

Fortum Kaukolämpö Plus järjestelmän päivitysprosessi

Jaakko Uusitalo

Opinnäytetyö
Tietojenkäsittelyn koulutusohjelma
2020



| | |
|---|--|
| Tekijä(t) Jaakko Uusitalo | |
| Koulutusohjelma Tietojenkäsittelyn koulutusohjelma | |
| Opinnäytetyön otsikko Fortum Kaukolämpö Plus järjestelmän päivitysprosessi | Sivu- ja liitesivumäärä 82 + 1 |
| Opinnäytetyön otsikko englanniksi Upgrading process of Fortum Kaukolämpö Plus system | |
| <p>Työskentelen järjestelmäasiantuntijana Fortum Oy:llä Espoon toimistossa keilaniemen kauniissa merimaisemassa kolmannessa kerroksessa osana Kaukolämpö Plus järjestelmän kehitys- ja ylläpitotiimiä. Työtehtäviini kuuluu Kaukolämpö Plus järjestelmän ylläpito ja uusien toiminnallisuuksien kehittäminen.</p> <p>Opinnäytetyö käsittelee Fortumin Kaukolämpö Plus järjestelmän päivitysprosessia. Kaukolämpö Plus järjestelmä on Fortumin intranetissä toimiva web-aplikaatio, jonka tarkoituksena on tukea kaukolämmön myyntiä Fortumilla. Päivitysprosessi sisältää Kaukolämpö Plus järjestelmän ohjelmien päivittämistä uusille versioille. Tätä prosessia on seurattu opinnäytetyössä 10 viikon ajan ja siihen on kuvattu tarvittavat muutokset, joita versioiden päivitys on aiheuttanut järjestelmään.</p> <p>Tavoitteeksi asetin oppia web-järjestelmän päivitysprosessin. Halusin ymmärtää, mitä web järjestelmän päivitysprosessi pitää sisällään ja miten se toteutetaan. Lisäksi halusin lisätä kokemusta projektinhallinnasta käyttäen ketteriä kehitysmenetelmiä.</p> <p>Seurantajakson jälkeen alussa asettamani tavoitteet oli saavutettu. Tutustui myös hyviin ohjelmointikäytäntöihin ja -tapoihin, joista uskon olevan paljon apua tulevaisuudessa. Seurantajakson jälkeen huomasin olevani paljon kriittisempi työni jälkeen. Koen keskittyväni olennaiseen ja olemaan entistä tehokkaampi työntekijä.</p> | |
| Asiasanat ColdFusion, Fortum, ohjelmistokehitys, päivitysprosessi, web-järjestelmä | |

Sisällys

| | | |
|------|---------------------------------------|----|
| 1 | Johdanto | 1 |
| 2 | Lähtötilanteen kuvaus | 3 |
| 2.1 | Oman nykyisen työn analyysi | 3 |
| 2.2 | Lähtötasoni | 5 |
| 2.3 | Kehittyminen | 5 |
| 2.4 | Sidosryhmät työpaikalla | 6 |
| 2.5 | Vuorovaikutustaidot työpaikalla | 7 |
| 3 | Päiväkirjaraportointi | 9 |
| 3.1 | Seurantaviikko 1 | 9 |
| 3.2 | Seurantaviikko 2 | 19 |
| 3.3 | Seurantaviikko 3 | 27 |
| 3.4 | Seurantaviikko 4 | 32 |
| 3.5 | Seurantaviikko 5 | 37 |
| 3.6 | Seurantaviikko 6 | 42 |
| 3.7 | Seurantaviikko 7 | 50 |
| 3.8 | Seurantaviikko 8 | 55 |
| 3.9 | Seurantaviikko 9 | 61 |
| 3.10 | Seurantaviikko 10 | 65 |
| 4 | Pohdinta ja päätelmät | 70 |
| 5 | Lähteet | 73 |
| 6 | Kuvat | 79 |
| 7 | Liitteet | 80 |
| | Liite 1. Lyhenteet | 80 |

1 Johdanto

Työskentelen järjestelmäasiantuntijana Fortum Oy:llä Espoon toimistossa keilaniemen kauniissa merimaisemassa kolmannessa kerroksessa osana Kaukolämpö Plus järjestelmän kehitys- ja ylläpitotiimiä. Tiimiin kuuluu minun lisäksi kaksi muuta jäsentä, jotka toimivat Kaukolämpö Plussan change managereina eli muutosten hallitsijoina. Kirjoitan opinäytetyönä päiväkirjaa, jossa tulen kuvaamaan Kaukolämpö Plus järjestelmän päivitystä.

Työtehtävissäni tarvittava tietoperusta koostuu lähinnä eri ohjelmistokieliin dokumentaatiosta. Kaukolämpö Plus järjestelmä on rakennettu Adoben ColdFusionin päälle, joten virallisen dokumentaation tunteminen on oleellista. Adobe ColdFusion on ketterä web-ohjelmistojen kehitysalusta. Adobe ColdFusionin dokumentaatio löytyy kokonaisuudessa heidän virallisilta verkkosivuiltaan. (Adobe a.) Adobe ColdFusionin tukena on käytetty CFWheels kehitysrunkoa eli framework:ia, jonka avulla on toteutettu järjestelmän sivut sekä toiminnallisuudet. Tästä syystä työssä on olennaisempaa hyödyntää CFWheelsin kuin ColdFusion dokumentaatiota (CFWheels). On hyvä silti huomauttaa, että CFWheelsin dokumentaatio on osittain aika vajavaista, jolloin täytyy tehdä omat päätelmät ColdFusionin dokumentaation pohjalta. Kaukolämpö Plus järjestelmän sivuilla käytetään Bootstrap 3 framework:ia sivujen kehityksessä, jotta niiden luominen olisi tehokkaampaa ja sivun ulkonäkö pysyisi mahdollisimman samanlaisena sivulta siirryttyä toisaalle.

Kaukolämpö Plus järjestelmässä käytetty data on tallennettu Microsoftin SQL Serverille. SQL Server on relaatiotietokanta, jossa dataa voidaan jakaa tauluihin ja sitten linkittää toisiinsa (Microsoft b; Technopedia 2017). Hyvä perusteellinen tuntemus yleisimmistä kyselykielistä ei ole pahitteeksi, sillä dataa liikutellaan käyttöliittymän sekä palvelimen välillä. Kyselykieli on tietokannan datan hallintoihin erikoistunut kieli (Webopedia). Datan hallinnoiminen käsittää tiedon lisäämisen, muuttamisen ja poistamisen.

Kaukolämpö Plus järjestelmää päivittäessä täytyy ottaa huomioon järjestelmän eri toiminnot, koska yhden toiminnon päivittäminen saattaa säteillä negatiivisesti toisiin toimintoihin. Järjestelmän kehittäminen tapahtuu omassa kehitysympäristössä, jossa pyörii kehityseen tarkoitettu jaettu tietokanta. Tämä tarkoittaa sitä, että tuotantoon ajettavat muutokset täytyy kahdentaa tuotantotietokantaan.

Tavoitteenani on oppia web-järjestelmän perusteellinen päivitysprosessi. Päivitysprosessi tehdään valituilla metodeilla pysyen aikataulussa ja hyödyntäen projektin hallintamenetelmiä. Prosessissa täytyy ottaa huomioon eri ohjelmien yhteensopivuudet ja nykyisten Kaukolämpö Plus järjestelmän toimintojen päivitysmahdollisuudet.

Opinnäytetyö alkaa kesäkuun ensimmäisenä päivä viikolla 23 ja päättyy siitä noin 10 viikon kuluttua elokuussa 21.08.2020. Päiväkirjatyyppisen opinnäytetyön raportointi tapahtuu päivittäisellä työtehtävien kuvaamisella ja viikoittaisella analyysillä.

Avainsanat avattu alla olevassa listassa:

| | |
|----------------------------|--|
| ColdFusion | Ketterä web ohjelmistojen kehitysalusta (ColdFusion) |
| ColdFusion Markup Language | ColdFusion Markup Language on ohjelmointikieli, joka on ColdFusionin ensisijainen ohjelmointikieli (2kmediat) |
| CFScript | CFScript on ohjelmointikieli, joka on ColdFusion Markup Languageen laajennus (2kmediat) |
| CFWheels | ColdFusionin web ohjelmistojen kehitykseen tarkoitettu kehitysrunko (CFWheels) |
| T-SQL | Itsenäinen SQL-kielen laajennus, joka tulee sanoista Transact Structured Query Language (Rouse 2019) |
| CRUD | Tarkoittaa kyselykielessä käytettäviä käyttökäskyjä. CRUD tulee englannin sanoista: Create, Read, Update and Delete (Codecademy b) |
| KIS | Tulee sanoista "Keep It Simple", joka kääntyy suomeksi "Pidä se yksinkertaisena" (CyberDefinitions) |
| Bootstrap | Web kehityksessä yleisesti tunnettu ja käytetty front-end kehitysrunko (Bootstrap) |
| Jira | Projektinhallintatyökalu (Atlassian d) |
| LESS | LESS on CSS:n laajennus, joka mahdollistaa muun muassa muuttujien ja funktioiden käytön (Lesscss) |
| SASS | SASS on CSS:n laajennus, joka mahdollistaa muun muassa muuttujien ja funktioiden käytön (Coron 2020) |

2 Lähtötilanteen kuvaus

2.1 Oman nykyisen työn analyysi

Työskentelen järjestelmäasiantuntijana (System Engineer) Fortumilla Espoon Keilaniemin kauniissa merimaisemassa. Tehtäväni on tukea Fortum Kaukolämpö Plus järjestelmän kehitys- sekä ylläpitotiimiä heidän tehtävissään. Työtehtäviini kuuluu uusien toiminnallisuuksien kehittäminen Kaukolämpö Plus järjestelmään ja sen vikojen ennaltaehkäiseminen sekä korjaaminen. Siksi on hyvä ymmärtää, miten sen eri prosessit käyttäytyvät ja miten käyttäjät niitä käyttävät.

Työtehtäväni vaatii ennen kaikkea ymmärtämistä imperatiivisen ja olio-ohjelmoinnin päälle, koska ColdFusion on rakennettu Javan päälle. Java-ohjelmointikieli on imperatiivista, deklarativista sekä olioperustaista (lamda 2016). ColdFusion tukee nykyään olio-ohjelmoinnin tekniikoita, mutta ei sitä täysin (Leeds 2012).

Hyödynnämme Kaukolämpö Plus kehityksessä tikettijärjestelmää, jonka avulla hallitsemme työtehtäviä. Tikettijärjestelmään pääsevät myös Kaukolämpö Plus järjestelmän käyttäjät. Huomatessaan Kaukolämpö Plus järjestelmässä vian tai mahdollisen kehitysidean, käyttäjät jättävät siitä tiketin. Tikit ovat käyttäjille "ostopyyntöjä" ja kehittäjille työtehtäviä. Ne ovat käyttäjille ostopyyntöjä siksi, että tikkien arvostelun jälkeen on käsitys siitä, onko mahdollista kehitystoivetta järkevä kehittää ottaen huomioon sen tarpeellisuus ja siitä koituvat kustannukset. Tämän hoitaa change manager eli muutosten hallitsija, joka on lähin esimieheni. Hänen tehtäviinsä kuuluu aiemmin mainitun lisäksi tehtävien delegointi tiimin muille jäsenille. Työtehtäviä delegoidaan yleensä tärkeys ja kiireys järjestyksessä. Tikein luojalle täytyy ilmoittaa tikettijärjestelmän käyttöliittymän kautta missä vaiheessa työtehtävä on. *Suunnittelu, toteutus, testaus ja implementointi* -vaiheille löytyy kolme välivaihetta, jotka ovat seuraavat: *voi aloittaa, toteutuksessa ja epäonnistui*. Aina siirtyessä vaiheesta toiseen tikettijärjestelmässä on syytä päivittää tikein tilaa, jotta change manager ja tikein luoja näkevät kuinka pitkälle kehitys on edennyt.

Uusien toiminnallisuuksien kehittäminen käynnistyy Kaukolämpö Plus järjestelmän käyttäjien jättämästä tiketistä. Change manager siirtää tikein nimiini, jonka jälkeen käymme yhdessä läpi, mitä muutoksia kehitysehdotus vaatii Kaukolämpö Plus järjestelmän käyttöliittymä- sekä palvelinpuoleen. Perusteellisen suunnittelun jälkeen voin aloittaa toiminnallisuuden kehittämisen.

Osa työstäni koostuu virheiden korjaamisesta tai niiden ennaltaehkäisystä. Kaukolämpö Plus järjestelmä lähettää sähköpostia pääkäyttäjille, silloin kun tuotantopuolella tapahtuu sovellusvirhe. Virheet ovat joko enemmän tai vähemmän kriittisiä. Enemmän kriittiset virheet tulevat yleensä nopeammin selville kuin vähemmän kriittiset, koska kriittisistä virheistä tulee automaattisesti sähköpostia. Sähköpostien avulla virheet pystytään nopeammin paikantamaan ja sitten korjaamaan. Vähemmän kriittiset virheet yleensä koostuvat käyttöliittymän rakenteen muodossa, jossa esimerkiksi jokin nappula saattaa käyttäytyä oudolla tavalla.

Ennaltaehkäisevät tehtävät ovat sellaisia missä koodipohjaa halutaan optimoida. Optimointi tapahtuu esimerkiksi refaktoroimalla olemassa olevia järkälemäisiä ohjelmistokoodilohkoja omiin pieniin funktioihin, joita voidaan hyödyntää muualla järjestelmässä tai parannetaan sen yleistä luettavuutta. (Kiiskinen 2018.) Lisäksi jotkut Kaukolämpö Plus järjestelmän prosessit ovat jätetty puolivalmiiksi, koska niihin ei ole haluttu käyttää liikaa aikaa tai sitten aikaa ei ole varattu riittävästi. Kaikki Kaukolämpö Plus järjestelmän päivitykset ovat tehty MVP-ratkaisuina (Liite 1). MVP tarkoittaa mahdollisimman pientä toimivaa tuotetta, jolla tyydytetään asiakkaan tarpeet (Agile Alliance). Tästä syystä koodissa saattaa ilmentyä toimintaa hidastavia tekijöitä, joita tulisi poistaa. Hidastavia tekijöitä voivat olla ohjelmistossa muun muassa esiintyvät toistorakenteet, joiden sisälle on tehty epätarkkoja tietokantakyselyitä.

Kaukolämpö Plus järjestelmä on rakennettu ColdFusionin päälle. ColdFusion on ketterä web sovelluksien kehitysalusta, joka käyttää ensisijaisesti CFML- tai CFScript-ohjelmistokieliä (2Kmediat) (Liite 1). Kehityksen tukena käytetään CFWheels-kehitysrunkoa, jonka avulla Kaukolämpö Plus järjestelmän sivut ja toiminnallisuudet ovat rakennettu. Sivujen kehityksessä käytetään Bootstrap 3 kehitysrunkoa, jonka avulla niiden luominen on tehokkaampaa ja ulkonäkö pysyy mahdollisimman samanlaisena sivulta siirryttyä toisaalle. Kaukolämpö Plus järjestelmässä käytettävä informaatio on tallennettu Microsoftin SQL Serverille. SQL Server on relaatiotietokanta, jossa dataa voidaan jakaa tauluihin ja sitten linkittää toisiinsa (Technopedia 2017).

Kaukolämpö Plus järjestelmää on käytetty ja kehitetty jo vuodesta 2006. Sitä on kehittänyt moni eri henkilö, jolla jokaisella on oma tapansa ohjelmoida ja oma näkemys siitä, mikä on hyvää ohjelmistokoodia. Tästä syystä työni vaatii ennen kaikkea kärsivällisyyttä sekä hyvää luetun ymmärtämistä, jottei ohjelmistoa optimoidessa tarvitsisi kirjoittaa kaikkea uudelleen tai ohjelmointivirheiden paikantamiseen kuluisi tuhattomasti aikaa.

Kommunikaatiotaidot tulevat tarpeeseen etenkin silloin kun suunnittelee Kaukolämpö Plus järjestelmään uutta toiminnallisuutta. Tiimini jäsenet tarjoavat tukea toiminnallisuuden kehityksessä ja ongelmatilanteissa. Sidosryhmiltä saan tarvittaessa tietoa, mitä tulisi ottaa huomioon esimerkiksi käyttöliittymää suunniteltaessa.

Työaikana pyrin kommunikoimaan muiden sidosryhmien välillä, jotta en olisi täysin riippuvainen change managerista. Tällä tavoin vähennän esimieheni työtaakkaa ja todennäköisesti saan antoisampaa informaatiota sen tullessa suoraan sidosryhmiltä. Yritän myös jatkossa käyttää enemmän aikaa dokumentaatioiden lukemiseen.

2.2 Lähtötasoni

Olen toiminut Fortumilla koko- ja osa-aikaisesti vuoden ajan. Kaukolämpö Plus tiimin liittyessäni tutustuin sen riippuvuuksiin, kuten ColdFusion, ColdFusion Markup ja -Scripting -ohjelmistokieliin. Olen myös lisännyt taitojani kyselykielissä, kuten muun muassa tutustumalla paremmin T-SQL:ään. T-SQL eli Transact Structured Query Language on SQL-kielen itsenäinen laajennus (Rouse 2019).

Olen taitava suoriutuja, koska olen toistaiseksi aina löytänyt keinon toteuttaa toivotut ominaisuudet. Onnistuminen ei sinänsä kerro koko totuutta. Tehtävät, joita minulle delegoidaan, menee aina change managerin kautta, joten hänellä on ensimmäinen ja viimeinen sana mitä toiminnallisuutta pääsen kehittämään.

Syy, miksi en sanoisi itseäni kokeneeksi asiantuntijaksi on se, että ympärilläni on enemmän Kaukolämpö Plus järjestelmän kokemusta omaavia toimijoita ja aikani järjestelmän parissa on vielä lyhyehkö. Käytän valitettavan vähän aikaa dokumentaatioiden lukemiseen ja uusien asioiden sisäistämiseen. Luen rivien välistä ja hyppään suoraan ohjelmointiin. Yleensä suunnittelen vasta ohjelmoidessa. Lyhyen kokemukseni pohjalta osaan silti perustella Kaukolämpö Plus projektissa, miten mikäkin toiminto kannattaisi koodipohjalta rakentaa ja miten sen täytyisi käyttäytyä, jotta mahdollistettaisiin hyvä käyttäjäkokemus. Rajaava tekijä tulee yleensä esiin heti, jos en tiedä tai ymmärrä sen käyttäjien toimintaprosessia.

2.3 Kehittyminen

Kiinnostuin ohjelmoinnista ammattikoulussa, josta asti olen ollut varsin aktiivinen harrastelija. Tietojen käsittely opintojeni alussa hain kesätöihin App Studios Finland Oy:lle. Työskentelin siellä vähän yli kaksi vuotta web kehittäjänä erilaisissa projekteissa. App Studios

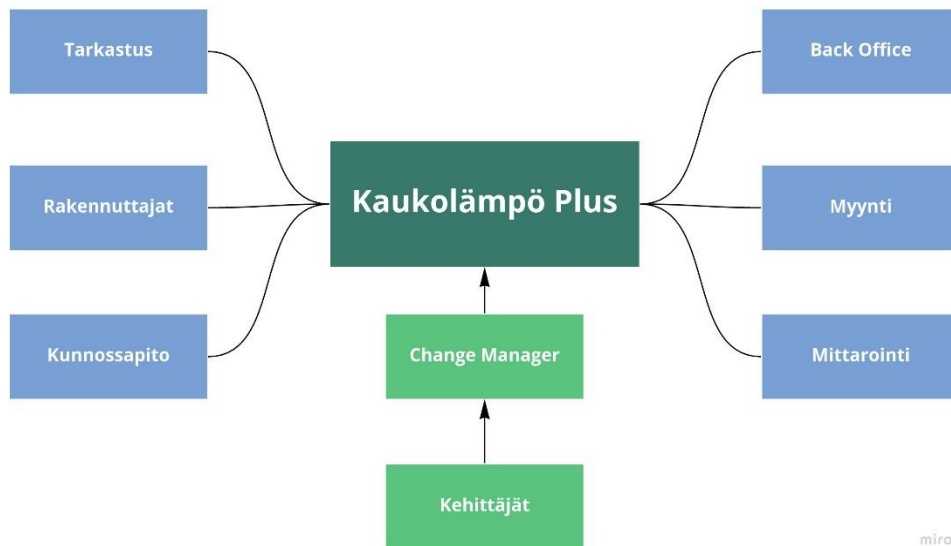
Finland Oy:n jälkeen ryhdyin kevytyrittäjäksi, josta sain myynti- ja asiakaskokemusta. Ke-
vällä 2019 hain kesätöihin Fortumille kehittämään Kaukolämpö Plus järjestelmää.

Olen tehnyt vapaa- ja työajalla useita eri projekteja eri tekniikoilla. Olen suuntautunut tie-
toisesti kehittämään web toteutuksia niiden kehittämisen mielekkyyden takia. Huomasin
jokunen aikaan sitten, että uusien teknologioiden omaksuminen vie huomattavasti vähem-
män aikaa kuin aikaisemmin. Uusien teknologioiden nopea omaksunta auttaa, kun täytyy
tietää mitä työvälineitä valitaan seuraavaan projektiin. Koko urani ajan olen kehittänyt
web-sivustoja tai sitten web-aplikaatiota.

Ohjelmoiminen edellyttää ennen kaikkea vahvaa tarkkaavaisuutta, sillä pienikin virhe saat-
taa aiheuttaa pahimmassa tapauksessa järjestelmien rampautumista. Silloin tällöin tulee
kirjoiteltua operaattoreita väärinpäin, jotka johtavat virheisiin. Tämä voi johtua siitä, että
ohjelmoin monesti kiireessä ilman kiirettä. Olen huomannut useaan otteeseen, että hätäil-
len turhaan yksinkertaisten asioiden kanssa. Minulla on tapana myös monimutkaistaa yk-
sikertaisia asioita. Pysin aina dynaamisuuteen, vaikka siihen ei olisi paikoin tarvetta. Asioi-
den monimutkaistaminen kuluttaa valtavasti aikaa. Dynaamisuudesta on toki paljonkin
hyötyä tietyissä tilanteissa, jos muistaa pitää sen joissain rajoissa. Sitä tarvitse vain pakot-
taa joka paikkaan. Yritykseni säästäisi kohdallani tässä asiassa rahaa, jos saisin KIS-me-
todin painettua takaraivoon (Liite 1).

2.4 Sidosryhmät työpaikalla

Kaukolämpö Plus järjestelmän päätarkoitus on tukea Kaukolämmön toimintaa Fortumilla
helpottamalla sen käyttäjien arkea digitalisoimalla ja automatisoimalla sidosryhmien toi-
mintaa. Kaukolämpö Plus järjestelmää käyttävät useat eri sidosryhmiin kuuluvat henkilöt.
Sidosryhmiä kuvattu kuvassa 1.



Kuva 1. Kaukolämpö Plus järjestelmän sidosryhmät

Sidosryhmät, kuten Myynti, tarkastukset ja mittarointi ovat aktiivisesti yhteydessä change manageriin. Ne ovat myös suoraan yhteydessä kehittäjiin, silloin kun change manager ei ole paikalla tai se ei vaadi change manageria. Kehittäjät suunnittelevat Change managerin kanssa ratkaisun muiden sidosryhmien toivomiin toimintoihin. Change managerilla on viimeinen sana, mitä toiminnallisuutta lähetään kehittämään Kaukolämpö Plus järjestelmään. Mainitut sidosryhmät myös työskentelevät lähellä kehittäjiä, joka mahdollistaa nopean kanssakäymisen. Uusista toiminnallisuuksista voidaan keskustella hyvin matalalla kynnyksellä.

2.5 Vuorovaikutustaidot työpaikalla

Change manager delegoi minulle työtehtäviä ja hoitaa kommunikoinnin sidosryhmien kanssa, jotta voisin keskittyä tekevään työhön. Change manager tekee myös alustavia määrittelyjä tehtävästä kehityksestä. Kehittäjät sidosryhmä on siis ainoa sidosryhmä, jonka mielipiteet ja intressit ovat keskeisiä työni kannalta. On hyvä huomauttaa, että poikkeuksiakin ilmenee esimerkiksi silloin kun kehitän uusia toiminnallisuuksia tai korjaan Kaukolämpö Plus järjestelmässä esiintyviä vikoja. Jos change manager ei ole saavutettavissa voin toki suoraan konsultoida asianomaisia. Pääasiassa työtehtävät kuitenkin suunnitellaan change managerin kanssa niin ettei asianomaisten konsultoinnille ole järin tarvetta. Kaukolämpö Plus järjestelmän palautteet kantautuvat yleensä korviini esimieheni kautta.

Joudun työssäni usein turvautumaan kollegoihin, sillä jotkut tehtävät vaativat laajempaa tietämystä eri sidosryhmien toiminnoista. Suurin haaste on yleensä ymmärtää Kaukolämpö Plus järjestelmän eri toiminnot ja sidosryhmien toimintaprosessit. On tärkeää tietää

sidosryhmien toimintaprosessit esimerkiksi suunniteltaessa käyttöliittymää. Jos puhutaan vaikka kirjausnäkömäästä olisi syytä tietää missä järjestyksessä käyttäjät asettaisivat tietoja järjestelmään ja onko tiedot jotenkin riippuvaisia toisistaan. Hyvä suunnittelu edellyttää hyvää tuntemusta Kaukolämpö Plus järjestelmän käyttäjien prosesseista. Change managerilla on vankka tietämys sidosryhmien eri prosesseista, jolloin heihin on hyvä turvautua silloin, kun tarvitsee tietoa.


3 Päiväkirjaraportointi

3.1 Seurantaviikko 1

Maanantai 01.06.2020

Aamulla olisi tavoitteena jatkaa muutaman tunnin mittaripuolelle tulevan skannerin ohjelmoinnista. Tarkoituksena olisi rakentaa ITF-viivakoodin checksum tarkastuskoodin tarkistus. Vapaasti käytettävä zxing-js ohjelmointikirjasto, jota luomani viivakoodiskanneri hyödyntää on puutteellinen (GitHub b). Siihen ei ole rakennettu ITF-viivakoodien tarkistuslukujen tarkastusta, joten se täytyy itse rakentaa. Kyseistä ITF-viivakoodia esiintyy ainakin UH-50 mittarilaitteissa. Skannerilla luetaan erilaisten laitteiden viivakoodeja, jotka kirjataan tietokantaan inventointivaihetta varten.

Luetaan laskijoita



| # | Sarjanumero | Lisätieto | Tila |
|---|-------------|-----------|------|
| 1 | 69870148 | Lataa... | ✘ |

Vaihda kameraa camera2 2, faci Palaa takaisin Vaihda laitetyp...

Kuva 2. Kaukolämpö Plus järjestelmän skanneri

Tänään olisi myös tavoitteena pitää aloituspalaveri kesän projektille, joka liittyy Kaukolämpö Plus järjestelmän päivitykseen. Sovimme viime perjantaina yhteisesti tiimin kanssa, että katsomme yhdessä Jira-projektihallintasovellusta (Atlassian b).

Sain aamulla ennen tiimin keskeistä palaveria viivakoodiskanneria työstettyä hyvin eteenpäin. ITF-viivakoodin tarkastus näyttää nyt toimivan niin kuin pitää. Kuvassa 2 näkyy

skannattava laite ja skanneri toiminnassa. ITF-viivakoodin tarkistus tehdään siksi, että varmistetaan viivakoodiskannerin lukeman sarjanumeron oikeamuotoisuudesta.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Check Digit | |
|---|-------|---|---|---|---|---|---|---|---|----|-------------|---|
| Luettu sarjanumero | 0 | 6 | 5 | 4 | 6 | 1 | 7 | 1 | 3 | 5 | | |
| 2.rivi | 0 | | 5 | | 6 | | 7 | | 3 | | | |
| 3.rivi | | 6 | | 4 | | 1 | | 1 | | | | |
| 2. rivi yht. | 21 | | | | | | | | | | | |
| 3. rivi yht. | 12 | | | | | | | | | | | |
| 2.rivi kerrotaan 3 | 63 | | | | | | | | | | | |
| 2.riviin lisätään 3.rivi | 75 | | | | | | | | | | | |
| Tulos pyöristetään ylöspäin lähimpää 10, joten 75 -> 80 | | | | | | | | | | | | |
| Pyöristetystä tuloksesta miinustetaan pyöristämätön tulos | 80-75 | | | | | | | | | | = | 5 |

Kuva 3. ITF-viivakoodin tarkistusnumeron laskentakaava

ITF-koodin tarkistusnumeron lasku aloitetaan aina viimeisestä numero. Sarjanumerosta lasketaan ensimmäiseksi yhteen joka toinen numero, josta tulee 21. Toiseksi lasketaan yhteen aloittaen toiseksi viimeisestä numerosta, josta tulee 12. Ensimmäisen rivin tulos kerrotaan kolmella, jolloin saadaan 63. Sitten lisätään toisen rivin tulos, josta tulee 75. Lopputulos saadaan pyöristämällä tulos 75 lähimpään ja ylimpään kymmenekseen, josta miinustetaan pyöristämätön luku 75. Tarkistusnumeroksi tulee 5, jota voidaan verrata sarjanumeron viimeiseen lukuun. Jos luvut täsmäivät sarjanumerolla on tarkistusnumero oikein. Edellä mainittu laskentakaava näkyy kuvassa 3.

Sain skanneriin lisättyä muutamia puuttuvia ominaisuuksia. Skanneri palauttaa nyt onnistuneen luvun jälkeen käyttäjälle luetun laitteen tietoja. Aikaisemmin tilalla oli joko tyhjää tai geneeristä tekstiä. Viivakoodiskanneri on nyt edistynyt siihen pisteeseen, että kohta sitä voidaan alkaa testailemaan. Muutamia ominaisuuksia kuitenkin vielä puuttuu, mutta uskon, että saan ne viimeistään huomenna valmiiksi. Palaveri katkaisi ajatuksen juoksun, mutta se oli odotettua ja välttämätöntä.

Puolikymmenen aikoihin kävimme kollegoiden kanssa Jiraa läpi. Loin Jiraan tehtäviä ja käyttäjätarinoita. Tehtävät kuvaavat projektissa esiintyviä tehtäviä ja käyttäjätarinoilla kuvataan toiminnallisuutta yksinkertaisuudessaan, jotta asiakas ymmärtää sen helposti (Lepäniemi). Palaverin jälkeen oli vaikea jatkaa skannerin työstämistä, koska ajatukset pompivat hyvin paljon tulevan projektin ja nykyisen tehtävni välillä.

Aloitin tänään kokoaikaisesti Fortumilla. Töiden määrä äkillisesti moninkertaistui ja pelkään erittäin kiireisen kevään aiheuttaman stressin palaavan. Nyt on tärkeää huolehtia myös itsestään, joten päätän olla miettimättä töitä liikaa vapaa-ajalla. Uskon, että Jira auttaa huolehtimaan projektin aikatauluttamisessa ja siksi aion panostaa sen opiskeluun mahdollisimman paljon. Täytyy jatkossa pyrkiä keskittymään yhteen asiaan kerrallaan. Töitä tulee jatkossakin olemaan paljon samanaikaisesti ja työt tulevat olemaan erilaisia. Tämä takia on siis edettävä päivä kerrallaan.

Tiistai 02.06.2020

Tänään olisi tarkoitus saada viivakoodiskanneri valmiiksi. Täytyy myös lisäksi korjata ticketijärjestelmään ilmoitettu bugi, joka liittyy Kaukolämpö Plus järjestelmän liisi-prosessiin. Kaukolämpö Plus järjestelmässä luotujen tehtävien uniikki id ei jostain syystä tallennu oikein tietokantaan. Uniikilla id:llä voi myöhemmin etsiä kyseisen liisin ja siihen liitetyn tehtävän. Kaukolämpö Plus järjestelmässä voidaan jakaa järjestelmän sisäisiä tehtäviä tietyille käyttäjille tai käyttäjäryhmille. Uskon ohjelmistovirheen johtuvan ajatusvirheestä. Täytyy katsoa se ajan kanssa, mutta ensin teen skannerin valmiiksi. Lopuksi jos aikaa jää tavoitteena on tutkia Jiraa ja lisätä sinne sopivia tehtäviä liittyen kesän projektiin, jos sellaisia tulee mieleen.

Nyt kun päivä on ohitse, voidaan todeta, että skanneri vei suurimman osan päivän työajasta. Tärkeintä on se, että se on nyt valmis ja poissa päiväjärjestyksestäni. Skannerin tekeminen alusta loppuun oli mielenkiintoista. Seuraavaksi mittaripuolen henkilöstö testaa sovellusta käytännössä.

Kehityksessä tapahtunut testaus on usein epätarkkaa, koska varsinkin omaan ohjelmistokoodiin sokeutuu. Sokeutumisella tarkoitan sitä, että testaukset tehdään hyvin kaavamaisesti. Sokeutuminen yleensä johtaa siihen, että jotain jää joko huomaamatta tai ei tule testattua riittävällä tarkkuudella. Siksi on hyvä saada MVP eli Minimal Viable Product ulos mahdollisimman nopeasti, jotta käyttäjät pääsevät sitä käyttämään ja antamaan palautetta. MVP tapaa ei hyödynnetty tässä projektissa, sillä sen toiminnallisuuksista tiedettiin riittävästi ja niitä ei ollut useita. Skanneri otetaan testaukseen vasta silloin kun kaikki edellytykset ovat täytetty.

Päivän aikana taistelin yhden bugin kanssa jonkun aikaa. Skanneri ei kyennyt lukemaan viivakoodeja ensimmäisen onnistuneen luvun jälkeen. Ohjelmistokoodiin oli ilmaantunut jokin uusi bugi uusien lisättyjen toimintojen takia. Muutoksia oli niin paljon, että en halunnut palata riviäkään taaksepäin.

Tämän viivakoodiskanneriprojektin aikana olen huomannut, mitä enemmän minulla on tehtäviä samanaikaisesti päällä, sitä hitaammin tuotan ohjelmistokoodia. Tahdon tehdä kaikkia samaan aikaan ja se ei fyysisesti tunnu olevan mahdollista ainakaan minun kohdallani. Huomenna aloitan seuraavan projektin, joka on koko kesän mittainen Kaukolämpö Plus 7.0 päivitys.

Keskiviikko 03.06.2020

Aamupäivän tavoitteena on tutkia aiemmin mainittua liisi-prosessiin liittyvää bugia. Tiketti-järjestelmään on kirjoitettu seuraavasti "Liisi-tehtävästä generoitava tehtävä ei päivitä kantaa tehtävän id:tä liisitehtävälle". Pelkästä tiketistä voidaan päätellä, että jotain menee tallennuksessa pieleen. Täytyy lukea jonkun verran CFWheels dokumentaatiota, koska uskon, että save-funktion parametri saattaa olla syytä uniikin id toimimattomuuteen. Save-funktiolla tallennetaan olio tietokantaan. Save-funktio on CFWheelsin funktio. Olio on olio-ohjelmoinnissa rakennuskomponentti, jolle voi määrittää ominaisuuksia tai funktiota (Kareinen). Kyseinen olio sisältää liisiin liittyviä tietoja.

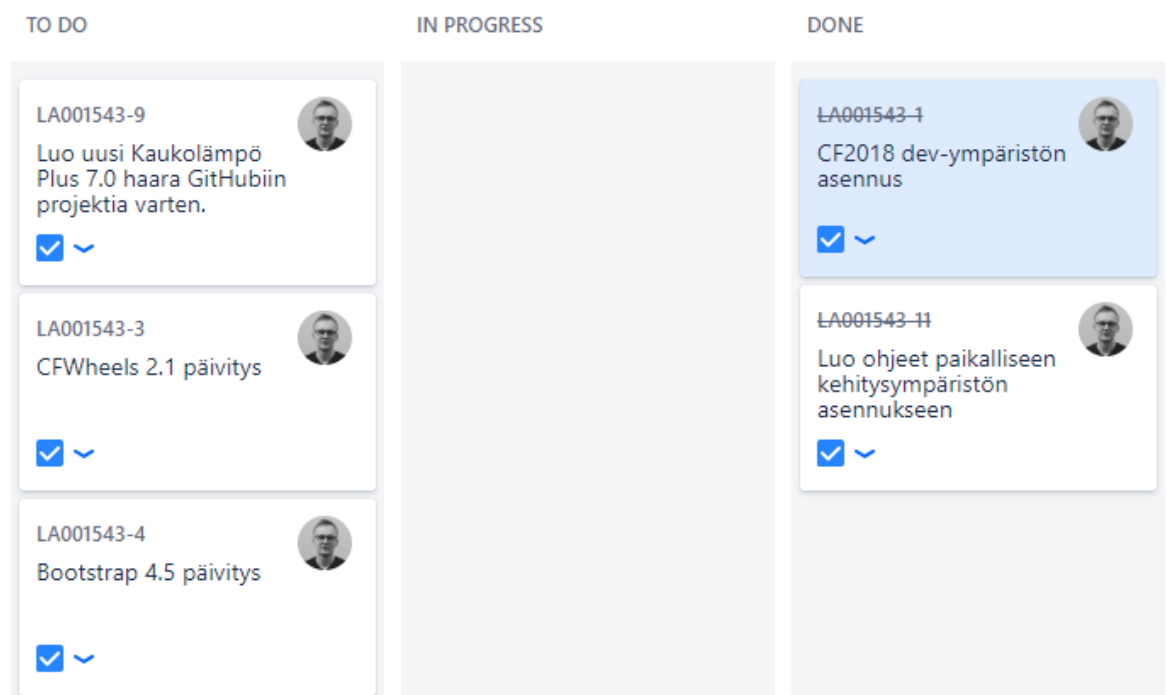
Loppupäivästä olisi tarkoitus katsoa kesän projektiin liittyviä asioita kuntoon. Aloitan todennäköisesti lukemalla ColdFusion version päivittämisen ohjeet perusteellisesti ja sitten perustamalla projektihakemisto uudelle versiolle omalle koneelleni. Projektihakemistoon kopioidaan nykyinen Kaukolämpö Plus versio 6.0, jonka päälle päivitys tehdään. Minulla tulee siis olemaan kaksi KL Plus järjestelmää koneella. Toinen vaihtoehto on GIT-haaran luominen, johon alan tekemään muutoksia. Haara on GIT-versiohallinnan yksi monista toiminnallisuuksista, jolla voi luoda omia koodinäkymiä. Koodinäkyvät voidaan ajatella täysin erillisenä projektista, koska haaran tehdyt muutokset eivät vaikuta muihin haaroihin. Haaroihin voidaan myöhemmin muun muassa liittää toisia haaroja (Atlassian a).

Sain bugin korjattua aamupäivän aikana kahvitunnilla. Ongelmana oli liian yksinkertaistettu koodi. Liisi-prosessin tehtävän luonnissa generoitava uniikki id täytyi siirtää omaan muuttujaan. Uniikki id pääteltiin ternary-operaattoria käyttäen. Ternary-operaattorilla pystyy lyhyesti kirjoittamaan muuttujan arvon dynaamiseksi. Esimerkiksi jos haluan antaa muuttujalle A arvon X jos Y on 1 ja muissa tapauksissa 2, kirjoitan seuraavasti: $A = X == "Y" ? 1 : 2$ (Nadel 2009). Tässä tapauksessa määritin RelObjectId muuttujan arvon ternary-funktiolla ennen kuin käytin sitä save-funktiossa.

Merkitsin tikettijärjestelmään ongelman ratkenneeksi ja mitä tuli korjattua. Kahvitunnilla kävi kanssa ilmi, että eilen valmistuneeseen skanneriin voisi lisätä vielä oikeudet.

Tarkemmin sanottuna halutaan lisätä estoja. Skanneri tulisi olla näkyvässä vain tietyillä käyttäjäryhmillä. Tämä vaati pieniä muutoksia tietokantaan sekä itse skannerin kontrolleriin. Kontrollerilla ohjataan toiminnallisuutta Kaukolämpö Plus järjestelmässä. Kaukolämpö Plus on rakennettu MVC-arkkitehtuurin mukaisesti. MVC tulee sanoista Model View Controller, jonka päätarkoitus on erottaa käyttöliittymä sovelluslogiikasta (Codecademy a).

Tänään käynnistin ensimmäisen Sprintin. Sprint on yleensä 1-4 viikon työskentelyjakso, jotka suunnitellaan aina etukäteen. Suunniteltaessa valitaan tehtäviä projektissa, jotka halutaan toteuttaa kehitysjonosta. Kehitysjono tarkoittaa listaa tuotteen kaikista toiminnoista (Helsinki).



Kuva 4. Jira-sovelluksen Sprint-näkymä

Haluan, että Kaukolämpö Plus järjestelmän päivittäminen sujusi mahdollisimman mutkattomasti. Päätin ennen ensimmäistä työpäivää, että käytän enemmän aikaa lukemiseen. Tällä kertaa aion käyttää siihen työaikaa. Luin ColdFusion 2018 version Migration Guide. Siinä oli kuvien lisäksi onnekseni vain 19 sivua (Adobe b). Migration Guide opastaa ColdFusion palvelimen päivityksessä. Lukemisen jälkeen tiesin olevani valmis asennusprosessiin, joten käynnistin asennusohjelman. Asentamisessa piti ottaa huomioon ensin sulkeamalla aiemmat ColdFusion version 2016 palvelut. Ne sai yksinkertaisesti sammutettua Windows 10 tehtävien hallinnan kautta. Aiemmasta versiosta tuli kuitenkin kopiota Kaukolämpöön liittyviä tietoa, joita halutaan käyttää 2018 versiossa. Kyseessä oli

järjestelmäasetuksia, kuten missä palvelimet sijaitsevat ja mitkä ovat niiden käyttäjien tiedot. Seuraavalla tarmokkaasti ohjeita asennus onnistui mutkitta.

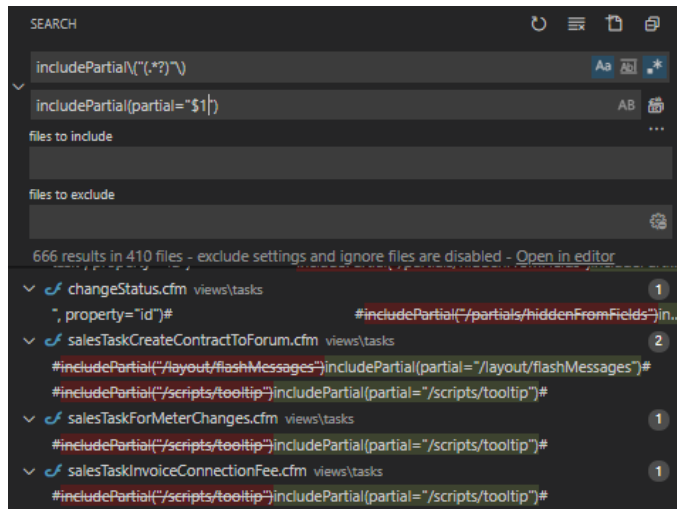
Change managerit pyysivät, että välittäisin asennusohjeet heille kirjallisessa muodossa. Loin Word-dokumentin asennusohjeista, jotka jaon jaetulle palvelimelle. Suomenkielinen ohje opastaa, miten versio 2016 päivitetään versioon 2018. Lisäsin jaetulle palvelimelle myös tarvittavat tiedostot, joita asennuksessa tarvittiin.

Päivän lopussa kävin kuittaamassa Jirassa, että sprintin ensimmäiset tehtävät oli saatu valmiiksi. Minulle jäi erittäin hyvä fiilis, sillä kuvittelin aamulla, että päivä toisi ongelmia tullessaan järjestelmävirheiden muodossa. Näin ei käynyt ja uskon sen johtuvan siitä, että jaksoin käyttää normaalia enemmän aikaa dokumentaation lukemisessa. Version päivitys ei ole mikään taikatemppu ja siksi kaukolämpö Plus pyörii nyt ColdFusion 2018 Update 9 päällä.

Torstai 04.06.2020

Seuraavaksi olisi vuorossa CFWheelsin asennus ja sen asentamiseen liittyvät ohjeet. Muita tavoitteita en aseta tälle päivälle, koska tiedän suurimman osan tulevista ongelmista liittyvän juuri CFWheelsin päivittämiseen. Tulen tänään päivittämään CFWheels 1.4.5 version 2.1 versioksi. En aio keskittyä Kaukolämpö Plus prosessien testaamiseen vaan pelkästään siihen, että Kaukolämpö Plussan yleiskuva säilyy. Prosessien läpi käyminen on tulevien päivien töitä. Päivitys on tosi iso oman kokemuksen mittakaavassa. Täytyy lukea tarkasti CFWheelsin Upgrade Guide, joka löytyy heidän verkkosivuiltaan. Siellä kerrotaan miten hakemistorakenteet muuttuvat ja mitä tiedostoja tulee lisää ja poistaa. Eniten aikaa kuluu varmasti kontrollereiden läpikäymiseen, sillä niitä on paljon. Niistä on hyvä aloittaa siksi, että ne ovat yksinkertaisimmat. Käyttöliittymästä käy nopeasti ilmi, jos controlleria ei ole reititetty oikein. Nimittäin silloin Kaukolämpö Plus järjestelmän sivu eivät aukea. Reitittäminen tarkoittaa osoitteiden kirjoittamista tiedostoille (Shukla, 2013).

Luin maltillisesti CFWheelsin Upgrade Guiden heidän virallisilta verkkosivuiltaan. Upgrade Guide ei ollut pitkä ja siinä oli hyvin esitetty muutokset, jotka piti ottaa huomioon päivittäessä. Siksi päivittäminen onnistui ilman ongelmia. Käytin apunani Regular Expression etsintäkaavaa, jolla voidaan etsiä tekstistä osumia (RegularExpressions). Sen avulla pystyin koko Kaukolämpö Plus järjestelmän kontrollereista sekä malleista etsimään "`<cfunction name="init">`" ja `includePartial`-funktiota.



Kuva 5. Regular Expression etsintäkaavan käyttöä Visual Studio Code:ssa

Kuten kuvassa 5 etsin kaikki includePartial-funktiot, joita kuvan mukaan oli 666. Ne olivat ripoteltu 410 eri tiedostoon. Toinen hakupalkki ylhäältä alaspäin pitää sisällään korvattavan tekstin. Dollari-merkki ja sen perässä oleva numero 1 kuvastaa Regular Expressionin ryhmiä, joihin voi taltioida tietoja. Ryhmät luodaan käyttäen sulkuja. CFWheelsin Upgrade Guiden mukaan jokaisessa IncludePartial-funktiossa tulisi olla "partial" niminen parametri. Parametrien avulla voidaan ohjata dataa funktioiden sisälle ja siten vaikuttaa niiden toimintaan (BBC). Regular Expressionin avulla sain nipistettyä kyseisestä tehtävän tekemisestä arvion mukaan yksi tai kaksi päivää. Onnekseni Visual Studio Code ja Regular Expression ovat olemassa. Visual Studio Code on ohjelmistokehitykseen tarkoitettu kehitysalusta (Visualstudio).

CFWheelsin päivitys ei kuitenkaan ole vielä täysin valmis, sillä en ole tehnyt ainuttakaan kokovaltaista testiä. Jouduin jättämään CFWheelsin kehitysfoorumille viestiä, jotta saisin AJAX ongelmaan apua (Google Groups; Uusitalo 2020a) (Liite 1). AJAX:illa voi keskustella tietokannan kanssa ja päivittää sivun tietoja lataamatta sivua uudelleen (Fowler 2020). Yllätyin, kuinka nopeasti sain asennettua CFWheelsin ja ratkaistua useamman siihen liittyvän ongelman.

Perjantai 05.06.2020

Tänään olisi tarkoitus jatkaa siitä mihin eilen jäätiin. Tavoitteena on saada Bootstrapin versio 3 päivitettyä viimeisimpään versioon, joka on 4.5. Lisäksi tavoitteena on saada CFWheelsin AJAX-funktiot toimimaan ja sitten päivittää trimparams niminen CFWheelsin lisäosa. Trimparams versio on suunnattu 1.4.5 versiolle ja siksi se on syytä päivittää. En tosin tiedä, onko kyseistä lisäosaa edes päivitetty tukemaan CFWheelsin 2.1 versiota.

Trimparams-lisäosa päivittäminen on oleellista, koska sitä käytetään jokaisessa kentässä Kaukolämpö Plus järjestelmässä. Lisäosa siivoa turhat välilyönnit tekstien ympäriltä. Bootstrapin päivittäminen on päätavoitteeni. Olen asettanut käynnistämäni kehitysjakson päättymään ensi maanantaina klo 12.

Trimparams lisäosaa ei ollut päivitetty tukemaan uusinta versiota tai ainakaan sitä ei löytynyt virallisilta CFWheelsin lisäosien lataussivulta. Katsoin lisäosan koodin läpi, jota oli noin 60 rivin verran. Lisäosa näytti siltä, että sen päivittäminen ei olisi iso työ. Täytyy jossain vaiheessa projektia päivittää kyseinen lisäosa.

Tutkin myös lisää CFWheelsin reititykseen liittyviä asioita. Siitä löytyi hyvä, mutta lyhyt tutoriaali videomuodossa. Video avasi hyvin, miten reititys täytyy rakentaa CFWheels ympäristössä, ja mitkä ovat reitittämisen hyvät käytännöt (NeoKing 2018a; NeoKing 2018b). Change manager ehdotti, että seuraavassa sprintissä rakentaisin testikontrollerin, jonka avulla voisin todentaa reitityksen toimintaa. Tekstikontrollerin avulla change managerit pääsevät ajan tasalle, miten jatkossa reititys täytyy rakentaa.

Bootstrap kehitysrungon kanssa oli paljon ongelmia päivän aikana. Uusimman version päivittäminen tuotti paljon manuaalista työtä. Bootstrap 3 käytettiin LESS-kirjastoa, joka tekee CSS:stä dynaamisen mahdollistamalla muuttujien sekä funktioiden ja matemaattisten operaatioiden käytön (Liite 1) (Lesscss). CSS on HTML:ssä käytetty tyylitiedostokieli, jonka avulla kerrotaan selaimelle miltä HTML elementtien tulisi näyttää (Liite 1) (Mozilla a; W3 a). Bootstrap 4 käyttää SASS, joten kaikki LESS-tiedostot täytyy muuttaa SASS:ksi (Bootstrap). SASS on LESS:in tapaan CSS:än lisäosa (Liite 1) (Coron). Bootstrap versiossa 3 oli vielä mahdollista käyttää pikseleitä esimerkiksi kirjasinkoon määrittämisessä. Bootstrap 4 vaatii rem-yksikön käyttöä. Jokainen dynaamiseksi asetettu arvo täytyi muuttaa rem-yksikköön. (Bootstrap.) Rem-yksikkö on CSS:ssä käytetty elementin relatiivinen pituuden määrittäminen, joka pohjautuu juurielementissä määritettyyn kirjasinkokoon (Liite 1) (W3 a).

LESS ja SASS olivat termeiltään ja syntaksiltaan minulle entuudestaan tuttuja, mutta ne olivat koskaan niitä kunnolla käyttäneitä. LESS-tiedostojen SASS-muodoksi kääntöprosessissa sain hyvän käsityksen, miten ne eroavat toisistaan ja miten kyseinen prosessi tapahtuu. Samalla mietin, millainen on hyvä hakemistorakenne Kaukolämpö Plus järjestelmään. Sitä tutkiessani kävi ilmi, että hakemistorakenteesta kannattaa tehdä selkeämpi ja nimetä tiedosto johdonmukaisesti. Minulla meni pitkään ennen kuin ymmärsin, mitkä tiedostoista ovat tärkeitä ja mitkä ovat turhia uudessa Kaukolämpö Plus versiossa.

Viikkoanalyysi

Viikolla 23 aloitettiin ja lopetettiin uusia ja vanhoja projekteja. Opettelin työn tekemistä uudelleen kiinnittämällä organisoimiseen enemmän aikaa. Jira on erittäin hyvä juuri tässä asiassa ja jatkossa aion hyödyntää sellaista käyttöä pienissäkin projekteissa. Aiemmin kaikki tekemäni työ Kaukolämpö Plus projektissa on tapahtunut enemmän tai vähemmän suullisen ohjauksen kautta. Nyt suullisen ohjauksen lisäksi minulla on turvana Jira.

Jiran käyttö on loppujen lopuksi erittäin yksinkertaista. Jiran hallintapaneeli on huomattavasti kätevämpi kuin Post it -laput. Päihittää heittämällä kynä ja paperi kombinaation. Mielestäni anteliaan ominaisuus siinä on se, että minulla on kaikki samassa paikkaa. Näin kaiken yhdeltä näytöltä ja tiedän mihin tuleen tänään keskittymään. Palataan vain kaksi viikkoa taaksepäin, jolloin työskentelin Fortumilla yhden päivän viikossa. Aina perjantaisin taistelin muistaa, mitä perjantaina tulee tehdä ja mitä tulee saada valmiiksi. Paperi, joka sisälsi muistiinpanoja ei riittänyt aina muistuttamaan päivän töistä. Jiran kaltaista palvelua olisi ollut hyvä hyödyntää jo skannerin kehityksessä.

Dokumentaatioiden lukeminen on todennäköisesti lisännyt tehtävien edistymisen vauhtia. Vielä on vaikea arvioida sitä tarkemmin, koska ryhtiilikkeestäni on kulunut vasta noin viikko. Olen perehtynyt ColdFusionin ja CFWheelsin rakenteisiin lukemalla niiden dokumentaatiota ja tutkimalla projektihakemistoa koneellani. Perehtyminen on antanut varmuutta kehittää projektia parempaa suuntaa.

Teknillisesti osaamiseni on kehittynyt tutustumalla LESS ja SASS-kieliin. Olin aiemmin jo tietoinen siitä, että Kaukolämpö Plus järjestelmässä käytetään LESS:iä tyylittelyssä. Kaikki tyylitiedostoihin tekemäni muutokset ohjasin change managerille, joka sitten käänsi CSS-tyylitiedoston LESS-muotoon. LESS ja SASS-kieliin tutustumisen ansiosta voin tulevaisuudessa päivittää tyylitiedostoja, jos sellaiseen tulee tarvetta. Ne ovat erittäin käytettyjä teknologioita, joten niiden osaaminen ei ole huono asia (Coyier 2016).

Regular Expression on hyvin pitkään tuntunut siltä niin kuin se olisi silkkaa taikaa. Ymmärrettävyyden rajan ulkopuolella, johon ei saa minkäänlaista kosketusta. Olen aina hyödyntänyt Googlea niissä tapauksissa, jossa valtavan suuresta tekstistä täytyy valita jokin tietty pätkä jokaiselta riviltä tiettyjen merkkien välistä. Vastauksena eri foorumeilla on ollut lyhyesti, että käytä Regular Expressionia, jonka perässä etsintäkaava. Uskon, että tämän viikon aikana opittu Regular Expression ryhmät aiheuttivat palon oppimaan lisää. Jatkossa aion pyrkiä ymmärtämään Regular Expressionin etsintäkaavan rakenteen ja ymmärtämisen kautta vähentää tarvetta etsiä vastauksia foorumeilta.

ITF-viivakoodin rakenne ja sen tarkistuskoodin laskenta oli ehkä mielenkiintoisin asia viikon aikana. Skanneria tehdessä piti ottaa huomioon erityyppiset viivakoodit. Tarkistuskoodia varten kirjoittamani tarkistusfunktio oli mielenkiintoisin osa skanneriprojektia. Sen kirjoittamisessa ei mennyt kovin kauaa ja se ei ollut vaikeaa. Pääsin pitkästä ajasta käyttämään koodissa matemaattisia funktioita. En muista, koska viimeksi olisin käyttänyt käänteistä toistorakennetta. Niitä tulee käytettyä erittäin harvoin ainakin nykyisessä työssäni. Skanneriprojektissa oli muitakin "viivakoodeja". Yleisesti tunnettua QR-koodia löytyi monista eri laitteista. Nämä laitteet eivät vaatineet sen suurempia tarkistuksia. Kuitenkin yhdessä laitetypissä QR-koodin sisälle oli tallennettu verkkosivun osoite, jonka yhtenä parametrinä oli skannatun laitteen sarjanumero. Sarjanumero täytyi poimia osoitteesta ja vasta sitten suorittaa tietokantatarkistus.

Ensiviikolla voisin perehtyä enemmän Jiran käyttöön. Jiran verkkosivuilta löytyvä Best Practices -oppaan lukeminen lisäisi ymmärrystä ja hyviä käytäntöjä. Tämän takia sieltä on hyvä aloittaa Jiran maailmaan tutustuminen (Atlassian b).

3.2 Seurantaviikko 2

Maanantai 08.06.2020

Kello 12 mennessä change managerien pyytämä päivitysopas tulisi olla valmis. En vielä osaa sanoa kuinka kattava sen täytyisi olla. Uskon, että laitan siihen vain vaiheet ColdFusionin asentamisesta. Asentamisen loppuosion voi hoitaa GIT:llä. Viimeviikon aikana puskin sitä mukaa muutoksia versionhallintaan, kun sain asioita valmiiksi.

Sitten olisi tarkoitus suunnitella uusi viikon mittainen kehitysjakso. Tiedän osittain mitä se voisi sisältää viime viikon keskusteluista change managerin kanssa. Nimittäin teen testikontrollerin uudella reitityksellä, jotta voin esittää reitityksen toimintaa tiimilleni.

Sprint 1 on nyt virallisesti ohi. Sain heti aamusta paranneltua ColdFusionin ja CFWheelsin päivityksen ohjeistusta, joka jäi viimeiseksi asiaksi tälle sprintille. Nyt täytyy odotella, mitä tiimin muut jäsenet siitä kommentoivat. Suoritin itsenäisesti lyhyesti sprintin retrospektiivin. Sen tarkoituksena on miettiä, mikä onnistui sprintissä ja missä voisi jatkossa parantaa (Hietaniemi 2019).

Olen tyytyväinen ensimmäisen sprintin tulokseen. Sain jokaisen kohdan ratkottua ja toteutettua johdonmukaisesti ja riittävällä tarkkuudella perustuen hyviin tietolähteisiin. Tietolähteitä toimivat virallisesti ColdFusionin, CFWheelsin ja Bootstrapin dokumentaatiot. Käytin myös huomattavasti enemmän aikaa tietolähteiden lukemiseen ja siitä oli suuri apu.

Seuraavana luvassa oli toisen sprintin suunnittelu ja käynnistäminen. Asetin seuraavalle sprintille päätavoitteen, jossa Kaukolämpö Plus järjestelmässä olisi testikontrolleri. Kontrollerille määritettäisiin perustoiminnallisuudet. Perustoiminnallisuuksilla tarkoitan CRUD-funktiota (Liite 1). Yleisesti CRUD-funktioita käytetään silloin, kun halutaan luoda, lukea, muokata tai poistaa dataa tietokannasta (Codecademy b). Luonnin jälkeen reitittäisin kontrollerin uudella "mapper"-funktioilla, koska vanhaa reititys tapaa ei enää tueta (CFWheels).

Asetin myös tavoitteeksi tutustua CommandBox nimiseen command line interface työkaluun. CommandBoxilla voisi hallinnoida Kaukolämpö Plus järjestelmää tehokkaammin paikallisesti. CommandBoxiin saa asennettua cfwheels-cli nimisen laajennuksen, joka mahdollistaa CFWheels projekteihin kontrollireiden, mallien ja näkymien luomista suoraan CLI:sta. (Liite 1) (CFWheels.)

Päivä oli loppujen lopuksi vähän sekava. Sprintin suunnittelun ja käynnistämisen jälkeen aloin järjestelmällisesti suorittamaan tehtäviä niiden kiireys- ja tärkeysjärjestyksessä. Olin jo vähän puolen päivän jälkeen saavuttanut päivälleni asettamat tavoitteet. Tavoitteiden saavuttamisen jälkeen tein vähän sitä ja tätä hyppien sprintin tehtävien välillä. En osannut ihan päättää, mitä olisi seuraavaksi hyvä saada valmiiksi. CommandBox ja kontrollerin luominen liittyy vähän toisiinsa tai ainakin niitä voisi tutkia samanaikaisesti. CommandBoxilla on mahdollisuus luoda testikontrollereita, joten ajattelin aluksi tutustua siihen. CommandBox vaikutti turhan monimutkaiselta muutaman seikan takia ja siksi päätin aloittaa testikontrollerin tekemisen vanhaan tapaan. Jossain vaiheessa kuitenkin päätin yrittää uudelleen CommandBoxilla, mutta sekin yritys kaatui. Merkitsin Jiraan, että katson CommandBoxia myöhemmin, jos sprintillä jää siihen aikaa.

Tiistai 09.06.2020

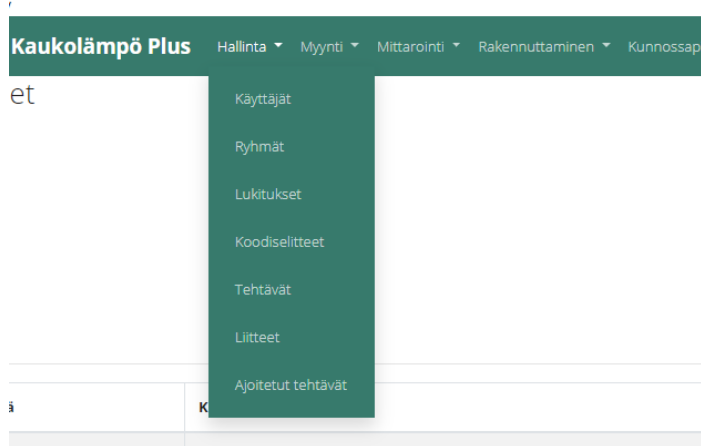
Testikontrollerin valmistuttua otan tavoitteeksi Kaukolämpö Plus järjestelmän visuaalisen ilmeen parantamisen. Nykyisen kehitysjakson yksi käyttäjätarina oli mahdollisuus nähdä AJAX-funktiot toiminnassa. En muistanut tehdä AJAX-funktiosta omaa tehtävää Jiraan. Keskenäiseen kehitysjaksoon ei voi lisätä tehtäviä tai käyttäjätarinoita. Kehitysjakso täytyisi keskeyttää, muokata ja aloittaa uudelleen. Siihen en nyt ryhdy vaan ensi kerralla keskityn tarkkaavaisuuteen ja pitkitän kehitysjakson aloittamista, kunnes kaikki kohdat ovat tarkastettu. Tavoitteeksi voisin lisätä yhden Kaukolämpö Plus järjestelmän prosessin kääntämisen uudelle versiolle. Tämä voisi olla AJAX-kontrolleri, jota yritin viime viikolla ratkoa ja josta jätin viestin foorumille. Ensimmäisen kehitysjakson tavoitteena oli Bootstrap 4.5 asentaminen. Asentaminen käsitti Bootstrapin tiedostojen päivittämisen, mutta ei kuitenkaan uusien ongelmien ratkaisemista. Uudessa Kaukolämpö Plus järjestelmässä on vielä useita eri ongelmia, kuten sellaisia, jotka estävät kokonaan joidenkin sivujen käytön.

Onnistuin tavoitteissani melko hyvin. Kyseinen vaihe on projektin sekaisin, sillä tässä kohtaa vasta testaillaan uusia asioita ja tutkitaan, mitä asioita järjestelmässä tulee muuttaa. Puhuin tiimiläisen kanssa projektista ja kerroin, että projektinhallinta järkkyy. Projekti ei järky kuitenkaan niin, että vielä täytyisi mihinkään suorastaan puuttua. Kerroin hänelle, että projektissa on nyt niin monta osa-aluetta, joista valita. En käyttänyt tarpeeksi aikaa kehitysjakson suunnitteluun. Pikaisen tilannekatsauksen jälkeen päädyin siihen, että käytän yhden tai kaksi päivää alkaen tänään sivulla esiintyvän navigaatiopalkin korjaamiseen.

Keskiviikko 10.06.2020

Tänään olisi tarkoitus jatkaa navigaationpalkin kanssa. Todennäköistä on, että navigaatiopalkki ei valmistu vielä lähipäivinä. Tiedostot, jotka ovat navigaatiopalkin ulkonäöstä riippuvaisia ei välttämättä ole lisätty projektiin onnistuneesti. Hankaluutena tyyli tiedostojen päivittämisessä on se, että ne saattavat olla riippuvaisia useista eri tiedostoista. Lisäksi, kun käytössä on SASS:in ja LESS:in kaltaiset muuttujia tukevat tyyli tiedostokielet työ hankaloiduu huomattavasti. Jokaisella muuttujalla ei ole välttämättä itseään kuvaava nimi tai on sijoitettu väärään paikkaan. Uskon, että projektin aikana tyyli tiedostoja tullaan useaan otteeseen hienosäätämään.

Päivän aikana sain edettyä navigaatiopalkin kanssa hyvin. Poistin Kaukolämpö Plus järjestelmästä SmartMenu nimisen navigaatiopalkin laajennuksen. Viimeisin SmartMenu päivitys tukee Bootstrap 4, mutta ei 4.5. Nykyinen navigaatiopalkki ei käytä laajennuksia.



Kuva 6. Kaukolämpö Plus 7.0 järjestelmän navigaatiopalkki

Kävimme myös testikontrollerin läpi tiimiläisten kesken. Avasin heille, miten nykyinen reititys toimii. Kuvastin myös, miten kontrollissa jatkossa olisi mahdollista käyttää "route" parametria kontrollerien funktioiden väliseen kommunikointiin. Aikaisemmin CFWheelsin funktiossa käytettiin vahvasti "controller" sekä "action" parametrejä, joille annettiin arvoksi halutun kontrollerin ja toiminnallisuuden nimi.

Samassa palaverissa keskustelimme projektin vaiheista. Sovimme, että uuden käyttöliittymän ei tarvitse olla täysin samannäköinen kuin edeltäjänsä. Hyvin usein noudatan liian ankarasti annettuja ohjeita. Päivitysprosessissa on osattava tehdä valintoja mahdollisuuksien mukaan. SmartMenu-laajennus oli hyvä merkki siitä. Pysin saamaan saman navigaation toimimaan uudessa versiossa. Navigaatiolaajennuksen päivittäminen osoittautui hankalaksi ja siihen kului turhaan aikaa. Kaiken aikaa olisin voinut etsiä toisen

navigaatiolaajennuksen, jossa ovat samat toiminnallisuudet. Onneksi projektissa on vielä sidonnaisia päivitettävänä. Näissä pyrin olla tekemättä samoja virheitä.

Torstai 11.06.2020

Tänään olisi tarkoitus lisätä navigaatiopalkkiin linkit. Linkkien lisäksi hyvän käyttäjäystävällisyysperiaatteiden mukaisesti linkit voisivat viedä käyttäjän toivomaansa paikkaan. Tämä tarkoittaa sitä, että reititykseen on saatava tolkkua.

Linkkien generoiminen navigaatiopalkkiin tapahtuu toistorakenteessa. Toistorakenteessa käydään läpi tietokannasta ladattuja rivejä, jotka sisältävät navigaatioon muodostettavien linkkien nimiä ja ikoneita. Linkit muodostetaan hyödyntäen CFWheelsin URLFor-funktiota ja FontAwesome-ikonikirjastoa. URLFor-funktion parametrien kanssa jouduin päivän mittaan taistelemaan.

```
<a id="#menuId#"
  href="#URLFor(controller=menu.controller, action=len(menu.action) ? menu.action : "index")#"
  data-toggle="dropdown"
  aria-haspopup="true"
  aria-expanded="false"
  class="nav-link dropdown-toggle">
  <cfif len(menu.iconClassName)>
    <span class="fas #menu.iconClassName# fa-fw"></span>
  </cfif>
  #menu.title#
</a>
```

Kuva 7. CFWheelsin URLFor-funktion käyttöä

Kuvassa 9 oleva funktio toisella rivillä ei jostain syystä halunnut toimia oletetulla tavalla. Dokumenttien mukaan se antaa käyttäjälle mahdollisuuden käyttää Controller, Action ja Route -parametrejä. URL-osoite, jonka URLFor-funktio generoi ei totellut määrittelyjäni. Funktio ohitti täysin action parametrin asetetun arvon. Oletan, että jos menu muuttujan action arvo on tyhjä niin klikatessa generoitua linkkiä käyttäjä ohjattaisiin kontrollerin oletuskäytävään. Oletuskäytävällä tarkoitan tässä tapauksessa etusivuja. Suurimmalla osalla kontrollereista on taulukko etusivulla. Oikean muotoisen linkin generoitumista ei kuitenkaan tapahdu vaan linkki jää generoitumatta kokonaan, jos menu muuttujan action arvo on tyhjä.

Perjantai 12.06.2020

Päivän tavoitteena on käsittää miksi mikään linkkeihin liittyvä ei toimi oletetulla tavalla. Navigaatiopalkin linkkien takia minulla on herännyt epäily, onko uusi version välttämättä se

versio, johon kannattaa päivittää. Aion käyttää tämän päivän linkkien tutkimiseen ja ymmärtää, miten reititys täytyy rakentaa. Dokumentaatio on epäselvä ja toivottu lopputulos ehkä mahdoton. Keskustelimme eilen tiimiläisten kesken päivityksen jälkeisestä kehityksestä. Uudessa tavassa tulisi ottaa huomioon reititys. Normaalien käytäntöjen jälkeen jokainen kontrolleri täytyisi reitittää mapper-funktiolla. Jokainen funktio, joka on määritetty kontrolleriin, täytyisi lisätä mapperiin oikeaan kohtaan. CFWheels 1.4.5 versiossa riitti vain, kun loi kontrollerin ja sen funktiot. Reititys tapahtui automaattisesti käyttäjän huomaamatta. Uusimmassa CFWheels versiossa automaattista reititystä kutsutaan wildcard-funktioksi. Wildcard-funktio on CFWheelsin reitityksessä käytettävä funktio, joka mahdollistaa kontrollerien käskyttämisen (CFWheels).

Wildcard-funktion käyttö on huono ajatus siitä syystä, että se saattaa paljastaa ei haluttuja reittejä, joiden avulla sivusto on haavoittuvainen CSRF-hyökkäyksille (Liite 1). CSRF-hyökkäyksiä tehdään muun muassa sähköpostissa esiintyvien linkkien tai kuvien kautta. Käyttäjä klikkaa jotain satunnaista linkkiä, joka näyttää ohjaavan palkintosivulle. Tosiasiassa linkki ohjaa järjestelmään, johon olet kirjautunut. Linkki saattaa sisältää ohjauksen lisäksi komennon, joka poistaa tietokannasta tietoja. CFWheelsin 2.1 reititys malli antaa paremman kontrollin, mitä mikäkin käyttäjän käskyttävä funktio saa tehdä (CFWheels).

Kaikista yleisin ja tehokkain tapa estää CSRF-hyökkäyksiä on käyttää CSRF-merkkiä sivulla tehtävissä kutsuissa (PortSwagger). CFWheelsissä sen käyttöönotto on tehty helppoksi. Kontrolleriin on lisättävä `protectsFromForgery`-funktio. Layout-tiedostoon tulee lisätä `authenticityToken` meta-merkki muodossa. AJAX-funktioihin on syytä lisätä sama `authenticityToken` omana muuttujana. (CFWheels.) Layout-tiedosto pitää sisällään Kaukolämpö Plus järjestelmän sivuston ja sen perusteet.

Kaukolämpö Plussassa on nyt mukautettu `textFieldTag`-merkki. Sen nimi on `dateTimeTag` ja se hyödyntää `textFieldTag`-merkin toimintaperiaatteita. Ne eroavat toisistaan siten, että uudessa merkissä Bootstrapin mukaisesti FontAwesome-ikoni esiintyy sen oikeassa laidassa. Ikoniksi on valittu kalenteri, sillä `dateTimeTag`-merkkiä tullaan käyttämään tilanteissa, joissa käyttäjälle halutaan esittää päivän ja aikataulun valinta. Sovellusta, joka hyödyntää luomaani `dateTimeTag`-merkkiä ei ole vielä valittu. Aikaisemmin käytössä ollut päivän ja aikataulun sovellus toimii vain Bootstrap 3.4 versiolla.

Muokkasin myös projektille kriittisiä ja Fortumin ulkopuolisille salaisia tiedostoja. Aikaisemmin useat määrittelyt olivat enemmän tai vähemmän samassa tiedostossa. Näissä tiedostoissa oli muun muassa tietoja salaisista avaimista, joiden avulla keskustellaan sovellusten kesken turvallisesti. Tiedostoissa oli myös määrittelyksiä, jotka voisivat olla

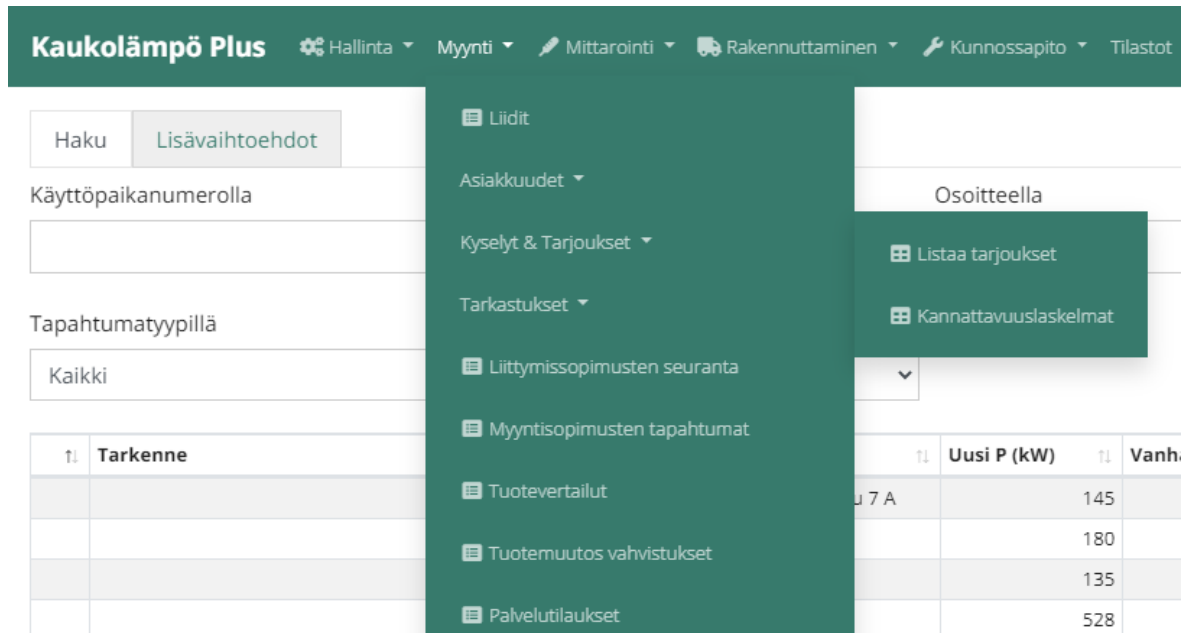
versionhallinnassa. Määrityksiä kuten, mitä CSS-luokkaa käytetään Kaukolämpö Plus järjestelmän kaikissa tekstikentissä. Nyt tiedostot ovat jaettu karkeasti kahteen ja tiedostot yhdistetään "<cfinclude> funktiolla settings.cfm nimisessä tiedostossa. Settings.cfm on nimensä mukaisesti asetustiedosto, jonka CFWheels lukee järjestelmän käynnistyessä.

Sovimme tiimiläisen kesken, että käytämme jatkossa CFWheelsin 2.1 reititysmallia. Reititysmalli lisää turvallisuutta järjestelmään, koska se mahdollistaa eri http-kutsujen erottelun. Muutostyö tulee vaatimaan hyvin paljon refaktorointia, mutta onneksemme projektiin on varattu monta viikkoa.

Viikkoanalyysi

Päivien ja viikko -tavoitteiden osalta viikko ei ollut varsin kiireinen eikä haastavin ainakaan tähän mennessä. Tästä syystä tekninen kehittyminen ei ollut varsin suurta. Viikon aikana jouduin selvittämään erilaisten Kaukolämpö Plus järjestelmän sidonnaissovelluksien yhteensoveltuvuuksia. Suuremmaksi ongelmaksi esiintyi CFWheelsin URLFor- ja LinkTo-funktiot. Kaukolämpö Plus järjestelmässä aikaisemmin käytettiin samoja funktiota, mutta eri parametreillä. Aiemmin käytetyt parametrit olivat "Controller" ja "action" nimisiä. Niiden toimintaperiaate on täysin sama kuin "route" parametrillä. Ne eroavat toisistaan siten, että "route" mahdollistaa dynaamisesti generoitujen reittien käyttämisen, kun taas "controller" ja "action" parametrit eivät. Suurin haaste oli ymmärtää, miksi "controller" ja "action" parametrit eivät toimineet CFWheels 2.1 versiossa.

Ymmärrettyäni reitityksen minulle jäi parempi käsitys, miten ja mitä Kaukolämpö Plus järjestelmästä tulee muuttaa. Vähemmän ongelmalliseksi osoittautui eri sidonnaissovellusten käyttöönotto. Aikaisemmassa Bootstrap versiossa käytetyt sidonnaissovellukset kuten SmartMenu laajennus ei toiminut suoraan nykyisessä Bootstrap versiossa. Tämän takia oli tarve löytää vastaava laajennus samoilla toiminnallisuuksilla tai sitten muokata aikaisempi toimimaan uudessa Bootstrap versiossa. Vähemmän aikaa vievä vaihtoehto osoittautui järkevämmäksi vaihtoehdoksi.



Kuva 8. Kaukolämpö Plus 7.0 järjestelmän navigaatiopalkki monitasoisuudella

Nykyisessä Kaukolämpö Plus järjestelmässä ei ole erillistä navigaatioon tarkoitettu sovelusta. Monitasoisuus näkyy kuvassa 8. Navigaatio toimii nyt Bootstrapin standardin mukaisesti ja sen tukena on yksinkertainen JQuery-skripti, joka mahdollistaa monitasoisuuden. JQuery on JavaScript kirjasto, jonka päätarkoitus on tehdä JavaScriptin kirjoittamisesta vähemmän aikaa kuluttavaa.

Viime viikolla asettamani tavoite tutustua Jiraan epäonnistui. En nähnyt tarpeelliseksi keskittyä Jiran oppimiseen kuitenkaan, vaikka aluksi näin ajattelin. En tutustunut Jiraan, siksi koska en rehellisesti sanottuna tiennyt, mitä siitä olisi hyvä tietää. Todennäköisesti olisi hyvä lukea aiemmin mainitsemani Jiran Best Practises guiden. Oppaan lukeminen unohtui täysin viikon muiden tehtävien priorisoinnin takia. Maanantaina ennen puolta päivää minulla on hyvää aikaa lukea opas, jos siltä tuntuu.

Kehitysjakson muut asettamani tavoitteet ovat täytyneet lähestulkoon täysin. Asettamani tehtävä parantaa Kaukolämpö Plus järjestelmän ulkoasua on vielä kesken. Tehtävä on hyvin laaja ja se on yleisesti ottaen huono asia. Cassidyn mukaan kehitysjakson tehtävän sopiva laajuus tulisi olla kahdesta kuuteentoista tuntiin, kun taas Francino kertoo, että tehtävien suorittamiseen saisi mennä korkeintaan yksi päivä (Cassidy 2016; Francino). Pieniä tehtäviä on helpompi hallinnoida. Ne myös helpottavat keskittymistä ja lisää läpinäkyvyyttä tiimin kesken (Francino).

Viikko oli loppujen lopuksi erittäin täyteläinen, vaikka välillä tuntui, ettei projekti liiku mihinkään suuntaan. Olemme siirtymässä projektissa suorittavaan vaiheeseen. Maanantaina

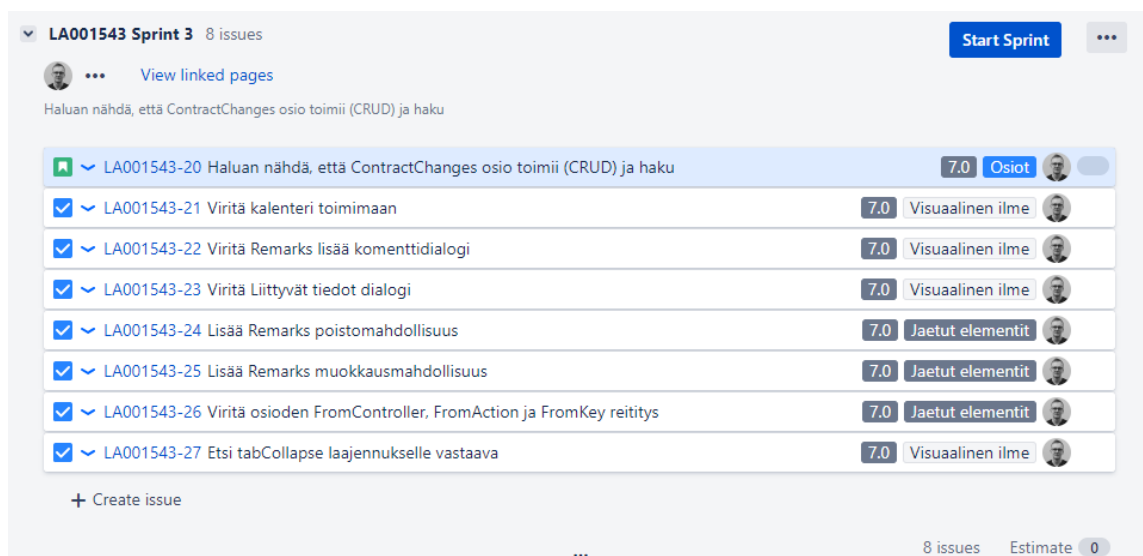
on tarkoitus lopettaa ja käynnistää uusi kehitysjakso. On vielä vaikea arvioida, mitä ensi viikko tuo tullessaan. Varmaa on, että Jiraan keskitytään tämänkin viikon edestä ja uuden kehitysjakson suunnitteluun käytetään enemmän kuin puolituntia.

3.3 Seurantaviikko 3

Maanantai 15.06.2020

Puolelta päivin kehitysjakso kaksi loppuu ja kolmatta aletaan suunnittelemaan. Päivän tavoitteena on viimeistellä edellisen viikon aiheet ja sitten keskittyä uuden kehitysjakson suunnitteluun. Yhdeksi tavoitteeksi ennen puolta päivää asetetaan Jiran oppaiden lukeminen. Ennen seuraavan kehitysjakson suunnittelua kehitysjaksosta yksi tulisi pitää retrospektiivi. Juuri alkaneella viikolla haluaisin oppia projektihallinnollisia asioita.

Tutkin aamulla Jiraan liittyviä asioita. Käytin oppimateriaalina muutamia videoita Youtubesta (Chuparkoff; Packt Video 2018). Niissä käsiteltiin Jiran eri toiminnallisuuksista sekä Agile toimintamallin periaatteita. Agile on yksi monista ketteristä projektinhallintamenetelmistä. Agile on iteratiivinen malli, jonka ideana on pilkkoa työtä helpommin hallintaviin osiin. Työtä suoritetaan yleensä 1 - 4 viikon kehitysjaksoissa (Atlassian c). Kuvassa 9 nähdään tiimiläisten kesken suunnitteleman seuraava kehitysjakso. Päätimme yhdessä, että seuraava vaihe on toteuttaa yksi olemassa oleva Kaukolämpö Plus järjestelmän prosessi tai "osio" kokonaisuudessaan. Kokonaisen osion päivittäminen antaa paremman käsityksen, mitä kaikkea on otettava huomioon ohjelmistokoodia päivittäessä.



LA001543 Sprint 3 8 issues

Start Sprint

View linked pages

Haluan nähdä, että ContractChanges osio toimii (CRUD) ja haku

| ID | Description | Label | Assignee |
|-------------|--|----------------------|----------|
| LA001543-20 | Haluan nähdä, että ContractChanges osio toimii (CRUD) ja haku | 7.0 Osiot | [Avatar] |
| LA001543-21 | Viritä kalenteri toimimaan | 7.0 Visuaalinen ilme | [Avatar] |
| LA001543-22 | Viritä Remarks lisää kommenttidiologi | 7.0 Visuaalinen ilme | [Avatar] |
| LA001543-23 | Viritä Liittyvät tiedot dialogi | 7.0 Visuaalinen ilme | [Avatar] |
| LA001543-24 | Lisää Remarks poistomahdollisuus | 7.0 Jaetut elementit | [Avatar] |
| LA001543-25 | Lisää Remarks muokkausmahdollisuus | 7.0 Jaetut elementit | [Avatar] |
| LA001543-26 | Viritä osioiden FromController, FromAction ja FromKey reititys | 7.0 Jaetut elementit | [Avatar] |
| LA001543-27 | Etsi tabCollapse laajennukselle vastaava | 7.0 Visuaalinen ilme | [Avatar] |

+ Create issue

8 issues Estimate 0

Kuva 9. Kolmannen kehitysjakson suunnittelua

Kaukolämpö Plus järjestelmän osion toteuttaminen antaa myös mahdollisuuden päivittää puuttuvia sidonnaisohjelmia, jotka ovat käytössä monessa eri osiossa. Packt Videon videossa kerrotaan epic:stä. Epic:it ovat kuin käyttäjätarinoita, mutta pidempiä. Niillä on selkeä alku ja loppu. Loin muutaman epic:n Jiraan näin aluksi ja lisäsin ne jokaiselle tehtävälle tai käyttäjätarinalle. Kuvassa 9 näkyy "Visuaalinen ilme", "Jaetut elementit" ja "Osiot".

Visuaalinen ilme-epic:n idea on sisältää kaikki tulevat Kaukolämpö Plus järjestelmän visuaaliset lisäykset ja parannukset. Jaetut elementit-epic sisältää kaikki sellaiset sivulla toistuvat elementit, kuten sidonnaisohjelmat. Osiot-epic sisältää kaikki tulevat käyttäjätarinat, tehtävät ja korjaukset, mitä liittyy Kaukolämpö Plus järjestelmän osioihin. Packt Videon videossa kerrottiin, että epic:ien käyttö parantaa mahdollisuutta seurata projektin kulkua.

Korjasin myös ongelmia liittyen navigaatiopalkin monitasaisuuteen. Viimeviikolla lisäämäni JQuery-skripti oli puutteellinen. Navigaatiopalkki sisältää listoja listojen alla ja ne eivät sulkeutuneet silloin kun käyttäjä painoi toista sisennettyä listaa. Kyseessä oli visuaalinen bugi, johon oli yksinkertainen ratkaisu. Täytyi vain määrittää sisennettyjen listojen sulkeutuminen klikatessa toista samalla tasolla olevaa sisennettyä listaa.

Huomenna olisi tarkoitus käynnistää hieman alle viikon mittainen kehitysjakso, jonka tavoitteena on saada ContractChanges-osio valmiiksi Kaukolämpö Plus järjestelmässä. Kuvassa 9 näkyy kolmannen kehitysjakson suunnitelua. En usko, että ylitse pääsemättömiä esteitä tulee vastaan. Minulla on hyvin hallussa ne seikat, jotka pitää ottaa huomioon Kaukolämpö Plus järjestelmää päivittäessä.

Tiistai 16.06.2020

Tavoitteena tänään on saada ContractChanges-osion editointi, luominen ja poistaminen toimimaan. Samalla korjaan jokaisen sivun vastaamaan Bootstrapin ohjeistusta. Ohjeistuksessa sanotaan, että sivun sisältö pitäisi olla aina Bootstrapin sarakkeiden alla. Sarakkeet on oltava rivien alla ja rivit eivät saa olla suoraan rivien alla (GetBootstrap).

Päivän aikana sain edistettyä ContractChanges-osiota hyvin. CFWheelsin reitityksen kautta kontrollereissa täytyisi käyttää new-funktiota, silloin kun halutaan luoda uusi objekti (CFWheels). Koko Kaukolämpö Plus järjestelmässä new-funktion tilalla on käytetty "add"-funktiota. Ne ajavat täysin saman asian, mutta ensimmäinen näistä on uuden reititysmallin automaattisesti luoma. Jatkossa puhutellaan ContactChanges-osion edit-funktiota "editContractsChange" reitillä. CFWheels siis generoi tämä automaattisesti, kun merkitsee reititysfunktion resources-funktiolla uuden kontrollerin. Resources-funktio luo edit, new, create, show, update ja delete päätepisteet tai "endpointit" annetulle nimelle (CFWheels). Tämä monen muun asian joukossa pitää ottaa huomioon jokaisessa tulevassa kontrollereissa.

Lisäsin puuttuvan sidonnaisohjelman nimeltä datepickerin ja korvasin BootstrapDialog-laajennuksen käyttämään natiivia Bootstrap 4.5 modal-ikkunaa (t1mon). BootstrapDialog-

laajennuksella pystyi helposti luomaan dynaamisia ikkunoita, joissa pystyttiin näyttämään informaatiota käyttäjälle. Nykyään se ei ole välttämätöntä, koska Bootstrapissa on sellainen sisäänrakennettuna (Bootstrap). Tosin ilman BootstrapDialog-laajennusta ikkunoiden luominen tulee olemaan työläisempää, sillä Bootstrap ei tarjoa dynaamista ikkunoiden luontia.

```
<cfif NOT contractChange.isNew() AND NOT locked AND model("user").hasPermission("delete")>
  <div class="col-auto">
    #deleteButtonTag(
      route="ContractChange",
      key=contractChange.id,
      buttonText="Poista myyntisopimustapahtuma",
      confirmText="Haluatko varmasti poistaa myyntisopimustapahtuman?"
    )#
  </div>
</cfif>
```

Kuva 10. DeleteButtonTag-merkin käyttö ohjelmistokoodissa

Loin uuden merkin nimeltä "deleteButtonTag", joka näkyy kuvassa 10. Sen tarkoitus on toimia nykyisen poistamiseen tarkoitetun napin tilalla. Sen käyttö tulee olla mahdollisimman helppoa, jonka toimintaa voidaan ohjata sille määritettävillä parametreillä. Lisäsin myös kyseisen merkin palautteeseen dynaamisen varmistusikkunan. Käyttäjälle esitetään aina kuvassa 10 näkyvä "confirmText" parametrin arvo ja kaksi painiketta. Toinen painikkeista poistaa poistettavan asian ja toinen peruuttaa toiminnon.

Keskiviikko 17.06.2020

Tänään olisi tarkoituksena jatkaa siitä, mihin eilen jäätiin. Lisäystoiminto on seuraava toiminto, jonka haluan lisätä ContractChanges-osioon. Samalla joutuu myös ajattelemaan reititystä ja sen helpottamista. Uuden ContractChanges-osio luonti edellyttää ConsumptionPoint-objektin. Käyttäjän painaessa Lisää-painiketta ContractChanges-osiossa käyttäjä ohjataan ensin valitsemaan sopiva ConsumptionPoint. ConsumptionPoints on lista käyttöpaikoista.

Sain lisättyä ContractChanges-osion lisäystoiminnon. Lisääminen oli yllättävän helppoa varsinkin nyt, kun ymmärtää reitityksen päälle. Uusi reititysmalli asettaa muutamia ongelmia vielä ratkottavaksi. Tänään kamppailin ymmärtää ratkaisua siihen, miten nykyisen kontrollereiden välisen liikkuminen kannattaisi rakentaa. Aiemmin kontrollereiden välillä käytettiin parametrejä, jotka kertoivat, mistä käyttäjä tulee. Sitä käytettiin silloin, kun käyttäjälle haluttiin antaa mahdollisuus palata edelliselle sivulle Takasin-painiketta painamalla. Myös silloin, kun haluttiin vaikuttaa, miten sivusto käyttäytyy riippuen käyttäjän lähteestä. Muutamassa osiossa on käytössä parametrit, jotka kuvastavat käyttäjän vielä

aikaisemman lähteen. Funktiot, jotka rakentavat näitä parametreja on muokattava siten, että ne tukevat ja käyttävät pääasiassa uutta route-parametria.

Torstai 18.06.2020

Tänään olisi tarkoitus saada lähes jokaisessa kontrollerissa esiintyvä Search-toiminta valmiiksi. Samalla on tarkoitus jatkaa eilisen reitityksen kanssa. Lopuksi jos aikaa jää niin Remarks-osio olisi tarkoitus virittää käyttämään jotain BootstrapDialogin tyylistä laajennusta. Joudun todennäköisesti muokkaamaan nykyistä BootstrapDialog 4.0 versiota siten, että se toimisi nykyiseissä 4.5 versiossa.

ContractChanges-osio on nyt 90 % valmis. Reititys näyttää toimivan niin kuin pitää. Kokeilin päivän aikana eri variaatioita, miten linkit olisi mahdollista rakentaa. Näiden generoimisessa oli eniten ongelmia. Hankaluuksia tuotti oikeiden parametrien käyttö ja niiden perusteella linkkien rakentaminen. Kaukolämpö Plus järjestelmään oli jo aikaisemmin rakennettu tähän tarkoitukseen funktio, mutta kyseinen funktio ei piitannut route-parametristä. Sitä funktiota käytetään lähes jokaisessa Kaukolämpö Plus järjestelmän osiossa. Kivuttommin ratkaisu oli siis muokata kyseinen funktio sellaiseksi, joka ottaa huomioon vain route-parametrin ja key-parametrin.

Route-parametri kertoo CFWheels:ssä, mihin kontrolleriin ja funktioon komento kohdistetaan (CFWheels). Key-parametri useimmissa tapauksissa tarkoittaa tietokannasta käsiteltävien rivien uniikkeja id:tä. Linkit generoidaan route ja key parametrejä käyttäen näin: `URLFor(route="editContractChange", key=key)`, joka tulostaa käyttöliittymään `"/index.cfm/contract-changes/[key]/edit"`. Index.cfm jälkeisen kenoviivan kenoviivojen sisältö kuvaa kyseistä kontrolleria, jota halutaan käsitellä. Linkki määräytyy URLFor:n route-parametrin mukaan. Hakasulkujen sisälle key:n tilalle tulee URLFor:n key-parametrin asetettu arvo. CFWheelsiin on rakennettu `$findRoute`-funktio, jota käytetään monessa muussa CFWheelsin funktiossa. Sillä voidaan etsiä annetuilla arvoilla mapperiin asetettu polku eli route ja siihen liittyvä kontrolleri ja action. Kyseessä olevan linkin loppuosa "edit" määräytyy myös route-parametristä.

Rakensin funktion, joka hyödyntää CFWheelsin `$findRoute`-funktioita. Funktio palauttaa käyttäjälle osoitteen, jota voi käyttää linkeissä. Osoite generoidaan annetuista parametreista. ContractChanges-osiossa Lisää-toiminto vaatii käyttäjän ensin valitsemaan myyntisopimustapahtumalle käyttöpaikan. Käyttäjälle esitetään lista käyttöpaikoista, joita klikkaamalla käyttäjä ohjataan luontisivulle, jossa käyttäjä voi asettaa tietoja koskien myyntisopimustapahtumaa. Myyntisopimustapahtumat koskevat aina käyttöpaikkoja. Tähän listaan

täytyi generoida ”palaa edelliselle sivulle” linkki rakentamalla funktiolla fromRoute-parametristä. Linkin täytyy myös sisältää tieto valitusta käyttöpaikasta. FromRoute-parametri kertoo järjestelmälle, miltä sivulta käyttäjä on navigoitunut nykyiselle sivulle.

Viikkoanalyysi

Asettamani tavoitteet seurantaviikolle 3 eivät täytyneet täysin. Maanantaina käytin muutama tunnin Jiran opiskeluun lähinnä opetelleen uusia Agile Scrum-aiheita. Scrum on sovelluskehitysmalli projektinhallintaan. Scrum keskittyy projektin vaiheistamiseen ja jatkuvaan kontrolliin (Meteoriitti). Epic:n käyttö mahdollistaa nähdä projektin kokonaiskuvan osina (Hygger). Aiemmin viikolla käydessäni Packt:n ja Chuparkoff:n videomateriaalia selvisi, että Epic:t ovat isoja käyttäjätarinoita niin kuin Hyggerin artikkelista kävi ilmi.

Maanantai päivän loputtua kolmannes kehitysjakso oli päättynyt. Kaikki tavoitteet saavutettiin hyvissä ajoin. Kehitysjakson suunniteluun käytettiin huomattavasti enemmän aikaa kuin edelliseen ja se näkyi työn teossa. Työn teko oli tehokkaampaa ja järjestelmällisempää. Uskon, että parannettavaa on kummassakin vielä paljon. Olisi mielenkiintoista tietää, miten työn tehokkuutta voisi lisätä muutenkin kuin suunnittelemalla ennen aloittamista.

Ensiviikolla selvitän työtehokkuuden lisäämiseen liittyviä asioita. Tiedän, että tauotonta työskentelyä kannattaisi välttää. Usein löydän itseni viettämässä ruoka- ja kahvitaukoja tietokoneen ääressä. Ruokatauot eivät välttämättä ole lakisääteisien mukaisia, vaan ruoka ahmitaan mahdollisimman nopeasti. Tauton työskentely aiheuttaa ongelmia keskittymiseen ja keskittymisvaikeudella on madaltavia vaikutteita työtehokkuuteen. Etätöitä tehdessä olen huomannut, että nämä asiat korostuvat entistä enemmän. Tauottoman työskentelyn lisäksi työpäivät venyvät etätöissä helpommin kuin toimistolla. Tähän haluaisin kiinnittää enemmän huomiota. Työhyvinvointi on oikea asia ja siitä kannattaa pitää huolta.

Ensiviikolla pääsen viimein aloittamaan Kaukolämpö Plus projektin osioiden uudelleen kirjoittamisen. ContractChanges-osion päivittämisestä kävi ilmi, että muutoksia ei ole kontrolleri tasolla välttämättä aivan järkyttävästi. Kontrollereita on tosin yli 80, mutta moni niistä on suhteellisen yksinkertaisia. ContractChanges oli niistä ehkä yksinkertaisin. Pitää myös miettiä, miten pilkon osioiden päivittämisen Jiraan. Tästä asiasta voisi löytyä internetistä esimerkki.

3.4 Seurantaviikko 4

Maanantai 22.06.2020

Tänään olisi tarkoitus pitää kolmannen kehitysjakson retrospektiivi. Sen jälkeen suunnitella uusi neljäs kehitysjakso. Tuleva kehitysjakso tulee koostumaan Kaukolämpö Plus osioiden päivittämisestä, yleisen ulkonäön hiomisesta ja muutamien puuttuvien sidonnaisohjelmien päivittämisestä.

BootstrapDialog-laajennus toimii nyt niin kuin pitää. Sen ohjelmistokoodia täytyy muuttaa siten, että laajennus generoi tiettyjä HTML-elementtejä tietyillä luokilla. Tämä on erittäin hyvä, sillä voimme hyödyntää jo ohjelmoituja Bootstrapin ikkunoita. Laajennus käyttää vielä Bootstrapin oletusteeman värejä, joten ne täytyy vielä vaihtaa. BootstrapDialogia täytyy vielä testailta pitkin viikkoa ennen kuin olemme siihen tyytyväisiä. Nykyiset korjaukset näyttävät lupaavilta.

Lisäsin puuttuvat AJAX-funktiot mapper-funktioon. Lisääminen oli kannattavaa tehdä tässä vaiheessa, sillä moni BootstrapDialogin ikkuna käyttää AJAX-kontrolleria, jossa kyseiset funktiot sijaitsevat. Asetin kaikki funktiot sallimaan vain GET-pyyntöjä näin aluksi, koska en muistanut, mitkä kaikki olivat GET tai POST-tyyppisiä. GET ja POST-pyyntöt ovat HTTP-protokollassa käytettäviä pyyntöjä, joilla voidaan ladata tai lähettää dataa (Liite 1).

Remarks-osio on myös osittain valmis ja hyödyntää nyt uutta BootstrapDialogia. Remarks-osio on Kaukolämpö Plus järjestelmässä esiintyvien osioiden kommenttipalsta. Käyttäjät voivat jättää kommentteja koskien esimerkiksi myyntisopimustapahtumia. Refaktoroin Remarks-kontrolleria muuttamalla add-funktion new-funktioksi ja erottamalla uuden new ja vanhan edit-toiminnallisuuden toisistaan omiksi näkymiksi. Remarks-osiossa on vielä bugi, joka estää käyttäjän näkemästä käyttäjien jättämiä kommentteja.

Parantelin myös Kaukolämpö Plussan ulkonäköä. Kirjasinkokoa pienennettiin huomattavasti. Tekstikenttien ja nappuloiden kokoja pienennettiin. Rivien ja sarakkeiden välejä kaivennettiin. Navigaatiopalkin värejä ja ikoneita korjattiin. Otin mahdollisimman paljon mallia aikaisemmasta Kaukolämpö Plus 6.0 järjestelmän teemasta. Keskityin lähinnä siihen, että käyttöliittymän elementit mahtuvat sivulle vanhaan tapaan.

Tiistai 23.06.2020

Päivän tavoitteena on jatkaa siitä mihin eilen jäätiin. Remarks-osioon täytyy vielä lisätä poistomahdollisuus sekä kommenttien ulkonäköä täytyy hiota. Remarks-osio vaatii myös muutamia elementtimuutoksia. Bootstrap 4.5 ei enää tue Panel-komponentteja (Bootstrap). Panel-komponenttia on käytetty useissa eri paikoissa Kaukolämpö Plus järjestelmän eri osioissa. Panel-komponentin tilalla tulee jatkossa käyttää Card-komponenttia. Remarks-osion valmistuttua voin siirtyä seuraavan osion päivittämiseen, joka on ConsumptionFlexibilities.

Remarks-osio on nyt päivitetty uusimmalle versiolle. Remarks-osioissa on Card-komponentti ja se vastaa Kaukolämpö Plus järjestelmän odotuksia. Sitä täytyy vielä testaila ennen kuin voidaan todeta, että se toimii niin kuin pitää.

Reitityksessä käytetty fromRoute-parametri aiheutti ongelmia linkkien luomisessa. Linkit eivät sitenkään generoitunut oikein niin kuin aluksi kuvittelin. Yksinkertaisin keino korjata ongelma oli luoda uusi parametri. Parametrin nimeksi valitsin "toRoute". Sen tarkoitus on kertoa käyttäjälle, mihin käyttäjä tulisi siirtää uudelleen ohjauksen jälkeen. Linkki voidaan jatkossa generoida toRoute-parametrin avulla uudella funktiolla. Funktioon annetaan tieto, mihin käyttäjä halutaan ohjata ja uniikki id, jos sellainen on saatavilla. Sen käyttäminen on yksinkertaista ja samantapaista kuin aikaisemmat samaan tarkoitukseen rakennetut funktiot. Uskon, että uusi funktio auttaa selkeyttämään ohjelmistokoodia.

Keskiviikko 24.06.2020

Tänään olisi tarkoitus jatkaa ConsumptionFlexibilities-osion päivittämistä. Päivitys tulee olemaan hieman isompi päivitys kuin edellinen, sillä ConsumptionFlexibilities-osio sisältää Remarksin lisäksi Attachments ja Contacts-osiot.

Sain edistettyä ConsumptionFlexibilities-osiota hyvin. Contacts ja Attachment-osiot valmistuvat hyvää vauhtia ConsumptionFlexibilities-osion kanssa. Olen miettinyt täytyisikö vaihtaa lähestymistapaa päivittämiseen seuraavassa kehitysjaksossa. Nykyinen tyyli toimii varmasti, mutta en tiedä, onko se kaikista tehokkain. Nykyinen tyyli on päivittää kontrolleri kerrallaan. Päivittää jokaisen kontrolleriin liittyvät sivut, joita on jokaisessa keskimäärin viisi. Olen saanut muutaman päivän sisällä melko tehokkaasti päivitettyä kolmea kontrolleria kerrallaan. Hidastava tekijä on se, kun joutuu jatkuvasti hyppimään eri aiheiden välillä. Vaihtoehtona olisi käydä jokaisessa kontrollerissa muokkaamassa yksi toiminto. Pääasiassa sellaiset toiminnot, jotka toistuvat kontrollerien välillä. Silloin tekeminen säilyisi

samankaltaisena. Tällainen voisi todennäköisesti lisätä tehokkuutta, koska voisin keskittyä yhteen toiminnallisuuteen kerralla.

Torstai 25.06.2020

Tänään olisi tarkoituksena käydä läpi, mitkä osiot on syytä ottaa huomioon ja mitä ei. Pidämme noin tunnin mittaisen palaverin, jossa käymme Kaukolämpö Plus järjestelmän läpi. Päivä jatkuu eiliseen tapaan päivittäen ConsumptionFlexibilities, Attachments ja Contacts-osiota.

Reititykseen jouduin taas kerran tekemään muutoksia, jotta se tukee dynaamisesti asetettuja muuttujia. Huomasin aiemmin viikolla tehdyn funktion vielä tuottavan ongelmia tilanteissa, joissa käyttäjä ohjataan toiseen paikkaan ja sitten takaisin lähtöpaikkaan. Nykyinen versio toimii ainakin kolmessa erilaisessa paikassa.

Lisäsin päivän aikana uusia funktiota mapperiin ja yritin pohtia, mitä funktiota voisi jättää päivittämättä tai poistaa kokonaan. Yhdeksän aikaan pidimme tiimipalaverin, jonka tarkoituksena oli selvittää pohtimaani kysymykseen vastaus. Kaukolämpö Plus järjestelmästä poistetaan muutamia kontrollereita, koska niiden käyttö on vähäistä tai siirretty toisaalle.

Pohdimme myös yhteisesti millä tavalla voisin lisätä tehokkuutta työssäni. Tulimme siihen tulokseen, että on parempi ajatella kokonaisuuksien kannalta. Tehdään yksi asia kerrallaan mahdollisimman valmiiksi ja sitten siirrytään seuraavaan. Tällä hetkellä tehokkuutta vähentää se, kun joutuu jatkuvasti hyppimään eri aiheiden välillä.

Perjantai 26.06.2020

Tänään olisi tavoitteena jatkaa siitä, mihin eilen jäin. Yritän saada tämän päivän aikana Contacts ja Attachments-osiot mahdollisimman valmiiksi. Muutamia JavaScript funktiota on vielä päivittämättä sekä ohjauksia kontrollereiden välillä.

ConsumptionFlexibilitiesin Contacts ja Attachments-osioiden yhdistävät tekijät ovat nyt valmiita. Päivittäminen vaati paljon rakenteellisia ja JavaScript-muutoksia. Nykyään ConsumptionFlexibilitiesille pystyy vanhaan tapaan lisäämään yhteyshenkilöitä ja liitteitä.

Työpäivän aikana ilmaantui mielenkiintoinen asia liittyen CFWheelsin toiminnallisuuteen. Kaukolämpö Plus järjestelmässä on kirjoitettu useassa kohdassa kaksi eri toiminnallisuutta saman funktion sisälle. Funktio saattaa ensin ohjata käyttäjän valitsemaan vaaditun

tiedon. Valitsemisen jälkeen käyttäjä kutsuu samaa funktiota painamalla linkkiä tai nappulaan, jonka jälkeen funktio käyttäytyy eri tavoin yleensä päivittäen valittua tietoa jollain tavalla. Funktiot ovat kirjoitettu yleensä siten, että toiminnallisuus on eri, jos "key" arvo löytyy URL-osoitteesta. Jossain tapauksissa on hyödynnetty isGet ja isPost-funktiota, jotka hyödyntävät CFWheelsin request scopea (CFWheels).

```
/**
 * Returns whether the request was a normal `GET` request or not.
 *
 * [section: Controller]
 * [category: Miscellaneous Functions]
 */
public boolean function isGet() {
    return request.cgi.request_method == "get";
}

/**
 * Returns whether the request came from a form `POST` submission or not.
 *
 * [section: Controller]
 * [category: Miscellaneous Functions]
 */
public boolean function isPost() {
    return request.cgi.request_method == "post";
}
```

Kuva 11. isGet ja isPost-funktiot CFWheelsin miscellaneous-tiedostossa

Kuvassa 11 esiintyvät funktiot palauttavat boolean arvon, onko request.cgi.request_method arvo "get" tai "post". Boolean on primitiivinen data tyyppi, jolla on kaksi mahdollista arvoa 0 ja 1 tai yleisimmin tunnettuna true ja false (Microsoft a). Samaan funktioon voidaan kirjoittaa eri toiminnallisuuksia näiden funktioiden avulla vähän samaan tapaan kuin "key" arvon mukaan. Tosin "key" vaihtoehdolla get ja post-pyyntöjen välillä ei ole eroa. Suositeltavaa olisi käyttää niiden yhdistelmää tai sitten jakaa funktio kahteen ja nimetä ne toisistaan erottuviksi.

HTML-lomakkeet tukevat vain GET ja POST-pyyntöjä (W3 b) ja useissa tapauksissa olisi tarvetta lähettää PATCH, PUT ja DELETE-pyyntöjä, mutta ne eivät ole natiivisti tuettuja. Rajoituksen voi kiertää CFWheelsissä siten, että asettaa lomakkeen metodin valitulle metodille. CFWheels generoi lomakkeisiin _method nimisen piilokentän, jonka arvona formiin asetettu method arvo. Kyseisen method arvon avulla isGet ja isPost-funktiot päättävät funktiossa, mitä halutaan tehdä. CFWheelsistä löytyy myös muutama muu samanlainen funktio kuten isPatch, isPut ja isDelete.

Haluan välttää koodipohjan uudelleen kirjoittamista siellä, missä se on tarpeetonta. Uudessa reititysmallissa täytyy erotella funktiot hyväksymään tiettyjä pyyntöjä. Useassa tilanteessa sama funktio täytyy kirjoittaa kahteen kertaan mapperiin, koska mapperi ei hyväksy POST-kutsuja reitteihin, johon on määritetty sallimaan vain GET-pyyntöjä. Ensimmäinen on yleensä GET-pyyntöjä varten ja toinen POST, PATCH, PUT tai DELETE funktiota varten.

Huomasin, että yhdessä tapauksessa isPatch-funktio ei kunnioita asettamaani lomakkeen metodia. Vaikka `_method` piilokentässä olisin "patch" arvo koodissa joutuu käyttämään isPost-funktiota eikä isPatch-funktiota. Tämä mielestäni epäjohdonmukaista ja siksi jätin tästä kysymyksen CFWheelsin kehityspalstalle. (Google Groups; Uusitalo 2020b.)

Viikkoanalyysi

Viimeviikon analyysissä asettamani tämän viikon tavoite ei täytynyt, sillä se unohtui täysin. Ajattelin perjantaina päivän lopussa, että jatkan nykyistä tavoitetta seuraavalle viikolle. Torstaina koko ajatus tavoitteesta tuli vasta ensimmäisen kerran mieleen. Perjantai kiirehti ohitseni ja koko tavoite jäi vain ajatuksen tasolle. Ensi viikolla katson työhyvinvointiin liittyviä asioita heti alkuvuikosta sen sijaan, että jättäisin ne viimeiseksi.

Alkuvuikosta työssäni ei ilmennyt ylitsepääsemättömiä ongelmia. Loppuvuikosta esiintyi epäjohdonmukaisuuksia CFWheelsin kanssa. `IsDelete`, `IsGet`, `IsPatch`, `isPost` ja `IsPut` funktiot eivät toimineet niin kuin oli annettu ymmärtää. Jätin tästä asiasta CFWheelsin kehitysfoorumille kysymyksen, johon odotan vastausta viikonlopun yli. Ongelmaan, jota yritän ratkaista on muutamia muitakin ratkaisukeinoja käytettävissä, mutta haluan ymmärtää miten kyseisiä funktiota täytyisi käyttää.

CFWheelsin dokumentaatioissa annettiin ymmärtää, että edellisessä kappaleessa mainituilla funktiolla voidaan tarkastaa ja erottaa tehty pyynnöt toisistaan (CFWheels). Kyseiset funktiot palauttavat boolean arvon riippuen siitä, onko pyynnön tyyppi sama kuin se oletettu tyyppi. `isGet` ja `isPost`-funktiot toimivat kuten pitää, mutta ongelmat ilmenivät käyttäessäni `isPatch`-funktiota. Mietimme asiaa hetken tiimiläiseni kanssa ja tulimme siihen tulokseen, että CFWheels:ssa on epäjohdonmukaisuuksia näiden funktion välillä. Jätin epäjohdonmukaisuudesta kehitysfoorumille kysymyksen. Kehitysfoorumilla toinen henkilö oli kysynyt samaa kysymyssivullani ja hän jäi myös odottamaan vastausta kysymääni kysymykseen. (Google Groups; Uusitalo 2020b.) Uskon, että epäjohdonmukaisuus ratkeaa viimeistään ensi viikon puoliväliin mennessä.

3.5 Seurantaviikko 5

Maanantai 29.06.2020

Tänään on tavoitteena saada isPatch-kysymykseen vastaus. ConsumptionFlexibilities-osio olisi saatava valmiiksi puoleen päivään mennessä, sillä kehitysjakso 4 päättyy silloin. Loppupäivästä suunnittelen uuden kehitysjakson, pidän retrospektiivin ja tutustun työhyvinvointiaiheisiin lukemalla ainakin muutaman artikkelin.

Jättämäni kysymykseen oli vastattu hyvin epämääräisesti. Luulen, että vastaaja oli ymmärtänyt kysymykseni väärin tai sitten vastannut kiireessä. Jätin toteamuksen kysymysmäisesti samaan kysymyssivuun ja toivon, että siihen reagoidaan huomiseen mennessä.

ConsumptionFlexibilities ei ole vielä valmis. Viimeisenä tehtävä on lisätä muutama funktio mapperiin ja sitten päivittää ne Bootstrapin ja CFWheelsin osalta. Jokainen tallettava funktio on GET-tyyppinen, joten ne pitää ensinnäkin muuttaa POST-tyyppiseksi, muuten reitittämisessä ei ole mitään järkeä. Tämä on ollut yleinen käytäntö lisätä eri toiminnallisuudet saman funktion sisälle.

Toinen yleinen käytäntö on ollut hyödyntää GET-tyyppisiä kutsuja tapauksissa, joissa on muokattu informaatiota tietokannasta. Yleisesti ottaen GET-tyyppisillä kutsuilla ei saisi tehdä tietokanta muutoksia. GET-tyyppiset kutsut mahdollistavat ennen aikaisen haun, bottien hyväksikäytön, GET-kutsut tallentuvat välimuistiin ja HTTP-standardi väittää niin. Näin Brian R. Brondy listaa StackOverFlow-sivulla (StackOverFlow).

Kehitysjakso jatkuu siis vielä ainakin huomiseen. Artikkeleiden lukeminen siirtyy myös huomiseen, jos saisin silloin järjestettyä itselleni aikaa. Voisin pyrkiä jaksottamaan työtäni paremmin käyttämällä tauot järkevimmin. Lukemalla huomenna työterveyteen liittyviä artikkeleita on hyvä tapa tehdä juuri näin.

Tiistai 30.06.2020

Tämän päivän tavoitteena on saada ConsumptionFlexibilities-osio viimein valmiiksi. Osi-ossa on vielä muutama funktio päivittämättä. Päivän aikana on myös lopetettava ja aloitettava uusi kehitysjakso. Päivän aikana tulisi lukea muutama artikkeli ja ottaa niistä löytyviä taukoaktiiviteettivinkkejä käyttöön.

Nyt ConsumptionFlexibilities-osio on valmis. Aloitin funktioiden päivittämisen melkein alusta. Eilen ehdin päivittelemään ConsumptionFlexibilities -osion funktioita aika pitkälle. Harmikseni tänään jatkaessani en muistanut, mihin jäin. Iski epävarmuus ja totaalinen ajatuskatko. En ollut enää varma kannattaako muokata kyseisiä funktioita vain luettavuuden kannalta. Päädyin aloittamaan melkein alusta. Väitän, että olisi kannattavampaa vain uudelleen kirjoittaa funktiot, mutta siihen ei ole nyt mielestäni aikaa ja uskon, että jaan tiimiläisten kanssa saman ajatuksen. ConsumptionFlexibilities-osioon liittyvien Attachments ja Contacts-osioiden funktiot ovat myös valmiita.

Aloitin myös uuden kehitysjakson, jonka tavoitteeksi asetin ConsumptionPoint-osion. Valitsin osion siksi, että se sisältää Attachments ja Contacts-osioiden liittyviä funktioita sekä muutamia muita. Tällä tavoin saan hiottua viimeisetkin virheet pois Attachments ja Contacts-osiosta, joita päivittäessä on saattanut ilmaantua. Kehitysjaksoon asetin päättämispäiväksi tulevan viikon perjantain. Uskon, että saan siihen mennessä ConsumptionPoint-osion valmiiksi, vaikka osio on isompi kuin kaksi edellistä yhteensä.

Luin muutaman artikkelin koskien työhyvinvointia. Ajanhallinnasta päällimmäiseksi jäi mieleen TTL:n fokuskello. Fokuskellolla voi yksikertaisesti jaksoittaa työtä oletuksena 25 minuutin pituisiksi jaksoiksi. Jaksoja seuraa 5 minuutin tauko. Jaksojen pituuksia voi muokata helposti sen intuitiivisella käyttöliittymällä. Fokuskello soittaa merkkiäänä aina jaksojen välissä ilmoittaen käyttäjälle alkavasta tauosta. (TTL.) Pyrin jatkossa käyttämään fokuskelloa tai sen tyyppistä kelloa työssäni, koska uskon, että se pelkästään lisää tehokkuuttani. Olen kiinnittänyt huomiota jo pidemmän aikaa, että aamuisin työni tehokkuus on huomattavasti korkeampi kuin loppupäivästä. Uskon taukojen olemattomuuden olevan osa syy tähän.

Keskiviikko 01.07.2020

Tänään olisi tavoitteena saada valmiiksi eilen aloitetusta ConsumptionPoint-osion HeatExchangers-osio. Eilen ehdin aloittamaan sen päivittämisen ja muokkaamistoiminnallisuus jäi keskeneräiseksi. Aion aloittaa sen päivittämisellä.

Päivän loputtua en ihan ehtinyt saamaan HeatExchangers-osion valmiiksi. ConsumptionPoint-osioon sisältyi toinen samankaltainen osio HeatTechnicalDatas. Päivitin kumpaakin samanaikaisesti, koska uskoin, että se olisi tehokkainta. Huomenna on todennäköistä, että kummatkin ovat valmiita. Osioiden valmistuminen olisi kehitysjakson asettamani tavoitteen kannalta hyvin tärkeää.

Torstai 02.07.2020

Tavoitteena on saada ConsumptionPoint-osion HeatExchangers ja HeatTechnicalDatas osiot valmiiksi. Niiden jälkeen olisi tarkoitus tehdä Liittyvät tiedot-partial, jota muokkasin aikaisemmin ContractChanges-osiossa, mutta löysin siitä vielä parannettavaa. Partial on CFWheels:ssä osa sivua (CFWheels). Yleisimmät käyttötapaukset niille ovat, kun halutaan sisällyttää jokin elementti tiettyyn kohtaan sivulle. Liittyvät tiedot-partial vaatii lähinnä visuaalisia muutoksia, joten niiden tekemisessä ei luulisi menevän kauaa. Aloitan päivän pitämällä retrospektiivin viime kehitysjakson sprintistä, jotta se tulee viimein tehtyä.

Suoritin retrospektiivin heti ensiksi. Retrospektiivi olisi pitänyt pitää heti kehitysjakson loputtua, sillä oli vaikea vastata asettamiini kysymyksiin. Seuraava kehitysjakson päättyy huomenna, mutta retrospektiivin teen siitä vasta ensi viikon maanantaina, koska en ole kotona tulevana viikonloppuna.

Harmikseni HeatExchangers ja HeatTechnicalDatas -osiot eivät vielääkään valmistuneet. Sain edistettyä niitä paljon, mutta edessä on vielä työtä. Suurin osa koostuu reittivirheistä tai sitten reittien asematta jättämisestä.

Jouduin päivän aikana tekemään useita muutoksia mapperiin, koska en oikein osannut päättää, miten mapperiin asetettavat resurssit tulisi laittaa. Mapper-funktio tukee Resources ja Resource-funktiota, joiden erona on se, että Resources vaatii "key" arvon. Tarkalleen ottaen "Primary Key" arvon. Resources-funktio generoi myös enemmän päätepisteitä kuin resource-funktio. Resource-funktiota käytetään niissä tilanteissa, kun puhutaan yksittäisistä "resursseista", kuten ostoskärri tai profiili. Myös silloin, kun "key" arvoa ei haluta joko näyttää ulkopuolisille tai sitä hallitaan jollain muulla tavalla. CFWheels dokumentaation mukaan resource-funktion toiminnallisuutta voidaan hyödyntää esimerkiksi muokattaessa käyttäjän profiilia. Profile-resurssin päätepiste näyttäisi silloin tältä `/profile/edit`. Jos kyseessä olisi ContractChanges-osio silloin resources-funktio olisi soveliaampi valinta, sillä pelkkä nimi kertoo sen sisältävän useita myyntisopimustapahtumia. (CFWheels.)

Resources ja Resource-funktiot mahdollistavat myös sisennetyt resurssit. HeatExchangers ja HeatTechnicalDatas -osiot kuuluvat silloin ConsumptionPointin sisälle, koska kummassakin on aina ConsumptionPoint. ConsumptionPoint eli käyttöpaikkaan kuuluu HeatExchangereitä ja HeatTechnicalDataa. Nöiden päätepisteet generoituvat sisennettyihin resursseihin asti. Päätepisteet tuottivat ongelmia, koska joissain tapauksissa oli "key" arvo ja joissain ei ollut "key" arvoa. CFWheelsin reititys ei hyväksynyt tätä ja siksi generoi siitä virheitä.

Päivän aikana myös mietimme yhteisesti tiimiläisen kanssa, miten DeleteButtonTag-merkistä voisimme saada vielä selkeämmän ja käytännöllisemmän. Tulimme siihen tulokseen, että painotamme JavaScriptiä HTML-tyylin ylitse. Luodessani kyseistä merkkiä BootstrapDialog ei vielä toiminut, joten loin Bootstrapin modaalin Bootstrap dokumentaation ohjeiden mukaisesti. Huomenna olisi tarkoitus muokata kyseinen merkki siten, että minimoidaan toistuvuutta lisäämällä JavaScriptiä ja vähentämällä merkin generoimaa koodia.

Perjantai 03.07.2020

Päivän ja kehitysjakson tavoitteena on saada ConsumptionPoints-osio valmiiksi. Jatkan ConsumptionPointin loppuun päivittämistä ja sitten refaktoroin DeleteButtonTag-merkin. Huomasimme kollegan kanssa torstaina toimistolla käydessäni, että DeleteButtonTagin ohjelmistokoodia voisi parantaa. Päädyimme siihen lopputulokseen, että siirrän sen toiminnallisuutta mahdollisimman paljon JavaScriptin puolelle. Nappulaa painaessa tehdään AJAX-pyyntö DeleteButtonTagiin määritettyjen parametrien avulla. Yksi parametreista määrää, mihin käyttäjä ohjataan AJAX-pyyntön jälkeen. Päätely tehtiin DeleteButtonTagin sisällä. Nyt se voidaan tehdä kontrollerin funktion tasolla, mihin AJAX-pyyntö osoitetaan.

ConsumptionPoints-osio on nyt siinä pisteessä, että voin siirtyä seuraavan kokonaisuuden pariin. HeatExchangers ja HeatTechnicalDatas-osiossa ei ollut paljoa tekemistä eilisen jäljiltä. Suurin osa oli vain virheellisesti asetettuja tai asettamattomia route-parametrin arvoja. ConsumptionPoint-osion sisällä oli paljon muita partialeja, joita käydessäni läpi korjailin. Korjaukset koostuivat lähinnä visuaalisista ja rakenteellisista muutoksista.

DeleteButtonTag toimii nyt jQueryn BootstrapDialogilla. Sen sijaan, että nappula vain näyttäisi piilotetun DeleteButtonTagissa luodun modaalin, DeleteButtonTagin generoi modaalin vasta nappulaa painaessa. Ensimmäinen DeleteButtonTag versioni toisti piilotettuja modaaleita sivulle niin monta kuin DeleteButtonTagia olisi sivulla. BootstrapDialog-metodilla voidaan vain generoida jQuery-funktio. Funktio kutsuu BootstrapDialog luonti funktiota annetuilla parametreilla, joka generoi kokonaisuudessa modaalin. Tällä tavoin sivun rakenne pysyy siistimäpää ja nopeampana, koska DeleteButtonTag ei luo modaaleita etukäteen ja moneen kertaan. Modaaleita ei välttämättä käytetty ollenkaan, joten siitäkin syystä se oli turhaa.

Viikkoanalyysi

Viikon tavoitteet saavutettiin kaikilta osilta. Otin selvää viimeviikolla tehdyn lupaukseni pohjalta työhyvinvointiin liittyviä asioita. TTL:n mukaan työtä kannattaa jaksoittaa, jotta työhön olisi helpompaa keskittyä (TTL). Tämä on totta, sillä olen itsekin huomannut, kuinka paljon sillä on merkitystä. Tällä viikolla en vain hyödyntänyt TTL:n ohjeita enkä myöskään artikkelissa mainittua fokukselloa. Huomenna otan sen käyttöön ainakin aamunpäivän ajaksi. Katson todennäköisesti vielä jotain muita tekniikoita, millä aivoja saisi stimuloitua taukojen aikana. Taukojumppakaan ei olisi pahitteeksi.

Olen itsestäni ylpeä. Sain ConsumptionPoints-osion valmiiksi ennätysajassa, vaikka muutoksia oli paljon. Viikon aikana keskityin noin yhteen kokonaisuuteen kerralla ja se selkeästi tuotti tulosta. IsPatch()-kysymykseen sain viimein vastaukseni. Vastaus ei ollut kaiken kattava, mutta riittävä. Tiedän nyt, miten edetä vastaavissa tapauksissa. Minulle kehoitettiin käyttämään funktiota juuri siten, miten olen niitä käyttänyt. IsPatch()-funktion kaltaisiin funktioihin ei kuitenkaan tullut selvennystä. Tämä ei ole onneksi ongelma, koska ongelman voi kiertää käyttämällä IsPost()-funktioita, joka periaatteessa käyttäytyy samalla tavalla. Funktion nimi on vain eri. Jätän ensiviikolla uuden kysymyksen tarkentaen ongelmaa, johon etsin vastausta. Ensiviikon haasteena on Customerships-osio, johon liittyy muutamia osiota, joten samankaltainen viikko on edessä.

Luen vapaa-ajalla tällä hetkellä Robert C Martinin kirjoittamaa Clean Code käsikirjaa, joka opettaa hyviä ohjelmistokehityksen periaatteita. Valitsin kyseisen aiheen siksi, että haluan olla parempi työssäni ja haluan tuottaa ”puhdasta” koodia, jota kollegani ymmärtävät. Jatkossa aion panostaa selkeämmän ohjelmistokoodiin tuottamiseen. Olen pitkään miettinyt eri tapoja, miten ohjelmistokoodista voi saada selkeämmän ja kirja ainakin tähän asti on pysynyt samoilla linjoilla. Uskon, että siitä tulee olemaan paljon apua.

3.6 Seurantaviikko 6

Maanantai 06.07.2020

Tämän päivän tavoitteena on pitää kehitysjakson 5 retrospektiivi niin kuin edellisenä perjantaina lupasin. Sitten on aloitettava uuden kokonaisuuden päivittäminen. Valitsin Customerships-osion, koska siinä on muitakin yhdistettyjä osioita. Samaan tapaan kuin ConsumptionPointsissa.

Uusi kehitysjakso alkoi tänään. Asetin viikon mittaiseksi tavoitteeksi saada valmiiksi Customerships-osion. Siihen sisältyy ConsumptionPoints-osion tapaan useita muita osioita. Aloitin Customerships-osiolla päivittämällä perusnäkyvät valmiiksi. Niiden jälkeen siirryin Activities-osion pariin.

Päivän aika ilmaantui muutamia ongelmia liittyen taas takaisin reititykseen. Käyttämäni fromRoute ja fromKey parametrit eivät ole riittäviä. Huomasin ongelman jo aikaisemmin, mutta kiinnitin siihen vasta nyt huomiota. ConsumptionFlexibilities-osiossa uuden ConsumptionFlexibilitien luonnin jälkeen käyttäjä ohjataan väärään paikkaan, koska annettu fromRoute parametri on väärä. Tähän olisi kiva löytää jokin ratkaisu, mutta juuri tänään en keksinyt sellaista. Onneksi se ei ole akuutti asia. Voin jatkaa muiden osioiden työstämistä vapaasti.

Takaisin-painike on nykyään oma merkkinsä. Kaukolämpö Plus järjestelmässä takaisin painiketta käytetään lähes jokaisella sivulla. Takaisin-painike on aikaisemmin tehty linkTo tai buttonTag-fukzioilla. Lisäksi funktioihin on annettu aina samat parametrit, kuten linkin ikoni, teksti ja CSS-luokat. Nyt riittää vain, että kirjoittaa backButtonTag, jossa on oletuksena ikoni, teksti ja CSS-luokat.

Tiistai 07.07.2020

Tavoitteena on saada Activities-osio Customerships-osion osalta valmiiksi. Eilen minulla jäi kesken edit-partialin muokkaaminen. Kyseisessä partialissa on monia uusia asioita, joita pitää ottaa huomioon. Eilen huomasin, että asentamani Air-datetimepicker ei olekaan soveltuva projektiin. Siinä ei ole mahdollista natiivisti avata kellon aika näkymää. Kokeilin muokata sitä hetken ja totesin ettei muokkauksessa ollut mitään järkeä. En halunnut käyttää useita tunteja sen muokkaamiseen. Latasin toisen datetimepicker:n, jonka otan tänään käyttöön, jossa on mahdollisuus valita päivämääränäkymän ja kellonäkymän välillä.

Select2 niminen monivalintakenttä täytyy päivittää uusimpaan versioon. Sitä on käytetty useassa eri Kaukolämpö Plus järjestelmän osiossa.

Päivitin Select2-nimisen laajennuksen sekä Bootstrap 3 typeahead laajennuksen. Kummatkin ovat käytössä useassa paikassa Kaukolämpö Plus järjestelmässä. Uusimmat versiot sai ladattua helposti niiden versionhallinta sivuiltaan. Kyseiset laajennukset ovat forkkavissa GitHubista omalle Fortumin GitHub tilille. Forkkaus eli forking tarkoittaa jaetun repository:n eli hakemiston moninaistamista omalle tilille, johon voi tehdä muutoksia muuttamatta alkuperäistä hakemistoa (GitHub a). Hakemistoihin olisi ideana tehdä muutoksia ja jatkossa käyttää niitä. Select2 laajennus vaati yhden oletusasetuksen yliajamisen. Laajennusta muokataan siksi, ettei käyttäjän tarvitsisi, joka kerta kirjoittaa haluttua teemaa. Teema halutaan määrittää etukäteen juuri tästä syystä.

Customerships-osion Activities-osio ei valmistunut tänään. Täytyy huomenna jatkaa siitä mihin tänään jäätiin. Katsoimme tiimiläisen kanssa vaihtoehtoja tapaa rakentaa sivu navigaation uusiksi. Ajattelimme hyödyntää sessiota sen rakentamiseen. Session eli istunto tarkoittaa käyttäjän ja palvelimen yhteistä aikajaksoa (ComputerHope). Nykyinen navigointi tapahtuu käyttäen "from" alkuisia parametrejä. Mielestäni se on likainen tapa määrittää ja siinä on omat ongelmansa. Tällä hetkellä käyttäjä voi ainoastaan palata takaisin maksimissaan kaksi sivua taaksepäin.

Sessiossa on vain ongelmana se, että sessio on käyttäjäkohtainen. Sama istunto jakautuu eri tabien kesken. Useat käyttäjät avaavat useita eri tabeja ja silloin, kun navigaatio on rakennettu sessioon navigaation jakautuisi tabien kesken. Ainoa tapa olisi ohjeistaa käyttäjää käyttämään vain yhtä tabia, mutta se olisi typerää.

Meillä jäi hyvä ajatus siitä, miten navigaatio tulisi toimia. Emme kuitenkaan keksineet, miten tabit pitäisi erottaa toisistaan. Kokeilin käyttää selaimen SessionStoragea. SessionStorage on yleisimpien selainten, kuten Google Chromen, Firefoxin ja uusien Microsoft Edgen yksi ominaisuus. Sinne voi tallentaa dataa, joka toimii vain siinä tabissa, jossa tieto on tallennettu. (Mozilla c.) Kysymykseksi jäi vain se, miten tieto saadaan ColdFusionin kontrollerin tasolle.

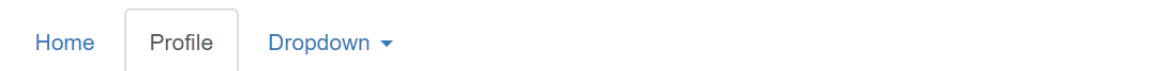
Keskiviikko 08.07.2020

Tänään jatkan Activities-osion parissa. Yritän keskittyä nyt Activities-osion päivittämiseen Customerships-osion osalta ja unohtaa kaiken muun ainakin hetkeksi, kunnes Activities-osio on valmis. Activities-osion jälkeen siirryn päivittämään Customerships-osioon liittyviä

kyselyitä, käyttöpaikkoja, asiakkuuksia jne. Jossain vaiheessa päivää katsomme todennäköisesti tiimiläisen kesken lisää navigaationjärjestelmän päivitysmahdollisuuksia.

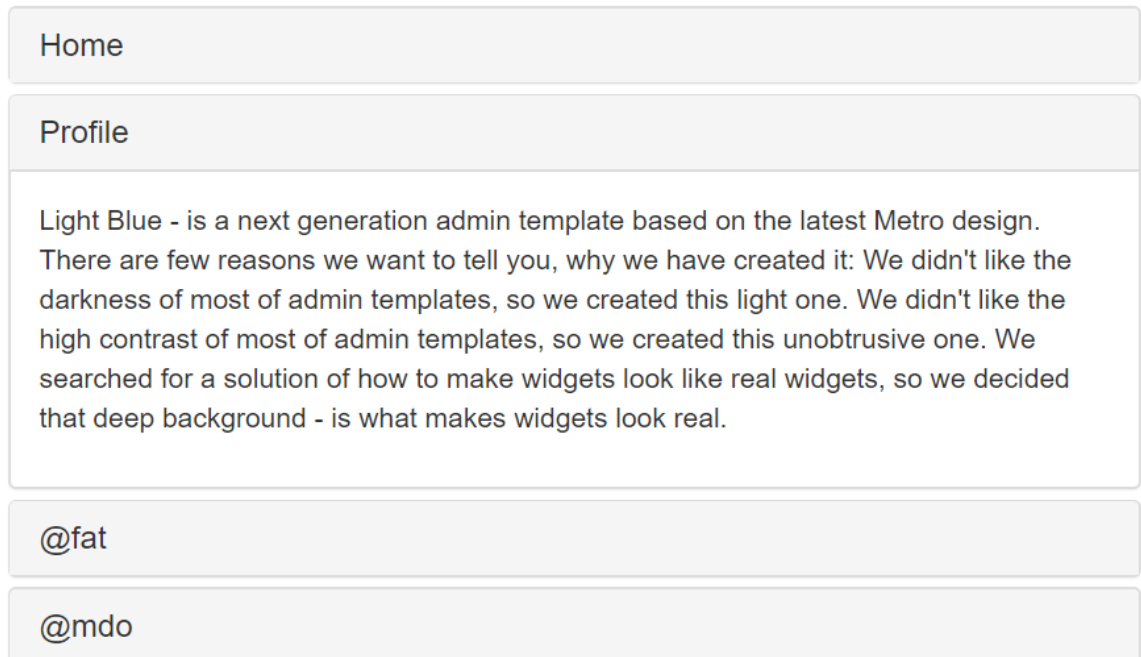
Työpäivä kului erittäin nopeasti loppuun. Sain melkein koko Customerships-osion valmiiksi. Työstin Activities-osio Customerships-osion osalta valmiiksi. Parantelin myös useita aikaisempia osiota, kun huomasin, että olin asettanut virheellisesti Bootstrapin luokkia Search-partialeihin.

Search-partialeissa olisi vielä yksi päivitettävä jQuery laajennus. Aikaisemmin Kauko-
lämpö Plus järjestelmässä oli käytössä TabCollapse niminen laajennus. Kuvissa 12 ja 13 Laajennus muuttaa halutun tab valikon paneeleiksi, kun käyttäjän selaimen viewport pienenty riittävästi (Kuva 12; Kuva 13). Viewportilla tarkoitetaan käyttäjän näkyvässä olevaa verkkosivun osaa (Technopedia).



Light Blue - is a next generation admin template based on the latest Metro design. There are few reasons we want to tell you, why we have created it: We didn't like the darkness of most of admin templates, so we created this light one. We didn't like the high contrast of most of admin templates, so we created this unobtrusive one. We searched for a solution of how to make widgets look like real widgets, so we decided that deep background - is what makes widgets look real.

Kuva 12. Bootstrap 3 version tabit



Kuva 13. Bootstrap 3 version tabit mobiilinäkymässä

Nykyinen TabCollapse laajennus ei valitettavasti toimi sellaisenaan Bootstrap 4.5 versiossa, koska paneelikomponentteja ei ole enää olemassa. Paneelit näkyvät kuvassa 13. Laajennuksen ohjelmistokoodia pitäisi muokkaa siten, että paneeli vaihdettaisiin card-komponenttiin. Linkkien uudelleen generointi pitäisi kirjoittaa uudelleen käyttämään uusia Bootstrap 4.5 version rakenteita ja luokkia. Tab ja Card-komponentit pitäisi asettaa käyttämään uusia visibility-luokkia. Visibility-luokilla pystyi piilottamaan ja näyttämään verkkosivulla HTML-elementtejä. (Bootstrap.)

Torstai 09.07.2020

Tänään on tavoitteena saada Customerships-osio valmiiksi. Samalla päivittäessä on hyvä siivoilla ja muokkaa funktiota. Katsomme tänään myös navigaatiojärjestelmää uudelleen.

Customerships-osio on nyt kokonaisuudessaan valmis mukaan lukien kaikki siihen liittyvät osiot. Osiossa ei loppujen lopuksi ollut paljoa muokattavaa. Lähinnä vain rakenteellisia muutoksia sekä ikonien ja luokkien päivittämistä. Jokaisessa näkymätiedostossa on päivitettävä ainakin muutama ikoni käyttämään uusia luokkia, joita uusi FontAwesome-ikonikirjasto vaatii.

Kehitimme tiimiläisen kanssa takaisin painikkeen navigaatiojärjestelmän perustumaan SessionStorageen. SessionStorageen tallennetaan linkit käydyillä sivuilla. Myös tieto siitä, onko käyttäjä siirtymässä takaisin edelliselle sivulle. Linkit sijaitsevat SessionStorageessa listassa, jota käsitellään lisäämällä tai poistamalla linkkejä. Käyttäjän navigoidessa sivulle tallennetaan nykyinen sivu SessionStorage listaan. Käyttäjän painaessa takaisin-painiketta listasta poimitaan toiseksi viimeinen linkki, jonka jälkeen linkki poistetaan listasta, ettei seuraavalla sivulla takaisin-painikkeessa ole edellisen sivun linkkiä. Näin estämme ikuisen silmukan. Ikuinen silmukka tarkoittaa ohjelmaa, joka jatkaa suoristustaan jatkuvasti pysähtymättä pysäyttämättä. (Jyväskylän yliopisto.)

Navigaatiojärjestelmä on tosin vielä keskeneräinen. Käyttäjän navigoidessa joillain muulla menetelmällä sivujen välillä saattaa sekoittaa SessionStorageen tallennetun linkkilistan. Uskon, että ongelma selviää huomenna.

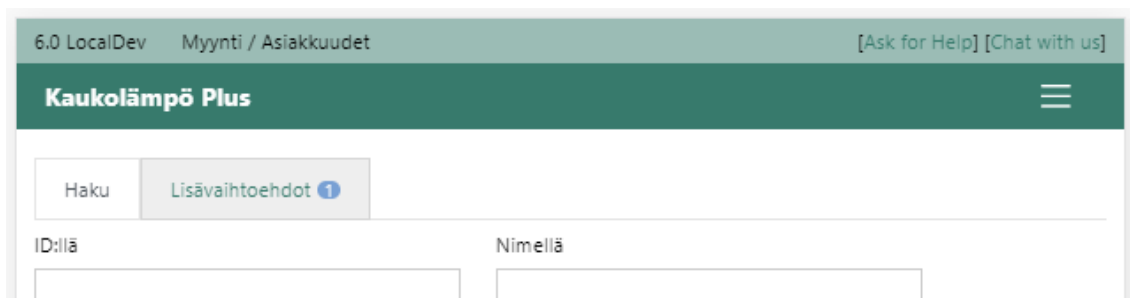
Perjantai 10.07.2020

Tänään on vuorossa kehittää TabCollapsen tyyppinen helppokäyttöinen jQuery laajennus. Lisäksi haluan saada Takaisin-painikkeen navigaatiojärjestelmän toimimaan oikein. Siinä

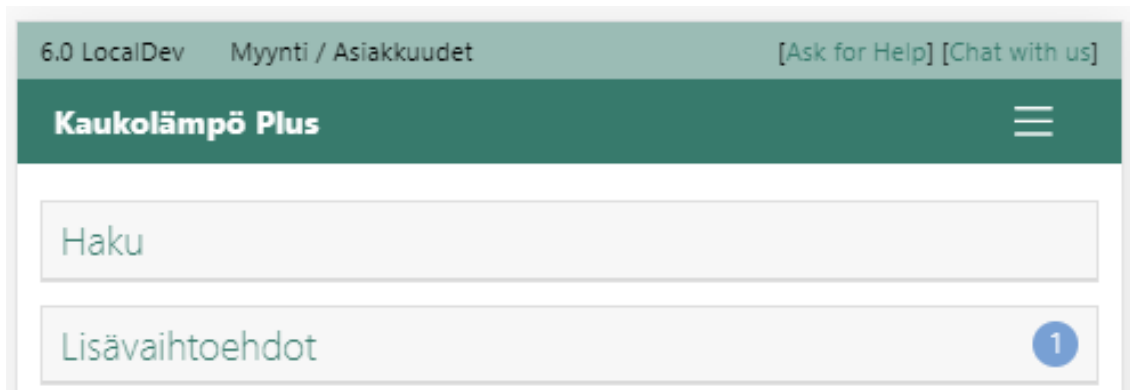
jäi vielä muutama asia selvittämättä, kuten tilanne, jossa käyttäjä ohjataan samalle sivulle, poistaa tietokannasta jotain tai käyttää jotain muuta keinoa siirtyä sivulta toisalle.

Ratkaisimme yhdessä tiimiläisen kanssa tilanteen, jossa käyttäjä täytyi ohjata päivityksen jälkeen takaisin samalle sivulle. URL-osoitteen tallentavaan funktioon täytyi lisätä ehto, joka vertaa edellistä URL-osoitetta nykyiseen URL-osoitteeseen. Tässä tapauksessa haluamme estää tallennuksen silloin, kun edellinen URL-osoite on sama kuin nykyinen. Jos listaan tallentuisi peräkkäistesti kaksi samanlaista URL-osoitetta Takaisin-painike ohjaisi käyttäjän aina takaisin samalle sivulle, jolloin käyttäjä joutuisi ikuisen silmukkaan. Toinen samankaltainen tilanne estettiin, kun käyttäjä poistaa tietoja tietokannasta tekemälläni DeleteButtonTagilla.

Päivitin TabCollapsen Bootstrap laajennuksen nyt tukemaan viimeisintä 4.5 versiota. Olen aiemmin maininnut muutaman otteeseen, että viimeisin Bootstrapin versio ei tue panel-komponentteja. TabCollapse laajennus on tarkoitettu Bootstrap versiolle, jotka ovat alle 4. Laajennuksesta täytyi siis muuttaa kaikki panel-komponenttia käyttävät kohdat card-komponenttia käyttäviksi. Tabit muuttuvat nyt card-komponenteiksi, kun käyttäjän viewportin leveys pienenee riittävästi (Kuva 14; Kuva 15).



Kuva 14. Bootstrap 4.5 version tabit



Kuva 15. Bootstrap 4.5 version tabit mobiilinäkyvässä

Uusimmassa Bootstrap versiossa Panel-komponenttien lisäksi ei visibility-luokkia ei ole enää olemassa. Luokkien nimet ovat vaihdettu sekä niitä on tullut lisää. TabCollapse laajennuksessa jäi vielä yksi kohta muokkaamatta, joka avaa ensimmäisen Card-komponentin sisältökohdan. Ongelma liittyy eventtien nimiin. Eventeillä hallitaan TabCollapsella rakennettuja elementtejä.

Viikkoanalyysi

Viikon aikana keskityin Seurantaviikolla 5 selvittämiini asioihin. Käytin kolmena päivän aktiivisesti ja vähemmän aktiivisesti fokukselloa. Huomasin sen auttavan taukojeni pitämisessä. Yleensä unohdan pitää taukoja ja työskentelen työpäivän alusta ruokataukoon tauotta. Välissä olisi hyvä pitää ainakin yksi tauko, jotta aivot saisivat vähän rauhoittua ja virkistäytyä. Huomasin niinä päivinä jaksavani enemmän ja pidempään, kun pidin selkeitä 15 minuutin taukoja tasaisin aikaväleihin. Kaikki tauot tai työpäivät eivät olleet tasapituisia, mutta suurin osa oli. Suurimman eron huomasin siinä, etten ollut täysin väsynyt työpäivän jälkeen. Aion jatkossakin kiinnittää huomiota tähän asiaan, koska sillä on selkeästi hyötyjä. Olen tiennyt ja huomannut aiemminkin saman asian, mutta en ole osannut kiinnittää siihen huomiota kesken mielenkiintoisen työpäivän.

Kehitin navigaatiojärjestelmän Robert C. Martinin ohjeiden mukaisesti. Hän kirjassaan luettelee erilaisia hyviä periaatteita ohjelmistokehitykseen. Ohjelmistokehityksessä on suotavaa ottaa huomioon käsiala. Kirjassaan hän mainitsee, että muuttujat, funktiot, argumentit, luokat, tiedostot, hakemisto ja paketit tulisi nimetä siten, että niistä käy ilmi, mikä on niiden tarkoitus (Martin 2019, 17-30). Nimeämistä ei kannata sivuuttaa turhapäiväsenä ja siihen olisi jokaisen suotavaa käyttävän riittävästi aikaa. Ohjelmoidessa ohjelmoijan lukevat jatkuvasti koodia. Ohjelmoijat käyttävät enemmän aikaa lukemiseen kuin itse ohjelmoimiseen. Hyvät nimet säästävät huomattavasti aikaa, koska ne kertovat lukijalle, miksi muuttujat, funktiot ja argumentit on olemassa.

```

function getLastHistoryUrlPath() {
  const urlPathHistory = getUrlPathHistoryArray();
  return urlPathHistory.length ? urlPathHistory[urlPathHistory.length - 1] : false;
}

function getPreviousHistoryUrlPath() {
  const urlPathHistory = getUrlPathHistoryArray();
  return urlPathHistory.length ? urlPathHistory[urlPathHistory.length - 2] : false;
}

function getUrlPathHistoryArray() {
  const urlPathHistory = window.sessionStorage.getItem(urlPathHistoryKey);
  if(urlPathHistory !== null){
    const parsedUrlPathHistory = JSON.parse(urlPathHistory);
    if(parsedUrlPathHistory.constructor === Array){
      return parsedUrlPathHistory;
    }
  }
  return [];
}

```

Kuva 16. Uuden navigaatiojärjestelmän funktiota

Robertin mukaan funktioiden tulisi olla ennen kaikkea lyhyitä (Martin 2019, 34). Mielestäni pienet funktiot ovat parempia siksi, että niitä on helpompi käsitellä. Isot funktiot myös useasti tuottavat herkemmin bugeja.

Loin funktiot käyttäen ”Do On Thing”, jonka Robert C Martin mainitsee kirjassaan sivulla 35. ”Functions should do one thing. They should do it well. They should do it only.” Jos huomaan funktion tekevän useampaa kuin yhden asian kannatan sen refraktoroimista. Refaktoroiminen täytyy kuitenkin tapahtua kohtuuden rajoissa. Jokaista ehtolauseketta ei välttämättä kannata kirjoittaa omaksi funktioksi. Muutos saattaisi vaikuttaa negatiivisesti ohjelmistokoodin lukemisen helppouteen ja lisätä abstraktion tasoa. (Martin 2019, 35-36.) Yleisesti ottaen ehtolauseen sisältö vaikuttaa suuresti siihen kannattaako se kirjoittaa omaksi funktioksi. Kuvassa 16 on kolme funktiota, jotka jokainen tekee vain yhden asian. GetLastHistoryUrlPath noutaa uusimman tai viimeisimmän URL-osoitteen, getPreviousHistoryUrlPath noutaa edellisen tai toiseksi viimeisen ja getUrlPathHistoryArray palauttaa kaikki URL-osoitteet listana. GetUrlPathHistoryArray-funktiota käytetään kahdessa muussa funktiossa, kuten kuvasta 16 näkee.

Olen myös siivonnut kontrollereita ja näkymätiedostoja aina niitä päivittäessä. Olen kiinnittänyt huomiota turhiin kommentteihin ja huonosti nimettyihin muuttujiin ja funktioihin, ylikirjoitettuun ohjelmistokoodiin ja epäjohdonmukaisuuksiin. Robert mainitsee kirjassaan, että usein ohjelmoijat paikkaavat ohjelmointitaitojaan kommentoimalla (Martin 2019, 55). Toiminnallisuus, joka mallinetaan funktioksi on kaukana kauneudesta ensimmäisellä kirjoituskerralla. Funktio kirjoitetaan nopeasti ja todetaan valmiiksi, koska se tekee, mitä siltä

odotetaan. Funktiolle annetaan nimi, lyhyt kuvaus ja tiedosto tallennetaan ja suljetaan. Funktioiden nimet saattavat olla epätarkkoja, muuttuvat vähemmän kuvaavia ja ehkä helposti harhaanjohtavia. Siinä on ainekset monelle ohjelmistovirheille, joita on vaikea jatkoissa löytää. Siksi on erittäin tärkeää käyttää aikaa nimeämiseen ja kuvaamiseen. Jos funktion kokonaisuus on selkeä ja muuttujat itsestään selviä niin kommentoimiselle ei ole tarvetta.

Pyrin kirjoittamaan mahdollisimman selkeää ohjelmistokoodia aina kun siihen on mahdollisuus. Toki joskus en saavuta korkeita odotuksiani, jolloin ohjelmistokoodini jää keskeneräiseksi. Puhuessani keskeneräisestä ohjelmistokoodista tarkoitan juuri näitä samankaltaisia asioita, mistä Robert kirjassaan puhuu. Ylikirjoitettu ohjelmistokoodi on myös turhaa nykypäivänä. Olemme eläneet jo pitkään versionhallinnan aikaa, joka mahdollistaa sen, että poistettu ohjelmistokoodi voidaan palauttaa historiasta. Siksi olen julkeasti poistanut tällaisia esiintymiä Kaukolämpö Plus järjestelmästä.

3.7 Seurantaviikko 7

Maanantai 13.07.2020

Tänään alkaa ja loppuu kehitysjakso. Kehitysjakson tavoitteena oli päivittää Customers-osio ja sen sisällä olevia osiota. Seuraavassa kehitysjaksossa on otettava kunnianhimoisempi tavoite, ettei projektin kanssa tule kiire. Toistaiseksi uskon olevani hyvin aikataulussa. TabCollapsen eventit täytyy vielä päivittää vastaamaan uusinta versiota. Niitä muuttamalla varmistetaan, että ensimmäinen card-komponentti aukeasi viewportin pienentyessä riittävän pieneksi.

TabCollapsen eventit on nyt uusittu. Vastaus löytyi juuri sieltä, mistä sen odotinkin löytyvän. Bootstrapin Collapse ja Navbar sivuilta Bootstrapin dokumentaatiosta. Pelkkä eventien muuttaminen ei riittänyt vaan koodia piti muokata muuttamalla tabien otsikoiden valitsijoita. Bootstrap 4.5 navigaatiossa rakenne on hieman erilainen ja siksi TabCollapse ei osannut noutaa oikeaa HTML-elementtiä.

Aloitin uuden kehitysjakson tänään. Käytimme sen suunnitteluun noin 15 minuuttia, joka taisi olla nopeusennätys tähän mennessä. Haluan jatkaa isojen kokonaisuuksien päivittämistä vielä, kun niitä on jäljellä. Isoissa kokonaisuuksissa on kivaa se, että niissä on monia muita kokonaisuuksia sisällä. Ajatukset pysyvät hyvin kasassa, kun voi keskittyä yhteen kokonaisuuteen kerralla hyppimättä täysin eri kokonaisuuksien välillä. Hyppimättä kokonaisuuksien välillä auttaa myös reittien kirjoittamisessa, kun on helppoa katsoa, mikä resurssi kuuluu minkäkin sisälle.

Tällä viikolla tarkoituksena on saada Kaukolämpö Plus järjestelmän hallintatyökalut valmiiksi. Niihin sisältyy käyttäjien ja ryhmien hallinta, niiden oikeuksien hallinta, liitteiden määrä, nykyhetken kirjautuneet käyttäjät ja osioiden lukitukset. Annoin itselleni aikaa noin viikon, mutta en usko näissä kuluvan niin kauaa.

Tiistai 14.07.2020

Tänään tavoitteena on saada Users-osio valmiiksi ja aloittaa Groups-osion päivittäminen. Ajattelin käyttää koko hallintapaneeli puolella uutta navigaatiojärjestelmää. Hallintapaneeli ei ole yhteyksissä mihinkään muuhun osioon, joten siellä on hyvä testata, miten navigaatiojärjestelmän toimii.

Sain Users-osion valmisteltua ja aloitin Groups-osion päivittämisen. Groups-osio oli hyvin samankaltainen kuin Users-osio ja siksi sen päivittäminen oli virtaviivaista ja helppoa. Locks-osio ehti valmistua myös, sillä se sisälsi vain muutaman toiminnon.

Keskiviikko 15.07.2020

Tavoitteeksi asetan tänään saada koko hallintapaneeli valmiiksi. Tällä hetkellä työn alla on Tasks-osio. Tasks-osiossa on näkymä, jossa pystyy hallinnoimaan Kaukolämpö Plus järjestelmän sisäisiä tehtäviä. Tehtävät on aina kohdistettu käyttäjälle tai ryhmälle.

En saanut tehtäviä valmiiksi syystä, että pidin tänään lyhyemmän päivän. Taistelinkin kauan reittien kanssa ja yritin keksiä tapaa kirjoittaa URLFor-funktioiden linkkejä mahdollisimman dynaamisesti. Tasks-osion etusivulla on käytössä BootstrapMenu-laajennus, joka vaatii myös päivitystä. BootstrapMenulla voidaan luoda tiputusvalikko linkkeineen helposti JavaScriptillä. Ongelmana, että haluaisin luoda URLFor-funktiolla kyseiset linkit ja sitten täydentää niitä JavaScriptillä niin kuin ennen on tehty. Aiemmin mainittu linkkien rakenteen muutos vaikuttaa kyseiseen koodiin siten, että key-arvo täytyy upottaa linkin keskelle. CFWheels 1.4.5 tyyli oli "[controller]/[action]/[key]" ja uusimmassa versiossa "[controller]/[key]/[action]". Tästä syystä key-arvoa ei voi vain lisätä linkin loppuun kauttaviivan jälkeen. Ratkaisuna oli asettaa reititys siten, ettei se vaadi key-arvoa suoraan vaan tieto siirretään params-parametrin avulla kohdistettavaan funktioon. Sama tilanne toistuu kaikissa funktiossa, joissa osoite halutaan generoida melkein kokonaan JavaScriptillä.

Torstai 16.07.2020

Tänään on saatava hallintapaneeli viimein valmiiksi. Haluan kiivaasti siirtyä toisen osion päivittämiseen. Olen pysynyt tasaisessa tahdissa ehkä vähän kiihdyttäen kohti projektin loppua. Ajattelin eilisen tiimiläisen kanssa pidetyn keskustelun pohjalta ottaa seuraavaksi Inspections-osion työn alle.

Hallintapaneeli on nyt valmis muutamaa funktiota lukuun ottamatta. Refaktoroin yhden ison CodeDescriptions-osion sivun. Sivulla näytettiin Kaukolämpö Plus järjestelmän osiot ja niiden selitteet Panel- ja listakomponenteilla. Panel-komponentit täytyi vaihtaa Card-komponenteiksi ja samat näytät enemmän- ja piilota toiminnallisuudet tuli palauttaa takaisin. Komponenttien vaihtamisen jälkeen erottelin Card-komponenttien luomisen omaksi apufunktioksi, johon voi asettaa tarvittavat tiedot parametreina. Näkymätiedosto on mielestäni nyt siistimmässä kunnossa.

Perjantai 17.07.2020

Tavoitteena on saada muutamat eilen kesken jääneet funktiot päivitettyä. Keskiviikkona huomaamani BootstrapMenu-laajennus liittyy Tasks-osioon ja moneen muuhun osioon. Bootstrap laajennus on hyvä tässä vaiheessa päivittää alta pois. Laajennuksessa on samankaltaisia ongelmia, jossa uusimmassa Bootstrap versiossa on poistettu jotain vanhempia komponentteja. Nyt pudotusvalikko ei ole enää vanhaan tapaan ul eli unordered list vaan pelkkä div-elementti, jossa on sama CSS-luokka. Uudessa versiossa kyseiset CSS-luokat kohdistuvat vain kyseessä olevassa tapauksessa vain kyseessä oleviin div-elementteihin. Unordered list on lista, joka ei ole numeroitu eli ei ole "järjestyksessä". Unordered list-elementin vastakohta on eli ordered list. Lisäksi valinnat pudotusvalikossa täytyy vaihtaa a-elementeiksi. A-elementit ovat linkkielementtejä.

Hallintapaneeli on nyt kokonaan valmis. Sen viimeistely vei kuitenkin suurimman osan päivästä. Bootstrapin ohjeiden mukaisesti jokaiseen näkymätiedostoon oli lisättävä row- ja col-luokilla varustetut div-elementit. Päivitin BootstrapMenun tukemaan Bootstrapin 4.5 versiota. Sen päivittäminen oli yksinkertaista, sillä se vaati vain muutamien elementtien vaihtamista toisiin.

Huomasin myös päivän aikana uudessa navigaatiojärjestelmässä mahdollisen ongelman. Ongelma liittyy tilanteeseen, jossa käyttäjä ajautuu virheeseen ladatessaan AJAX-kutsussa kohdefunktion näkymätiedoston väärin. Navigaatiojärjestelmässä oletetaan tällä hetkellä, että sitä halutaan käyttää joka paikassa. AJAX-kutsussa virheellinen palvelimen vastaus saattaa ladata saman sivun uudelleen, jolloin sivun sisällä on kaksi samanlaista sivua. Navigaatiojärjestelmän funktio on jQueryn "on document ready" funktion sisällä, joka tarkoittaa käytännössä sitä, että siihen liitetyt funktiot käynnistyvät heti "DOM":n käynnistyttyä. DOM eli Document Object Model on ohjelmointikäyttöliittymä HTML- ja XML-dokumenteille (Mozilla b). JavaScriptillä voidaan dynaamisesti muokata DOM:ia. AJAX-virhe ei sinänsä ole ongelma, koska kyseinen virhe on mahdollista tapahtua ainoastaan silloin, kun ohjelmoija on itse tehnyt virheellisen ohjauksen controllerissa. Kuitenkin olisi hyvä varautua tilanteeseen, jossa kyseinen skenaario voisi tapahtua. Siksi uskon, että navigaatiojärjestelmä pitäisi olla mahdollista deaktivoida sivukohtaisesti ja ottaa täten käyttöön vain siellä, missä sitä halutaan käyttää. Tämä tulee todennäköisesti selviämään ensi viikon aikana.

Viikkoanalyysi

Viikon aikana mietimme tiimiläisen kanssa, miten navigaatiojärjestelmää voisi parantaa. Käytössä oleva navigaatiojärjestelmä on sellainen, jossa luetaan URL-parametrejä. URL-parametrit siis kertovat sovellukselle, mistä käyttäjä on tullut ja sen avulla mahdollistaa käyttäjän paluun takaisin lähtöpaikalleen. Jos käyttäjä navigoituu sivulle ja sitten vielä seuraavalle sivulle käyttäjä voidaan palauttaa normaaliin tapaan lähtöpaikalle toissijaisella URL-parametrillä. Mitä tapahtuu silloin, kun käyttäjä haluaa mennä vielä yhden askeleen eteenpäin? Käyttäjä voi toki mennä askeleen eteenpäin, mutta sieltä ei ole paluuta. Nykyinen navigaatiojärjestelmä siis kantautuu vain kahden askeleen päähän. Tämä on ollut rajoite suunnittelussa ja toteutuksessa pitkään. Uusi järjestelmä ei käytä URL-parametrejä vaan se tallentaa linkin kokonaisuudessa selaimen välilehtikohtaiseen sessionStorageon. Takaisin-painike noutaa edellisen sivun linkin aina sivun latautuessa.

Ne eroavat siten, että aktiivisessa käytössä oleva vaihtoehto vaatii ohjelmoijan kirjoittavan jokaiseen painikkeeseen tai funktioon parametri joko käsin tai hyödyntäen funktiota. Uudella tapaa käyttäen ohjelmoijan tarvitsisi kutsua pelkkää backBtn-funktiota haluamassa kohtaan, johon Takaisin-painikkeen haluaisi generoida. Uudet funktiot ja painikkeet hoitavat lopun, eikä Takaisin-painikkeiden parametrejä tarvitse jatkossa enää kirjoittaa. Uusi navigaatiojärjestelmä on tosin vielä testauksen alla ja sitä joutuu todennäköisesti vielä muokkaamaan, jotta se tyydyttää jokaisen tilanteen tarpeen.

Kirjoitin navigaatiojärjestelmän Robert C. Martinin kirjan ohjeiden mukaisesti. Mielestäni hän on perustellut kantansa hyvin selkeään ohjelmistokoodiin. Kirjassaan hän mainitsi, että esimerkiksi muuttujien ja funktioiden nimet tulisi olla selkeitä. Funktiot tulisi olla lyhyitä ja tehdä vain yhden asian. (Martin 2019, 17-30, 35.) Olen pitkään ajatellut useista hänen esiiin tulleista asioista samalla tavalla ja nyt menneellä viikolla ajattelin näitä asioita entistä enemmän. Kuvassa 16 näkyy funktiota, joissa on otettu huomioon luettelemia asioita.

Olen myös kiinnittänyt huomiota omiin ohjelmointitapoihin. Olen jo kauan aikaa sitten huomannut, että kirjoitan asioita monesti suoraan muistista. Muistiaukkojen takia yksinkertaiset vertailut saattavat tuottaa virheitä. Yleisimmät virheet tulevat juuri yksinkertaisimmista vertailulauseista. Tähän haluaisin kiinnittää jatkossa huomiota. Lisäksi kirjoita hyvin lyhyesti, jonka jälkeen yleensä testaan, toimiiko ohjelma. Tämä tapa hidastaa erittäin paljon, varsinkin kun ColdFusionin ja CFWheelsin päällä pyörivä Kaukolämpö Plus järjestelmä ei ole maailman nopein uudelleen ladattava ympäristö. Muutoksien välissä täytyy aina odottaa useampi sekunti ennen kuin muutokset voi nähdä ruudulla. Minun pitäisi keskittyä kirjoittamaan funktio yhdellä kerralla ja sitten katsoa toimiiko, eikä niin, että testaan jokaisen rivin jälkeen, tuleeko virhettä.

Ensi viikolla pyrin parantamaan tapojani. Kirjoitan funktiot yhdellä kertaa ja sitten katson, mikä meni vikaan, jos ollenkaan. Uskon, että verkosta tai Robert C. Martinin kirjasta löytyisi helpotuksia ongelmiini menettelytapojen muodossa. Haluisin olla tehokkaampi ja virheettömämpi ohjelmoimissani.

3.8 Seurantaviikko 8

Maanantai 20.07.2020

Tänään on tarkoitus jatkaa perjantaina aloittamaani Inspections-osion päivittämistä ja samalla aloittaa InspectionEvents-osio, koska se liittyy Inspections-osioon. Ajattelin myös lukea muutaman artikkelin liittyen ohjelmointiin. Clean Code-kirjasta on ollut tähän mennessä paljon apua ja se onkin avartanut katsettani entistä enemmän, miten ohjelmistokoodia kannattaa tuottaa ja mitä asioita tuottamisessa kannattaa ottaa huomioon. Kaikkia näitä käytäntöjä en tosin pääse hyödyntämään työssäni, koska ne eroaisivat muiden käytännöistä erittäin paljon. Hyvä esimerkki on sellainen, jossa yksi funktio tekee useamman kuin yhden asian. Voisin väittää, että nykyisin tekisin sen mielelläni vaikka kolmeen eriin funktioon, jos se olisi tarpeellista ihan vain sen takia, että luettavuus paranisi. Muutos olisi siinä mielessä ongelmallinen, että minun olisi suotavaa monistaa tapaa esimerkiksi kontrolleri tasolla muissa funktiossa. Tämä veisi erittäin paljon aikaa pois nykyisestä projektistani. Nykyisen projektin tarkoitus on lähinnä päivittää, jokaisen ohjelman versio eikä kirjoittaa jokaista funktiota uudelleen.

Inspections ja InspectionEvents-osiot ovat hyvällä mallilla. Huomenna saan ne todennäköisesti valmiiksi. Päivällä ilmaantui ongelma ohjelmistokoodissa, jota en saanut ratkaistua. Mainitsin siitä tiimiläiselle ja hän tuumasi, että tehdään sama asia toisella tavalla. Sitten muistin, että kyseisen ongelman voi myös kiertää poistamalla yritystiedot tarkastukselta. Olimme joskus muutamia viikkoja sitten sopineet, että yritystiedot voidaan poistaa kaikkialta, koska niitä ei suoraan tarvita missään. Tänään puhuttiin, että yritystiedoista siirretään yrityksen nimi suoraan asiakastietoihin, jos asiakkaalta löytyy yrityksen tunniste.

Tiistai 21.07.2020

Tänään on tavoitteena saada Inspections ja InspectionEvents-osiot valmiiksi, mutta on hyvin todennäköisestä, ettei ne valmistu. Jos aikaa jää niin lisätavoitteena on suunnitella uusi kehitysjakso. Toistaiseksi en ole aloittanut viime perjantain jälkeen uutta kehitysjaksoa ja syytä siihen en tiedä.

Inspection-osio ja InspectionEvents on muutamaa funktiota ja testausta vaille valmis. Korjailin useita epähuomioissani unohtuneita reittejä edellisistä funktiosta ja päivitin tekemääni MakeToRouteUrl-funktiota tukemaan muitakin parametrejä kuin pelkkiä from ja to-alkuisia. Nested Resources eli sisällytetty resurssi, kuten HeatExchangers on ConsumptionPointille ja InspectionEvents Inspectionsille. Sisällytetyt resurssit vaativat aina ylemmän tason key-

arvon. CFWheels nimeää "nestatuissa resursseissa" ylemmän tason key-arvon automaattisesti siten, että osion nimi lisätään yksikkömuodossa key-termin eteen. InspectionKey esiintyy siis kaikissa InspectionEvents reiteissä ja samaan tapaan ConsumptionPointKey jokaisessa HeatExchangers reiteissä. Edellinen MakeToRouteUrl-funktio mahdollisti vain yhden "key"-arvon eli se ei kääntänyt annettuja arvoja generoitavaan linkkiin. Tämä johti reittivirheisiin, jotka on nyt onneksi korjattu.

Keskiviikko 22.07.2020

Tänään jatketaan eilen kesken jääneiden osioiden kanssa. Haluan myös käyttää aikaa tänään koko järjestelmän näkymätiedostojen massamuokkaamiseen. Massamuokkaamisessa tarkoitan, että yhdellä komennolla saisin muokattua useamman rivin kerralla. MakeToRouteUrl ja makeFromArguments-funktioita täytyy myös muuttaa siten, että ne toimisivat vähän dynaamisemmin tilanteissa, joissa käsitellään sisällytettyjä resursseja. DeleteButtonTag-merkki tulisi myös refaktoroida, koska sen sisällä on paljon siihen kuulumatonta logiikkaa, kuten funktio, joka päättelee fromRoute-arvon perusteella, mitä argumentteja kyseinen reitti tarvitsee.

Päivän lopussa olin saanut paljon hyvää aikaiseksi. Sivutuotteena syntyi myös jonkun verran pahaa. Siirsin AddContact-funktion kontrollerin ylemmälle tasolle. AddContact-funktiota käytettiin muistaakseni seitsemässä eri osiossa. Jokaisessa osiossa AddContact-funktio oli hyvin samanlainen. Funktioissa oli erona vain se, että mitä tekstejä funktiot tuostivat tai mihin ne ohjasivat onnistumisissa tai epäonnistumisissa. Nyt järjestelmässä on yksi ja sama funktio, jota kutsutaan kontrolleritasolla tietyillä parametreillä. Parametrejä ovat mitä tekstejä käyttäjälle näytetään. ChangeContact-funktio on ihan samanlainen. Mielestäni tämänkaltaiset toistuvat funktiot kannattaa aina kirjoittaa omaksi funktioksi, jota sitten kutsutaan tarpeen tullen. Tämä olisi Robert C. Martinin mukaan selkeää koodia.

Yritin osan päivästä keksiä tapoja saada linkkien generoivia funktiota toimimaan oikein. Olen erittäin lähellä sitä pistettä, jossa funktiota voi jaella ympäri järjestelmää. Olisi hienoa, että järjestelmästä löytyisi funktio, joka generoisi paluulinkin fromRoute tai toRoute-parametrien mukaan. Tässä funktiossa pääasiana on CFWheelsin sisäinen funktio \$findRoute. Sille annetaan route-arvo, joka noutaa kaikki siihen reittiin liittyvät tiedot, kuten mitä key-arvoja reitti tarvitsee. Tämä on erittäin tehokasta, koska voin kirjoittaa tarkistusfunktio tai generointifunktion, joka hyödyntää tätä tietoa generoidessaan linkkiä. Lopuksi ohjelmoijan täytyy vain ymmärtää antaa funktiolle reitin vaadittavat key-arvot. Tällä hetkellä key ja fromKey-arvot tuottavat ongelmia niiden samankaltaisuuden takia. En saanut ongelmaa ratkaistua, mutta uskon, että huomenna siihen löytyy vastaus.

Torstai 23.07.2020

Tänään olisi tarkoitus saada Inspections ja InspectionEvents-osiot valmiiksi. Suurin syy miksei niitä eilen saatu valmiiksi oli se, että reittien generoiminen ja määrittäminen ei onnistunut aiemmin luoduilla funktiolla. Lisäksi muutamat funktiot ohjasivat käyttäjät väärään paikkaan päivittämisen tai poistamisen jälkeen. Nykyisessä Kaukolämpö Plus versiossa tapahtuu näin ja ajattelin, että parempi korjata ohjautumaan oikein. Siksi reitityksiin liittyviä funktioita täytyi päivittää.

Muokkasin reititysfunktioita ja kehittelin niiden logiikkaa tukemaan sisällytettyjä resursseja. Funktio toimii nyt niin kuin edellinen samankaltainen funktio. Aiemman toiminnallisuuden lisäksi funktiossa käytetään CFWheelsin sisäistä funktiota päättelemään, mitä muuttujia reitti tarvitsee ja sen perusteella lisää olemassa olevista parametreista ne uudelleen parametreiksi. Siinä on tosin vielä parannettavaa varaa. Haluaisin muokata siitä sellaisen, joka siivoaa ylimääräiset parametrit pois. Tällä hetkellä reititysfunktio saattaa palauttaa tiettyissä tilanteissa ylimääräisiä "from" tai "to" parametreja.

Tein massamuokkauksia eri tiedostoihin. Haluan nopeuttaa työtäni, ettei ihan jokaista riviä tarvitsi muokata erikseen. Kaikkien osioiden delete-funktiota kutsuvat painikkeet ovat korvattu käyttämään DeleteButtonTagia. DeleteButtonTag ei vaadi route arvoa enää, sillä asetin sen käyttämään kontrollerin nimeä oletusarvona. Nimestä piti ottaa viimeinen "S" kirjain pois, koska reittien nimet, jotka muokkaavat resurssia ovat automaattisesti yksikkömuodossa.

CFWheelsin 2 versiosta lähtien reitit käyttävät "add" funktion sijasta "new" funktiota. Kutsuessaan funktiota CFWheels lataa funktion nimisen tiedoston käyttäjälle näkyviin. Kaikki "add" nimiset tiedostot kannattaa muuttaa "new" nimisiksi. Tiedostoja on monta, sillä läjes jokaisella osiolla on sellainen. En halua tai jaksa muokata niitä yksittäisesti. Komentorivin funktiolla saan helposti käännettyä kaikki add.cfm nimiset tiedostot new.cfm nimisiksi. "for /r %x in (add.cfm) do ren "%x" new.cfm" komento kääntää kaikki kansion sisällä olevat add.cfm tiedostot new.cfm tiedoksi.

Perjantai 24.07.2020

Kokeilen nyt yhden päivän niin, että muokkaan toistuvia asioita yhdellä kertaa. Muistan puhuneeni eri tavoista, miten lähestyä projektia ja ajattelin silloin, että asioita kannattaa ajatella kokonaisuuksina. Päivittää yksi osio kerralla. Minulla on sellainen tunne, että

haluan kuitenkin tehdä tällaiset toistettavat asiat kerralla alta pois, koska on turhauttavaa ja hidasta jokaisessa kontrollerissa tehdä samat asiat aina samassa järjestyksessä. Osioiden välillä ehtii unohtamaan päivittävään asiaan liittyvät pikkuseikat ja sitten täytyy joka tapauksessa virkistää muistia ottamalla mallia jo päivitetystä osiosta. Joudun hyppimään sivun välillä tiuhaan tahtiin ja se tuntuu hidastavan työtäni huomattavasti. Yritän tänään välttää sitä. Asetan päivän tavoitteeksi päivittää kaikki etsintätoiminnon partialit.

Sain kaikki etsi-partialit muokattua, joten niitä joutuu tuskin jatkossa enää muokkaamaan. Käytin tänään erittäin paljon Regular Expressionia apuna muokkaamisessa. Sain useita tiedostoja muokattua samanaikaisesti Visual Studio Coden etsi ja korvaa-toiminallisuudella. Kyseinen toiminto löytyy hyvin useista eri IDE:stä. IDE:n tarkoitus on käsitellä tekstiä tiedostoissa. Etsi-kenttään asetetaan hakuehto ja korvaa kenttään korvausteksti. Etsimisessä ja korvaamisessa voi myös käyttää Regular Expressionia. Se on erityisen hyvä sellaisissa tapauksissa, jossa esiintyy paljon samankaltaisuutta.

Päivän aikana etsin tällaisia tapauksia. Korvasin vanhoja FontAwesomen kirjaston ikoneita, muutin kaikki etsi-partialin `#startFormTag()` parametrin vastaamaan uusia "route" määrittäjiä, liitteiden lisäys funktioihin lisäsin puuttuvat "fromRoute" parametrin, luo uusi-painikkeiden action muutin kohdistumaan uusiin "new" funktioihin.

Kaukolämpö Plus järjestelmässä oli määritetty eri sivujen HTML-formit generoitumaan seuraavasti `#startFormTag(id="company", action="update")#`. Näihin funktioihin oli päästävää käsiksi, mutta halusin välttää jokaisen tiedoston yksittäisen muokkaamisen, joten kehitin yksinkertaisen hakulausekkeen `#startFormTag(id="(.*)", action="update")#`. Kyseiset hakuparametrit hakevat koko Kaukolämpö Plus järjestelmän tiedostoista `startFormTag`-merkit, jotka täyttävät hakulausekkeen määrittäykset. Sulut määrittävät Capture Groupin ja sen sisälle määritetty `".*"` valitsee kaikki merkit lukuun ottamatta `"\n"`, joka tarkoittaa uuden rivin alkua (Regexpr; TechTerms 2011). Numerot varustettuna dollarimerkillä mahdollistaa Capture Groupin tuloksen valitsemisen. Korvaavassa lauseessa voimme käyttää Capture Groupin valitsijoita ja asettaa ne haluamaan järjestykseen. Visual Studio Coden korvaavan lauseen kenttään voidaan asettaa seuraavasti `#startFormTag(id="$1", route="$1", key=$1, method="patch")#`. Painettaessa `replace`-painiketta etsitty rivi korvaantuu korvauksi asettamalla tekstillä. Dollarimerkillä varustetut numerot saavat tässä tapauksessa `id`-parametrin sisällön. Lisäksi lisäsin `method="patch"` lisätään `startFormTag`in loppuun.

Viikkoanalyysi

Kuluneen viikon aikana kehitin itseäni kohti oikeaan suuntaan. Lukiessani Martinin Clean Code-kirjaa olen entistä enemmän kiinnittänyt huomiota tuottamaani ohjelmistokoodiin. Olen kiinnittänyt erityisesti huomiota tuottamaani ohjelmistokoodin laatuun. Haluan, että jatkossa tuottamastani ohjelmistosta näkee logiikan ja tarkoituksen selkeästi.

Luin viikolla Regular Expressionista ja yritin käyttää sitä tilanteissa, joissa siitä oli apua. En ole vielä erityisen tehokas taikka taidokas Regular Expressionin käyttäjä, mutta nykyiseen projektiin nähden riittävän osaava. Suurimassa osassa tapauksista, joissa Regular Expressionia on hyödyllistä käyttää, koskee yksinkertaisia etsi ja korvaa funktioiden nimiä tai niiden parametrejä. Massamuokkauksessa Regular Expressionista oli erittäin paljon apua.

Reititysfunktiot ja reititysasiat näyttäisi olevan nyt viimein hallinnassa. MakeToRouteUrl-funktiossa oli ongelmana se, ettei siinä ollut mahdollista ottaa huomioon ne tilanteet, joissa on useampi key-arvo. Asia ratkaistiin yksinkertaisesti niin, että CFWheelsin sisäisen funktion avulla haettiin annetun reitin vaadittavat muuttujat ja sitten toistorakenteessa poimitaan MakeToRouteUrl-funktioon annettujen parametrien joukosta oikeat arvot. Arvoja käytettiin URL-osoitteiden generoinnissa.

AddContact-funktio siirrettiin ylemmälle tasolle, koska kyseinen toiminnallisuus toistuu Kaukolämpö Plus järjestelmässä useita kertoja. Funktio nimettiin uudelleen ja sitä kutsutaan nyt kontrolleritasolla niiden omissa funktioissa. Mielestäni tämä oli hyvä muutos ja sitä tulisi tehdä jatkossa aina tilanteissa, joissa esiintyy toistoa.

Kaukolämpö Plus projektissa on osioiden osalta ajateltu niin, että yksi osio on yksi kokonaisuus ja tämä osio tulisi olla täysin riippumaton muista kokonaisuuksista. Osion tulisi siis toimia täysin itsenäisesti. Tämä on yksi syy, joka voi johtaa toiminnallisuuksien moninkertaistumiseen. Muokatessani ohjelmistokoodia olen useassa kohdassa ajatellut, että nuo funktiot voisi hyvin siirtää ylemmälle kontrolleritasolle yhdeksi funktioksi, jotta tulevaisuudessa ohjelmoijan ei tarvitsisi tehdä samaa muutosta kymmeneen eri funktioon. Funktiot ovat yleensä hyvin samanlaisia. Tiettyjä funktiota on määritetty globaaleiksi funktioiksi, joita on tarkoitus käyttää missä tahansa. GetRoute, getController, getAction ja getKey-funktiot ovat juuri sellaisia funktiota. AddContact-funktio on nyt sellainen funktio. Ainoana "ongelmana" näen, että funktio ei kuulu mielestäni "oikeaan" paikkaan. Funktio kuuluisi mielestäni olla jossain omassa kontrollerissa, joka voitaisiin periyttää käsiteltävään kontrolleriin. Muistan jostain lukeneeni, ettei ColdFusion-ohjelmistokieli tunnista moniperitymistä ja tämä voi olla ongelma, jos asiaa haluttaisiin viedä eteenpäin mainitsemaani suuntaan.

3.9 Seurantaviikko 9

Maanantai 27.07.2020

Löysin viime perjantaina bugin koskien DeleteButtonTagia. Asettamani route-parametrin oletusarvo ei toimi. Oletusarvo haetaan käyttämällä getController-funktiota. Tuloksesta riisutaan viimeinen kirjain pois muutamalla nimi yksikkömuotoon. Inquiries muuttuu Inquire nimiseksi ja sellaista ei löydy mapperista. Inquiries tapauksessa yksikkömuodon tulisi olla Inquiry. Tavoitteena tänään on korjata kyseinen ongelma. Lisätavoitteena on saada Inquiries-osio valmiiksi.

Löysin CFWheelsin sisäisen singularize-funktion, joka muuttaa annetun nimen yksikkömuotoon. Funktio oli juuri sellainen, jota etsinkin. Ajattelin aluksi käyttää \$findRoute ja jotain omaa funktiota yksikkömuodon päättelyyn, mutta onneksi tajusin tutkia CFWheelsin tiedostojen sisältöä.

Inquiries-osio valmistui myös ja siirryin saman tien uuden osion päivittämiseen. Se olisi sen verran pieni osio, ettei sen päivittämisessä mennyt kauaa. Osioden päivittämisessä yleensä vie aikaa reittien lisääminen mapperiin sekä uusien reittien lisääminen näkymätiedostoissa vanhojen paikalle. Monessa tapauksessa parametrit ja key-arvoja joutuu pyörittelemään ja lukemaan itse funktiot kertaalleen läpi, jotta tietää minkälaiseksi reitti määritetään.

Tiistai 28.07.2020

Tavoitteena tänään on saada Contracts, ConnectionPlans ja ContractDates-osiot valmiiksi. Tänään käyn myös toimistolla Keilaniemen kauniissa merimaisemassa. Meillä on palaveri tiimin kesken, missä katsomme, kuinka projekti etenee.

En saanut osiota valmiiksi. Aamupäivä meni keskustellessa Kaukolämpö Plus versio 7 ominaisuuksista, päivitetyistä ja päivittämättömistä osioista. Tiimin muut jäsenet käyttävät seuraavat päivät testaillessa projektin päivitettyjä osiota. Näin varmistamme, onko toiminnallisuudet ennallaan. Päätimme kerätä virheet Jiraan.

Keskiviikko 29.07.2020

Tavoitteena tämän päivän aikana on saada eilen mainitsemani osiot valmiiksi. Niistä yhdessä osiossa on useita reittejä lisäämättä ja toisessa vain muutamia.

Osiot ovat nyt valmiita. Aloitin heti News-osion päivittämisen. Kyseistä osiota käytetään järjestelmän etusivulla. Koko osion idea on mahdollistaan pääkäyttäjille helppo tapa tuottaa järjestelmän sisäisiä uutisia sen käyttäjille luettavaksi. Uutisia, kuten uudet ominaisuudet tai järjestelmämuutokset.

Etusivulla on käyttäjän mahdollista siirtyä muokkaamaan omaa profiiliaan. Käyttäjä voi siinä vaihtaa tietoja kuten, nimensä ja työnimikkeensä. Pääkäyttäjällä on lisävalinta, jolla voi määrittää, onko kyseinen käyttäjä pääkäyttäjä. Kokeilimme seuraavaa tiimiläisen kanssa; kirjauduimme peruskäyttäjänä sisään järjestelmään, muokkasimme sivulla esiintyvää lomaketta lisäämällä sen sisälle piilotetun "onko pääkäyttäjä" valinnan tekstikenttänä totta-arvolla ja painomme tallenna-painiketta. Päivitys meni läpi ja peruskäyttäjä oli hetkessä pääkäyttäjä. Nyt käyttäjällä oli täydet oikeudet järjestelmään. Tämä luetaan tietoturvariksi.

Yksinkertaisena ratkaisuna oli muuttaa käyttäjäoliota siten, että käyttäjän päivittäessä käyttäjätietoja järjestelmään tarkistetaan, onko muutoksen tekevä käyttäjä pääkäyttäjä tietokannasta käsin. Jos käyttäjä ei ole pääkäyttäjä tietoa ei päivitetä. Nyt järjestelmä on asetta turvallisempi.

Torstai 30.07.2020

Tavoitteena tänään on saada News-osio ja etusivun kaikki muut toiminnot valmiiksi. Lisäksi myös korjata navigaatiopalkista löytyvä ohjelmointivirhe. Ohjelmointivirhe liittyy toisen tason linkkeihin, joita käyttäjä ei voi valita. Valinta on estetty silloin, kun toisen tason alta löytyy kolmannes taso.

Sain News-osion päivittyä valmiiksi heti aamusta. Siinä oli vain yksi funktio, joka ei halunnut toimia deleteButtonTagin kanssa. Asetin joku aikaa sitten deleteButtonTag-merkin luontifunktioon kontrolleriargumentin. Jos käyttäjä ei määritä merkin luontivaiheessa kontrolleria se haetaan parametreista oletukseksi. Tämä tehtiin siksi, että käyttäjälle voitaisiin määrittää palautumispaikka. Käyttäjä ohjataan aina palautuspaikkaan onnistuneen tai epäonnistuneen poistamisen jälkeen. Ongelmaksi osoittautui, että mapperiin määritetty News-osio taipuu eri tavalla kuin muut osiot. En paikantanut syytä, miksi News-nimenä taipuu eri tavalla. Luulen sen johtuvan siitä, että sanalla "News" ei ole yksikkömuotoa. Se ei voisi olla "New", koska "New" on varattu termi jokaiselle osiolle. Mainitsin aikaisemmin, että CFWheels päättelee sanasta monikkomuodon ja yksikkömuodon siihen kehitetyillä funktioilla. CFWheelsin vastaus kyseiseen ongelmaan on lisätä sana "Index" osion

perään. Oletuspolku tai osion etusivulle vievä reitin nimi olisi silloin "NewsIndex". Funkti-
oon, joka generoi deleteButtonTagissa paluupaikan osoitteen käyttäisi kyseessä olevan
nimeä "News". Tässä tapauksessa "News" reittinä ei johtaisi minnekään. Yksinkertaisin
tapa oli poistaa "News" resurssista "delete" funktio käyttämällä "except" määrittystä ja sit-
ten määrittämällä oma "DeleteNews" funktio, joka osoittaa kyseessä olevan kontrollerin
delete-funktioon. Lopuksi DeleteButtonTagiin täytyi vielä määrittää "DeleteNews" reitiksi.

Navigaatiopalkissa löytyi ohjelmointivirhe. Navigaation toisen tason linkit, joissa oli kol-
mannen tason valintoja ei sallinut käyttäjää valitsemaan toisen tason linkkiä. Kuvassa 8
näky toisen tason navigaatiopalkki. Käyttäjä painoi Tarkastukset-linkkiä, jonka olisi tarkoi-
tus viedä käyttäjä tarkastukset sivulle, mutta sen sijaan käyttäjälle avautui kolmannen ta-
son linkit. Kolmannen tason linkit toimivat moitteettomasti. Ratkaisuna ongelmaan oli
muuttaa navigaation rakennetta siten, että toisen tason linkeistä, joissa oli kolmannen ta-
son valintoja, otettiin linkit pois. Linkit siirrettiin tekstin data-attribuuteiksi. Lisäsin Ja-
vaScriptillä funktion, joka kuuntelee klikkauksia toisen tason tekstissä. Tekstiä klikatessa
data-attribuuteista poimitaan osoite ja käyttäjä ohjataan sivulle.

Perjantai 31.07.2020

Tänään on tavoitteena saada Activities-osio valmiiksi. Olen korjailut sitä vähän pitkin ke-
sää, sillä se on liittynyt muutamiin tekemiini osioihin. Jos aikaa jää aloitan SalesLeads-
osion päivittämisen. Vielä on vaikea arvioida, kauanko yhden osion päivittämisessä kuluu,
koska jokainen osio on hyvin erilainen suuruudeltaan. Siksi on vaikea sanoa, onko tavoite
edes saavutettavissa. Pitää kuitenkin muistaa, että tässä projektissa tavoitteet eivät ole
kriittisiä. Tämän tyyppisissä tavoitteissa voi joustaa tarpeen tullen.

Päivän aikana saavutin suunnitellut tavoitteet. Tavoitteiden lisäksi sain valmiiksi muitakin
osioita. Activities-osion lisäksi sain valmiiksi SalesLeads-osion. Disconnections-osio on
hyvin lähellä valmistumista, kuten myös siihen liittyvät DisconnectionSteps-osio. Perjan-
taina on hyvin tuottelias.

Viikkoanalyysi

Viimeviikolla jätin avoimen kysymyksen liittyen kontrollerien moniperiytymiseen. Keskuste-
lin asiasta tiimiläiseni kanssa ja hän sanoi, ettei ColdFusionissa ole mahdollista monipe-
riyttää kontrollereita. Tämä oli kylläkin tiedossa, sillä noin vuosi sitten käydyssä projektissa
halusimme hyödyntää saman tyyppisiä funktiota tietyssä kontekstissa. Loimme uuden
kontrollerin, jota käytimme kyseisessä projektissa. Luodulla kontrollerilla pystyi

hallitsemaan tiettyjä laitteita. Se sisälsi funktioita, kuten liitä tai irrota kyseinen laite mittarilta. Jos Kaukolämpö Plus järjestelmästä haluaisi tehdä poistaa kahdentumaa, täytyisi järjestelmässä hyödyntää useita eri kontrollereita tai käyttöliittymiä eli interface:ia. Interface:lla olio-ohjelmoinnissa tarkoitetaan listaa toiminnallisuuksista, joita elementillä täytyy olla toimiakseen (University of Utah).

Olen yrittänyt karsia aikaa vieviä prosesseja päivittäisestä työstäni etsimällä siihen soveltuvia työkaluja. Yksi tällainen työkalu on ollut Regular Expression. Se on mahdollistanut jomoneen kertaan mainitsemani massamuutokset. Regular Expression on hyödyllinen työkalu silloin, kun olen halunnut vähentää manuaalista toistuvaa työtä. Sen käyttö on melko yksinkertaista tai ainakin on ollut niissä tapauksissa, joissa olen sitä hyödyntänyt. Esimerkiksi etsin koko järjestelmän kaikki search-partialit ja muokkasin niiden sisältä löytyvät Bootstrapin tabit ja niiden sisällä olevat linkit käyttämään oikeita linkkejä. Visual Studio Code:lla näkee vielä ennen hakuosumien korvaamista, mitkä osumat muuttuvat muutoksen tehdessä. Työkalun hyödyntäminen on nopeuttanut huomattavasti tiedostojen läpi käyntiä. Nyt monessa osiossa riittää, kun muuttaa vain näkymätiedostoissa esiintyvät linkit käyttämään reittejä, jotka määritetään samalla mapperiin.

Projektissa on vielä jäljellä melko monta osiota päivittämättä. Tiedän, että vauhti väistämättäkin jatkaa kiihtymistään. Minulla on aika hyvin päivitysprosessi hallussa Kaukolämpö Plus järjestelmän osalta. Tiimiläiset liittyivät joukkoon testaamaan jo päivitettyjä osiota. He listaavat ohjelmointivirheeni ja unohtuneet määritykset Jiraan. En ole vielä ajattelut korjata Jiraan koottuja virheitä. Ajattelen, että niiden korjaaminen tapahtuu vasta projektin loppuvaiheessa.

Ensiviikolla on lyhyempi viikko. Aion heti maanantai aamusta määrittää kolmen päivän pituisen kehitysjakson. Kehitysjaksot ovat olleet tauolla muutaman viikon. Syy johtuu, siitä etten ole ihan varma, mitä sinne kirjoittaisin. Loppujen lopuksi osioiden perusteellinen kirjaaminen Jiraan vie huomattavan määrän aikaa ja näen sen osittain turhaksi. Ensimmäisissä kehitysjaksoissa kuvasin jokaisen toiminnallisuuden omaksi "kortiksi", kuten kuvassa 4 näkyy. Parempi olisi vain listata osiot, jotka tulee päivittää toimintoiheen kehitysjakson aikana. Projektin kokonaistavoitetta ja kehitysjakson tavoitteita olisi ehkä helpompi seurata. Olen myös ajattelut lisääväni kaikki osiot listana Jiraan, johon voisin sitten merkata, mitä olen päivittänyt ja mitä on vielä päivittämättä.

3.10 Seurantaviikko 10

Maanantai 03.08.2020

Tänään on tavoitteena määrittää kolmen päivän pituinen kehitysjakso. Jätän turhan perusteellisen kuvauksen toiseen kertaan ja keskityn siihen, että Jirasta löytyy minimaalinen tarvittava tieto, jolla saa kehitysjakson suoritettua. Disconnections ja DisconnectionSteps-osiot ovat ainakin tämän päivä listalla, jotka olisi hyvä saada tehtyä.

Aloitin kehitysjakson tänään. Määritin sen mahdollisimman suppeasti ilman turhia kuvauksia ja tarina tai tehtävä erotteluja. Jirassa lukee yksinkertaisesti neljän osion nimet tyypillisessä tarinamuodossa. Kehitysjakson tekemiseen meni noin viisi minuuttia, koska sen sisältö oli ennaltaan tiedossa.

En saanut Disconnections tai DisconnectionSteps-osioita valmiiksi, koska keskityin tänään kaikkeen muuhun. Kaukolämpö Plus versio 6:ssa oli vielä mahdollista käyttää linkTo ja buttonTo-tageissa "confirm" parameteria. Confirm-parameteriin annettiin arvoksi teksti, joka haluttiin näyttää käyttäjälle. Sen tarkoitus on mahdollistaa kehittäjälle helppo käyttöliittymä varmistusikkunaan. Confirm-parametrilla esitetään käyttäjälle ikkuna, jossa varmistetaan, onko käyttäjä varma varmistettavasta asiasta. Näitä tapauksia löytyi Kaukolämpö Plus järjestelmästä noin 90. Nykyään confirm-parametri ei toimi ja CFWheelsin Upgrade Guidessa kerrottiin, että on suositeltavaa käyttää jotain siihen tarkoitettua lisäosaa. Lisäosia oli kaksi saatavilla ja kummatkin niistä sopi mainiosti uuteen versioon. Ne tosin erosivat nykyisen käyttöliittymän ulkoasusta.

Päätin tehdä oman vastaavanlaisen lisäosan. Otin mallia JS Confirm lisäosasta (King). Muokkasin JS Confirm lisäosaa siten, että se tukisi BootstrapDialog lisäosaa. Monistin JS Confirmin uudelle nimelle. Valitsin nimeksi "BootstrapConfirm". Kyseessä on hyvin yksinkertainen lisäosa, joka yliajaa normaalin linkTo ja buttonTag-merkit. Samalla se lisää tarkistuksen, jossa katsotaan, löytyykö merkin sisältä confirm-argumenttia. Jos argumenttia ei löydy argumenttilistasta, se käyttäytyy normaaliin tapaan.

```

public string function $jsStripHrefTag(required string rv){
    local.href = REMatch('href="\([^"]*)', arguments.rv);
    local.href = local.href[1] & '';
    local.href = REMatch('"(.*)"', local.href);
    local.href = local.href[1];
    local.href = Left(local.href, -1);
    local.href = Right(local.href, -1);
    return local.href;
}

```

Kuva 17. HREF-attribuutin sisällön riisuminen a-merkistä

Ohjelmistoon oli lisättävä uusi funktio, joka riisuu href-attribuutin sisällön a-merkistä. Href-attribuutin sisällä on osoite, johon käyttäjä halutaan ohjata, jos BootstrapDialogin valinta on positiivinen. Loin tähän tarkoitukseen uuden funktion ”\$jsStripHrefTag”. Funktio näkyy kokonaisuudessaan kuvassa 17. Se etsii a-merkistä href-attribuutin sisällön REMatch-funktiolla. Etsinnässä hyödynnetään Regular Expressionia. Osoite siivotaan ylimääräisistä merkeistä ja palautetaan pääfunktion local.href muuttuun.

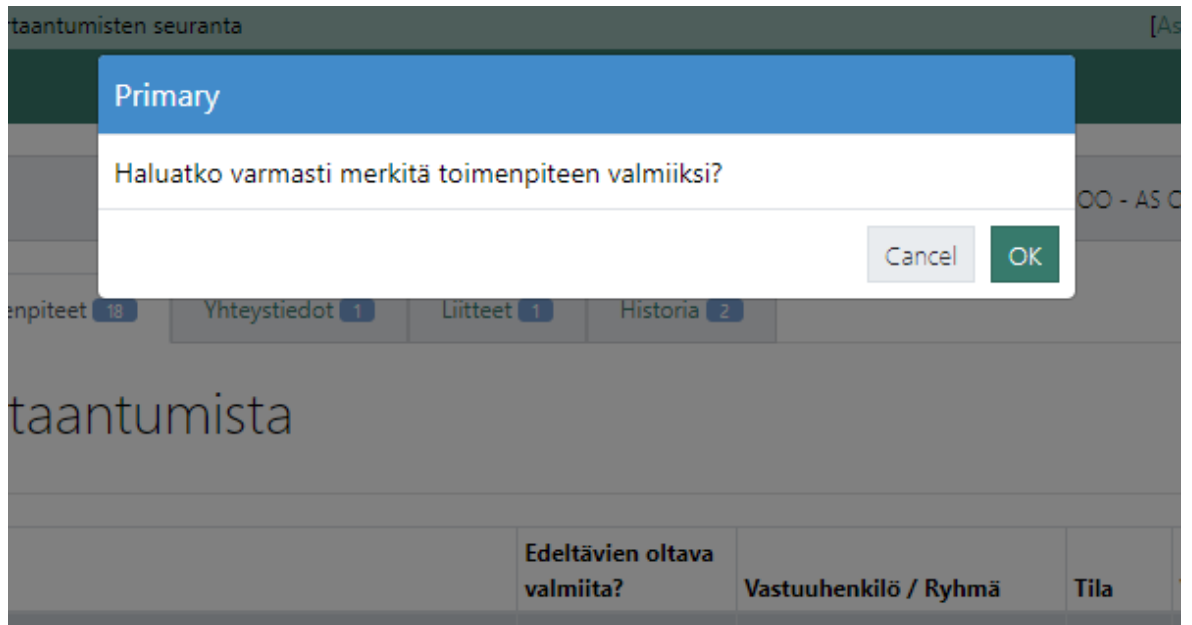
```

public string function $jsBootstrapConfirmAdd(required string name, required string rv, required string confirm) {
    if (Len(arguments.confirm)) {
        local.href = $jsStripHrefTag(arguments.rv);
        local.methodWithUrl = "window.location.href='" & local.href & "'";
        local.js = "event.preventDefault();BootstrapDialog.confirm('#JSStringFormat(arguments.confirm) #', function(result){if(result){local.methodWithUrl#}});";
        local.check = " onclick=""";
    }
}

```

Kuva 18. BootstrapConfirm toiminnallisuuden aktivointi

Pääfunktio on kuvattu osittain kuvassa 18. Muuttuja monistetaan uudeksi ja siihen lisätään JavaScriptin window.location.href-metodi. Metodilla ohjataan käyttäjä osoitteeseen. Local.js muuttuun tallennetaan myös pieni skripti, jonka tarkoituksena on mahdollistaa BootstrapDialogin käyttö. Event.preventDefault-metodi estää linkin normaalin toiminnan. Tässä tapauksessa sillä varmistetaan, ettei varmistusikkuna vain välittäisi käyttäjälle. Lisäksi se estää myös käyttäjän ohjautumasta uudelle sivulle.



Kuva 19. BootstrapConfirmin generoima varmistusikkuna käytännössä

BootstrapConfirm-lisäosa toimii nyt jokaisessa noin 90 kohdassa. Se on paras kolmesta muusta vaihtoehdoista. Toinen vaihtoehdoista olisi ollut poistaa kaikki varmistusviestit. Kolmas olisi ollut kirjoittaa jokaiseen tapaukseen oma JavaScript funktio, joka näyttää vahvistusviestin. Nyt kun se on valmis voin taas keskittyä muihin asioihin. BootstrapConfirmiin täytyy tosin vielä asettaa tekstioletukset. Englanninkieliset nappulat ja otsikkopalkki täytyy suomentaa. Ne voi hyvin todennäköisesti asettaa oletuksiksi sivun yleiseen JavaScript-tiedostoon.

Tiistai 04.08.2020

Työpäivän tavoitteena on vaihtaa PickADate-lisäosa toiseen samankaltaiseen "datetimepickeriin" ja saada Disconnections sekä DisconnectionSteps-osio valmiiksi. Osoiden valmistujen jälkeen minulla olisi vuorossa Integrations, Interruptions sekä InterruptionLetters-osiot.

PickADate-lisäosa vaihdettiin toiseen. Uudeksi datetimepickeriksi valitsimme tiimiläisen kanssa TempusDominuksen datetimepickerin (TempusDominus Bootstrap 4). Vanha datetimepicker ei toiminut niin kuin halusimme. Tämä on jo kolmas kerta, kun vaihdamme samankaltaista lisäosaa järjestelmässä. Onneksi heti alussa loin DateTimeTagin, koska nyt kaikkien kenttien vaihtaminen on nopeaa. Riittää vain kuin muuttaa yhtä funktiota yhdessä tiedostossa. TempusDominuksen kanssa on ollut myös vaikeuksia. Sen käyttöönotto on epäselvä ja ohjeita ei juurikaan löydy. Sen funktioiden käyttö on hieman epäjohdonmukaista, sillä se ei tunnu noudattavan mitään normeja. Haluaisin esimerkiksi, että

tekstikentän arvon voisi alustaa tai noutaa jollain funktiolla. PickADate-funktiolla oli tähän tarkoitukseen set ja get-funktiot. Tutkiessani TempusDominuksen dokumentaatiota en löytänyt mitään vastaavaa (TempusDominus Bootstrap 4). Nämä funktiot ovat mielestäni erittäin yleisiä ja siksi on hyvin epänormaalia, jos sellaiset puuttuisivat. Voi tosin olla, että jostain jäi huomaamatta.

Disconnections ja DisconnectionSteps-osiot valmistui heti aamusta. Niissä ei juurikaan ollut muokkaamista jäljellä. Näiden jälkeen siirryin päivittämään Integrations-osiota, joka oikeastaan vaati vain mapperiin resurssin merkitsemisen. Sen jälkeen vuorossa oli päivittää Interruptions ja InterruptionLetters-osiot. Yksi tiimiläisistämme kertoi jokin aikaa sitten, että kyseisen osion kontrollerista ja sen näkymätiedostoista voi poistaa index sekä search-funktiot. Tänään keskustelimme asiasta uudelleen, koska halusin varmistaa muistini. Päädyimme siihen, että koko kontrollerin voisi nimetä uudelleen ja sisällyttää siihen vain tietyt funktiot vanhasta. Samalla saamme osion nimestä vähän kuvaavamman.

Keskiviikko 05.08.2020

Tänään siivoan Interruptions-osion ja annan sille toisen nimen. Maanantaina aloittamani erittäin lyhyt kehitysjakso päättyy tänään, joten kaikki listalla olevat olisi saatava valmiiksi. Tämä ei ole ongelma, sillä nämä kaksi osiota ovat yhdessä valmiita tämän aamupäivän aikana. Loppupäivän ajan minulla on aikaa korjata tiimiläisen keräämiä ohjelmointivirheitä tai unohdettuja toiminnallisuuksia. TempusDominus olisi myös hyvä saada toimimaan.

Kehitysjakson tavoitteet saavutettiin aamupäivästä. Osiot olivat hieman pienempiä kuin esimerkiksi viimeviikolla työn alla olleet osiot. Vielä on muutama osio päivitettävänä ennen kuin siirryn niistä suurimman pariin. Olen tarkoituksella jättänyt mittaristoon liittyvät osiot viimeiseksi. Uskon, että ensiviikon jälkeen viimeistään siirryn niiden pariin.

TempusDominus-laajennus toimii nyt Kaukolämpö Plus järjestelmässä. Sen asentamisessa meni isompi osaa päivästä. Se vaatii huomattavasti muutoksia dateTimeTagiin sekä moneen eri osion näkymätiedostoihin. Eniten vaikeuksia tuotti Activities-osion edit-partialissa määritetyt JavaScript-funktiot. Sivulla on kaksi päivämääräkenttää ja kaksi kellonaikakenttää. Kolmeen niistä oli liitetty funktio, joka kuuntelee muutoksia jokaisen omassa tekstivalintakenttässä. Mainitsin eilen, että dokumentaatio on puutteellinen ja olen edelleen sitä mieltä. Sain kuitenkin pienellä väännöllä kentät toimimaan. Lisäosasta tosiaankin puuttuu hyvin oleellisia funktioita, kuten "set" tyyppinen funktio, joka asettaisi valittuun datatimepickeriin tietyn arvon. Lisäksi koko laajennus toimii vähän kummallisella tavalla. Datatimepicker liitetään samalla tavalla elementtiin kuin mikä tahansa muu lisäosa,

mutta tässä niitä käsitellään hieman eri tavalla. Kaikki käsittely tapahtuu data-attribuutin kautta. Asettaessani datetimepickerin elementtiin joudun joka kerta kutsumaan data-metodia ensin ja sitten datatimepickerin funktiota. Funktioita kutsutaan niiden nimillä antamalla datepicker-metodiin arvo. Datetimepickerin arvojen asettamisessa on käytettävä JQueryn val-funktiota. Arvon noutaminen onnistuu datetimerpicker-metodin "viewDate"-arvolla tai sitten datetimepickerin datetimepicker-muuttujan date-muuttujalla. Ohjelmistokoodissa jälkimmäinen lyhin vaihtoehto näyttäisi tältä: `$("#element").data('datetimepicker').date`. Elementillä tarkoitetaan HTML-elementtiä, johon datetimepicker on lisätty.

Viikkoanalyysi

Uutena asiana tällä viikolla tuli CFWheelsin lisäosien ohjelmointi. BootstrapDialogin tekeminen oli yksinkertaista ja nopeaa. Osittain siksi, että pystyin ottamaan mallia toisesta lisäosasta. Eniten aikaa kului Regular Expressionin kanssa. Href-attribuutin poimiminen amerkistä oli melko hankalaa. Vaikeuksia aiheutti merkkijonon eskapointi. Eskapoinnilla tarkoitetaan merkkijonojen merkkien muuttamista erikoismerkityksistään "tekstiksi" (IBM). Eskapoinnissa lisätään kauttaviiva erikoismerkin eteen. Tässä tapauksessa muutoksia vaati href-attribuutin arvon kauttaviivat ja attribuutin ympärillä olevat puoli- ja lainausmerkit.

Kaukolämpö Plus järjestelmän päivitysprosessi on hyvällä mallilla. Päivittämistä ja ohjelmointivirheiden korjaamista vielä riittää tuleville viikoille. Vauhti on kiristynyt huomattavasti menneillä viikoilla. En tiedä johtuuko se siitä, että monet osiosta ovat on valmiita käytettäväksi vai siitä, että olen työn teossa harjaantunut sellaiseksi, ettei jatkuvasti tarvitse katsoa ohjeita tai kerrata tehtyjä muutoksia.

Päiväkirjan kirjoittaminen päättyy tänään. Päiväkirjaprosessin aikana olen pysynyt hyvin kärryllä, mitä on tehtävä seuraavaksi. Se on auttanut minua keskittymään työhöni ja asettamaan selkeitä tavoitteita. Tavoitteisiin on ollut helppoa pyrkiä ja niiden seuraaminen on ollut yksinkertaista. Päiväkirjaa en tule jatkossa kirjoittamaan työn ohella, koska tämä on aikaa vievää, vaikka olen kirjoittamisessakin kehittynyt huomattavasti. Pidin aikaisemmin "To do" -listaa aktiivisesti työn ohella. "To do" -listan ylläpitäminen oli myös vaivalloista. Vaivaantumisesta tuskin pääsen koskaan eroon tässä asiayhteydessä, mutta se on loppujen lopuksi erittäin pieni hinta ajan tasalla sekä kartalla pysymisestä.

4 Pohdinta ja päätelmät

Alussa kerroin, kuinka vanhojen tapojeni mukaisesti saatoin monimutkaistaa funktiota tarpeettomasti. Saatoin välillä käyttää funktioiden dynaamiseksi kirjoittamiseen tuhottomasti aikaa. Halusin, että funktioita voisi käyttää useissa eri tilanteissa. Useissa tilanteissa dynaamisuudelle ei ole tarvetta. Olen tiedostanut edellisen ja yrittänyt parantaa tapojani. Raportoinnin aikana pyrin keskittymään olennaiseen. Asettamalla selkeät tavoitteet jokaiselle päivälle vähensi turhan tekemistä. Usein työpäivän jälkeen suunnittelin seuraavan päivän asioita valmiiksi. Kirjoitin lyhyen kappaleen, jota sitten täydensin seuraavana päivänä ennen työpäivän aloittamista. En muista unohtaneeni kertaakaan raportoinnin aikana, mitä olin tekemässä. Tämän kanssa kamppailin vielä vähän ennen opinnäytetyötä. Tosin silloin työskentelin vain yhtenä päivänä viikossa. Minulla oli käytössä erilaisia listaohjelmia, joihin kirjasin asioita siihen järjestetykseen, mistä oli seuraavaksi jatkettava. Se oli ihan hyvä ratkaisu, mutta väitän, että päiväkirjan tapainen muistio toimii paremmin.

Tutustuin kirjailijaan nimeltä Robert C. Martin. Hän on kirjoittanut useita kirjoja liittyen ohjelmointiin. Hänen Clean Code-kirjastaan oli paljon apua. Minulla oli jo entuudestaan polttava tarve lukea, jokin ohjelmointiin liittyvä kirja. Kirjassaan hän listaa hyviä ohjelmointitapoja ja pistää lukijan pohtimaan omaa ohjelmointityyliään. Huomasin useissa kohdissa ajattelevani hyvinkin samalla tavalla, mutta Robert onnistui viemään asian vielä askeleen pidemmälle. Hän on saanut minut ajattelemaan ohjelmointityyliäni uudelleen. Kiinnostukseni kirjaan todennäköisesti tulistui, kun tajusin, että haluan olla hyvä ohjelmoija. Sellainen ohjelmoija, joka ottaa muut ohjelmoijat huomioon ja ei pelkää tehdä ohjelmistokoodiin muutoksia. Muutoksia, jotka saattavat rikkoa koko järjestelmän. Haluan ymmärtää, mikä tekee esimerkiksi funktiosta hyvän funktion.

Kuluneiden viikkojen jälkeen tunsin kehittyneeni ohjelmoijana. Minulle oli jo entuudestaan karttunut työhön riittävästi tarvittavaa osaamista. Tästä syystä pystyin keskittymään itseni kehittämiseen eri tavoin. Clean Code -kirja oli tässä isossa osassa. Tunnen olevani huomattavasti kriittisempi omaan jälkeeni. Käytän huomattavasti enemmän aikaa esimerkiksi funktioiden kirjoittamiseen. Haluan, että ne ovat loogisia, yksinkertaisia ja mahdollisimman kuvaavia. Suosin yksinkertaisuutta dynaamisuuden yli. Kirja on saanut minut ajattelemaan näitä asioita uudelleen. Kaukolämpö Plus päivitysprojektin aikana refaktoroin useita funktiota juuri tästä syystä. Useat funktiota ovat silti päivittämättä, koska Kaukolämpö Plus järjestelmän päivityksessä tähän ei ryhdytty. Kaiken uudelleen kirjoittaminen veisi tuhottomasti aikaa. Järjestelmä oli tarkoitus päivittää uusimpiin versioihin eikä korjata jokaista olemassa olevaa epä johdonmukaisuutta tai mahdollisesti sekaannuttavaa asiaa.

Kaukolämpö Plus järjestelmän päivitysprosessin aikana pääsin kokeilemaan muutamia uusia asioita. Päälimmäisenä niistä on suoraan päivitysprosessiin liittyviä seikkoja. Lähihinä koko ColdFusion palvelimen pystyttäminen, CFWheels kehitysrungon asentaminen ja edellisten versioiden sekä Bootstrapin version päivittäminen. Minulla oli aluksi ajatus, että ColdFusion ja CFWheels tuottaisivat eniten muutoksia järjestelmään, mutta olin väärässä. ColdFusion ja CFWheels eivät juurikaan tuottaneet suoraan koodin muutoksia. CFWheels vaati muutoksia ainoastaan reittien ja muutamien tiedostojen sijaintien osalta. Bootstrapin päivitys vaati eniten huomiota. Bootstrap 3 ja Bootstrap 4 versioiden väliin oli mahtunut erittäin paljon muutoksia ja niiden tuottaminen koodin vei suurimman osan ajasta.

Päivittäessäni näkymätiedostoja tutustuin myös CFWheelsin lisäosiin. Projektin aikana törmäsin tilanteeseen, jossa vanha toiminnallisuus oli helpompi saattaa uudelle versiolle luomalla sitä varten CFWheelsin lisäosa. Otin mallia jo olemassa olevasta lisäosasta, jonka päälle tein meille sopivat muutokset. Halusin, että lisäosa näyttää Kaukolämpö Plus-salta, joten sisällytin siihen BootstrapDialog-laajennuksen. Laajennustahan käytetään lähes jokaisessa järjestelmän osiossa, joten yhteneväisyys säilyy tämän osalta.

Tutustuin myös CommandBox nimiseen työkaluun. Minulla oli ajatus projektin alkupäässä, että kyseisellä työkalulla saisin helposti hallintoa Kaukolämpö Plus järjestelmää ja tätä kautta lisättyä tehokkuutta työssäni. Sen käyttöönotto oli helppoa ja siitä löytyi lähes tyhjentävät ohjeet. CommandBoxilla olisi voinut luoda kontrollereita, modeleita ja näkymätiedostoja yhdellä lauseella suoraan käyttöliittymästä. Käyttöönotto tyssäsi lopullisesti siihen, kun tajusin, ettei CommandBoxiin lisättävällä CFWheels lisäosalla ollut mahdollisuutta tuottaa oikean tyyppistä ohjelmointikieltä. Ainakin natiivisti CommandBox käytti CFScriptiä. Kaukolämpö Plus järjestelmä on kirjoitettu CFML:llä.

Keräsin ymmärrystä Regular Expressionista menneiden viikkojen aikana. Olen pitkään katsonut ja pysynyt siitä tiedoin tahdoin kaukana, koska sen syntaksi on vaikuttanut erittäin epäselvältä ja siksi pelottavalta. Tutustuin siihen vasta nyt oikeastaan ensimmäisen kerran. Sen käyttämisestä on ollut projektin aikana mahdottomasti apua. Aktiivisesti tiedostava käyttäminen on todennäköisesti säästänyt minulta useiden päivien edestä manuaalista työtä. Syntaksissa on silti paljon aukkoja. Se on täynnä merkkejä, joiden merkitystä en vielä ymmärrä. En anna sen kuitenkaan estää intoani käyttää sitä. Verkko on täynnä palveluita, joissa voi helposti testata, miten Regular Expressionin merkkijono toimii käytännössä. Useat näistä sivuista tarjoavat myös esimerkkejä, miten Regular Expressionilla saa yleisimpiä ongelmia ratkottua. Aion jatkossa hyödyntää Regular Expressionia siellä, missä siitä on apua.

Mielestäni päiväkirjan tekstiä on melko vaikeaa hyödyntää suoraan missään jatkokehityksessä tai tutkimuksessa. Uskon, että päiväkirjan teksti yksinkertaisesti on liian ominainen, ettei sitä voi soveltaa muissa samantyyppisissä projekteissa. Päiväkirjasta saa yleiskuvan siihen minkälaista on päivittää suurehko web-järjestelmä sen ohjelmiseen, mutta teksti käsittelee vaan ja ainoastaan ColdFusion, CFWheels ja Bootstrap ympäristöä. Tästä syystä hyödyntäminen onnistuu lähinnä vain silloin, jos samankaltainen ympäristö vaatii päivitystä.

Viikoittaiset analyysit ja päivittäiset kirjoitukset saivat minut ajattelemaan tapojani työssä. Päätin opinnäytetyön alussa keskittyä töissä töihin ja vapaa-ajalla aivan muuhun. Tämä oli minun osaltani mahdollista siksi, että pidin kyseessä olevaa päiväkirjaa. Pysyin aikataulussa Kaukolämpö Plus projektissa, koska jokainen päivä oli suunniteltu etukäteen. Väitän, että suunnitellut tavoitteet vähensivät työpaineita. Fokuskello, johon tutustuin opinnäytetyöprojektin puolesta välissä auttoi minua jaksamaan päivittäiset työtehtäväni tehokkaammin loppuun. Taukojen pitäminen osoittautui erittäin olennaiseksi työhyvinvoinnin kannalta, kuka olisi arvannut.

Asetin itselleni tavoitteeksi ymmärtää, mitä web-järjestelmän päivitysprosessi pitää sisällään ja miten se suoritetaan. Halusin hyödyntää ketteriä kehitysmenetelmiä, joiden avulla päivitysprosessi pysyisi hyvin hallinnassa. Projektia kokonaisuudessa hallittiin Jiralla. Seurantaviikkojen aikana pidin useita kehitysjaksoja, joille jokaiselle asetettiin omat projektiin liittyvät tavoitteet.

Opinnäytetyö oli opettavainen kokemus. Asettamani tavoitteet saavutettiin. Toisaalta koen silti, että päivitysprosessin toteuttamisessa olisi parannettavaa. Tulevaisuudessa käyttäisin enemmän aikaa ympäristön tutkimiseen ja asettaisin projektille selkeät rajat. Päivitysprojektin aikana ilmaantui yllättäviä asioita, joihin olisi voinut varautua perusteellisella perehtymisellä etukäteen. Teknillistä kokemusta en juurikaan saanut lisättyä, koska projekti, jota opinnäytetyön aikana tein liittyi hyvin pitkälti päivittäiseen työhöni. ColdFusionin, CFWheelsin ja muutamien muiden osalta tutustuin muutamiin uusiin käsitteisiin, mutta muuten työ oli pääosin tuttua. Teknillisen kokemuksen lisääminen on tärkeää, mutta uskon opittujen hyvien ohjelmointikäytäntöjen ja -tapojen tuottavan enemmän hyvää pitkällä juoksulla.

5 Lähteet

Painetut lähteet:

Martin, R. 2019. Clean Code. Pearson Education, Inc. United States of America

Elektroniset lähteet:

2Kmediat. ColdFusion. Luettavissa: <https://www.2kmediat.com/internetohjelmointi/coldfusion.asp>. Luettu: 30.05.2020

Adobe a. About ColdFusion. Luettavissa: <https://helpx.adobe.com/coldfusion/using/about-coldfusion.html>. Luettu: 28.05.2020

Adobe b. ColdFusion 2018 Migration Guide. Luettavissa: www.adobe.com/content/dam/acom/en/products/coldfusion/pdfs/cf2018/cf2018-migration-guide.pdf&usq=AOvVaw1RJBHYTMz9SSpl4mzCU0b8. Luettu: 03.06.2020

Agile Alliance. Minimum Viable Product (MVP). Luettavissa: <https://www.agilealliance.org/glossary/mvp/>. Luettu: 02.06.2020

Atlassian a. Git Branch. Luettavissa: <https://git-scm.com/book/en/v2/Git-Branching-Branched-in-a-Nutshell>. Luettu: 03.06.2020

Atlassian b. Jira Software best practices. Luettavissa: <https://www.atlassian.com/software/jira/guides/getting-started/best-practices>. Luettu: 07.06.2020

Atlassian c. The Agile Coach. Luettavissa: <https://www.atlassian.com/agile>. Luettu: 15.06.2020

Atlassian d. What is Jira used for? Luettavissa: <https://www.atlassian.com/software/jira/guides/use-cases/what-is-jira-used-for>. Luettu: 01.06.2020

BBC. Functions, procedures and modules. Luettavissa: <https://www.bbc.co.uk/bitesize/guides/z9hykqt/revision/4>. Luettu: 04.06.2020

Bootstrap. Migrating to v4. Luettavissa: <https://getbootstrap.com/docs/4.4/migration/>. Luettu: 05.06.2020

Bootstrap 3. Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. Luettavissa:

<https://getbootstrap.com/docs/3.3/>. Luettu: 01.06.2020

Cassidy, D. 2016. Size Matters in Agile. Luettavissa: <https://www.frontrowagile.com/blog/posts/72-size-matters-in-agile>. Luettu: 14.06.2020

CFWheels. API. Luettavissa: <https://api.cfwheels.org/v2.1>. Luettu: 28.05.2020

Chuparkoff, D. 2018. A New Introduction to Jira & Agile Project Management. Katsottavissa: <https://www.youtube.com/watch?v=TsG3OWTDAFY>. Luettu: 15.6.2020

Codecademy a. MVC: Model, View, Controller. Luettavissa: <https://www.codecademy.com/articles/mvc>. Luettu: 03.06.2020

Codecademy b. What is CRUD? Luettavissa: <https://www.codecademy.com/articles/what-is-crud>. Luettu: 08.06.2020

ComputerHope. Session. Luettavissa: <https://www.computerhope.com/jargon/s/session.htm>. Luettu: 12.07.2020

Coron, T. 2020. What is Sass? Your guide to this top CSS preprocessor. Luettavissa: <https://www.creativebloq.com/web-design/what-is-sass-111517618>. Luettu: 06.06.2020

Coyier, C. 2016. Poll Results: Popularity of CSS Preprocessors. Luettavissa: <https://css-tricks.com/poll-results-popularity-of-css-preprocessors/>. Luettu: 07.06.2020

Cyberdefinitions. What Does KIS Mean? Luettavissa: <https://www.cyberdefinitions.com/definitions/KIS.html> Luettu: 14.06.2020

Domantas, G. 2019. What is HTML? The Basics of Hypertext Markup Language Explained. Luettavissa: <https://www.hostinger.com/tutorials/what-is-html>. Luettu: 14.06.2020

GetBootstrap. How it works. Luettavissa: <https://getbootstrap.com/docs/4.0/layout/grid/>. Luettu: 16.06.2020

GitHub a. Getting Started with GitHub. Luettavissa: <https://docs.github.com/en/github/getting-started-with-github/fork-a-repo>. Luettu: 12.07.2020

GitHub b. zxing-js. Luettavissa: <https://github.com/zxing-js/library>. Luettu: 01.06.2020

Google Groups. CFWheels. Luettavissa: <https://groups.google.com/forum/#!forum/cfwheels>. Luettu: 28.06.2020

Fowler, T. 2020. What Is Ajax and What Is It Used For? Luettavissa: <https://career-karma.com/blog/what-is-ajax/>. Luettu: 06.09.2020

Francino, Y. How detailed should tasks within a user story be for agile teams? Luettavissa: <https://techbeacon.com/app-dev-testing/how-detailed-should-tasks-within-user-story-be-agile-teams>. Luettu: 14.06.2020

Helsinki. Scrum. Luettavissa: <https://kehmet.hel.fi/menetelmalaari/scrum/>. Luettu: 03.06.2020

Hietaniemi, J. 2019. Scrum pähkinän kuoressa. Luettavissa: <https://gofore.com/scrum-pahkinankuoressa/>. Luettu: 08.06.2020

IBM. Escape sequences. Luettavissa: https://www.ibm.com/support/knowledgecenter/SSLTBW_2.3.0/com.ibm.zos.v2r3.cbclx01/escape.htm. Luettu: 02.09.2020

Jyväskylän yliopisto. Ikuinen silmukka. Luettavissa: http://users.jyu.fi/~vesal/kursit/ohj2/moniste/html/m-10_13.htm. Luettu: 02.09.2020

Kareinen, A. Olio. Luettavissa: <https://www3.uef.fi/fi/web/anja.kareinen/olio-ohjelmointi>. Luettu: 06.06.2020

Kiiskinen, A. 2018. If it works, fix it! Luettavissa: <https://contribyte.fi/2018/05/08/if-it-works-fix-it/>. Luettu: 30.05.2020

King, T. 2017. CFWheels JS Confirm. Luettavissa: <https://www.forgebox.io/view/cfwheels-js-confirm>. Luettu: 09.08.2020

Kukhnavets, P. 2018. Why do We Need Scrum Epics in Software Development? Luettavissa: <https://hygger.io/blog/need-scrum-epics-software-development/>. Luettu: 21.06.2020

- Leeds, D. 2012. Is ColdFusion Object-Oriented Language. Luettavissa: <http://www.hit-thebits.com/2012/10/is-coldfusion-object-oriented-language.html>. Luettu: 31.05.2020
- Leppäniemi, P. Ketterät menetelmät: Scrum. Luettavissa: <http://ohjelmistotuotanto.panu-leppaniemi.com/agile-scrum/>. Luettu: 01.06.2020
- Lesscss. It's CSS, with just a little more. Luettavissa: <http://lesscss.org/>. Luettu: 06.06.2020
- Meteoriitti. Ketteryys haltuun: Scrum Pähkinänkuoressa. Luettavissa: <https://meteo-riitti.com/2013/06/06/ketteryys-haltuun-scrum-pahkinankuoressa/>. Luettu: 21.06.2020
- Microsoft a. Boolean Data Type. Luettavissa: <https://docs.microsoft.com/en-us/office/vba/language/reference/user-interface-help/boolean-data-type>. Luettu: 02.09.2020
- Microsoft b. SQL Server technical documentation. Luettavissa: <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>. Luettu: 29.05.2020
- Mozilla a. CSS: Cascading Style Sheets. Luettavissa: <https://developer.mozilla.org/en-US/docs/Web/CSS>. Luettu: 06.06.2020
- Mozilla b. Introduction to the DOM. Luettavissa: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction. Luettu: 19.07.2020
- Mozilla c. Window.SessionStorage. Luettavissa: <https://developer.mozilla.org/en-US/docs/Web/API/Window/sessionStorage>. Luettu: 12.07.2020
- Nadel, B. 2008. Learning ColdFusion 9: The Ternary Operator. Luettavissa: <https://www.bennadel.com/blog/1643-learning-coldfusion-9-the-ternary-operator.htm>. Luettu: 03.06.2020
- NeoKing 2018a. CFWheels Routing in 2.x - Part 1. Katsottavissa: <https://www.youtube.com/watch?v=BnPGApAvMVQ>. Katsottu: 05.06.2020
- NeoKing 2018b. CFWheels Routing in 2.x - Part 2. Katsottavissa: <https://www.youtube.com/watch?v=0CiGxJyJEIQ>. Katsottu: 05.06.2020

Packt Video. 2018. Creating and Running an Agile Project in JIRA: Epics, Stories, Bugs, and Tasks | packtpub.com. Katsottavissa:

<https://www.youtube.com/watch?v=LO8s6WM4Zo0>. Katsottu: 15.06.2020

Regexr. Luettavissa: <https://regexr.com/>. Luettu: 26.07.2020

Rouse, M. 2019. T-SQL (Transact-SQL). Luettavissa: <https://searchsqlserver.tech-target.com/definition/T-SQL>. Luettu: 31.05.2020

Shukla, A. 2013. Routing in MVC. Luettavissa: <https://www.codeproject.com/Tips/573454/Routing-in-MVC>. Luettu: 04.06.2020

SQLCourse. What is SQL? Luettavissa: <http://www.sqlcourse.com/intro.html>. Luettu: 14.06.2020

StackOverFlow. Why shouldn't data be modified on an HTTP GET request? Luettavissa: <https://stackoverflow.com/questions/705782/why-shouldnt-data-be-modified-on-an-http-get-request>. Luettu: 29.06.2020

Technopedia. Viewport. Luettavissa: <https://www.techopedia.com/definition/32644/viewport>. Luettu: 06.09.2020

Technopedia. 2017. What does Relational Database (RDB) mean? Luettavissa: <https://www.techopedia.com/definition/1234/relational-database-rdb>. Luettu: 29.05.2020

TechTerms, 2011. Newline. Luettavissa: <https://techterms.com/definition/newline>. Luettu: 02.09.2020

Tempus Dominus Bootstrap 4. Luettavissa: <https://tempusdominus.github.io/bootstrap-4/>. Luettu: 09.08.2020

TTL. Ajanhallinta. Luettavissa: <https://www.ttl.fi/tyontekija/tyoaika/ajanhallinta/>. Luettu: 30.06.2020

T1m0n. Air Datepicker. Luettavissa: <http://t1m0n.name/air-datepicker/docs/>. Luettu: 16.06.2020

Uusitalo, J. 2020a. Ajax controller routing problems - Upgraded from 1.4.5 to 2.1. Luettavissa: <https://groups.google.com/forum/#!topic/cfwheels/ILh9k8D3I94>. Luettu: 28.06.2020

Uusitalo, J. 2020b. isPatch() is not working. Luettavissa: <https://groups.google.com/forum/#!topic/cfwheels/mkZSjQ58AGI>. Luettu: 26.06.2020

Visualstudio. Code editing. Redefined. Luettavissa: <https://code.visualstudio.com/>. Luettu: 06.06.2020

Webopedia. What Is Query Language? Luettavissa: https://www.webopedia.com/TERM/Q/query_language.html. Luettu: 29.05.2020

W3 a. CSS Values and Units Module Lengths. Luettavissa: <https://www.w3.org/TR/2013/CR-css3-values-20130730/#font-relative-lengths>. Luettu: 05.06.2020

W3 b. Form submission. Luettavissa: <https://www.w3.org/TR/html4/interact/forms.html#h-17.13.1>. Luettu: 28.06.2020

6 Kuvat

| | |
|--|----|
| Kuva 1. Kaukolämpö Plus järjestelmän sidosryhmät..... | 7 |
| Kuva 2. Kaukolämpö Plus järjestelmän skanneri | 9 |
| Kuva 3. ITF-viivakoodin tarkastusnumeron laskentakaava..... | 10 |
| Kuva 4. Jira-sovelluksen Sprint-näkymä | 13 |
| Kuva 5. Regular Expression etsintäkaavan käyttöä Visual Studio Code:ssa | 15 |
| Kuva 6. Kaukolämpö Plus 7.0 järjestelmän navigaatiopalkki..... | 21 |
| Kuva 7. CFWheelsin URLFor-funktion käyttöä..... | 22 |
| Kuva 8. Kaukolämpö Plus 7.0 järjestelmän navigaatiopalkki monitasoisuudella..... | 25 |
| Kuva 9. Kolmannen kehitysjakson suunnittelua | 27 |
| Kuva 10. DeleteButtonTag-merkin käyttö ohjelmistokoodissa | 29 |
| Kuva 11. isGet ja isPost-funktiot CFWheelsin miscellaneous-tiedostossa..... | 35 |
| Kuva 12. Bootstrap 3 version tabit | 44 |
| Kuva 13. Bootstrap 3 version tabit mobiilinäkymässä | 44 |
| Kuva 14. Bootstrap 4.5 version tabit | 46 |
| Kuva 15. Bootstrap 4.5 version tabit mobiilinäkymässä | 46 |
| Kuva 16. Uuden navigaatiojärjestelmän funktiota..... | 48 |
| Kuva 17. HREF-attribuutin sisällön riisuminen a-merkistä..... | 66 |
| Kuva 18. BootstrapConfirm toiminnallisuuden aktivointi..... | 66 |
| Kuva 19. BootstrapConfirmin generoima varmistusikkuna käytännössä | 67 |

7 Liitteet

Liite 1. Lyhenteet

CFML (sanoista ColdFusion Markup Language)

CFScript (ColdFusion Scripting Language)

CFWheels (sanoista ColdFusion On Wheels)

SQL (sanoista Structured Query Language)

T-SQL (sanoista Transact Structured Query Language)

IDE (sanoista Integrated Development Environment)

KIS (sanoista Keep It Simple)

REM (sanoista root em)

LESS (sanoista Leaner Style Sheets)

SASS (sanoista Syntactically awesome style sheets)

CRUD (sanoista Create, Read, Update and Delete)

CLI (sanoista Command line interface)