

## Koodarin päiväkirja

Julius Porkka

Opinnäytetyö  
Tietojenkäsittelyn koulutusohjelma  
2018



<b>Tekijä(t)</b> Julius Porkka	
<b>Koulutusohjelma</b> Tietojenkäsittelyn koulutusohjelma	
<b>Opinnäytetyön otsikko</b> Koodarin päiväkirja	<b>Sivu- ja liite-sivumäärä</b> 3+26
<b>Opinnäytetyön otsikko englanniksi</b> Diary of the programmer	
<p>Koodarin päiväkirja käsittelee kirjoittajan työhönliittyvää arkea. Kirjoituksessa seurataan oman osaamisen käyttämistä sekä uusien asioiden soveltamista liittyen arkisiin pulmatilanteisiin. Kirjoittaja on töissä it alan yrityksessä ohjelmistokehitys -yksikössä. 10 viikon seurantajakson aikana henkilö opettelee uusia koodikirjastoja joita uudet projektit vaativat.</p>	
<b>Asiasanat</b> Java, JavaScript, React, Front-end, Back-end, Apache POI	

## Sisällys

1	Johdanto .....	1
2	Lähtötilanteen kuvaus .....	2
2.1	Oman nykyisen työn analyysi .....	2
2.1.1	Työtehtäväni .....	2
2.1.2	Tarvittava osaaminen .....	3
2.2	Osaamiseni .....	4
2.3	Kehittyminen .....	4
2.4	Sidosryhmät työpaikalla .....	5
2.5	Vuorovaikutustaidot työpaikalla .....	5
2.6	Työviikkoni .....	6
3	Päiväkirjaraportointi .....	7
3.1	Seurantaviikko 1 .....	7
3.2	Seurantaviikko 2 .....	10
3.3	Seurantaviikko 3 .....	14
3.4	Seurantaviikko 4 .....	15
3.5	Seurantaviikko 5 .....	18
3.6	Seurantaviikko 6 .....	20
3.7	Seurantaviikko 7 .....	23
3.8	Seurantaviikko 8 .....	24
3.9	Seurantaviikko 9 .....	25
3.10	Seurantaviikko 10 .....	26
4	Pohdinta ja päätelmät .....	28
	Lähteet .....	29

# 1 Johdanto

Opinnäytetyö alkaa 19.02.2018, jolloin raportointi työpäivistä alkaa. Tämä tulee kestämään 12 viikkoa eteenpäin.

Päiväkirjatyypin opinnäytetyön raportointi tapahtuu päivittäisellä työtehtävien kuvaamisella ja viikoittaisella analyysillä.

Työtehtävissä tarvittava tietoperusta on tietynlaisten alkeiden osaaminen ohjelmoinnista ja tieto niiden kehittämiseen. Itsensä kehittäminen on tässä työssä pääroolissa. Ammattikirkallisuutta on olemassa ohjelmistokehitykseen liittyen, mutta omalta osaltani tämä kirjallisuus perustuu Stack overflow sivuston läpikäymiseen, sekä ohjelmakielen dokumentaation läpikäymiseen. Näiden lähteiden taustalla on se, että tässä roolissa törmää asioihin joita on ennakkoon vaikea valmistautua.

Toimin IT –alan yrityksessä tittelillä ”Software Developer”. Työympäristöni on konttori, joka kuuluu yritykselleni.

## 2 Lähtötilanteen kuvaus

### 2.1 Oman nykyisen työn analyysi

#### 2.1.1 Työtehtäväni

Työtehtäviäni on kuluvaan projektiin ennalta määritettyjen tehtävien toteutus. Projektiin on määritetty tietyt toteutettavat ohjelmapalaset jotka tulee toteuttaa projektin haluttuun lopputulokseen päästessä.

Tärkeä rooli on vuorovaikutus testaaajien kanssa. Ohjelmapalasen X ollessa mielestäni valmis testaaajat lähtevät testaamaan sitä ja tämän jälkeen vuorovaikutus testaaajien kanssa liittyy toimivuuteen, heidän löytämiin havaintoihin ja mahdollisten määrittelyiden porsaanreikien selvittämiseen sekä läpikäyntiin.

Tuotantoon siirtymiseen liittyvät läpikäynnit toteutusten osalta kuuluu usein tehtäviini. Projektissa koodin mennessä tuotantoon on käytävä läpi, että kaikki mahdolliset toiminnalliset palaset sekä niihin liittyvät materiaalit ovat mukana matkassa tekemistäni kilkkeistä, jottei tuotannossa tule yllätyksiä. Asennusohjeiden läpikäynti omien ohjelmapätkien kohdalta kuuluu samaan vastuualueeseen. Asennusohjeet toimitetaan Cloud asentajille jotka ovat vastuussa koko Cloud palvelusta jossa ohjelma pyörii.

Tuotannossa vastaantulevien virheiden/bugien korjaus koodiin, kuuluu omien tuotosteni kannalta tehtäviini. Tuotannossa voi tulla ilmi asioita joita ei testi ympäristöissä kohdattu, koska tuotanto on täyspäiväisessä käytössä käyttäjillä. Bugien ilmetessä niistä tulee korjauspyyntö, jonka toteutan liittyen omatekemiini ohjelmapätkiin. Bugin ollessa korjattu siitä siirretään korjattu versio tuotantoon.

Ohjelmakomponenttien dokumentointi on tärkeää, koska kukaan ei ole projektissa ikuisesti ja ohjelmisto jää jatkokehitykseen elämään. Todennäköisesti tulevaisuudessa tulee asioita esille tuotannossa joihin ei osattu varautua, niin kun aikani projektista jättää niin sen hetkisten tekijöiden on jollain tasolla saatava dokumentti käteen jossa kerrotaan edes karkealla tasolla mitä sen ohjelmapätkän on tarkoitus tehdä jotta korjaus olisi jotenkin inhimillistä tehdä käyttämättä viikkoja koodin ymmärtämiseen.

Olen aktiivinen yrityksen sisällä ja tykkään tehdä erilaisia asioita, välillä minua pyydetään tekemään erinäköisiä asioita jotka ei liity millään tapaa ohjelmointiin, enemmänkin ns "Promo" -asioihin, jotka tuovat kivaa vaihtelua silloin tällöin normaaleihin työrutiineihin.

Uusien henkilöiden perehdytys on ollut tehtäväni samoihin hommiin tuleville koodareille. Projektiin tulee jatkuvasti uusia tekijöitä ohjelmoimaan erilaisia palikoita ja mikäli tulevat samaan hommaan kanssani niin olen opastanut että mitä missäkin ympäristössä menee ja miksi jotta pääsevät mahdollisimman nopeasti alkuun ja toteuttamaan ohjelmistoa.

### **2.1.2 Tarvittava osaaminen**

Työtehtävissäni tarvitaan google -taitoja, niin kuin varmaan jokaisessa koodaus-tehtävässä jota kuka tahansa ohjelmoija tekee. Googlauksen ABC -kirja tulee olla luettuna. Vuorovaikutus tiimiläisten kesken tämänhetkisessä projektissa on välttämätöntä, agilella kun mennään niin määrittelyt saavat lisäyksiä lennossa tai saattavat muuttua kun huomataan että ei tämä näin voi toimia. Koodikolleegat, määrittelijät sekä testaajat ovat läheisimmät pelaajat tässä projektissa omalta osaltani.

Tähän projektiin liittyviin ohjelmointitehtäviin liittyviä osaamisalueita ovat, SQL tietokantojen osaaminen, Java ohjelmointi, sekä SVN versionhallinta.

Työn tekemisessä oma jatkuvan kehityksen parantaminen on kaikki kaikessa, melkeinpä jokapäivä törmää asiaan jota pitää google -kollegoilta kysyä.

Tietoja joita tarvitsen tietääkseni mitä teen on jonkinlainen näkemys että mitä koodipätkän pitäisi käytännössä tehdä ja mitä se sitten siellä lopullisessa ohjelmistossa tekee. Tämän lisäksi pitää ymmärtää termit joita tässä projektissa käytetään. Määrittelyt on tehty word dokumenteiksi jokaisesta ohjelma-palasesta, mutta jos jotain asiaa ei ymmärrä niin määrittelijää voi aina häiritä. Tekijän on ymmärrä määrittelyksestä suoraan mitä pitää tehdä, muuten määrittely on puutteellinen.

Tässä työtehtävässä tarvitsee vuorovaikutusta toisten tiimiläisten kanssa, sekä ongelmanratkaisutaitoja.

## 2.2 Osaamiseni

Osaamiseni vuoden jälkeen tehtävissäni on pessimistisen optimistisella ajattelulla hyvä suoriutuja, joka toki korreloi saamani tehtävän kanssa liittyen projektiin. Tietokannan päässä johon toteutetaan business logiikkaa ORACLE PL/SQL:llä alkaa olemaan osaltani ammattimaisella tasolla. En keksi asioita joita en pystyisi toteuttamaan tietokannan päähän jollain konsteilla, toki sellaisiakin varmasti löytyy. Java puolella on vielä paljon opiske- lua, en usko että java opiskelu loppuu koskaan. Kun tulin koulusta työelämän java-maail- maan niin ymmärsin että täällä ollaan paljon tarkempia että koodi on laadukasta sekä loo- gista. Enää ei toimi menetelmä "sain sen jotenkin toimimaan", joka on järkevää ja siten it- sekin olen kehittänyt loogiseksi koodariksi javan suhteen. Java -tasolla voin sanoa hyvin mielin olevani juniori. Kykenen javalla kehittämään tarvittavia palikoita, mutta usein kon- sultoin joissain määrin kokeneempia henkilöitä ja kysyn asioita joita googlesta en löydä. Konkarit auttavat mielellään ainakin meidän java velhot ovat kaikki hyviä tyyppejä joita voi häiritä aina.

Vuorovaikutustaidoissa alan olemaan jo osaaja, välillä törmää siihen että puhutaan sa- masta asiasta mutta keskustelun jälkeen molemmille jää täysin erikäsitykset, joten niitä on aina hyvä kehittää. Vuoropuheluissa tärkeintä on että osaa käyttää seuraavia fraaseja:

"Mitä ihmettä tarkoitat tuolla",

"Miksi tällä tavalla",

"Nyt en ymmärrä yhtään mitään, otetaan ihan step by step",

"Ymmärsitkö mitä hain? Hyvä, eli mitä hain?, eli et ymmärtänyt anna kun se- litän paremmin"

## 2.3 Kehittyminen

Ammatillisessa kehityksessäni olen tasolla junior. Koen että koodaamisessa hyvä koodari on koodari joka osaa toteuttaa alusta-asti itse asiat kaiken niistä ymmärtäen ilman suu- rempia murheita tai seinään törmäämisiä, sellainen en lähellekkään ole. Olen tehnyt pääni sisällä seuraavanlaisen koodarierarkian:

Aloittelija<Junior<Koodari<Senior<Nörtti<Velho

Toiminnassani oma tasoni näkyy siinä että osaan toteuttaa, asioita mutta törmään myös usein seiniin joista kuitenkin päästään läpi ja tarvitsen välillä kokeneemmilta konsultointia.

Tulevaisuudessa minun tulee panostaa ymmärtämiseen. Ymmärtäminen on tärkeintä, ei itse se että saa asian toimimaan epämääräisellä keinolla. Ennen törmäsi siihen että sai palikan toimimaan mutta ei osannut selittää miksi se nyt toimii sillä ratkaisulla, jolloin ei opi mitään eikä voi sanota että osaa kyseisen asian tämä on niin kutsuttua apinointia joka ei tee ihmisestä koodaria.

## **2.4 Sidosryhmät työpaikalla**

Omalta osaltani työni koskettaa tiimiläisiä sekä cloud tiimiä, jonka sovellusta räätälöimme asiakkaalle. Asiakkaalta tulee mahdollisia muutospyyntöjä, mutta en ole itse heidän kanssaan kontaktissa. Asiakasrajapinnasta olen saanut pysyä kaukana, joten tieto minulle tulee joko projektipäällikön tai määrittelijöiden kautta.

## **2.5 Vuorovaikutustaidot työpaikalla**

Vuorovaikutus projektilaisten kanssa hoituu 90 % ajasta skypellä, sillä tiimi on jakautunut ympäri suomea. Päivittäiset daily palaverit joissa kerrotaan mitä on työnalla ja onko ongelmia. Välillä vuorovaikutus tapahtuu kasvokkain konttorilla, jos projektipäällikkö sattuu löytämään minut tilasta jossa yleensä työskentelen. Oikean esimieheni kanssa palavereja sekä vuorovaikutuksia on erittäin harvoin.

Omalta osaltani asiakaspalveluun ja muihin sidosryhmiin kuin omiin tiimiläisiini vuorovaikutus on olematonta. Viestit asiakkaalta minulle toteutusmielessä kulkevat välikäden kautta jona toimii projektipäällikkö. Työtehtäviin liittyen kommunikoin täten ainoastaan projektilaisten tai räätälöimämme sovelluksen tuen kanssa kun on jotain tuotantoon liittyviä tai muita huomioitavia asioita.

Haasteet vuorovaikutuksessa omalta osaltani ovat, että ymmärrys ei aina ole yhteinen vaikka niin luullaan, kun on nörttien kanssa tekemisissä.

## **2.6 Työviikkoni**

Omalta osaltani työviikkoni on normaali 37,5h/viikko jonka lisäksi on kaksi kiky päivää. Työpäiväni koostuu kahdeksan tunnin rupeamasta, joka on joustavaa. Mikäli ajatus ei kulje keskellä päivää, niin on mahdollista lähteä harrastamaan ja palata sorvin ääreen illemmalla kun uskoo taas olevansa tehokas.

## 3 Päiväkirjaraportointi

### 3.1 Seurantaviikko 1

*Maanantai 27.8.2018*

Saavuin konttorille 08.30, olin saanut itseni aikaisin ylös sängystä. Konttorille tultuani katsoin livekartasta (älykäs rakennus), missä päin nuoret kollegat oleskelevat ja liityin heihin.

([https://campaigns.tieto.com/empathicbuilding?qclid=EA1aIQobChMI37SjjJjd3AIVWO-WaCh2KJQaMEAYASAAEqKvevD\\_BwE](https://campaigns.tieto.com/empathicbuilding?qclid=EA1aIQobChMI37SjjJjd3AIVWO-WaCh2KJQaMEAYASAAEqKvevD_BwE))

#### Base



Your space digital twin, 3D Model of your space layout, different areas and zones visualized, location based content management (infos, services, instructions, security)

#### Space



Find free desks and know when they have been used, Find available rooms and know where they are, Find free parking place, Know your utilization rates, Navigate

#### People



Manual or real-time location sharing, Find your colleagues, Find your guests/visitors, Create and manage profile, Share your skills and wills, Share your documentation, Heat map of space usage, share your thoughts, have fun and collaborate, Notifications

#### Service tickets



#### Air Quality



#### Voice



Tällä hetkellä projektinani on tuottaa asiakkaalle hakemus/lomake web – applikaatiota. Tänään työstämäni koodi hakee front-endistä JSON tietoa Java pohjaiseen back-endiin, jolla on tarkoitus parsea JSON, etsiä sieltä oleellista dataa jolla luodaan PDF raportteja. Eli logiikkana on:

1. Lähetä JSON dataa Back-endiin
2. Pura JSON data
3. Riippuen raporttityypistä, joka ilmenee JSON tiedosta, luodaan Javalla erilaisia PDF raportteja

*Tiistai 28.8.2018*

Tiistai alkoi saman projektin merkissä kuin edellinenkin päivä. Tänään tehtäväksi otin erilaisten excel raporttien tuottamisen käyttäen JSON dataa sekä Javan Apache POI – kirjastoa. Käyttäjän klikatessa UI sivulla erilaisia vaihtoehtoja. Esimerkki:

"luo raportti vuosittaisesta rahamäärästä", niin koodin pitäisi luoda oikean tyyppinen excel tiedosto, joka lähetetään back-endiltä takaisin front-endiin. Käyttäjän selain ehdottaa tallenna/lataa vaihtoehtoa, jolloin käyttäjä saa tämän tiedoston itselleen.

Aloin työstämään uudenlaista raporttia johon pitäisi tulla 3 eri sheettiä, joihin jokaiseen eri datasta muodostettuja chartteja. Kahteen ensimmäiseen välilehteen tulee luoda bar chartit ja kolmanteen pie chart. Olen aikaisemmin luonut jo raportteja käyttäen Apache POI kirjastoa joka on tehty juuri tätä varten, en kuitenkaan ole tehnyt chartteja joka alkuun vaikuttaa monimutkikkaammalta kuin normaalin tekstin syöttö sarakkeisiin.

Lähdin googlettamaan erilaisia vaihtoehtoja hakusanoin "Apache poi bar chart example", josta päädyin: ([https://www.programcreek.com/java-api-examples/index.php?source\\_dir=ScratchPad-master/src/org/apache/poi/hssf/usermodel/HSSFChart.java](https://www.programcreek.com/java-api-examples/index.php?source_dir=ScratchPad-master/src/org/apache/poi/hssf/usermodel/HSSFChart.java))

Paremmän esimerkin löysin: (<https://stackoverflow.com/questions/38913412/create-bar-chart-in-excel-with-apache-poi>)

Jonka vastauksessa oli suurinpiirtein suoraan kopioitava esimerkki joka luo taulun datasta joka oltiin tähän esimerkkiin kovakoodattu. Lähdin testailemaan tätä oikealla datalla josta nuo chartit tulee tehdä. Päivä ehti loppua kesken ja lähdin nauttimaan auringosta.

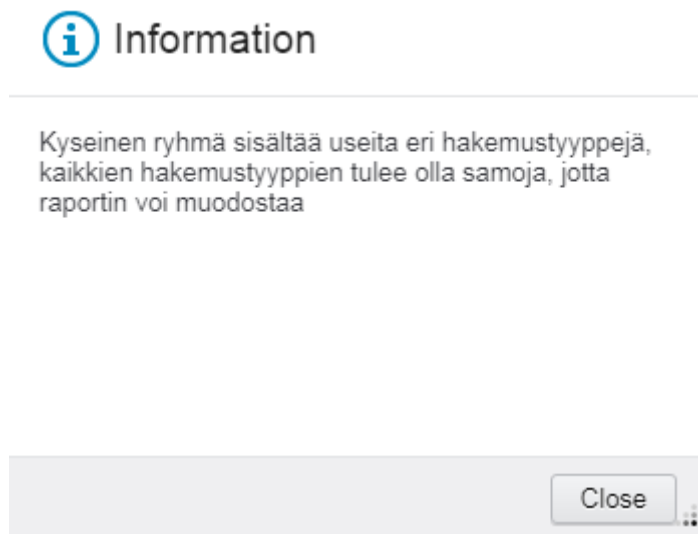
*Keskiviikko 29.8.2018*

Keskiviikkona en jatkanut charteilla, vaan jätin ne hautumaan koska niiden deadline ei ole vielä tulossa. Tänään katsoin käyttötapausta jossa oltiin kerrottu:

Käyttäjän pitää pystyä luomaan excel rapotti niin että on mahdollista valita haluamansa kentät siihen. Kyseinen raportti eroaa muista raporteista siten, että ei voida kovakoodata exceliin etukäteen. Kentät tulee lähettää käyttäjän valitsemina parametreina back-endiin

josta niihin haetaan tiedot ja muodostetaan excel. Aloin toteuttamaan logiikkaa jossa UI:ssa aukeaa modal window käyttäjälle, joka sisältää hakemusryhmään liittyvien hakemusten kentät. Nämä kentät tulevat checkbokseina joita käyttäjä ruksii. Kun tämä oli tehty niin aloin rakentamaan validointia. Validoinnin jälkeen tarkoituksena on toteuttaa tietojen haku sekä excel luonti, tämä checkbox valinta tulee myös pystyä tallentamaan tietokantaan jotta käyttäjällä on mahdollista jatkossa luoda samanlaisia raportteja ettei hänen tarvitse eriskeen aina checkboksista ruksia haluamiaan kenttiä vaan on jo valmiina erikseen nimetty pohja.

Kuva validoinnista.



*Torstai 30.8.2018*

Päivä meni apache poi esimerkkeihin perehtyessä. Sain tehtyä minimaalisia toteutuksia, jotta ymmärsin kirjaston logiikan. Toteutin tämän ohella raportteja.

*Perjantai 31.8.2018*

Perjantaina meni poikkeuksellisissa merkeissä, sillä pidin vapaapäivän. Seuraavalla viikolla jatkuu ohjelmointi siitä mihin torstaina jäi.

### ***Viikkoanalyysi viikko 1***

Viikko oli monipuolinen ja tekemistä riitti, tutustuin uusiin asioihin koodien sekä valmiskirjastojen puolella. Viikolla riitti toteutettavaa ja jiran backlog on vielä pitkä. Aika kului paljon googlailuun uusien asioiden osalta ja niiden ymmärtämiseen.

## 3.2 Seurantaviikko 2

*Maanantai 03.9.2018*

Toteutin front-endiä jossa käytetään DOJO nimistä JavaScriptia. Tarkoitukseni on luoda uuteen ikkunaan käyttäjälle eri vaihtoehtoja (kenttiä henkilötiedot, pankkitiedot, lainat jne.)

Jokainen tietotyyppi sisältää eri määrän kenttiä esimerkiksi pankkitiedot voivat sisältää: Henkilö nimi, henkilötunnus, pankkinumero

Jos kaikki kentät laitettaisiin tuohon ikkunaan allekkain niin niitä olisi saman tien siinä auki useita satoja, siksi päätin jaotella ne omien otsikoidensa alle. Eli tarkoitukseni on tuottaa ikkunaan painikkeet allekkain esim "Pankkitiedot" josta aukeaa checkbox vaihtoehdot käyttäjälle joiden otsikkoina on eri pankkitiedot. Tämän tulisi toimia niin että painamalla näkyy vaihtoehdot ja uudelleen painaessa ne menevät piiloon.

Löysin tähän JS kieleen seuraavasta sivusta hyvän esimerkin kuinka buttoni ja functiot toimivat DOJO:ssa: (<https://dojotoolkit.org/reference-guide/1.10/dijit/form/Button.html>)

Oikealla on lähtötilanne eli olin laittanut kenttiä vain allekkain ja vasemmalla on button josta kentät aukeavat onClick jälkeen.

Ennen onClick:

Reporting Fields

**Hakija**

Pankki

**Kasittelija**

- Hakemusnumero
- Hakemustyyppi
- Avustuslaji
- Palvelukokonaisuus
- Tila
- Toimiala
- Valmistelija
- Vastuualue

Add Items Cancel

Jälkeen onClick:

Reporting Fields

**Hakija**

Pankki

- Hakijan nimi
- Hakijan tyyppi
- Henkilötunnus
- Rekisterinumero
- Y-tunnus

**Kasittelija**

- Hakemusnumero
- Hakemustyyppi
- Avustuslaji
- Palvelukokonaisuus
- Tila
- Toimiala
- Valmistelija
- Vastuualue

Add Items Cancel

Törmäsin ongelmaan jonka virheilmoitus oli " Uncaught Error: Tried to register widget with id but id is already registered". Tämä tapahtui, mikäli koitin mennä kuvan mukaiseen näkymään useita kertoja eri hakemuksilta. Virhe johtui siitä että, sessioniin oli jo luotu kyseinen

id, eikä se tuhoudu vaikka kyseisen ikkunan sulkee. Googlelta löytyi kuitenkin tähän ratkaisu jolla kyseisen id:n sai tuhottua. Lisäsin tämän tuhoamisen functioon jossa tämä tuhoetaan aina ennen klikkausta.

(<https://dojotoolkit.org/reference-guide/1.8/dojo/dom-construct.html>)

```
require(["dijit/form/Button", "dojo/dom", "dojo/domReady!"], function(Button, dom){
    require(["dojo/dom-construct"], function(domConstruct){
        // Destroy a node byId
        domConstruct.destroy("bankButtonNode");
    });

    // Create a button programmatically:
    var bankButton = new Button({
        label: "Pankki",
        onClick: function(){
            // Do something:

            console.log("Runs bank button");
            var x = document.getElementsByClassName('hakijaPankkiButton');
            if (x.CheckBoxListContainer1.style.display === "none") {
                x.CheckBoxListContainer1.style.display = '';
            } else {
                x.CheckBoxListContainer1.style.display = "none";
            }
        }
    }, "bankButtonNode").startup();
});
```

*Tiistai 04.9.2018*

Aloitin päivän menemällä asiakastestiympäristöön, jonne lomalta tullut senior oli eilen ilta-päivästä asentanut tekemäni sovellusmuutokset. Asiakastestiympäristössä pääsen katsomaan koodin toimivuutta relevantimmalla datalla kuin mitä minulla on omassa development ympäristössä. Eiliset JavaScriptit saivat mennä hetkeksi sivuun ja focus meni PDF raporttien luomiseen. Olin toteuttanut edellisellä viikolla eteenpäin PDF raportteja, eli asiakkaalla on 20 erilaista hakemusta joista jokainen eroaa X tavalla. Kävin katsomassa näkykö kukin hakemus niin kuin pitäisi. Osa Json parsetuista tiedoista ei ilmestynyt pdf hakemuksille, joten tutkin asiaa. Virheeksi paljastui, että json parsemiseen oli jäänyt whitespace.

(Metodin nimi yliviivattu, mutta voisi olla esimerkiksi: getTestiData)

```
public void getTestiData(JSONObject compensation) throws JSONException {
    if (compensation.has("compensationInfo") && compensation.get("compensationInfo")!=null) {
        JSONObject compensationInfo = (JSONObject) compensation.get("compensationInfo");
    }
}
```

Muita havaintoja oli varmistaminen, että data parsitaan oikeasta paikasta Json objectin alta, mikäli data ei ilmestynyt pdf raporttiin niin vertasin Chromen devtoolilla verraten java koodiin, että onko sama polku datan parsimiseen.

Chromen työkalu:

```
▼ <div class="md-resize-wrapper">
  <textarea readonly html-decoder ng-model=
    "data.compensation.compensationData.purpose" md-maxlength rows="1" class="ng-
    pristine ng-valid md-input ng-not-empty ng-valid-md-maxlength ng-touched" id=
    "input_101" aria-invalid="false" ng-trim="false" style="height: 30px;">
  </textarea> == $0
```

Json parsija Java:

(Oikeat nimet korvattu sanalla Testi)

```
public void getTestiInformation(JSONObject compensation) throws JSONException {
    if (compensation.has("compensationData") && compensation.get("compensationData")!=null) {
        JSONObject compensationData = (JSONObject) compensation.get("compensationData");
        addToListTesti("Testi", compensationData.optString("purpose"));
    }
}
```

*Keskiviikko 05.9.2018*

Keskiviikko jatkui siitä mihin tiistaina jäin, eli PDF raporttien tarkasteluun asiakastestiym-  
päristöstä. Iltapäivällä treenasin React.js kirjastoa katsomalla tutoriaaleja:

<https://www.youtube.com/watch?v=Ke90Tje7VS0>

Yritän päivä päivältä oppia mahdollisimman paljon uutta, jotta osaamiseni kasvaisi.

*Torstai 06.9.2018*

Torstaina kävin läpi samoja PDF raportteja joihin keskiviikkona jäin. Raporttien tarkaste-  
luun ja testailuun kului aikaa, tämä ei ole mielekkäintä tekemistä, mutta ne täytyy kuitenkin  
käydä läpi.

*Perjantai 07.9.2018*

Perjantaina tutustuin lisää JavaScriptiin sekä toteutin checkbox vaihtoehtoista raporttia  
eteenpäin. JavaScript on edelleen minulle asia jota en ole aikaisemmin toteuttanut, joten  
sen ymmärtämiseen kului aikaa.

## ***Viikkoanalyysi viikko 2***

JavaScriptin harjoittelu on mieleistä asiaa varsinkin kun tähtään full-stack kehittäjäksi, joten oli hyvä että sain käytettyä sen ymmärtämiseen aikaa. Viikko kului sekä JS/Javan merkeissä. Sain toteutettua asioita, opittua uutta. Raporttien tarkastelu oli melko puuduttavaa työtä, mutta muuten viikko oli mieleinen.

### **3.3 Seurantaviikko 3**

*Maanantai 10.9.2018*

Maanantaina tiimin testaaja sekä projektipäällikkö oli palannut kesälomalta. Kävimme läpi testausvalmiita asioita jotka menisivät testaajalle. Projektipäällikön kanssa kävimme läpi toteutettuja palasia ja pientä aikataulutusta milloin kaikki task listaltani olisi valmiina. Elokuun loppuun mennessä suurin osa tekemistäni asioista menisi tuotantoon joten aloin priorisoida niitä. Suurin osa päivästä kului PDF sekä excel -raporttien toteutukseen/korjaamiseen.

*Tiistai 11.9.2018*

Tiistaina tein 2 erilaista PDF raporttia jotka olivat jääneet aikaisemmin tekemättä. Nyt kaikki tarvittavat PDF raportit ovat valmiita (20kpl), toki testeissä tulee löytymään korjattavia asioita, mutta enää viilausta vailla. PDF raporttien jälkeen tein 2 kpl erilaisia excel raporttia valmiiksi. Nyt kaikki on valmista torstaita varten kun nämä menevät asiakastestiympäristöön.

*Keskiviikko 12.9.2018*

Keskiviikko oli vielä raporttien viimeistelyä. Raporttien ollessa toteutettuna jatkoin checkbox Excelin tekoa. Logiikan pohtimiseen ja koodin minimoimiseen sain käytettyä monta tuntia.

*Torstai 13.9.2018*

Aamulla varmistin että on olemassa toimivat versiot koodeista (Jar sekä JS) jotka menivät asiakasympäristöön päivällä. Asennuksen jälkeen kävin ympäristössä katsomassa

asiakasdatan avulla, että näyttääkö oikealta ja toimiiko toteutukset. Raporteista oli tullut asiakkaalta testituloksia että muutamia tietoja jäi puuttumaan. Lisäsin nämä asiakkaiden tekemät havainnot tulostusraporttien toteutukseen.

Perjantai 14.9.2018

Asiakkaalta oli tullut muutamia havaintoja edellispäivän asennuksen jälkeen, joihin paneuduin liittyen raporttien lopputulemaan. Tarkastin bugit, joihin ehdin vielä perjantain aikana toteuttaa korjaukset sekä vietyä ne versiohallintaan.

### ***Viikkoanalyysi viikko 3***

Moni henkilö tuli lomalta takaisin joten oli paljon läpikäyntiä nykyisen tilanteen puolesta. Saimme julkaistua uutta versiota asiakkaan ympäristöön joten asiakkaatkin saivat uutta ihailtavaa. Viikolla en oppinut uusia asioita, vaan aika kului lähinnä jo opittujen asioiden hyödyntämisessä.

### **3.4 Seurantaviikko 4**

*Maanantai 17.9.2018*

Aloin rakentamaan logiikkaa jo aikaisemmin aloittamaani ohjelmapalaseen.

1. Käyttäjä valitsee x määrän hakemuksia järjestelmästä
2. Käyttäjä valitsee x määrän tietoja jokaiselta hakemukselta excel raporttiin
3. Front-end lähettää käyttäjän valitsevat hakemukset sekä valitut kenttien nimet back-endille
4. Back-end hakee jokaiselta hakemukselta käyttäjän valitsevat tiedot
5. Back-end muodostaa näiden hakemusten tiedoilla excel raportin
6. Back-end toimittaa excel raportin Base64Byteinä front-endille
7. Front-end muodostaa Base64Byteistä back-endissä tehdyn excelin takaisin exceliksi
8. Selain kysyy käyttäjää tallentamaan/avaamaan excel tiedoston

Eniten mietintää aiheutui siitä millä tavalla excel vastaanottaa tiedon kentistä jotka käyttäjä haluaa raportille näkymään hakemuskohteisesti.

Lähdin kokeilemaan Javan Map rakenteella niin, että Key arvona toimii string ja jokainen key sisältää listan tietoa jotka ovat käyttäjän valitsevat kentät excel sarakkeisiin.

```
System.out.println(reportItemsMap);
for (Entry<String, ArrayList<String>> entry : reportItemsMap.entrySet()) {

    // if rownum=1 then we want to add headers for first line of excel workbook

    if (rownum == 1) {
        System.out.println(entry.getValue());
    }

    // rownum>1 means that headers are not placed and we can implement real data
    if (rownum > 1) {
        System.out.println(entry.getValue());
    }
    rownum++;
}
}
```

Console kuvakaappaus ensimmäinen rivi sisältää koko Mapin sisältäen key ja value arvot. 2-4 rivi sisältää tiedot jaoteltuna for loopissa joista excelin tulee muodostua.

```
{Otsikot=[Rekisterinumero, Hakemusnumero, Ytunnus], Line1=[020020, 199022, 0123], Line2=[040404, 666, 111J]}
[Rekisterinumero, Hakemusnumero, Ytunnus]
[020020, 199022, 0123]
[040404, 666, 111J]
```

Esimerkki excel -tiedostosta jonka Java on luonut. Mapin purkaminen excel-tiedostoksi on onnistunut. Ensimmäisestä rivistä tehtiin otsikot ja seuraavien key valueiden listat on purettu ABC sarakkeisiin.

	A	B	C	D	E
1	<b>Rekisterinumero</b>	<b>Hakemusnumero</b>	<b>Ytunnus</b>		
2	020020	199022	0123		
3	040404	666	111J		
4					
5					
6					
7					
8					
9					

Kuvakaappaus chromen dev työkalun console riviltä. Kyseinen Array lähetetään back-endille. Tämä Array sisältää käyttäjän valitsevat kentät excel-tiedostoon.

```
▶ (3) ["Henkilötunnus", "Rekisterinumero", "Y-tunnus"]
```

```
---
```

Javan vastaanottaessa kyseisen Arrayn se on purettava listaa jotta on mahdollista loopata myöhemmin läpi. Googletin kuinka lisätä array suoraan listaan parametrilla:

(<https://stackoverflow.com/questions/36560522/add-string-array-to-arraylist>)

Kuvakaappaus ratkaisusta (linkki), jota käytin oman koodin toteutukseen myös.

```
ArrayList<String> aList = new ArrayList<String>(Arrays.asList(question1));
```

*Tiistai 18.9.2018*

Tiistaina jatkoin siitä mihin maanantaina jäin. Tein valmiiksi logiikan jolla java osaa hakea objectstoresta json datasta oikeat tiedot kenttiin esim. "Rekisterinumero, Hakemusnumero". Eilisessä versiossa data oli kovakoodattua.

*Keskiviikko 19.9.2018*

Keskiviikkona logiikka toimii, excel raporttiin tulee kenttäotsikoiden perusteella hakemuksen tiedot jsonista. Tämän jälkeen jätin mvp version tästä odottamaan huomista testiympäristöön viemistä varten jotta asiakas voi kuitata olenko oikeilla jäljillä vai en. Määrittelyt tästä palikasta oli melko onnettomat, siksi on hyvä varmistaa. Asiakkaalta oli tullut edellispäivänä testaustulosta liittyen PDF raportteihin sekä excel raportteihin. Loppu päivän korjasin excel raportteja.

*Torstai 20.9.2018*

Asiakkaalta tuli tieto että edellispäivänä jättämäni mvp versio checkbox raportin luomisesta olisi oikeilla jäljillä ja lähdin jatkokehittämään sitä.

*Perjantai 21.9.2018*

Päivä kului saman raportin jatkamisessa, sekä muutamien bugien korjaamisessa jo toteutetuista raporteista.

#### ***Viikkoanalyysi viikko 4***

Tällä viikolla sain työskennellä suurimman osan ajasta javan parissa. Back-endin teko on miellyttävää ja logiikan tekemisestä siihen saa eniten "ahaa" -elämyksiä. Viikolla riitti tekemistä koodin parissa ja tällä viikolla siihen myös sai käytettyä tosi paljon aikaa. Yleensä

viikolla tulee paljon oheista tehtävää ohjelmoinnin ohelle, mutta tällä viikolla siltä vältyttiin hyvin. Kokonaisuus jota tällä viikolla tein oli tekemiseltään mielenkiintoista.

### 3.5 Seurantaviikko 5

*Maanantai 24.9.2018*

Excel raportteihin on määritetty erilaisia sorttauksia, joiden mukaan datan tulisi olla järjestyksessä esimerkiksi A-Z. Lähdin googlettamaan miten Apache POI:lla luotuihin excel tiedostoihin saisi erilaista sorttausta/filtteröintiä. Googlettamalla selvisi ettei Apache POI -kirjastossa ole sitä tukevaa tekniikkaa. Etsinnän jälkeen päätin toteuttaa tämän valmiiksi javan listan sisälle, johon löysin seuraavanlaisen ohjeen:

<https://stackoverflow.com/questions/2784514/sort-arraylist-of-custom-objects-by-property>

For sorting an `ArrayList` you could use the following code snippet:

```
Collections.sort(studList, new Comparator<Student>(){
    public int compare(Student s1, Student s2) {
        return s1.getFirstName().compareToIgnoreCase(s2.getFirstName());
    }
});
```

Kokeilin kyseisellä logiikalla simppeleä versiota jonka toteutus alla:

```
ArrayList<User> userList = new ArrayList<User>();

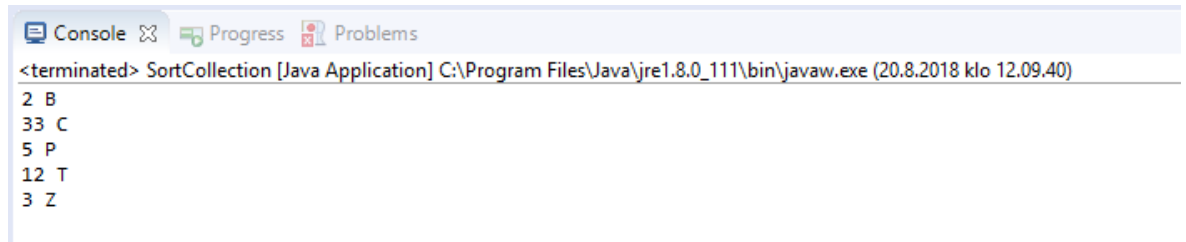
User u1 = new User ("3","Z");
User u2 = new User ("2","B");
User u3 = new User ("5","P");
User u4 = new User ("12","T");
User u5 = new User ("33","C");

userList.add(u1);
userList.add(u2);
userList.add(u3);
userList.add(u4);
userList.add(u5);

Collections.sort(userList, new Comparator<User>(){
    public int compare(User s1, User s2) {
        return s1.getName().compareToIgnoreCase(s2.getName());
    }
});

for (User u: userList) {
    System.out.println(u.getId()+" "+u.getName());
}
}
```

Kyseinen koodi uudelleen järjestää userit userListan sisällä joten saan haluamani lopputuloksen. Tässä esimerkissä järjestin ne nimen mukaiseen järjestykseen A-Z:



```
Console Progress Problems
<terminated> SortCollection [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (20.8.2018 klo 12.09.40)
2 B
33 C
5 P
12 T
3 Z
```

Lisäsin oikeaan toteutukseen seuraavalla logiikalla palikan. Toteutuksen palikka tosin sisältää oikean luokan (ei User joka on tässä demo -mallissa)

```
Collections.sort(userList, new Comparator<User>(){
    public int compare(User s1, User s2) {
        return s1.getName().compareToIgnoreCase(s2.getName());
    }
});
```

*Tiistai 25.9.2018*

Korjailin excel raportteja sekä PDF raportteja. Korjausten jälkeen jatkoin checkbox excel raportin toteutusta.

*Keskiviikko 26.9.2018*

Tuotin dokumentointia word muotoon tekemistäni raportti koodeista. Dokumentointi sisälsi logiikan ja ohjeistuksen kuinka raportti prosessi menee alusta loppuun asti.

*Torstai 27.9.2018*

Dokumentointi jatkui.

*Perjantai 28.9.2018*

Kolmas päivä dokumentointeja koodeista, jotta tulevan henkilön olisi helppo jatkokehittää kun vaihdan projektia. Dokumentoinnista tuli valmis.

### ***Viikkoanalyysi viikko 5***

Viikko oli onnistunut, sillä uusien ominaisuuksien toteuttaminen jo luotuihin toteutuksiin onnistui. Vanhoja jiroja oli vielä backlogissa joita jatkoin sen verran kun kerkesin. Viikko koostui sekä JavaScriptin sekä Javan toteuttamisesta. Loppuviikosta toteutin dokumentaatiot toteuttamistani kokonaisuuksista tämän projektin osalta.

### **3.6 Seurantaviikko 6**

*Maanantai 01.10.2018*

Uusi viikko alkoi suurilla haaveilla varustettuna. Viimeviikkoisen dokumentoinnin jälkeen olin valmis aloittamaan uudessa projektissa. Matkan varrella olin käynyt koulusta hake-  
massa react.js alkeet, jonka johdosta olin pystynyt ilmoittamaan tietäväni reactista perus-  
teet.

Uudessa projektissani esiintyvät seuraavat teknologiat/ympäristöt joita pääsen käyttä-  
mään fullstack devaajana:

Java (spring boot),  
React.js,

Kafka,  
MongoDb,  
Docker,  
Kubernetes,  
Azure.

Päivä kului tutustuessani uuteen arkkitehtuuriin sekä asentelemalla tarvittavia ympäristöjä ja työkaluja ajokuntoon.

*Tiistai 02.10.2018*

Projekti lähti tiistaina osaltani käyntiin toteuttamismielessä. Ensin luotiin Jhipster työkalua käyttäen valmis pohja Spring boot projektille (<https://www.jhipster.tech/>). Jhipster työkalu luo valmiin projektipohjan sisältäen tarvittavat dependencyt.

*Keskiviikko 03.10.2018*

Käyttöliittymästä oli olemassa demoversio jonka näköinen siitä pitäisi tulla. Tämä oli toteutettu mobirise työkalulla, jolla on helppo luoda käyttöliittymiä. Käyttöliittymä sisälsi dynaamisia formeja, joita oli generoitu Form.io työkalulla, tutustuin kyseiseen formi työkaluun sekä katsoin html koodista oleellisia osia.

*Torstai 04.10.2018*

Edellisenä päivänä olin perehtynyt käyttöliittymä esimerkkiin joka oli toteutettuna. Loin create-react-projektilla uuden react projektipohjan jonka päälle aloin rakentamaan toteutettavaa käyttöliittymää. Reactin SPA tulisi sisältämään laskutukseen liittyvän CRUD:in. Suurin osa laskuista tulee valmiina ja tämän uuden UI:n tarkoituksena on mahdollistaa käyttäjien manuaalisten erikoislaskujen lisääminen.

*Perjantai 05.10.2018*

React UI:n tekeminen jatkui perjantaina. Minulla ei ollut juurikaan kokemusta vielä reactista, joten tämän avulla pääsisin siihen kunnolla kiinni. Katsoin arkkitehdin kanssa mitä react "teknologioita" tähän olisi hyvä liittää ja Redux termi tuli eteeni. Perehdyin loppupäivän Reduxiin jotta saisin järkevän ymmärryksen siitä.

### ***Viikkoanalyysi viikko 6***

Tällä viikolla aloitin kolmannessa projektissani nykyisessä firmassa. Uusi projekti vaikutti teknologioiltaan mielenkiintoiselta ja uusien asioiden oppiminen motivoi minua kehittäjänä. Edellisessä projektissa kehitin uusia toimintoja jo olemassa olevaan projektiin ja nyt alkaneessa pääsin mukaan jo heti alusta eli taustalla ei ole muiden tekemää koodia. Tällä viikolla opettelin lisää reactia hyvien osaajien avustuksella ja haluan kehittyä tämän projektin myötä siinä osaajaksi.

### **3.7 Seurantaviikko 7**

*Maanantai 08.10.2018*

Maanantaina sain lisättyä redux:in projektiin. UI:sta oli nyt mahdollista luoda laskuja. Seuraavaksi otin työn alle olemassa olevien laskujen fetchaamisen apista.

*Tiistai 09.10.2018*

Toteutin datan hakemisen UI komponenttiin, joka renderöisi kyseisen datan taulunäkymään. Taulunäkymään käytin react-bootstrapia, jotta siitä tuli visuaalisesti kauniimpi.

*Keskiviikko 10.10.2018*

Tänään työnalla oli olemassa olevien laskujen muokkaus. Edellispäivänä tekemääni tauluun lisäsin jokaiselle riville "edit" buttonin. Päädyin ratkaisuun että "edit" buttonin onClick tapahtumasta aukeaisi modal ikkuna joka sisältää formin, joka populoidaan kyseisen laskun olemassa olevilla tiedoilla. Näitä tietoja tulisi olla mahdollista muuttaa sekä tallentaa.

*Torstai 11.10.2018*

Torstai kului arkkitehtuuriin tutustumisessa sekä siitä keskustellessa.

*Perjantai 12.10.2018*

UI:n rakentaminen jatkui ja sain tehtyä laskujen muokkauksen kuntoon. Seuraavana työnalle otin "delete" tapahtuman luomisen joka tulisi "edit" napin tavoin jokaisen rivin kohdalle.

### **Viikkoanalyysi viikko 7**

Tällä viikolla pääsin luomaan CRUD -operaatioita front-endin puolelle. Redux oli tullut minulle uutena asiana ja sen sisäistämisessä oli mennyt hetki. Oli kiva saada asioita tehtyä ja onnistua niiden teossa. Projekti on lähtenyt vauhdikkaasti käytiin sekä päivä päivältä opin uutta asiaa. Viikko oli erittäin onnistunut.

### **3.8 Seurantaviikko 8**

*Maanantai 15.10.2018*

Aamulla esittelin tekemiäni UI ratkaisuja laajemmalle yleisölle. Käytännössä olen tehnyt tätä kahdestaan arkkitehtien kanssa jotka ovat antaneet omia näkemyksiään matkan varrella. Tänään tähän tutustui myös toinen tiimi joka tekee samalle asiakkaalle sovelluksia. Kävimme läpi millä tavoin asiakkaan ulkoasujen tulisi olla yhtenäisiä.

*Tiistai 16.10.2018*

Käyttöliittymän CRUD operaatiot on valmiita (MVP) versiot. Seuraavaksi lähdin tutustumaan mahdollisiin autentikointi vaihtoehtoihin. Arkkitehti suositteli että lähtisin tutustumaan auth0 vaihtoehtoon (<https://auth0.com/>), joka käyttää JWT tokenia.

*Keskiviikko 17.10.2018*

Katsoin javascript puoleen liittyen tutorialeita, jotta pystyisin liittämään auth0 autentikaation UI ratkaisuumme.

*Torstai 18.10.2018*

Torstaina sain toteutettua auth0 ratkaisun react UI:hin. Auth0 ratkaisulla jwt token sekä käyttäjän kirjautumistiedot talletetaan selaimen "Local Storageen", jossa ne säilyvät tallessa sekä ovat käytettävissä.

*Perjantai 19.10.2018*

Perjantaina matkasin arkkitehtini kanssa Tampereen konttorille, jossa pääsin tutustumaan syvemmin käyttämäämme arkkitehtuuriin. Tampereen agendana oli tehdä esitys kirjoista/työkaluista/tuotteista joita yksikkömme käyttää softakehityksessä.

### **Viikkoanalyysi viikko 8**

Auth0 tutustuminen ja sen toteuttaminen react projektiin vei suurimman osan tämän viikon ajasta. En ollut aikaisemmin käyttänyt JWT tokeneita missään toteutuksessa. Samassa organisaatiossa työskentelevät tekijät opastavat mielellään näiden asioiden kanssa jos

törmään jossain kohtaa seinään. Tampereen arkkitehtuuri tapaamisessa törmäsin tosi monen uuteen termiin/teknologiaan joita meillä on käytössä.

### **3.9 Seurantaviikko 9**

*Maanantai 22.10.2018*

Reactista minulla ei ollut paljoa kokemusta ennen tätä projektia, mutta onneksi kollegoilta löytyy mielenkiintoa opastaa ja neuvoa "best practise" -tapoja. Tänään kävin läpi react koodiani, sekä sen rakennetta osaavampien henkilöiden kanssa.

*Tiistai 23.10.2018*

Muokkasin projektin reactin Redux rakennetta best practisen mukaiseksi (<https://github.com/erikras/ducks-modular-redux>). Edellä mainitulla rakenteella koodin jaottelu sekä tekeminen ovat visuaalisesti paremman näköistä, kuin action sekä reducerien tekeminen jokaiselle komponentillensa erikseen.

*Keskiviikko 24.10.2018*

Keskiviikkona aloin katsomaan miten saisin tehtyä crud Formeista/Resulteista dynaamisia, jotta sovelluksesta saisi "low coding effort" -tyylisen. Projektissa käytämme formeissa Form.IO työkalua ([www.form.io](http://www.form.io)), jonka api dokumentaatio löytyy (<https://documenter.getpostman.com/view/684631/formio-api/2Jvuks#intro>). Tämän projektin on tarkoitus olla POC, sekä uudenlainen alusta jolla pystymme tuottamaan asiakkaille nopeasti uusia SPA sovelluksia vähemmällä koodaamisella.

*Torstai 25.10.2018*

Torstai meni arkkitehdin mukana tutustuen syvemmin yksikköme arkkitehtuuriin. Käytännössä se tarkoittaa sitä että kolme arkkitehtia samasta organisaatiosta kokoontuu saman pöydän ääreen jakamaan näkemyksiään sekä kokemuksiaan.

*Perjantai 26.10.2018*

Perjantaina siirryin pois Windowsista käyttämään OSX käyttöjärjestelmää. Asensin uuteen läppäriini tarvittavat ohjelmat sekä ympäristöt.

## ***Viikkoanalyysi viikko 9***

Viikko oli mielekäs ja opin uusia vaihtoehtoja kauniimpien ratkaisuiden toteutukseen liittyen reactiin. Arkkitehtuuripalaverissa opin jälleen syvemmin koko pipe-line putkea. Opin uutta liittyen Azureen ja kafkaan. Tietoa tulee koko ajan lisää ja sitä mukaan koitan itse oppia soveltamaan sitä.

### **3.10 Seurantaviikko 10**

*Maanantai 29.10.2018*

Tänään katsoimme API:en toimivuuksia eteenpäin. Spring boot backendistä tulisi saada yhteys IBM:n tarjoamaan sääntökoneseen jonka kautta laskut menevät läpi. IBM:ssä tarkistetaan laskuille suunnattuja sääntöjä ja se palauttaa vastauksen onko lasku validi. Saimme testeillä GET metodin toimimaan, mutta POST:in toimivuudessa oli haasteellisuksia jsonin hankalan rakenteen vuoksi.

*Tiistai 30.10.2018*

Tiistai jatkui backend apien yhteyksien kanssa. Projektin backendillä tulee olla mahdollisuus tietojen hakuun usealta eri toimijalta (APE:ista), jotta kaikki tarvittavat tiedot saadaan käsittelyyn meidän sovelluksessamme.

*Keskiviikko 31.10.2018*

Keskiviikkona saimme tehtyä loppuun API yhteydet eri palveluihin. Meidän sovelluksemme saa haettua dataa ja vietyä sitä eri lähteisiin. Tämän jälkeen katsoimme unit testejä kuntoon.

*Torstai 01.10.2018*

Backend puolen toteuttaminen oli mennyt hyvin eteenpäin joten jatkoin tänään UI:n tuottamisella react projektissa. Katsoin autentikointia eteenpäin sekä, tutustuin miten JWT tokenin pitää kommunikoida backendin kanssa, jotta käyttäjän kirjautumisen voisi todentaa luotettavasti.

*Perjantai 02.10.2018*

Autentikoinnin tekeminen JWT tokenilla yhteensopivaksi backendin kanssa ei edellisen päivän jälkeen vielä toiminut joten jatkoin sitä perjantaina. Aiemmin minulla oli kokemusta springin securitystä, mutta jwt tokenia käyttäen etsin ratkaisuja googlaamalla.

### ***Viikkoanalyysi viikko 10***

Tämä viikko kului pääsääntöisesti backendiin tehtävien API:en ja niistä tehtävien kutsujen kanssa. Backend koodaus tapahtui yhteistyössä arkkitehdin kanssa joka opetti minulle hyviä best practise tapoja. Kokemukseni sekä osaamiseni kehittyi viikko viikolta ja viihdyn tällä hetkellä projektissani.

## 4 Pohdinta ja päätelmät

Tämän seurantajakson aikana olen oppinut paljon uutta, sekä tutustunut vanhoihin osamiseen syventävästi. Kaavat joilla etsin ratkaisua ovat pysyneet työn läpi samoina kuin aikaisemmin. Ennen näitä viikkoja minulla ei ollut kokemusta javascriptistä, jota päädin lopulta tekemään. Tämän kymmenen viikon jälkeen en osaa sanoa missä näen itseni viiden vuoden päästä muuta kuin koodaamassa.

## Lähteet

Empaattinen rakennus. Luettu 27.08.2018

[https://campaigns.tieto.com/empathicbuilding?gclid=EAlalQobChMI37SjjJjd3AIVWO-WaCh2KJQaMEAAAYASAAEgKvevD\\_BwE](https://campaigns.tieto.com/empathicbuilding?gclid=EAlalQobChMI37SjjJjd3AIVWO-WaCh2KJQaMEAAAYASAAEgKvevD_BwE)

Apache poi Chart esimerkki. Luettu 28.08.2018

[https://www.programcreek.com/java-api-examples/index.php?source\\_dir=ScratchPad-master/src/org/apache/poi/hssf/usermodel/HSSFChart.java](https://www.programcreek.com/java-api-examples/index.php?source_dir=ScratchPad-master/src/org/apache/poi/hssf/usermodel/HSSFChart.java)

Apache poi Bar chart esimerkki. Luettu 28.08.2018

<https://stackoverflow.com/questions/38913412/create-bar-chart-in-excel-with-apache-poi>

Dojo.js button esimerkki. Luettu 03.09.2018

<https://dojotoolkit.org/reference-guide/1.10/dijit/form/Button.html>

Dojo.js domConstruct esimerkki. Luettu 03.09.2018

<https://dojotoolkit.org/reference-guide/1.8/dojo/dom-construct.html>

React tutorial. Luettu 05.09.2018

<https://www.youtube.com/watch?v=Ke90Tje7VS0>

Java arraylistojen käsittely esimerkki. Luettu 17.09.2018

<https://stackoverflow.com/questions/36560522/add-string-array-to-arraylist>

Java objectien sorttaus esimerkki. Luettu 24.09.2018

<https://stackoverflow.com/questions/2784514/sort-arraylist-of-custom-objects-by-property>

Jhipster dokumentti. Luettu 02.10.2018

<https://www.jhipster.tech/>

Auth0 dokumentti. Luettu 16.10.2018

<https://auth0.com/>

Redux ducks github. Luettu 23.10.2018

<https://github.com/erikras/ducks-modular-redux>

Formio dokumentti. Luettu 24.10.2018

<https://documenter.getpostman.com/view/684631/formio-api/2Jvuks#intro>

