätoiminnosta, jonka voisin tämän päivän aikana lisätä siihen. Keskustelimme myöskin uuden projektin aloituksesta ja aiheista mitä olisi hyvä tutkia ja pohtia seuraavien päivien aikana. Päivä tulee kulumaan lisätoiminnon teossa. Tehtyäni sen valmiiksi aloitan tutkimaan hieman harjoittelijan kanssa, millä tavoin olisi hyvä toteuttaa uuden projektimme back-end-puoli.

Työpäivän lopussa:

Päivän aikana sain tehtyä lisätoiminnon tekemääni komponenttiin. Tehdessäni sitä jouduin muuttamaan hieman front-end:iä sekä komponentin TypeScript-tiedostoa. Muutoksien vuoksi kohtasin muutaman bugin testausta tehdessäni, joten jouduin muuttamaan aiemmin tekemieni toimintojen logiikkaa sekä rakennetta hieman. Ehdin tänään tutkimaan vähän uuteen projektiin liittyviä asioita. Jatkan tämän tehtävän parissa huomenna.

Tiistai 15.10.2019

Työpäivän alussa:

Jatkan tänään uuden projektin pohjatyötä. Siihen lukeutuu eilen mainitsemani back-end-puolen tekniikka, sekä projektin ja sen rakenteen yleinen pohdinta. Koetan tämän päivän aikana aloittaa vanhan komponentin muokkaamisen.

Työpäivän lopussa:

Sain päivän aikana tutkittua useampaakin eri vaihtoehtoa projektin back-end:lle. Kirjasin jokaisesta eri tekniikasta ylös hyviä sekä huonoja puolia. Käyn harjoittelijan kanssa tutkiamme asioita läpi tämän viikon perjantain kokouksessa, jossa käydään myös muiden projektin jäsenien tutkimia aiheita.

Keskiviikko 16.10.2019

Työpäivän alussa:

Aloitan tämän päiväkirjamerkinnän kirjoittamisen jälkeen muokkaamaan toisia komponentteja. Tarkoituksena on muuttaa näitä komponentteja, jotta niiden toimintoja voitaisiin hyödyntää uuden tuottamani ominaisuuden eri osa-alueissa. Esimerkiksi pystyttäisiin päivittämään tai poistamaan jokin asia, joka on luotu uutta ominaisuutta käyttäen.

Työpäivän lopussa:

Päiväni meni muokkauksia sekä muutamia lisäyksiä tehdessäni. Sain yhden vanhemmista komponenteista muutetuksi ja aloitin hieman seuraavan komponentin muokkaamista. Muokkaustehtävissä tulee menemään ainakin loppuviikko ja mahdollisesti suurin osa ensi viikostakin.

Torstai 17.10.2019

Työpäivän alussa:

Jatkan tänäänkin samoja asioita, kuin eilenkin. Koetan saada päivän aikana muutettua seuraavan komponentin. Yritän korjata eilen muuttamastani komponentista hieman sen toimintalogiikkaa, sillä ymmärsin eilen kotiin kävellessäni, että tein turhaan erään tietokantapyynnön. Olisin pystynyt yhdistämään tämän tietokantahaun toiseen tietokantapyyntöön.

Työpäivän lopussa:

Aloitinkin tänään päiväni tietokantapyynnön muutoksella, tein testauksia kyseisessä komponentissa ja kohtasin erikoisia ongelmia. Ongelmat koostuivat siitä, että tietokantapyyntö ei onnistunut, mutta asia ei aiheuttanut minkään kaltaista virhetilannetta eikä tuottanut virhetiedostoon mitään selitystä, mistä tämä olisi johtunut. Sain virhetilanteen selvitettyä ennen kotiinlähtöäni ja syy tämänkaltaiseen ongelmaan oli yllättävä. Kaikki johtui, että käytössämme on sovellus, jonka tietokantaan tallennetaan osa tiedoista samalla, kun ne tallennetaan paikalliseen kantaan. Minulla oli vanhentunut tämän sovelluksen lisenssi, jonka takia en pystynyt tallentamaan näitä tietoja tämän sovelluksen kautta, mikä johti siihen, että sain aiheutettua itselleni virhetilanteen. Kaikki ongelmat korjaantuivat, kun lisäsin uuden voimassa olevan lisenssin sovellukselle.

Perjantai 18.10.2019

Työpäivän alussa:

Tänä aamupäivänä teen vanhojen komponenttien muokkauksia. Tavoitteena olisi saada seuraavan komponentin tietokantahakuun lisättyä mukaan tarvittavat tiedot uuden ominaisuuden toimintoja varten. Haluaisin saada hyödynnettyä suoraan nykyisiä hakuja, jotta

välttyisin turhalta toistolta. Iltapäivällä minulla tulee olemaan alkuviikosta mainitsemani palaveri uuden projektin tiimoilta. Palaverissa käsittelemme projektin eri osa-alueita ja yhdessä harjoittelijan kanssa esitämme ehdotuksemme siitä, millä tavoin back-end voitaisiin toteuttaa.

Työpäivän lopussa:

Päivä meni pitkälti suunnitelman mukaan. Aamusta tein komponenttien muutostehtäviä, jotka sain tavoittelemaani kohtaan asti. Aamun aikana autoin kollegaani Angular:n sekä siihen liittyvän tietokantapyynnön kanssa. Iltapäivästä meillä oli projektinpalaveri, jossa esittelimme ideoita sekä ehdotuksia back-end:stä, sekä monesta muustakin aiheesta kuten front-end:stä ja kirjastoista, joita voisimme hyödyntää. Palaverin jälkeen meillä oli vielä työpäivää jäljellä, joten päätimme projektiryhmämme kanssa aloittaa projektin tarkemman suunnittelun.

Viikkoanalyysi

Viikko vierähti nopeasti. Se oli laaja-alainen ja sisälsi osittain ohjelmointia sekä osittain uusien aiheiden tutkimista. Ohjelmoinnin puolesta tein paljon muokkaustehtäviä jo olemassa oleviin komponentteihin, jotta saisin uuden ominaisuuden tuottamaa tietoa hallinnoitua sekä mahdollisuuksia mukaan muokattua. Tämä tehtävä ei kuitenkaan vielä tullut valmiiksi asti, vaan tulen jatkamaan tämän parissa vielä ensi viikollakin.

Viikon aikana tapahtui suuri määrä oppimista useammastakin aiheesta. Tutkiessa toteutustapoja back-end:lle löysin suuria määriä mielenkiintoisia artikkeleita, joiden pohjalta pystyin hyvinkin helposti sekä yksinkertaisesti saamaan mielikuvan, millä tavoin meidän tuleva projektimme olisi hyvä rakentaa ja mitä sen kuuluisi sisältää. Lisäksi tällä viikolla opin hieman lisää Angular:sta, kun autoin kollegaani eräässä hänen ongelmassaan. Se tuntui aluksi erittäin selkeältä ongelmalta, mutta paljastuikin paljon monimutkaisemmaksi. Saimme kuitenkin asian ratkaistuksi useamman tunnin yrittämisen jälkeen, mutta kuitenkin tämän pohjalta ymmärsin hieman enemmän, siitä millä tavoin Angular pystyy keskustelemaan back-end:n kanssa ja mitä kaikkea pyynnöistä tapahtuu.

Viikon kohokohdaksi pystyin helposti nostamaan uuden projektin alkututkimukset sekä perjantain palaverin, jossa pääsimme hyvin mukaan siihen, mitä meidän halutaan toteuttaa. Projektin aloitusajankohdaksi määriteltiin ensi viikon alku ja se tulee jatkumaan useampia kuukausia. Perjantaina palaverin päätteeksi teimme alkuvalmisteluja, kuten luomalla projektia varten kanban-työkalun, jonne pystymme tulevaisuuden lisäämään erilaisia tehtäviä kanban-kortteina. Lisäksi pohdimme millä tavoin olisi hyvä aloittaa tämän

kaltainen projekti, sillä kukaan ryhmämme jäsenistä ei ole ennen ollut laajoissa projekteissa mukana. Tutkin eri lähteistä hyviä tapoja lähteä liikkeelle projektissa, ja hetken tutkimisen jälkeen löysinkin useammankin artikkelin, joiden pohjalta aloimme yhdessä käymään keskustelua sekä kirjaamaan ylös keskustelun aiheita ja päätöksiä sekä ehdotuksia.

Käytimme hyväksemme muun muassa Bacrania:n (Bacrania J. 2015) kirjoittamaa artikkelia, missä hän listaa viisi erilaista suunnitteluehdotusta. Ehdotuksissa käsitellään projektin määrittelyä, aikataulua, suunnittelua, ryhmän välistä kommunikointia sekä tilanteita, joissa projektin saatetaan lopettaa. Näistä ehdotuksista eniten kävimme läpi kohtaa 3, jonka ideana on tehdä sivun mittainen suunnitelma projektista, jossa mietitään Bacrania:n (Bacrania 2015) listaamia asioita, joista yksi on riskien kartoittamiseen liittyvä kysymys. Kaikki hänen listaamansa kysymykset olivat todella keskustelua herättäviä sekä niiden pohjalta pystyimme saamaan hyvän pohjan suunnitelman projektiamme varten. Lisäksi hyödynsimme Dymondi:n (Dymond J 2015) artikkelia ja vertailimme sen sisältöä Bacrania:n kirjoittamaan sisältöön ja huomasimme, että kummassakin näissä on käsiteltyä samoja aihealueita hieman eri suunnasta. Kummatkin näistä artikkeleista ajoivat tärkeänä aiheena riskejä ja niiden kartoitusta. Vertailun sekä artikkelien pohjalta pääsimme ryhmänä siihen yhteisymmärrykseen, että meidän tarvitsee ensi viikon alussa pitää uusi palaveri, jonka tarkoituksena on keskustella vielä enemmän artikkeleissa esitellyistä aiheista sekä yleisesti projektin kulusta. Keskustelun pohjalta tulisi laatia kunnollinen ja sisältörikas dokumentti, jonka sisältö koostuu projektin jokaisesta eri aihealueesta sekä kaikista mahdollisista tilanteista, jotka voivat esiintyä meidän projektimme aikana. Esimerkkinä tilanteista voidaan ottaa Dymond:n (Dymond J 2015) mainitsema tilanne, jossa uusi jäsen tulee tiimiin, jolloin olisi hyvä olla kaikki asiat dokumentoituna, jotta uusin jäsen pääsisi mukaan työhön mahdollisimman nopeasti sekä vaivattomasti.

3.8 Seurantaviikko 8

Maanantai 21.10.2019

Työpäivän alussa:

Uusi työviikko on aluillansa ja tällä viikolle on jo suunniteltuna työtehtävät, jotka minun tarvitsee toteuttaa. Tänään jatkan komponentin muokkaamista, sekä osan päivästä käytän uuden projektin suunnitteluun ja aloitukseen.

Työpäivän lopussa:

Päivän aikana muokkasin vanhan komponentin toimivaksi uuden ominaisuuden tiedoilla ja muutin hieman tyyllittelyä aiemmin muokkaamasta komponentista, jotta se toimisi paremmin mobiilinäkömässä. Lisäksi jatkoin kanban-taulumme täyttämistä ja määrittelin millä tavoin esitän tiettyjä asioita. Esittelytavoiksi valitsimme tiimin kanssa värikoodauksen, joka on erittäin yksinkertainen tapa lajitella kortteja tietyn aihealueen alle. Kanban-taulumme sijaitsee avoimen lähdekoodin applikaatiossa nimeltä Wekan.

Tiistai 22.10.2019

Työpäivän alussa:

Muokkaan viimeisen vanhan komponentin toimivaksi uuden ominaisuuden kanssa. Oletan että tämän työtehtävän kanssa minulla menee vain aamupäivä ja iltapäivästä pystyn keskittymään uuteen projektiin.

Työpäivän lopussa:

Muokkauksessa kului aamupäivä, jokainen komponentti on muokattu toimivaksi. Huomenna olen ajatellut testaavani kaikki komponentit, jotta ne toimivat halutulla tavalla eikä ilmaantuisi minkäänlaisia virhetilanteita. Iltapäivä minulla oli erilaisia dokumentointitehtäviä. Suurin osa projektimme alkupään tehtävistä on dokumentointia sekä keskustelua. Päätämme useista eri aiheista, kuten ohjelmointisäännöistä sekä versionhallinnasta.

Keskiviikko 23.10.2019

Työpäivän alussa:

Tänään aion suorittaa testauksia muokkaamiini komponentteihin. Niiden tarkoituksena on tarkistaa, että kaikki toiminnot ovat kunnossa ja toimivat halutulla tavalla. Erilaiset virheet sekä bugit kirjaan ylös dokumenttiin ja kaikkien testauksien jälkeen korjaan mahdolliset ongelmat.

Työpäivän lopussa:

Sain käytyä läpi kaikki komponentit ja löysin useammankin ongelman, jotka korjaan huomisen aikana. Olin kokonaan unohtanut aamusta, että meillä oli iltapäivästä palaveri uudesta projektista. Palaverissa keskustelimme viime viikon analyysissa mainitsemistani asioista kuten projektin kulusta ja dokumentoinnista. Lisäksi puheenaiheeksi palaverissa nousivat ohjelmointikielet, jäsenien rooli projektissa sekä työskentelytavat.

Torstai 24.10.2019

Työpäivän alussa:

Päiväni kuluu eilen löytämieni bugien sekä virhetilanteiden korjauksissa. Tavoitteenani on saada kaikki korjatuksi päivän aikana. Korjausten jälkeen pystyisin saamaan valmiiksi version, jota voitaisiin esitellä kehitysympäristön kautta. Version esitleminen olisi tärkeää, sillä sen myötä saisin henkilöiltä mielipiteitä millä tavoin tekemääni ominaisuutta voisi parantaa tai muuttaa.

Työpäivän lopussa:

En saanut tänään tehtyä versiota kehitysympäristöä varten, mutta sain korjattua suurimman osan virheistä. Virhetilanne mikä minulta jäi tänään tekemättä, oli roolitus ja siihen liittyvät haasteet. Aion korjata tämän virheen huomisen aikana ja lisäksi koetan saada tehdyksi kehitysympäristöön version valmiiksi maanantaita varten.

Perjantai 25.10.2019

Työpäivän alussa:

Jatkan tänään roolituksen parissa. Korjattuani tämän ongelman aion tehdä projektistani version kehitysympäristöön, jotta pystyisimme esitlemään tekemääni ominaisuutta ja saamaan siitä mielipiteitä. Mielipiteiden ja ehdotusten pohjalta tulen muokkaamaan ominaisuutta ja tekemään siitä seuraavan version.

Työpäivän lopussa:

Päiväni ei kulunut suunnitelmien mukaan. Aamulla tein Wekan:ssa sijaitsevaan kanbantauluun muutaman lisäyksen linkkaamalla siihen tietoja toisesta taulusta, tämä aiheutti ongelmatilanteen, joka rikkoi taulumme kokonaisuudessaan. Tämän ongelman ratkaisemisessa minulta kului koko työpäivä, enkä pystynyt keskittymään uuden ominaisuuden työtehtäviin.

Viikkoanalyysi

Tämäkin viikko on sisältänyt monenlaisia työtehtäviä, joista jokainen on ollut omanlaisensa. Sain viikon aikana tehtyä uuden ominaisuuden lähes valmiiksi asti, ainoastaan rooliutus ja sen tuomat vaikeudet jäivät minulta tekemättä muiden työtehtävien takia. Lisäksi uuden projektin myötä olen pystynyt laajentamaan työtehtäviäni sekä tekemään niistä monipuolisempia.

Projektissa työskenteleminen alusta asti on ollut vaikuttavaa ja opettavaista, sillä ainoat projektikokemukseni tätä ennen on ollut kouluissa käytävistä projektikursseista. On ollut mielekästä nähdä millä tavoin työmaailmassa edetään projekteissa sekä mitä kaikkia alkuvalmisteluja ja dokumentaatio tarvitsee tehdä ennen kuin pystytään aloittamaan konkreettinen ohjelmointityö. Projektimme ei vielä edennyt siihen asti, että pystyisimme aloittamaan ohjelmoinnin. Olemme tekemässä päätöksiä useastakin eri aiheesta, päätösten jälkeen pystymme valmistelemaan projektin rungon eli asentamaan esimerkiksi tiedossa olevat kirjastot sekä komponentit, joita aioimme hyödyntää.

Tämän viikon perjantaina kohtaamani ongelma Wekanin kanban-tilaus oli mielenkiintoinen ja hämmentävä. Ongelman syyksi lopulta paljastui, että linkatessani korttia, kortti ei saanut viiteavainta-arvoa, jolloin tauluun tullessa Wekan ei pystynyt hakemaan tai näyttämään taulussa olevia kortteja. Ongelman ratkaiseminen vaati huomattavan määrän tutkimista useastakin eri lähteessä, joihin kuului muun muassa Wekanin GitHub:n ongelmasivostio (Wekan 2019) sekä Stack Overflow -foorumi. Monen tunnin lukemisen ja tutkimisen jälkeen ymmärsin, että ongelman syynä oli viiteavaimen-arvo, joka oli jäänyt tyhjäksi Wekanin käyttämään MongoDB-tietokantaan. MongoDB on kauan aikaa laajassa käytössä oleva tiedostopohjainen ei SQL-syntaksi käyttävä tietokanta. Tietokannan merkittäviä ominaisuuksia on muun muassa se, että tietokanta koostuu kokoelmista, joiden sisällä sijaitsevat tiedostodokumentit. (Guru99 2019).

Wekanimme pyörii Linuxissa ja pääsemme käsiksi sen sisältöön PuTTY-ohjelman kautta. PuTTY:n kautta kirjauduin sisään Linuxiin ja siirryin Wekanin kansioon, missä sijaitsee kaikki siihen liittyvät asiat kuten tässä tapauksessa etsimäni tietokanta. Yritin tutkia tietokannan sisältöä, mutta aluksi en oikein ymmärtänyt, miksi komentoni eivät toimineet, sillä en ollut ennen ollut tekemisissä MongoDB:n kaltaisen tietokannan kanssa. Hetken tutkimisen jälkeen ymmärsin, millä tavoin komentoja täytyy antaa tietokannalle, että sieltä saataisiin haluttua tietoa. Hain Wekanin tiedoista `db.cards.find().pretty()` – komennolla näkyville kaikkia tietokannassa olevien korttien tiedot ja sieltä etsin kortin jolla oli kenttä viiteavaimelle, mutta se oli tyhjä. Löydettyäni oikean kortin otin id:n talteen ja suoritin komennon `db.cards.remove("_id" : "ID")`, jonka avulla pystyin poistamaan kortin kannasta.

Tämän jälkeen käynnistin Wekanin vielä uudestaan "sudo systemctl restart snap wekan.wekan"- komennolla, minkä jälkeen kanban-taulu alkoi taas toimimaan.

Kokonaisuudessaan viikko on ollut mieleenpainuva ja kiinnostava. Viikon aikana opin uusia asioita monesta aiheesta kuten MongoDB:stä ja PuTTY:n käytöstä. Ongelmien ja virheiden myötä oppiminen tapahtuu parhaiten, sillä korjausta tehdessä pääsen keskittymään aiheeseen syvällisesti, jolloin käsiteltävä aihe jää hyvin tietoisuuteen. Tämä myöskin kasvattaa osaamisaluettani, jonka pohjalta ammatillinen kehitykseni jatkuu.

3.9 Seurantaviikko 9

Maanantai 28.10.2019

Työpäivän alussa:

Korjaan päivän aikana viime viikolla huomaamani ongelman roolituksessa. Rooleja ei ollut alun perin tarkoitettu lisätä useaa komponenttiin, joten joudun muuttamaan tämänhetkistä rakennetta. Uskon, että saan päivän aikana tehdyksi toimivan version, jonka pystyn laittamaan kehitysympäristöön iltapäivällä tai viimeistään huomenna.

Työpäivän lopussa:

Sain tehdyksi haluamani muutokset rooleihin sekä roolitukseen. Nyt voidaan käyttää useampaa roolia samassa komponentissa. Henkilön roolien avulla määritetään minkälainen ulkoasu on komponentilla, mitkä asiat ja toiminnot ovat näkyvissä. Sain myöskin tehdyksi version valmiiksi kehitysympäristöä varten.

Tiistai 29.10.2019

Työpäivän alussa:

Tänään lisään eilen tekemäni version kehitysympäristöön. Tämän jälkeen aloitan testaamaan kaikkia toimintoja sekä varmistan, että responsiivisuus on kunnossa. Mahdollisten ongelmien ilmaantuessa, kirjaan nämä ylös ja korjaan tilaisuuden tullen.

Työpäivän lopussa:

Sain lisättyä version aamusta kehitysympäristöön, kirjautuminen ei toiminut. Ongelmaksi ilmaantui, että olin unohtanut projektin web.config-tiedostoon lisätä yhden rivin. Lisäsin tämän kohdan tiedostoon, jonka jälkeen pystyin taas kirjautumaan sisään normaalisti ja suorittamaan testauksia. Huomasin muutaman responsiivisuusongelman, jotka kirjasin ylös. Pyysin myös kollegaani testaamaan toimintoja ja hän puolestaan löysi useammankin toiminnallisen bugin, jotka olisivat mielekästä korjata ennen kuin esittelemme tätä isommalle käyttäjäkunnalle.

Keskiviikko 30.10.2019

Työpäivän alussa:

Korjaan päivän aikana eilen löydettyt bugit sekä ongelmat. Niiden korjaamisen jälkeen teen uuden version kehitysympäristölle ja pyydän kollegoitani tekemään uuden nopean testauskierroksen. Lisäksi yritän päivän aikana keskittyä uuden projektin tehtäviin.

Työpäivän lopussa:

Aamupäivän aikana sain korjattua kaikki ilmentyneet ongelmat sekä laitoin uuden version kehitysympäristöön pyörimään. Testausta tehdessä versiosta ei löytynyt minkäänlaisia ongelmia. Tulemme jakamaan tämän version tietyille käyttäjryhmälle loppuviikosta, jos mitään muuta ei ilmaannu aiheeseen liittyen. Iltapäivä minulla kului uuden projektin tehtävissä. Yksi tehtävistä, jonka otin työkseni oli TSLint:n tutkiminen, asennus sekä ohjelmointisääntöjen määrittelemine. Tämä tehtävä ei tullut vielä valmiiksi asti, joten jatkan tästä aiheesta huomenna.

Torstai 31.10.2019

Työpäivän alussa:

Aloitan päiväni TSLint:n kanssa. Eilen sain tutkittua sekä asennettua TSLint:n uuteen projektiimme. Tänään aion määritellä perussäännöt ohjelmointitavoille. Tavoitteenani on saada TSLint toimimaan perussääntöjen kanssa.

Työpäivän lopussa:

Päiväni kului uuden projektin parissa, tällä hetkellä olen määritellyt perussäännöt ohjelmointikäytännöille. Käytäntöihin kuuluu muun muassa pakolliset puolipisteet sekä aaltosulkeet, kun käytetään metodeja if-ehto, switch-ehto tai for-looppia. Lisäksi päivän aikana huomasin, että projektissamme olevat npm-paketit eivät olleet uusimmissa versioissa, joten suoritin kaikkiin näihin päivityksen. Olemme sopineet projektiryhmämme kanssa, että yritämme pitää kaikki paketit uusimmassa versiossa niin kauan kuin se vain on mahdollista projektin kannalta. Pakettien päivitys tuotti hieman vaikeuksia, sillä niiden päivittäminen ei onnistunut vain yksinkertaisilla komennoilla. Jouduin tekemään useita eri vaiheita, jotta pystyin saamaan paketit uusimpiin versioihin.

Perjantai 01.11.2019

Työpäivän alussa:

Jatkan tänään uuden projektissa ja laitan kehitysympäristöön päivittämäni version jakoon muutamalle käyttäjälle ohjelmistotuessa. He ovat ensimmäisiä henkilöitä, jotka tulevat testaamaan uutta luomaani ominaisuutta. Yritän saada heiltä mielipiteitä mihin suuntaan ominaisuutta olisi hyvä kehittää. Ajattelin aamun aikana kirjoittaa eilen tekemästäni pakettien päivityksestä ohjeen muille projektimme jäsenille, että jokainen meistä pystyisi itsenäisesti suorittamaan tarvittavat päivitykset tulevaisuudessa.

Työpäivän lopussa:

Aamusta sain kirjoitettua ohjeen pakettien päivityksestä. En kuitenkaan antanut ohjelmistotuen henkilöille oikeuksia päästä testaamaan uutta ominaisuutta. Kollegani oli keskustellut muutaman henkilön kanssa tekemästani ominaisuudesta sekä sen tämän hetkisistä toiminnoista. He olivat tulleet siihen tulokseen, että ominaisuus vaati muutaman lisätoiminnon, ennen kuin sitä kannattaa edes esitellä talon sisäisesti. Nämä toiminnot ovat pitkälti tarkoitettu helpottamaan asiakkaan käyttöä. Tulen työskentelemään näiden toimintojen kanssa ensi viikosta alkaen.

Viikkoanalyysi

Toiseksi viimeinen päiväkirjan seurantaviikko on ohitse. Viikossa oli monipuolisuutta, sillä työtehtävät jakautuivat vanhoihin tuttuihin tehtäviin sekä uusiin tehtäviin uudessa projektissa. Osa viikostani kului viimeistellessäni uuteen ominaisuuteen liittyviä asioita kuten roolitusta. Saatuani tämän valmiiksi pääsin tekemään versiota kehitysympäristöön, joka on aina kiinnostavaa, sillä sitä tehdessä kohtaa mitä erilaisempia ongelmatilanteita. Uuden projektin tiimoilta pääsin tutustutamaan TSLint:iin, minkä ideana on määritellä ohjelmointisääntöjä projektille. Sääntöjen avulla pystytään pitämään kaikkien ohjelmoijien koodi samantapaisena. Lisäksi viikon aikana päivitin projektissamme olevat npm-paketit uusimpaan versioon ja tein tästä ohjeen, jonka avulla muutkin pystyvät toteuttamaan päivityksen.

Mielestäni tällä viikolla työtehtäväni ovat olleet mielekkäitä tehdä. Tehtävien pohjalta olen päässyt oppimaan uusia asioita kuten npm-komentoja sekä Angular-projektin rakennetta. On hienoa oppia uusia komentoja hyödyntäen npm, sillä monet komennoista nopeuttavat sekä helpottavat ongelmatilanteissakin. Tällä viikolla päivittäessäni paketteja en pystynyt päivittämään paketteja millään yksinkertaisella komennolla kuten esimerkiksi `npm update` tai `ng update`. Tutkin tätä asiaa netistä ja löysin hyvän artikkelin, jonka avulla pystyin saamaan työtehtävän tehdyksi. Flavio Scopes:n (Flavio Scopes 2018) artikkelissa kerrotaan yksinkertaisesti, millä tavoin saadaan kaikki paketit päivitettyksi. Aluksi pitää ajaa komento `npm-outdated`, jonka avulla nähdään, jos jokin paketti ei ole uusimmassa versiossaan. Tämän jälkeen asentamaan paketti nimeltä `npm-check-updates` ja käytetään komentoja `ncu -u`, joka päivittää kaikki tarpeelliset tiedot `package.json` tiedostoon sekä muihin tarpeellisiin paikkoihin. Näiden toimenpiteiden jälkeen voidaan käyttää komentoja `npm update` sekä tarpeen mukaan `npm install`, jolloin saadaan päivitetty tai ladattua kaikki uusimmat versiot projektissa olevista paketeista. Tämä artikkeli oli isona apuna yksinkertaiseen työtehtävään ja sen pohjalta tein ohjeen projektimme jäsenille.

Viikon aikana työskentelin myös TSLint parissa. Olin itse valinnut tämän työtehtävän itseleni, sillä omasta mielestäni tämä aihealue on kiinnostava ja olen kanssa tarkkana koodin muotoilusta sekä yleisestä formatoinnista. Tämäkin aihe oli entuudestaan tuntematon minulle. Lähdin tämän kanssa liikkeelle tutkimalla TSLint:n GitHub-sivustoa (TSLint 2019), sekä sieltä löytyvää listaa erilaisista ohjelmointisäännöistä. Tein tehtävää varten dokumentin, johon kirjasin kaikki säännöt, jotka mielestäni olisi hyvä lisätä projektiimme tehtyäni listan valmiiksi esittelin nämä säännöt projektimme jäsenille. Heidän mielestään kaikki valitsemani säännöt olivat järkeviä, sekä auttavat koodin ylläpitämisessä. Seuraavaksi asensin TSLint:n ja määrittelin jokaisen säännöt tslint.json- tiedostoon, jonka jälkeen tein testauksia, joilla varmistin, että jokainen sääntö toimii halutulla tavalla.

TSLint sekä muut ohjelmointisääntötyökalut ovat tärkeitä tukitoimintoja monellekin ohjelmointiprojektille. Näiden avulla pystytään hallinnoimaan koodia sekä sen muotoa ja tuottamaan yhteneväistä koodia, vaikka ohjelmoijia olisi useampiakin. Sääntöjen avulla estetään turhia formaatti-virhetilanteita muun muassa silloin kuin tehdään toimintojen yhdistämistä eri versiohallinnan haaroista. Bellairs (Bellairs R 2019) mainitseekin artikkelissaan, että ohjelmointisääntöjen työkalujen avulla pystytään automaattisesti huomaamaan mahdolliset ongelmatilanteet koodissa itsessään sekä muotoilussa. Hänen mielestään näiden työkalujen käyttäminen on tärkeää. Niiden myötä vähentää virhetilanteita, parannetaan kokonaisuudessaan koodin laatua sekä nopeutetaan kehitystä, koska ongelmatilanteet tulevat heti esille. Alapuolella taulukko Bellairs:n (Bellairs R 2019) mainitsemista hyvistä sekä huonoista puolista ohjelmointisääntöjen työkaluihin viitaten (Taulukko 2).

Taulukko 2. Ohjelmointisääntötyökalujen hyvät ja huonot puolet

Hyvät puolet	Huonot puolet
Tarkistaa suuren määrän erilaisia asioita	Voi tuottaa ongelmia sekä virheitä turhista asioista
Muotoilun tarkistus vastaa aina parhaita muotoilutapoja	Tekee tarkistuksia, mutta ei niin sanotusti opeta kehittäjälle, miten muotoilu kuuluisi tehdä
Edullinen	Vaikuttaa kehittäjän tuottavuuteen

3.10 Seurantaviikko 10

Maanantai 04.11.2019

Työpäivän alussa:

Viikkoni alkaa viime perjantaina keskusteltujen lisätoimintojen tekemisellä. Lisätoiminnot helpottavat loppukäyttäjän työtä ja nopeuttavat tiettyjen asioiden tekemistä. Tavoitteenani on saada yksi lisätoiminnoista valmiiksi tänään.

Työpäivän lopussa:

Päivän kuluessa sain haluamani lisätoiminnon valmiiksi. Testasin sitä usealla eri käyttötavalalla ja tulin siihen tulokseen, että se toimii oikein. Lisätoiminto hakee henkilölle listaan useasta eri paikasta kaikki hänelle tarpeelliset tiedot. Näitä tietoja pystytään hyödyntämään uudessa ominaisuudessa.

Tiistai 05.11.2019

Työpäivän alussa:

Tänään teen muut tarvittavat lisätoiminnot valmiiksi. Jos saan nämä valmistuvat, niin pystyn saamaan uuden version kehitysympäristöä varten. Tämän version tarkoituksena on esitellä uutta ominaisuutta lisätoimintoihin talon sisäisesti. Palautteen perusteella tehdään mahdollisesti muutoksia ennen kuin esittelyä asiakkaille.

Työpäivän lopussa:

Sain päivän aikana lähes kaikki halutut toiminnot tehdyksi. Ainoa toiminto mitä en saanut tehdyksi oli aikaan perustuvat säännöt eli ajankohdat, milloin jokin toiminto voidaan suorittaa. Esimerkiksi nappia A ei voi painaa, jos tietyistä toiminnosta on kulunut x aikaa. Teen tämän toiminnon loppuun huomenna ja lisään toimivan version kehitysympäristöön.

Keskiviikko 06.11.2019

Työpäivän alussa:

Jatkan eilen kesken jäänyttä aikasääntöjen tekemistä. Tehtyäni tämän valmiiksi ja testatuani sen teen version kehitysympäristöön. Saadessani version kehitysympäristöön suoritan vielä muutaman testauksen, nähdäkseni, että kaikki toiminnot ovat kunnossa.

Työpäivän lopussa:

Aikasääntöjen tekemisessä minulla kului suurin osa päivästä. Sain kuitenkin tehdyksi uusimman version kehitysympäristöön sekä suoritettua erilaisia testauksia. Kaikki tämän hetkiset toiminnot ovat toiminnassa ja huomenna koetan saada asetettua sekä annettua muutamalla henkilölle oikeudet tekemääni versioon. Oikeuksien myötä he pääsevät testaamaan uutta ominaisuutta toimintoihin ja esittelemään sitä muille käyttäjille.

Torstai 07.11.2019

Työpäivän alussa:

Päiväni kuluu henkilöiden oikeuksien sekä kirjautumisen kanssa. Keskustelin kollegani kanssa, että voisimme mahdollistaa kirjautumisen tekemääni versioon Azure AD:n kautta. Tämän myötä olisi helpompi antaa määritellä uusia käyttäjiä sekä myöntää heille oikeuksia. Päiväni tavoitteena on mahdollistaa kehitysympäristön versioon kirjautuminen Azure:n kautta.

Työpäivän lopussa:

Päiväni ei mennyt aivan suunnitellusti en saanut Azurea kokonaisuudessaan toimimaan. Henkilö pystyy tällä hetkellä kirjautumaan Azureen oikeilla tunnuksilla ja hänet ohjataan kehitysympäristöömme, mutta sisäänkirjautuminen sovellukseen estetään tuntemattomasta syystä. Osa päivästäni kului uuden projektin parissa, sillä päätimme aloittaa käyttämään Git:ä versionhallintatyökalunamme ja jokainen ryhmämme jäsen otti sen päivän aikana käyttöönsä. Huomenna osa päivästäni menee Azure-ongelman ratkaisemisessa ja osa uuden projektin parissa sekä Git:n lopullisessa määrittelyssä.

Perjantai 08.11.2019

Työpäivän alussa:

Tarkoitukseni oli aloittaa päivä Azure-ongelman kanssa, mutta katsoin äsken sähköpostiani ja kollegani oli korjannut tämän ongelman eilen illalla. Tämän takia päiväni kuluu uuden projektin parissa sekä Git:n määrittelyssä.

Työpäivän lopussa:

Päiväni meni pitkälti Git:n määrittelyssä ja toimintojen opetteluun kanssa. Kohtasimme tiimin kanssa useita ongelmia, kun yrittäessämme saada kloonattua projektiamme jokaisen henkilön koneelle. Ongelmien takia aloimme myös miettimään, kannattaisiko meidän vielä miettiä versionhallintatyökaluamme. Tulimme lopulta siihen tulokseen saadessamme ongelmamme korjattua, että Git on paremmalla pohjalla oleva versionhallintatyökalu meidän

projektillemme. Lisäksi mielestämme Git tarjoaa paremman käyttökokemuksen komentojen oppimisen jälkeen.

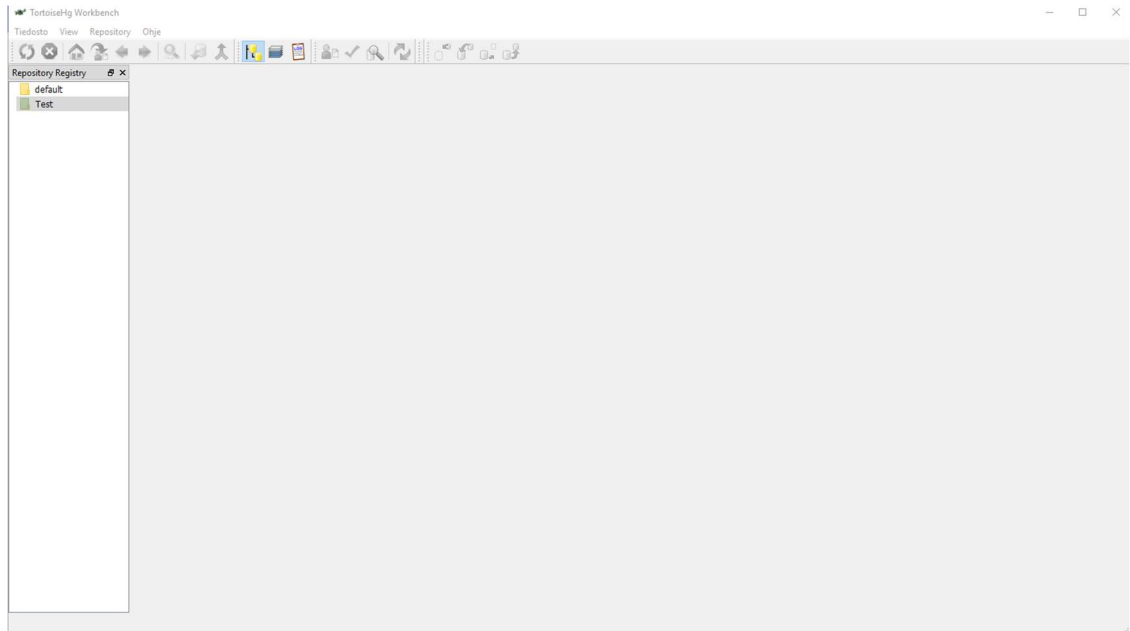
Viikkoanalyysi

Viimeisen päiväkirjaviikkoni aikana tehdyt työtehtävät olivat suurimmaksi osin vanhoja tuttuja työtehtäviä. Nämä sisälsivät front-endiä Angular:lla sekä C# back-endiä. Uudempina työtehtävinä viikolla oli Azure AD:n - tehtävät sekä versiohallinnan eri osa-alueet. Kaikista parhaiten mieleeni viikon tehtävistä jäi Azure sekä versiohallinta-työtehtävät. Kummattakin näistä aiheista ovat olleet ennen tiedossani, mutta en ole ennen perehtynyt kumpaankaan niin paljon kuin tällä viikolla. Versiohallinta on ollut käytössäni koko työsuhteeni ajan ja olen tottunut käyttämään TortoiseHq-nimistä versiohallintatyökalua. Tämän työkalun käyttäminen on itselläni sisältänyt pelkästään perustoimintoja eli projektin tiedostojen hakua, viemistä sekä yhdistämisoperaatioita. Tästä syystä on ollut avartavaa tutkia Git:n toimintaa ja nähdä sen tuomat hyödyt uudelle projektillemme.

Versiohallinnalla on merkittävä merkitys nykypäivän ohjelmistokehityksessä. Versiohallinnan avulla pystytään hallinnoimaan sovelluksen eri versioita. Osa voi olla kehityksessä olevia uusia ominaisuuksia sisältäviä tai julkaisukunnossa olevia versioita. Chacon ja Straub (Chacon S & Straub B 2019) mainitsee kirjansa alussa, että versiohallinta on järjestelmä, joka tallentaa tiedostoissa tapahtuvia muutoksia. Näihin muutoksiin pystytään pääsemään myöhemminkin, jos siihen on tarvetta. Kirjassa myös mainitaan yhdestä merkittävästä ominaisuudesta, jonka avulla on helppoa ja nopeaa palauttaa tehtyjä muutoksia.

Perjantaina yrittäessämme määritellä Git:n toimivaksi projektin jäsenillä, aloimme keskustelemaan vaihtoehtoisesti TortoiseHq:n käytöstä Git sijaan. TortoiseHq on kokoelma työkaluja sekä laajennuksia ja se toimii käyttöliittymä Mercurial-nimiselle versiohallintajärjestelmälle (TortoiseHq 2019). Keskustelun myötä päätimme tutkia aihetta, sekä tehdä yhteenvedon ja päätöksen, kumpi näistä kahdesta oli parempi meidän käyttöömmek. Tutkimme aiheita useilta keskustelufoorumeilta, sekä luimme dokumentaatiota kummastakin. Lisäksi otimme tämän puheenaiheeksi vanhempien kollegoiden kanssa. Keskustelun ja lukemisen myötä pääsimme siihen lopputulokseen, että projektin versiohallinta kannattaa suorittaa Git:llä. Syyksi miksi Git tuli valituksi oli, että sillä pystytään paremmin hallinnoimaan työhaaroja ja niiden versioita versiohistoriassa. Lisäksi kommentoja löytyy suuria määriä ja näiden avulla pystytään suorittamaan mitä erilaisempia toimintoja, jotka helpottavat muun muassa yhdistämisoperaatioissa. Alapuolella kuva TortoiseHq:n käyttöliittymästä (Kuva 6).

Kuva 6. TortoiseHg käyttöliittymä



Azure:n kanssa työskentely viikon aikana on ollut opettavaista, sillä en ole ennen päässyt perehtymään Azureen ja sen toimintaan. Azure AD on tunnistautumispalvelu, joka sopii toiminnoiltaan eri kokoisille yrityksille aina mikroyrityksistä suuriin yrityksiin asti. Azure:n avulla pystytään mahdollistamaan henkilön pääsy applikaatioihin kirjautumisen kautta. Applikaatiot voivat olla Microsoftin omia palveluita tai muita pilvipalveluissa olevia sovelluksia (Owen 2019). Ropponen (Ropponen S. 2019) on myös kirjoittanut samasta aiheesta ja hän mainitsee kirjoituksessaan, että Azure on globaalikirjautumisalusta. Sen tuomat lisäpalvelut tarjoavat mahdollisuuden moneen muuhunkin toimintaan. Mielestäni molempien artikkelien pohjalta saadaan hyvä kuva, siitä minkälaisia käyttömahdollisuuksia Azure AD:ssa on. Tulen tulevaisuudessa hyödyntämään paljon enemmän kaikkia sen lisäpalveluita sekä niiden tarjoamia mahdollisuuksia toteuttaa asioita.

Kokonaisuudessaan viimeinen päiväkirjaviikkoni sisälsi laajan määrän erilaisia työtehtäviä, joista osa oli niin sanotusti normaaleja tehtäviä ja osa aivan uusia. Opin viikon aikana uusia piirteitä Angular:sta sekä ymmärsin millä tavoin Azure AD ja versionhallinta toimivat. Viikko on ollut hyödyllinen tulevaisuuteni kannalta, sillä tulen varmasti kohtamaan työtehtäviä, jotka sijoittuvat joko Azureen tai versionhallintaan.

4 Pohdinta ja päätelmät

Päiväkirjaopinnäytetyöni on lopuillaan. Olen kirjoittanut yhdentoista viikon ajan päiväkirjaa työtehtävistäni. Opinnäytetyön kirjottamisen alussa olin ollut töissä harjoittelujakson sekä kesätyön ajan ja olin siirtynyt vakituisen työsopimuksen alaiseksi. Vakinaistamisen myötä sain mahdollisuuden laajentaa työtoimenkuvaani ja tällä hetkellä olen vastuussa yrityksen mobiilisovelluksen kehittämisestä. Päiväkirjaviikkojen aikana opin merkittävästi uusia asioita tutuista ohjelmointikielistä sekä frameworkeista.

Kehitys:

Opinnäytetyön aikana kehittymiseni on ollut merkittävää. Mainitsinkin, että kirjoittamisen alussa olin siirtynyt vakituisen työsuhteeseen ja saanut mahdollisuuden uusin työtehtäviin. Tehtävien myötä opin uusia asioita Angular:sta, TypeScript:stä, C#:sta sekä tietokannasta ja tietokantapyynnöistä. Lisäksi ymmärsin millä tavoin front- sekä back-end keskustelevat keskenänsä.

Viime kuukausien aikana pääsin luomaan uusia ominaisuuksia, sekä toimintoja, joista osa on lähitulevaisuudessa asiakkaiden käytössä. Näitä tehdessäni pääsin opiskelemaan monesta eri aiheesta teoriaa, jota pystyin myös lopulta hyödyntämään viikoittaisissa analyysseissäni. Merkittävimpiä aiheita, joita kohtasin opinnäytetyöni aikana, olivat kaikki autentikointiin sekä salasanoihin liittyvät alueet. Nämä alueet ovat mielestäni erittäin kiinnostavia ja haastavia. Tämän myötä olen päättänyt keskittyä laajasti yrityksen mobiilisovelluksen sekä uuden aloittamamme projektimme kirjautumispuoleen. Tulen toteuttamaan kumpaankin näistä kaksivaiheisen autentikoinnin sekä mahdollisuuden kirjautua sisään usealla eri tavalla.

Ratkaisumallit ja menetelmät

Päiväkirjan kirjoittamisen aikana olen oppinut käyttämään muutamaa työmenetelmää. Aloitin muutaman viikon kirjoittamisen jälkeen pitämään dokumenttia, johon kirjaan ylös keskustelujen myötä ilmaantuvia työtehtäviä sekä korjausehdotuksia. Tämän avulla pystyn pysymään mukana, mitä kaikkea minun tarvitsee tehdä ja unohtamatta tehdä jotakin tehtävää. Uuden projektimme pohjalta olen alkanut hyödyntämään kanban-taulua myös omissa työtehtävissäni. Loin uuden projektimme alussa itselleni yksityisen taulun, minne kirjaan ylös ideoitani sekä ajatuksiani.

Oppiminen

Päiväkirjan kirjoittamisen aikana opin paljon useasta aiheesta. Jouduin opinnäytetyön takia sisäistämään monia aiheita paremmin, pystyäkseni kirjoittamaan niistä

päiväkirjamerkinnöissä sekä viikkoanalyseissä. Opinnäytteen kirjoittamisen myötä olen pystynyt, kirjoittamaan laajemmin sekä kattavammin tekstiä uuden projektimme aloitusdokumentteihin.

Uudet aiheet ja tulevaisuuden hyödyt

Viimeisten kuukausien aikana olen huomannut laajan määrän erilaisia itselleni uusia aihepiirejä. Näihin lukeutuu muun muassa eri ohjelmointikielet, tietokannat sekä niiden toiminta. Olen saanut mahdollisuuden työskennellä usean aiheen kanssa, ja tämä on kasvattanut motivaatiotani kehitystyöhön. Yksi aiheista, johon tulen tulevaisuudessa keskittymään laajemmin on autentikointi sekä liittyvät aiheet kuten salasana- ja salausalgoritmit. Tämä aihe on aina ollut minusta kiinnostava ja olen aina halunnut päästä sisäistämään ai-
hettä syvällisemmin. Työni pohjalta olen nyt tilanteessa pystyäkseen keskittymään haluamiini aiheisiin. Aion parantaa kehittämäni sovellusten autentikointipuolta huomattavasti. Tulen työskentelemään muidenkin aiheiden piirissä kuten front-end:ssä Angular:n kanssa. Kuukausien ajan Angular:n on ollut keskipisteessä työtehtävissäni, joten kaikki mitä olen tähän mennessä oppinut siitä, tulee olemaan suureksi avuksi tulevaisuuden työtehtävissä.

Jatkokehitys

Työni avulla pystytään saamaan hyvä käsitys, minkälaista ohjelmoijan arki on yrityksessä. Viikkoanalyseissä on käsitelty viikon aiheita kokonaisuudessaan sekä jokaisessa analyysissä on esillä tietty teema kuten IP-osoitteet. Kirjoittamiani analyyssejä sekä niiden lähteitä voidaan hyödyntää tulevaisuudessa muissa opinnäytetoissa tai artikkeleissa.

Analysoinnin hyödyntäminen

Viikoittaiset päiväkirjamerkinnät sekä viikkoanalyysit ovat olleet erittäin hyödyllisiä työni kannalta. Olen usein palannut tarkistamaan opinnäytetyöstäni päivittäisiä merkintöjäni, nähdäkseni milloin olen tehnyt mitään työtehtävää ja mihin tilanteeseen kyseinen työtehtävä on edennyt. Lisäksi olen käyttänyt viikkoanalyseissa olevia lähteitä päivittäisissä työtehtävissäni, sillä monet löytämistäni lähteistä ovat olleet monipuolisia sekä esittäneet aihe-alueen laaja-alaisesti. Tulen varmasti myös tulevaisuudessakin palaamaan tähän dokumenttiin, sillä uskon, että jossakin työtehtävissäni tulen kohtamaan tilanteita, joita olen käsitellyt kirjoittaessani tätä opinnäytetyötä.

Lähteet

Argon2 2019. The Password hash Argon2, Winner of PHC

Luettavissa: <https://github.com/p-h-c/phc-winner-argon2> Luettu: 13.09.2019

Bacrana J. 2015. 5 Project-Planning Tips to Help You Meet Your Goals.

Luettavissa: <https://www.entrepreneur.com/article/246435> Luettu: 19.10.2019

Bellairs R 2019. Why is Linting Important? And How To Use Lint Tools.

Luettavissa: <https://www.perforce.com/blog/qac/why-linting-important-and-how-use-lint-tools> Luettu: 02.11.2019

Chacon S & Straub B 2019. Pro Git – Everything you need to know about Git.

Luettavissa: <https://git-scm.com/book/en/v2> Luettu: 09.11.2019

Dan Arias 25.04.2018. Hashing Password: One-Way Road to Security.

Luettavissa: <https://auth0.com/blog/hashing-passwords-one-way-road-to-security/> Luettu: 13.09.2019

Dan Arias 31.05.2018. Hashing in Action: Understanding bcrypt

Luettavissa: <https://auth0.com/blog/hashing-in-action-understanding-bcrypt/> Luettu: 13.09.2019

Dymond J 2015. 7 Things to Do Before Starting Any Project

Luettavissa: <https://innovationmanagement.se/2015/03/09/7-things-to-do-before-starting-any-project/> Luettu: 19.10.2019

Flavio Scopes 2018. Update all the Node dependencies to their latest version.

Luettavissa: <https://flaviocopes.com/update-npm-dependencies/> Luettu: 02.11.2019

Goalkicker 2018. Angular 2+ Notes for Professionals

Luettavissa: <https://goalkicker.com/Angular2Book/> Luettu: 07.09.2019

Guru99 2019. What is MongoDD? Introduction, Architecture, Features and Example

Luettavissa: <https://www.guru99.com/what-is-mongodb.html> Luettu: 26.10.2019

Jose Carlos 2016. Understanding two-way data binding in Angular 2.

Luettavissa: <https://medium.com/@bartsis/understanding-two-way-data-binding-in-angular-2-a8eafa850637> Luettu: 07.09.2019

Joseph D. Booth 2019. Angular Succinctly.

Luettavissa: <https://www.syncfusion.com/ebooks/angular-succinctly/scoring> Luettu: 07.09.2019

Margaret Rouse 2019. Single Sign-On (SSO)

Luettavissa: <https://searchsecurity.techtarget.com/definition/single-sign-on> Luettu: 28.09.2019

Michael Fitzgerald 2012. Chapter 1. What is a Regular expression .Introducing Regular Expressions by Michael Fitzgerald

Luettavissa: <https://www.oreilly.com/library/view/introducing-regular-expressions/9781449338879/ch01.html> Luettu: 20.09.2019

Miloni Shah 2017. What exactly is Angular 2?

Luettavissa: <https://www.quora.com/What-exactly-is-Angular-2> Luettu: 07.09.2019

Nodejs 2018. What is npm?

Luettavissa: <https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/> Luettu 12.10.2019

Owen 2019. What is Azure Active Directory (AAD).

Luettavissa: <https://www.otava.com/reference/what-is-azure-active-directory-aad/> Luettu: 09.11.2019

Patel Priyesh 2018. What exactly is Node.js?

Luettavissa: <https://www.freecodecamp.org/news/what-exactly-is-node-js-ae36e97449f5/> Luettu: 12.10.2019

Ropponen S. 2019. Azure AD on globaali kirjautumisalusta – ja paljon muuta.

Luettavissa: <https://www.tietokeskus.fi/azure-ad-on-globaali-kirjautumisalusta-ja-paljon-muuta/> Luettu: 09.11.2019

Stack Overflow 2018. Ng: Command not found while creating new project using angular-cli

Luettavissa: <https://stackoverflow.com/questions/37227794/ng-command-not-found-while-creating-new-project-using-angular-cli/55179821#55179821> Luettu: 12.10.2019

SAP Community 31.05.2013. What is Saml2?

Luettavissa : <https://blogs.sap.com/2013/05/31/what-is-saml2/> Luettu: 28.09.2019

Svetlin Nakov and Team 2013. Fundamentals of Computer Programming with C#

Luettavissa: https://introprogramming.info/english-intro-csharp-book/read-online/chapter-1-introduction-to-programming/#_Toc362296373 28.09.2019

TortoiseHq 2019. TortoiseHq - Documentation Chapter 2.

Luettavissa: <https://tortoisehg.readthedocs.io/en/latest/intro.html> Luettu 09.11.2019

TSLint 2019. TSLint

Luettavissa: <https://github.com/palantir/tslint> Luettu: 02.11.2019

Wekan 2019. Wekan – Open Source Kanban

Luettavissa: <https://github.com/wekan/wekan/issues> Luettu: 26.10.2019

WhatIsMyIPAddress 2019. Without IP Addresses, the Internet Would Disappear

Luettavissa: <https://whatismyipaddress.com/ip-address> Luettu: 04.10.2019