

lida Nikupeteri

SYSTEEMISUUNNITTELIJAN PÄIVÄKIRJA

SYSTEEMISUUNNITTELIJAN PÄIVÄKIRJA

Iida Nikupeteri
Opinnäytetyö
Kevät 2020
Tietojenkäsittely
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittely, Web-sovelluskehittäjä

Tekijä: Iida Nikupeteri

Opinnäytetyön nimi: Systeemisuunnittelijan päiväkirja

Työn ohjaaja: Ritva Virkkala

Työn valmistumislukukausi ja -vuosi: Kevät 2020

Sivumäärä: 42

Opinnäytetyö on tutkielma, joka on kirjoitettu päiväkirjamuotoiseksi. Tämän opinnäytetyön aiheena on kuvata systeemisuunnittelijan työtehtäviä. Tutkielma sisältää työpaikan ja -tehtävien kuvauksen, päiväkirjamerkinnot kymmeneltä viikolta sekä pohdinnan. Päiväkirjamerkintöjen lisäksi jokaisen viikon jälkeen on laadittu viikkoanalyysi, jossa käydään läpi tarkemmin teoriaa lähteitä käyttäen. Toimeksiantajana on ruostumatonta terästä tuottava Outokumpu Stainless Oy, jonka Tornion toimipisteessä työskentelin kesätyöntekijänä.

Tämän päiväkirjamuotoisen opinnäytetyön tavoitteena on kuvailla systeemisuunnittelijan työtehtäviä tehdasympäristössä. Lähteinä viikkoanalyseissä käytän toimeksiantajan sisäistä intraa, suullisia tiedoksiantoja sekä artikkeleita ja manuaaleja internetistä.

Asiasanat: Tietojenkäsittely, päiväkirja, systeemisuunnittelija

ABSTRACT

Oulu University of Applied Sciences
Business Information Systems, Web-application development

Author: Iida Nikupeteri

Title of thesis: Diary of a system designer

Supervisor: Ritva Virkkala

Term and year when the thesis was submitted: Spring 2020 Number of pages: 42

This thesis is a study that is written in diary format. The subject of this thesis is to narrate daily work life of a system designer. Study includes description of workplace and work assignments, diary entries for ten weeks and a reflection. In addition of the diary entries there is also a weekly analysis where the theory is examined in more detail based on sources. My employer is Outokumpu Stainless Oy, which is a company that produces stainless steel.

The goal of this thesis is to describe work life as a system designer in factory settings. As sources I mainly use company's own intranet, spoken declaration and internet articles and manuals.

Keywords: Information technology, diary, system designer

SISÄLLYS

1	JOHDANTO	6
2	NYKYTILANTEEN KUVAUS	7
2.1	Työpaikka	7
2.2	Oman nykyisen työn analyysi	8
2.2.1	Työtehtävien esittely	8
2.2.2	Arviointi	9
2.2.3	Kehittyminen	9
2.3	Sidosryhmät työpaikalla.....	10
2.4	Vuorovaikutustaidot työpaikalla	11
3	KÄSITTEET	12
4	PÄIVÄKIRJARAPORTOINTI	14
4.1	Viikko 1 – Tuulimittari	14
4.2	Viikko 2 – Kumiankka virheenjäljityksen apukeinona	19
4.3	Viikko 3 – Relaatiotietokantojen käsittely.....	21
4.4	Viikko 4 – AOD	24
4.5	Viikko 5 – JpGraph	26
4.6	Viikko 6 – Saavutettavuus värien avulla	27
4.7	Viikko 7 – Järjestelmäkaavio	29
4.8	Viikko 8 – ODBC	32
4.9	Viikko 9 – Työturvallisuus.....	35
4.10	Viikko 10 – Tietoturvallisuus.....	37
5	POHDINTA.....	40
	LÄHTEET.....	41

1 JOHDANTO

Tämän opinnäytetyön tavoitteena on kuvailla terästehtaan systeemisuunnittelijan työtehtäviä 10 viikon ajalta. Raportoinnin aikaväli on 17.6.2019-23.8.2019. Jokaisen viikon jälkeen analysoidaan projekteissa ilmenneitä ongelmia, ratkaisuja tai käytettyjä tekniikoita sekä tarkastellaan omaa oppimista ja kehittymistä. Yrityssalaisuuksien piiriin kuuluvat yksityiskohdat on jätetty pois tai sensuroitu salassapitovelvollisuuden vuoksi.

Opinnäytetyön toimeksiantajana on Outokumpu Stainless Oy, jonka Tornion toimipisteessä työskentelin kesän 2019. Organisaatio on minulle jo ennestään tuttu, sillä olen ollut siellä kesäharjoittelijana useampana kesänä. Vuosina 2014 ja 2015 olin kesätöissä tuotannon puolella kylmävalssaamalla prosessinohitajana. Kesällä 2018 tulin L2-automaatio-tiimiin suorittamaan ammatillisen harjoitteluni ja tänä kesänä pääsin uudestaan kesätöihin samaan tiimiin, jonka aikana suoritin tämän opinnäytetyön.

Systeemisuunnittelijan työtehtävissä tarvittavia taitoja ovat laajojen tietokantojen käsittely, web-ohjelmointitekniikat, UI-suunnittelu ja projektityöskentely. Koska teemme projekteja lähinnä organisaation sisälle, täytyy meidän ymmärtää myös tuotannon eri vaiheet teräksen valmistamisessa. Tehdasalueella työskenteleminen vaatii myös turvallisuuden tarkkaa noudattamista, standardien ymmärtämistä sekä tilannetajua.

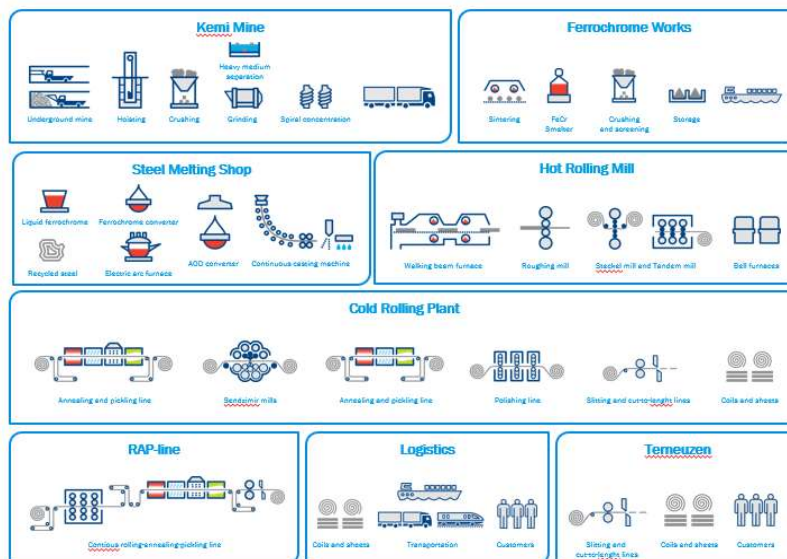
2 NYKYTILANTEEN KUVAUS

2.1 Työpaikka

Outokumpu on yksi johtavimpia ruostumattoman teräksen tuottajia maailmalla. Outokummulla on tehtaita Suomessa, Saksassa, Ruotsissa, Iso-Britanniassa, Yhdysvalloissa sekä Meksikossa. Myynti- ja palvelukeskukset ovat levittyneet ympäri maailmaa. (Outokumpu Oyj, 2019, viitattu 27.8.2019.)

Työpisteeni sijaitsee Tornion tehtaan alueella. Tornion tehdas on täysin integroitu, eli siihen kuuluu sulatto, kuumavalssaamo ja kylmävalssaamo. Tehtaan alueella toimii myös ferrokromitehdas sekä satama. Tornion tehtaan läheisyydessä on myös Kemissä sijaitseva kromikaivos, josta ruostumattoman teräksen valmistukseen tarvittavaa kromia saadaan (kuvio 1). (Outokumpu Oyj, 2019, viitattu 27.8.2019.)

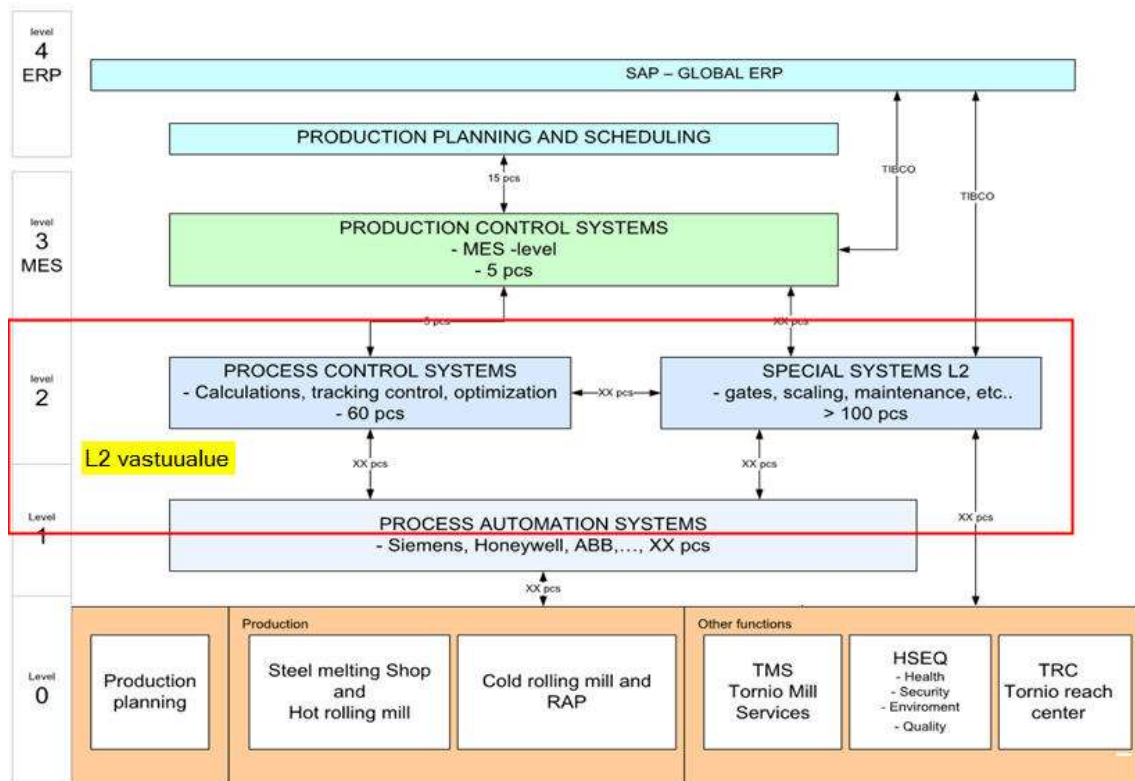
From chrome ore to stainless steel



8/27/2019

Kuvio1 Outokummun ruostumattoman teräksen tuotannon eri vaiheita. (Outokummun sisäinen intra, 2019, viitattu 29.8.2019)

Organisaatiossa kuulun tehdaspalveluissa toimivaan automaatio L2-tiimiin (kuvio 2), jonka vastuualueina ovat 3- ja 4-tasojen alapuoliset sovellukset ja tietojärjestelmät Outokummun Tornion tehtailla. Tiimin vastuualueisiin kuuluu järjestelmien käytettävyyden ylläpito, palvelimien päivitys ja huoltotyön sekä tuotannon työasemat, uudet projektit ja muutostyöt, terässulaton, kuuma- ja kylmävalssaamon raportointi. Tiimi vastaa yli 160 järjestelmästä, joihin kuuluu muun muassa erilaisia prosessiohjaus-, kunnossapito- ja erikoisjärjestelmiä.



Kuvio 2 Automaatio L2-tiimin vastuualueet kaaviossa (Outokummun sisäinen intra, 2019, viitattu 29.8.2019)

2.2 Oman nykyisen työn analyysi

2.2.1 Työtehtävien esittely

Työtehtäviini kuuluu ohjelmoida tarvittavia raportteja, jotka käsittelevät tietokannoista saatavia tietoja luettavaan muotoon. Raportit voivat sisältää listoja ja kuvaajia eri kriteereiden mukaan. Raportit voivat esimerkiksi toimia työvälineenä prosessilinjilla tai jäsenellä prosessien eri tietoja insinööreille, jotka tarvitsevat dataa kehitysprojekteihinsa. Raportit ohjelmoidaan osastollamme pääasiassa PHP-, HTML-, CSS-, SQL- ja JavaScript-tekniikoita käyttäen. Toisena tehtävänäni oli laatia osastollemme päivitetty järjestelmäkuvakaavio, joka tulee tehdä Microsoftin Visio-ohjelmaa

käyttäen ja tulostaa valmis kuva A0-kokoisena. Kaavioon tulee molempien sulattojen järjestelmät, joista 2-taso on vastuussa.

Web-ohjelmoinnin sekä tietokantojen käsittelyn lisäksi työtehtävissä täytyy osata noudattaa organisaation standardeja, tiukkoja turvallisuusmääräyksiä sekä ymmärtää tietoturva. Näiden lisäksi kommunikointitaidot ovat tärkeitä, sillä raportin tilanneet henkilöt eivät välttämättä osaa kertoa oikein minkälaista tietoa he haluavat, jolloin pitää osata lukea ”rivien välistä” ja kysyä tarkentavia kysymyksiä.

2.2.2 Arviointi

Suoriuduin minulle annetuista työtehtävistä hyvin, mutta joissain asioissa tarvitsin vielä kokeneempien työntekijöiden apua. Esimerkiksi monimutkaiset SQL-haut piti joko tarkistaa tai antaa vakituisten tehdä, sillä virheet saattavat olla kohtalokkaita, kun käsitellään tuotannossa kiinni olevia tietokantoja. Tehtävissä, missä ei tarvinnut tietoja tuotannon monimutkaisemmista asioista, selviydyin hyvinkin itsenäisesti, sillä viime vuoden työkokemus ja aiempi osaaminen tukivat niiden suorittamista. Näitä tehtäviä olivat ne, joissa tarvittiin PHP-, HTML- ja CSS-tekniikoita sekä muita web-ohjelmointiin liittyviä osa-alueita. Osaamiseni vuoksi kykenin avustamaan ja opastamaan toista harjoittelijaa, joka oli samoissa työtehtävissä kanssani.

2.2.3 Kehittyminen

Opinnäytetyön aikana kehityin eniten PHP:n ja tietokantojen käsittelyn parissa. Myös itsenäinen työskentely ja dokumentointi olivat alueita, joissa huomasin kehitystä esimerkiksi viime kesään verrattuna. Tietojenhaku nopeutui viime vuoteen verrattuna, sillä osasin etsiä oikeilta sivuilta ja oikeilla hakusanoilla. Osasin myös tehokkaammin käydä aineistoa läpi, jonka ansiosta löysin useampia lähteitä ja ratkaisuja ongelmille ja pystyin niistä valitsemaan sopivimman. Perusfunktioissa en enää tarvinnut katsoa mallia tai kysyä apua.

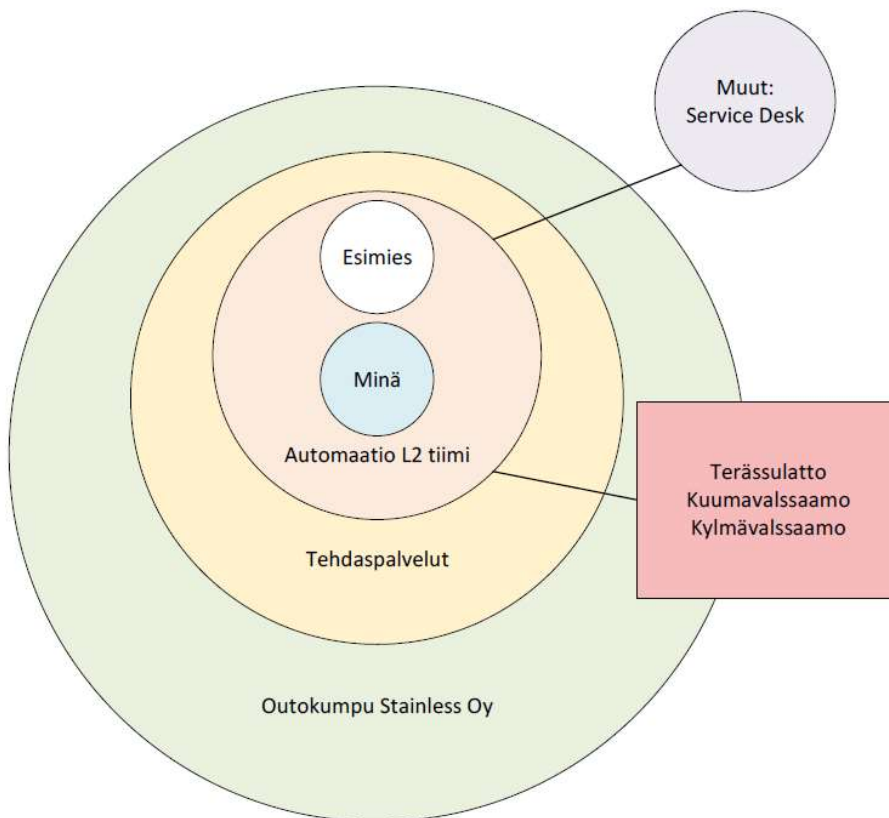
Ohjelmointikieliä oppiessani olen huomannut, että uusien tekniikoiden omaksuminen on nyt nopeampaa ja vaivattomampaa kuin ennen, sillä perusteet alkavat olla hyvin hallussa. Olen nyt mielestäni junior-kehittäjän tasolla web-ohjelmoinnissa ja valmis astumaan kunnolla työelämään

oppimaan lisää. Suurin oppiminen on tapahtunut juurikin työtehtävien kautta, jossa ilmenneitä ongelmia ja niiden ratkaisuja on voinut soveltaa muihinkin työtehtäviin ohjelmoinnissa.

Jatkossa minun tulee panostaa omiin taitoihini luottamista, ajankäytön hallintaan sekä dokumentointitaitojen kehittämiseen. Unohdan helposti kirjata ylös perusteellisesti mitä olen tehnyt, ja jos en tee niin unohdan herkästi yksityiskohdat. Jatkossa aion tehdä työtehtävistäni tarkempia muistiinpanoja. Projektityöskentelytekniikoita en päässyt opinnäytetyötä tehdessäni harjoittamaan juurikaan, joten sen osalta on vielä paljon opittavaa käytännössä.

2.3 Sidosryhmät työpaikalla

Sidosryhmä koostui tiimistäni, tehdaspalveluista, tuotanto-osastoista, käyttö-, kehitys- ja turvallisuus-osastoista sekä It-tuesta eli Service Desk:sta (kuvio 3).



Kuvio 3 Sidosryhmät työpaikalla

2.4 Vuorovaikutustaidot työpaikalla

Osastollamme oli viikoittainen tiimipalaveri, jossa käytiin läpi tiimiläisten työtehtävien tilanteet ja ajankohtaiset asiat. Kaikilla työntekijöillä on oma työhuone, jossa pystyi käymään kysymässä neuvoja tai apua, jos heillä ei ollut kiireitä. Kommunikointi tapahtui sen lisäksi Skypen, sähköpostin tai työpuheluiden kautta.

Enimmäkseen kielenä käytettiin suomea, mutta englannin kielen taito tuli tarpeeseen, sillä IT-tuessa ei ollut aina suomenkielistä tukea käytettävissä. Osastollamme kaikki puhuivat suomea, mutta koska itse organisaatio on monikansallinen, oli hyvä olla valmis kommunikoimaan myös englanniksi. Ohjelmoinnissa koodin kommentoiminen tapahtui pääasiassa englanniksi.

3 KÄSITTEET

AOD (Argon Oxygen Decarburization): Menetelmä, joka mahdollistaa alhaisen hiilipitoisuuden saamisen ruostumattomaan teräkseen ilman, että kromi hapettuu runsaasti.

AOD-konvertteri: Konvertteri, jossa valmistetaan terästä AOD-menetelmällä ja johon prosessikaasut johdetaan alaosassa olevien suuttimien läpi.

CSS (Cascading Style Sheet): Ohjelmointikieli, jolla määritellään web-dokumenttien ulkoasun muotoilua, kuten fontteja, värejä ja elementtien asettelua.

Foreach: Silmukka käy jokaisen rivin/objektin ja tekee kaikille saman mikä silmukkaan on määritetty.

Funktio: Koodin pätkä, joka suorittaa ennalta määritellyn toiminnon ja jota voidaan kutsua myös muualla koodissa ilman, että sen tarvii kirjoittaa uudelleen.

HTML (Hyper Text Markup Language): Ohjelmointikieli www-dokumenttien määrittelyyn.

Javascript: Ohjelmointikieli, jota käytetään dynaamisen sisällön lisäämiseen verkkosivulle.

Jpgraph: Olioperustainen kaaviokirjasto PHP5:lle.

Jquery: JavaScript-kirjasto.

JTSU: Terässulatto

KUVA: Kuumavalssaamo

KYVA: Kylmävalssaamo

Luokka: Luokka antaa määrytykset olion rakenteelle ja käyttäytymiselle.

Luokkakirjasto: Kokoelma valmiiksi ohjelmoituja luokkia.

Olio: Luokan ilmentymä.

Olio-ohjelmointi: Koostuu kokoelmasta olioita, jotka toimivat yhteistyössä keskenään.

Oracle Database: Oracle Corporationin kehittämä relaatiotietokantojen hallintajärjestelmä.

Merkkijono (String): Ohjelmointikielissä käytetty tietotyyppi, joka on jono peräkkäisiä merkkejä.

Microsoft Visual Studio: Microsoftin ohjelmointikehitysympäristö.

PHP (Hypertext Preprocessor): Avoimen lähdekoodin ohjelmointikieli dynaamisten ja interaktiivisten web-sivustojen tekemiseen.

PDF (Portable Document Format): Ohjelmistosta, laitteistosta ja käyttöjärjestelmistä riippumaton tiedostomuoto.

Responsiivisuus: Sivusto mukautuu käytettävän laitteen mukaan.

SQL (Structured Query Language): Kyselykieli, jolla voidaan tehdä erilaisia hakuja relaatiotietokantoihin.

SQL Server: Microsoftin relaatiotietokantojen hallintajärjestelmä.

VKU (Valokaariuuni): Lämpö saadaan sähköenergiasta uunin katon läpi tulevien grafiittielektrodien ja panoksen välille syntyvistä valokaarista.

Toad for Oracle: Tietokantojen hallintajärjestelmä

Tooltip: Selittävä teksti, joka ilmestyy, kun hiiren osoittimen vie elementin kohdalle.

ZedGraph: C#-kielellä ohjelmoitu luokkakirjasto, käyttäjä- ja web-käsittelijä .NET-kirjastolle. Pääasiassa erilaisten kaavioiden tekemistä varten.

4 PÄIVÄKIRJARAPORTOINTI

4.1 Viikko 1 – Tuulimittari

Maanantai 17.6.2019

Viikko alkoi viime vuonna aloitetun häkäkartta-raportin tuulimittaria tehtäessä. Tuulimittarin on tarkoitus hakea ilmansuunta sekä tuulen nopeus tietokannasta ja näyttää tiedot www-sivulla toimivassa reaaliaikaisessa häkävaroituskartassa. Löysin tähän tarkoitukseen sopivan valmiin CSS-koodin, jolla oli tehty kompassin muotoinen tuulimittari (kuvio 4) (Melcher S, 2015, viitattu 17.6.2019). Mittari näyttää tekstin lisäksi myös nuolella tuulen suunnan.

Toteutin mallin mukaisen ratkaisun, mutta siihen piti itse soveltaa tietokannasta haku ja varmistaa, että se päivittää tietoja automaattisesti. Malli löytyi codepen.com-sivustolta. Käänsin myös tekstit suomeksi sekä hienosäädin koordinaatteja sekä raja-arvoja raporttiin soveltuvaksi.



Kuvio 4 Tuulimittarin ulkoasu

Jatkoimme toisen kesätyöntekijän kanssa jo aikaisemmin tehtäväksi saatua kuumavalssaamon romutusraporttia, joka hakee tarvittavat tiedot tietokannasta ja generoi PHP:lla ohjelmoidun www-sivun PDF:ksi.

Kerkesin myös tehdä elektrodivaurio-raporttia eteenpäin. Se koostuu kahdesta osasta, joista ensimmäinen osa hakee tietokannasta tuoreimman rivin ja ajastetun toiminnan avulla se lähettää

automaattisesti generoidun PDF-tiedoston sähköpostilla valituille vastaanottajille. Toinen osa on kooste, johon kerätään esimerkiksi kuukauden ajalta kaikki tehdyt raportit ja ne näytetään www-sivuna, joka voidaan joko tulostaa tai ladata ja viedä Exceeliin. Aikaisemmalla viikolla saimme toisen kesätyöntekijän kanssa valmiiksi sivun pohjan sekä SQL-haut, mutta ongelmaksi nousi raporttimoottorin ajastettu toiminto, jota käytämme raportissa. Sivun näyttää itsessään hyvältä ja toimii, mutta sähköpostiin lähetettävä PDF-kuva vaatii vielä säätämistä, jotta siitä saadaan lukuystävällisempi. Tällä hetkellä sivun asettelu ei toimi sekä tekstit ovat liian tiiviit ja pienet, eikä niistä saa tarpeeksi hyvin selvää.

Iltapäivällä oli viikoittainen tiimipalaveri, jossa käytiin osastollemme kuuluvia asioita. Myös meidän kesätyöntekijöiden työstä käytiin läpi.

Tiistai 18.6.2019

Tuulimittari toimii kuten pitääkin, mutta koska tuulitiedot päivittyvät tietokantaan viiveellä, ei sitä voi käyttää häkäkartta-raportissa. Häkäkarttasivulla pitäisi näkyä reaaliajassa päivittyvät tuulitiedot, joten tietokanta ei sovi sen tarpeisiin.

Kuumavalssaamon romutusraportti saatiin melkein valmiiksi. Raportissa ilmennyt ongelma saatiin korjattua foreach-silmukalla, jonka avulla saatiin generoitua useampi PDF-sivu jokaiselle romutusraportille erikseen. Muokkasin SQL-hakua hakemaan aina edellinen tehdasvuorokausi, joka siis alkaa aamukuudelta ja päättyy seuraavana aamuna kello kuudelta. Seuraavaksi edessä on vielä automaattinen sähköpostin lähetys, jonka pitää lähteä joka päivä kello 05:59 ja pitää myös sovittaa raportti raporttimoottoriin sopivaksi.

Elektrodivaurioraportti eteni sen verran, että optimoin haun moottoriin sopivaksi. Se olikin hieman vaikeampaa kuin aluksi ajattelin, mutta sen ansiosta opin enemmän raporttimoottorin ominaisuuksista, kun jouduin kaivelemaan viime kesän tekemiäni raportteja ja tutkimaan myös itse moottorin rakennetta. Vanhaa muistellessa ja uutta oppiessa sain sitten raportin rakenteen ja toiminnalliset osat yhteneväksi muiden raporttimoottorin raporttien kanssa.

Keskiviikko 19.6.2019

Sain poistettua ylimääräisen taulukon elektrodivaurioraportin päätaulukon alta, johon tulostui raportin run-ID sekä ylimääräiset aikarajat. Aluksi en löytänyt koodista syytä, miksi kyseinen taulukko tulostui varsinaisen päätaulukon alle, mutta tarkemmin tutkittuani koodia löysin vihdoin syyn ongelmaan. Ylimääräinen taulukko oli ollut osa esimerkkiraporttia, jota käytin raportin pohjana, sillä siinä oli tarvittavat funktiot ja luokat. Tämän ansiosta opin kuitenkin enemmän raporttien ja niiden ominaisuuksien rakenteesta.

Kuumavalssaamon romutusraportin kanssa ilmeni hieman ongelmia, kun raporttimootorin ajastettu sähköpostitoiminto ei hyväksynytään jo valmiiksi muunnettua PDF-sivua, joten jouduimme ottamaan hieman taka-askelia sen suhteen. Raporttimootorin sähköpostitoiminto muuntaa html-sivun jo itse PDF:ksi, joten päätimme muuttaa raportin runkoa ja muokata ulkoasua vastaamaan suunniteltua.

Torstai 20.6.2019

Elektrodivaurioraportin kanssa ilmeni ongelmia, kun SQL-lauseeseen ei saanut toimimaan muuttujaa, joka hakisi päivämäärän kahden valitun päivämäärän väliltä. PHP:lla tapahtuva oliohjelmointi ei ole vielä täysin hallinnassa, mutta tämän raportin myötä olen oppinut siitä enemmän. Löysin tähän ongelmaan vaihtoehtoisen ratkaisun, joka toimii, jos raportti halutaan lähettää esimerkiksi joka kuukausi kuun ensimmäinen päivä, jolloin lause hakee edellisen kuukauden ensimmäisestä päivästä kuun viimeiseen päivään ja palauttaa taulut niiden väliltä. Tässä tavassa käytettiin SQL-hakuun DATEPART- ja DATEADD-syntakseja (Kuvio 1.6) Yritän tehdä vielä toimivan oliopohjaisen PHP-ratkaisun, mutta ainakin on jotain toimivaa valmiina.

Kuumavalssaamon romutusraportin ulkoasun muuttaminen toimii vain, jos style-attribuutit upottaa suoraan elementtien sisään html:n puolella. Ulkoinen CSS-tiedosto tai "<head>"-tunnisteen sisällä toimivat attribuutit eivät toimineet haluamallamme tavalla, joten jouduimme syöttämään manuaalisesti kaikki tarvittavat tyyliattribuutit. Sivuston kokoa täytyy vielä muuttaa staattiseksi, jotta elementit pysyvät koko ajan samoilla paikoilla, vaikka tulisi useampi PDF-sivu, sillä koodissa oleva silmukka saattaa vaikuttaa elementtien sijoitteluun.

Perjantai 21.6.2019

Juhannusaatto, ei työpäivää.

Viikkoanalyysi – Tuulimittari

Tämä viikko oli projektien suhteen monipuolinen, kun työn alla on nyt monta raporttia yhtä aikaa, jotka ovat kaikki eri vaiheissa. Viikko sisälsi Oraclen ja Microsoftin SQL-lauseiden pyörittelyä ja tietokantojen tutkimista, oliopohjaisen PHP-ohjelmoinnin opiskelua ja raporttimoottorin ajastustoimintoon tutustumista. Tuli kerrattua vanhaa ja opeteltua myös ihan uusia asioita.

Vaikka tuulimittaria ei käytettykään raportissa, opin silti uutta CSS-, PHP- ja jQuery-ohjelmoinnista sekä tietokantojen käytöstä niiden ominaisuuksien kanssa.

```
while (oci_fetch($result))
{
    $ilmansuunta = oci_result($result, 'WIND_DIRECTION');
    echo "Ilmansuunnan asteet : " . $ilmansuunta ;

    $nopeus = oci_result($result, 'WIND_SPEED');

    //Jos ilmansuunta on pohjoinen
    if ($ilmansuunta < '23' && $ilmansuunta >> '338'){
        $lyhenne = "n";
        $suunta = "Pohjoinen";
    }
    //Jos ilmansuunta on itä
    elseif ($ilmansuunta > '68' && $ilmansuunta < '113'){
        $lyhenne = "e";
        $suunta = "Itä";
    }
    //Jos ilmansuunta on etelä
    elseif ($ilmansuunta > '158' && $ilmansuunta < '203'){
        $lyhenne = "s";
        $suunta = "Etelä";
    }
    //Jos ilmansuunta on länsi
    elseif ($ilmansuunta > '248' && $ilmansuunta < '293'){
        $lyhenne = "w";
        $suunta = "Länsi";
    }
    //Jos ilmansuunta on koillinen
    elseif ($ilmansuunta > '23' && $ilmansuunta < '68'){
        $lyhenne = "ne";
        $suunta = "Koillinen";
    }
    //Jos ilmansuunta on kaakko
    elseif ($ilmansuunta > '113' && $ilmansuunta < '158'){
        $lyhenne = "se";
        $suunta = "Kaakko";
    }
    //Jos ilmansuunta on lounas
    elseif ($ilmansuunta > '203' && $ilmansuunta < '247'){
        $lyhenne = "su";
        $suunta = "Lounas";
    }
    //Jos ilmansuunta on luode
    elseif ($ilmansuunta > '293' && $ilmansuunta < '338'){
        $lyhenne = "sw";
        $suunta = "Luode";
    }
}
```

Kuvio 5 Tuulimittarin ehtolauseita PHP-kielillä

```

<div class="content">
  <div class="compass">
    <div class="direction">
      <!--Tähän tulee ilmansuunta ja vauhti--><p>
        <?php
          echo $suunta;
        ?>
        <span>
          <?php
            echo $nopeus;
          ?> m/s
        </span>
      </p>
    </div>
    <!--Tähän tulee suunta nuolen muodossa-->
    <?php
      echo "<div class='arrow ";
      echo $lyhenne;
      echo "'>";

      echo "</div>";
    ?>
  </div>
</div>

```

Kuvio 6 Tuulimittarin HTML-koodia

Kompassin nuoli on asetettu "absolute"-position, kun taas sen isäntäelementti on asetettu "relative"-position. "Absolute" tarkoittaa sitä, että elementti voidaan määrittellä juuri siihen kohtaan mihin se halutaan. Jos isäntäelementille ei määritellä arvoa "relative", periytyy asettelu suoraan html-sivuun itseensä. (Boskovic T, 2019, viitattu 3.3.2020.) Tämän ansiosta nuoli pysyy oikealla paikallaan, vaikka kuvaa skaalattaisiin isommaksi.

CSS:n transform-ominaisuus mahdollistaa elementille esimerkiksi kiertämisen, skaalautumisen tai animoinnin (W3Schools. 2020, viitattu 3.3.2020). Tässä raportissa transform-ominaisuutta käytettiin nuolen oikean position määrittämiseen. Nuolen asento riippuu siitä, mikä ilmansuunta on kyseessä. Jos ilmansuunta olisi koillinen, transform-arvoksi tulisi "rotate(45deg)", eli nuolen kulmaksi määräytyy 45 astetta (kuvio 7).

```

.compass .arrow.nne {
  transform: rotate(22.5deg);
}
.compass .arrow.ne {
  transform: rotate(45deg);
}
.compass .arrow.ene {
  transform: rotate(67.5deg);
}

```

Kuvio 7 Tuulimittarin nuolten suunnan määrittelyä

4.2 Viikko 2 – Kumiankka virheenjäljityksen apukeinona

Maanantai 24.6.2019

Aamu alkoi kuumavalssaamon romutusraportin ajastustoiminnon lähettämän sähköpostin tutkimisella, joka oli tullut oikeilla tiedoilla. Muotoilu PDF-kuvassa ei silti ollut vielä täydellinen, mutta oikein tulostuva data kertoo, että olemme oikeilla jäljillä.

Elektrodivaurioraportin ajastettu sähköposti tuli oikean näköisenä ja muotoilukin oli kohdallaan, vaikka tekstikenttiä olikin paljon. Asettelu sivulla on leveä, mutta lyhyt, joka johti tekstin näkymiseen liian pienenä. PDF-tiedoston muotoilu on vaikeaa, joka johtunee siitä, että raportointiympäristössä käytetään jotain valmista kirjastoa tai vastaavaa, emmekä uskalla mennä liikaa koskemaan mihinkään, ettemme vahingossa sotke aikaisempia ajastettuja raporteja.

Kehittelin elektrodivaurioraporttiin SQL-hakuun osan, joka hakee edellisen päivän sekä edellisen kuukauden tiedot tietokannasta.

```
WHERE DATEPART(dd, [taulun_nimi]) = DATEPART(dd, DATEADD(dd, -1, getdate())) AND DATEPART(yyyy, [taulun_nimi]) = DATEPART(yyyy, DATEADD(dd, -1, getdate())) .
```

Kuvio 8 Haku hakee edellisen vuorokauden tiedot.

```
WHERE DATEPART(m, [EDA_DATE]) = DATEPART(m, DATEADD(m, -1, getdate())) AND DATEPART(yyyy, [EDA_DATE]) = DATEPART(yyyy, DATEADD(m, -1, getdate()))".
```

Kuvio 9 Haku hakee edellisen kuukauden tiedot.

Muokkasin myös elektrodivaurioraportin ulkoasua helpommin luettavaan muotoon. Samasta raportista voisi tehdä kaksi identtistä versiota, joista toinen hakee edellisen kuukauden tiedon ja toinen edellisen vuorokauden.

Tiistai 26.5.2019

Aamupäivällä muokkasin elektrodivaurioraportin ulkoasua CSS-muotoiluilla. Ulkoasu alkaa olla melkein valmis ja vaatii enää pientä hiomista. Iltpäivällä aloitimme RRR-raportin korjausta, johon pitää tehdä lf-lause, jolla tarkistetaan, että saman sulatusnumeron omaava virhe ei laske sen painoa kahdesti, joka vaikuttaa muihin raportin laskutoimituksiin, joka johtaa siihen, että tulos ei ole tarpeeksi luotettava.

Keskiviikko 26.6.2019

Sain uudeksi projektiksi tehdä raportin AOD:n allokoimattomasta ajasta, johon voin ottaa mallia vastaavasta valokaariuunille tehdystä raportista. On mukavaa saada projekti, jonka pääsen tekemään melkein alusta loppuun asti itse. Raporttiin tulee alkuaika, loppuaika, kesto, luokka, koodi, vuoron koodi, tonnit ja raportin pääasia; allokoimaton aika. Loppuun tulee yhteenveto, jossa lasketaan kalenteriaika, tehojen päällä oloajan yhteenlaskettu aika, häiriöt, allokoimaton aika ja sen osuus prosenteissa. Käytän tämän päivän valokaariuunille tehdyn raportin tutkimisessa, jotta osaan hyödyntää samaa tekniikkaa tähän. Minulle projektin antanut työkaveri hoitaa tarkemmin vaativan SQL-haun ja minä hoidan loput.

Torstai 27.6.2019

Sain lisätietoja AOD2-raporttiin, johon pitää rakentaa logiikka SQL-lauseeseen, johon yhdistetään tietoja useammasta taulusta UNION-lauseella. Ensiksi pitää rakentaa SQL-lause, joka löytää oikeat tiedot ja tarkistaa, että se etsii oikean määrän dataa käyttäjän asettaman aikaikkunan mukaan. Jos alkanut aika alkaa ennen käyttäjän asettamaa aikaa, mutta päättyy aikaikkunan sisälle, se tulisi laskea osittain mukaan muuten paitsi loppulaskennassa. Näin laskut saadaan täsmäämään eikä tule vääristynyttä dataa.

Perjantai 28.6.2019

Aamulla tuli kommentteja romutusraportista, joissa oli pari korjausehdotusta. Toinen virheistä oli suhteellisen yksinkertainen logiikkavirhe, eli eräessä If-lauseessa oli puutoksia. Toinen virhe oli raportointiympäristöstä johtuva ominaisuus, joka tulee ensin käydä läpi jonkun kanssa, joka tietää siitä enemmän. Raportin ”tilaaja” halusi myös sähköpostiin tulevien PDF-raporttien lisäksi selainsivun, josta tarkastella raportteja kuukausittain. Tämän sivun kasasimme saman päivän aikana, sillä meillä oli oikeastaan tähän jo koodit valmiina, kun olimme aiemmin luulleet, että tällainen pitääkin tehdä.

Viikkoanalyysi – Kumiankka virheenjäljityksen apukeinona

Viikko sisälsi raportin viimeistelyn ja testivaiheessa löytyneiden vihreiden korjaamista, uuden projektin aloittamisen sekä vanhojen raporttien tutkimista. Opin taas lisää käyttämästämme raportointiympäristöstä sekä tietokantayhteyden muodostamisesta. Viime vuonna onnistuin kyllä

muodostamaan yhteyden, kun katsoin mallia muista raporteista, mutta nyt tuntuu, että ymmärrän jo tekniikkaakin sen takana.

Kuumavalssaamolle menevää raporttia sekä elektrodivaurioraporttia työstin pääasiassa yhdessä toisen kesätyöntekijän kanssa. Projektit etenivät hyvin, kun sai ratkoa ongelmia yhdessä ja hyödyntää molempien tietotaitoa ja osaamista. Sovellamme enimmäkseen ”rubberduck-debugging” :n eli kumiankka debuggauksen sekä pariohjelmoinnin metodeja työskentelyssämme.

Ohjelmoitaessa tulee ottaa huomioon se, että tietokoneen logiikka on suoraviivaista, eikä siinä ole samanlaisia vivahteita kuin ihmisen logiikassa. Kun ohjelmoitaessa tulee eteen ongelma, jota ei millään meinaa saada ratkottua, voi kokeilla kumiankka debuggausta. Kumiankka debuggaus on käytännössä sitä, että ongelma selitetään yksityiskohtaisesti joko objektille tai henkilölle, jonka tarkoituksena on saada itse huomattua virhe koodissa. (Hayes D, 2019, viitattu 24.7.2019.)

Yhdessä ohjelmoitaessa tulee huomattua toisen virheet helpommin kuin omansa, jolloin projekti voi edetä nopeampaa kuin yksin tehdessä. Omiin huolimattomuusvirheisiin tulee helposti ”sokeaksi”, mutta toisen tekemät virheet tunnistaa herkemmin. Henkilökohtaisesti opin paremmin silloin, kun työskentelen yksin. Kun joutuu itse etsimään tietoa ja ratkaisuja ongelmiin, tulee opittua enemmän ja syvemmin asioita, kun asiaan paneutuu kunnolla ilman häiriötekijöitä. Myös paineet suoriutua ovat suuremmat, kun vastuuta ei jaa toisen kanssa vaan kantaa sen kokonaan yksin.

4.3 Viikko 3 – Relaatietietokantojen käsittely

Maanantai 1.7.2019

Viikko alkoi allokoiaton aikaraportin SQL-lauseiden säätämällä, monitorin vian etsimisellä sekä viikoittaisella tiimipalaverilla. Sain viikonloppuna huonoja uutisia, joiden vuoksi oli hankala keskittyä työhömiin täysin, mutta yritin silti tehdä parhaani. Tavoitteeksi tälle viikolle muodostui AOD2:sen allokoiaton aikaraportin kehittäminen siihen pisteeseen, että suurin osa siitä olisi valmiina perjantaihin mennessä, sillä työtehtävän minulle antanut tiimiläinen jää kesälomalle silloin.

Myös monitorin responsiivisuus-ongelman selvittäminen jää tauolle, sillä myös tämän projektin antanut tiimiläinen jää lomalle ja tarvitsen aiheesta lisäselvityksiä. Alan keskittymään myös järjestelmäkuvien päivittämiseen heti ensi viikosta lähtien.

Tiistai 2.7.2019

AOD2:n allokoimaton aikaraporttiin sain työkaverin viimeistelemän uuden SQL-haun. Hän myös opetti minulle, miten ja miksi käytetään UNION-syntaksia ja milloin sitä kannattaa käyttää UNION ALL- ja DISTINCT-syntaksin sijasta, kun halutaan haun tuottavan uniikkeja tuloksia. Tein raportin pohjan uusiksi, koska siitä halutaan yhtenevä ja identtinen ulkoasultaan VKU2:n vastaavan raportin kanssa, jotta niitä voisi tarkastella rinnakkain esimerkiksi Excelissä.

Keskiviikko 3.7.2019

Aamupäivän tutkiskelin molempia allokoimaton aikaraportteja ja yritin saada AOD:n tiedot näkyviin, mutta en oikein onnistunut haluamallani tavalla. Ilmeisesti ongelma on SQL:n ja sivun välillä, koska tiedot eivät kohtaa. En myöskään saanut päivämäärää näkymään kunnolla.

Tiimiläinen sai viimeisteltyä tarvittavan SQL-lauseen iltapäivällä ja sainkin raportin näkymään raporttiympäristössä tarvittavine tietoineen. Koodiin pitää vielä saada haku käyttäjän asettamien aikarajojen mukaan kovakoodatun päivämäärän sijasta, mikä on tällä hetkellä SQL-haussa. Taulukossa on vielä pari epäkohtaa, jotka pitää korjata. Muokkailin lopuksi myös vähän Excel-määrittäjiä, jotta se näyttäisi halutut tiedot oikeanlaisina.

Torstai 4.7.2019

Tämän päivän tavoitteeksi päätin ottaa päivämäärähaun viimeistelyn. Jos saan sen valmiiksi, voin siirtyä laskujen ja kaavojen tarkistamiseen.

Päivän aikana sain raportin etsimään tiedot käyttäjän asettamilla aikarajoilla. Muutin str-replace-syntaksia käyttämällä SQL-haun alkuajan ja loppuajan muuttujaksi, jonka se ottaa käyttäjän syöttämänä päivämääränä. Nyt on enää hoidettavana laskutoimitusten paikkansapitävyys ja kaavojen tarkistaminen. Loppusummat eivät alkuperäisillä kaavoilla ole lähellekään oikeat, joten

ne pitää tehdä uusiksi. Jatkan niiden tarkistamista huomenna ja tavoitteena on saada raportti mahdollisimman hyvälle mallille.

Perjantai 5.7.2019

Selvitin raportin laskentaongelmia, jonka syyksi epäiltiin, että kyseessä on luultavasti pilkkupiste-ongelma desimaalilukuja käsiteltäessä. Tämä tuntuu olevan aika yleinen ongelma, sillä desimaalien käsittely laskennoissa voi vaihdella eri tekniikoiden välillä.

Vaikka en saanut tätä ongelmaa tällä viikolla ratkaistua, ei se haitannut työtehtävän minulle antanutta työkaveria, sillä hänen mielestään raportti oli jo hyvillä malleilla ja katsoo lomien jälkeen mihin asti olen raportin kanssa päässyt. Jatkan ongelman ratkaisua heti maanantaina viikonlopun jälkeen ja uskon, että raportti on seuraavan viikon loppuun mennessä valmis osaltani.

Viikkoanalyysi – Relaatietietokantojen käsittely

Jo ensimmäisenä opiskeluvuonna opiskelin tietokantojen käsittelyä ja tietokannanhallintajärjestelmänä toimi silloin avoimen lähdekoodin MySQL. Nyt töissä käytän Toad for Oracle- ja Microsoft SQL Server-ohjelmistoa, jotka taas ovat suljetun lähdekoodin ohjelmistoja. Vaikka ohjelmistot ovatkin erilaiset, sisältö on samanlaista mihin olen tottunut. Ohjelmistojen käyttöliittymät vaativat hetken oppimista, mutta niihin pääsi kiinni suhteellisen nopeasti.

Käsittämäni tietokannat ovat relaatiotietokantoja. Relaatietietokantojen tarkoitus on jakaa tieto käsitteisiin, joiden välille luodaan yhteyksiä. Taululle annetaan yksilöivä tunnus, jota kutsutaan pääavaimeksi. Kun tietoon halutaan viitata, käytetään viiteavainta. (Vihavainen & Luukkainen, 2017, viitattu 9.3.2020.)

Koulun kursseilta sai hyvän perustan tietokantojen käsittelyyn, mutta jo ensimmäisien työprojektien aikana opin nopeasti, millaista niiden käsittely on työelämässä. Joistakin kannoista on tehty testikäyttöön tarkoitettuja kopioita, mutta koska kaikista ei ole, piti työtehtävissä noudattaa erityistä varovaisuutta ja huolellisuutta, sillä käsittämäni tiedot ovat kiinni tuotannossa käytettävissä tietokannoissa.

4.4 Viikko 4 – AOD

Maanantai 8.7.2019

Ongelmana AOD2:n allokoimaton aikaraportissa on se, että raportti laskee ainoastaan kokonaisluvut, eikä huomioi desimaaleja ollenkaan. Tämän päivän tavoitteena on siis selvittää tätä ongelmaa ja yrittää etsiä ratkaisu siihen.

Kokeilin `str_replace`- sekä `number_format` -funktioita, mutta kumpikaan ei korjannut ongelmaa aivan täysin. `Number_format` -funktiota käytetään, kun formatoidaan numero näyttämään tuhansien erottelu oikeassa muodossa (W3resource, 2019, viitattu 8.7.2019). `Str_replace` -funktiota käytetään, kun halutaan korvata merkkijono toisella merkkijonolla (PHP Manual 2019, viitattu 8.7.2019). Yritin näiden funktioiden avulla muuttaa numeron erottamaan desimaalit pisteellä pilkun sijasta. Molemmilla laskutoimitusten tulokset eivät olleet paikkansapitäviä.

Tiistai 9.7.2019

Eilinen desimaaliongelmia ratkesi sillä, kun käytin `str_replace`- ja `number_format` -funktiota yhdessä. Testasin laskujen paikkansapitävyyden vielä Excelillä sekä manuaalisesti laskimella ja ainakin niiden mukaan laskut ovat oikein. Raportti on nyt valmis ja nyt pitää vain odotella, että työtehtävän minulle antanut työkaveri palaa lomilta töihin, jolloin raportti voidaan ottaa käyttöön.

```
$kesto = str_replace(',', '.', $kesto) ;  
number_format($allokoimaton,3, '.', '');
```

Kuvio 10 Desimaalien käsittelyyn käytetyt PHP-funktiot

Keskiviikko 10.7.2019

Viimeistelin vielä viimeisiä yksityiskohtia AOD:n raportista, jotta se vastaa VKU:n raporttia. Aloitin myös järjestelmäkuvien uudelleen piirtämisen. Päätin tehdä niistä yksinkertaisemmat ja tulostusystävällisemmät, joissa 3D-tyyliset kuvat ja muodot korvataan 2D-kuvioilla. Värimaailma kaipaa myös yksinkertaistamista.

Torstai 11.7.2019

Päivän tavoitteena oli saada järjestelmäkuvien ulkoasu suunniteltua valmiiksi. Muutin edellistä väripalettia erilaiseksi, johon kysyin myös esimiehen mielipiteen. Hänelle valitut värit kelpasivat, joten päätin ottaa ne käyttöön.

Perjantai 12.7.2019

Saldovapaa

Viikkoanalyysi – AOD

AOD-raportti ilmaisee halutut kvantitatiiviset eli mitattavissa olevat tiedot. Raportti toimii välikätenä tuotannon ja hallinnon välillä helpottaen päivittäisjohtamista antamalla yksityiskohtaisia ja mahdollisimman virheetömiä kehitysraportteja. Raportin tarve syntyi siitä, kun prosessin kaikista vaiheista ei saada tarkkoja raportteja, koska kaikki vaiheet eivät ole täysin automatisoituja, vaan vaativat operaattorilta manuaalisia toimenpiteitä. Manuaaliset mittaustulokset eivät aina ole tarpeeksi tarkkoja, joten työvaiheiden kaikkien vaiheiden kestoja ei saada mitattua tarpeeksi tarkasti, jonka vuoksi työvaiheiden kehittäminen on vaikeaa. Tämän raportin myötä raportoimaton eli allokoimaton aika saadaan raportoitua, jolloin voidaan esimerkiksi paikantaa vuoroja, jotka eivät merkitse häiriöitä tuotannonohjausjärjestelmään.

AOD on menetelmä, jolla teräkseen saadaan matala hiilipitoisuus ilman, että kromi hapettuu paljon. Ensimmäisessä vaiheessa prosessia AOD-konvertertiin kaadetaan sulaa terästä. Sulaan puhalletaan happea, typpeä ja argonia, joiden avulla saadaan kemiallisia reaktioita. Hiili hapetetaan teräslaadun vaatimusten mukaisesti, jonka jälkeen tapahtuu mellatus, eli reaktio, jossa ylimääräinen hiili poltetaan pois. Reaktion aikana sulaan syötetään seosaineita, joilla saadaan haluttu tilauksen ja teräslajin alkuainekoostumus. Prosessoinnin lopuksi suoritetaan pelkistysprosessi, jossa arvometallit pelkistetään syntyneestä kuonasta takaisin teräkseen. Tämän jälkeen otetaan lämpötila-, teräs- ja kuonanäyte. (Auer, keskustelu 4.7.2019.)

Raporttien ohjelmoimisen lisäksi on kiinnostavaa sekä tärkeää tutustua erilaisiin prosesseihin ja syihin, miksi raportti tarvitaan, keille se on suunnattu ja mitä hyötyä tehdystä raportista on.

4.5 Viikko 5 – JpGraph

Maanantai 15.7.2019

Viikko alkoi järjestelmäkuvien jatkamisella. Yhteyksien ja mittasuhteiden viilausta kiinnittäen myös huomiota väreihin ja tekstin näkyvyyteen.

Tiistai 16.7.2019

Päivä meni järjestelmäkuvien piirtämisellä, ei oikeastaan tapahtunut mitään tavallisesta poikkeavaa, samalla kaavalla mentiin kuin eilinenkin.

Keskiviikko 17.7.2019

Tämäkin päivä meni rutiinomaisesti järjestelmäkuvia piirtämällä. Vaikka tuntuukin, ettei tekemistä olisi enää paljon jäljellä, löytyy aina jotain pientä mitä pitää muuttaa tai hienosäätää.

Torstai 18.7.2019

Päivä kului kuten alkuviikkokin, mutta iltapäivällä juuri ennen kuin tein lähtöä sain uuden ohjelmointitehtävän. Siinä pitää tehdä sivu, joka näyttää graafin, joka hakee tiedot joka aamu-, iltaja yövuorolta tietokannasta ja piirtää halutuista arvoista sen mukaan kuvaajan. Tähän voisi käyttää JpGraphia, jota käytettiin yhdessä projektissa viime kesänäkin. Tutkin raportin speksejä ja suunnittelin sen toteutusta.

Perjantai 19.7.2019

Aamulla piirsin järjestelmäkuvia. Toinen kesätyöntekijä otti eilisen raportin hoidettavakseen, koska hänellä ei ollut muuta tehtävää sillä hetkellä ja hän oli ennenkin käyttänyt JpGraph-kirjastoa, joten raportti sujusi hänellä tehokkaimmin. Päätin sitten jatkaa vielä kaavioiden päivittämistä ennen viikonlopun viettoon lähtemistä.

Viikkoanalyysi – JpGraph

Tämä viikko oli aika lailla järjestelmäkuva-painotteista, mutta tuli myös opiskeltua jonkin verran JpGraphin dokumentaatiota, ennen kuin tiesin, että toinen työkaveri otti säröraportin hoitaakseen. Tämä ei kuitenkaan mennyt hukkaan, sillä JpGraph on hyödyllinen PHP-kirjasto, jolla voi tehdä dynaamisia kuvaajia käyttäen tietokannasta saatua dataa.

JpGraph piirtää kuvaajan kuvatiedostoon, joka näytetään -tunnisteessa. Määrittely kuvaajan piirtämiseen tehdään php-komennoilla. Vaikka kirjastolla on kätevä tehdä kuvaajia web-sivustolle, todella raskaat ja laajat kuvaajat kannattaa kuitenkin tehdä jollain muulla tavalla kuin PHP-skriptiä käyttäen. (Asial Corporation, 2016, viitattu 19.7.2019.)

4.6 Viikko 6 – Saavutettavuus värien avulla

Maanantai 22.7.2019

Kokeilin uutta väripalettia järjestelmäkuviin, kun nykyinenkään värimaailma ei mielestäni ollut tarpeeksi hyvä. Tutkin artikkeleita UI-suunnittelusta ja väripaletteista, jotka sopisivat myös henkilöille, joilla on vaikeuksia erottaa värejä. Netistä löytyikin väripaletteja, joita voisi kokeilla kuviin. Kiinnitin huomiota myös kontrasteihin, jotta kuviot ja viivat erottuisivat mahdollisimman hyvin taustasta ja toisistaan.

Tiistai 23.7.2019

Päivän tavoitteeksi asetin koetulostuksen, jotta voisin tarkastella miten muotoilut näkyvät tulostettuna, sillä pelkkään näytön esikatseluun ei voi luottaa täysin. Ruokailun jälkeen sain uudeksi tehtäväksi selvittää, voisiko järjestelmäkaavion tehdä järjestelmille, jotka kuuluvat automaatio L2:lle, mutta joiden sijainnit eivät välttämättä ole kaikille selkeät. Tähän voisi myös liittää tehdasalueen kartan, johon merkittäisiin kyseisten järjestelmien sijainnit.

Päivä meni myös tulostimen kanssa värkkäillessä, kun piti tulostaa päivitetymmät versiot järjestelmäkuvista, jotta pystyisi kyselemään mielipiteitä ja tarkistamaan, ovatko tiedot niissä oikein.

Yritin myös tulostaa karttakuvaa tehdasalueesta, mutta PDF-tiedosto ei jostain syystä mennyt läpi tulostuspalvelimesta. Järjestelmäkuvien tulostaminen onnistui lopulta moitteetta, joten tulostimessa ei pitäisi ainakaan olla vikaa, vaikka aluksi epäilin, että syy ongelmiin voisi olla esimerkiksi vanhentuneissa ajureissa.

Keskiviikko 24.7.2019

Päivän tavoitteena oli selvittää eilinen tulostusongelma. Karttatulostukseen liittyen tiimiläinen neuvoi tallentamaan PDF-tiedoston ensin omalle koneelleni verkkolevytä tulostamisen sijaan, sillä verkkolevy on niin paljon hitaampi etenkin, jos kyseessä on iso tiedosto. Tämä ratkaisi tulostusongelman ja karttakuva tulostui ongelmitta ja halutun kokoisena.

Tulostin molemmat versiot tehdasalueen kartasta, jotta voisin vertailla kumpaa projektissa kannattaa käyttää. Niistä voi myös helpommin tarkastella mitä pitää muuttaa, päivittää tai poistaa sekä millä tavalla järjestelmät merkitään niihin. Tehtävän minulle antanut tiimiläinen merkitsi tussilla kaikki järjestelmät, jotka tulisi kartassa näkyä.

Päätin myös aloittaa järjestelmäkuvien tekemisen alusta, sillä vanhoihin kuviin tehdyt muutokset eivät toimineet ihan saumatta, joten oli helpompi aloittaa piirustusten tekeminen puhtaalta pöydältä. Nyt on helpompi muuttaa mittasuhteet, jotta kuva olisi selkeämpi eivätkä vanhat muotoilut jääneet kummittelemaan.

Torstai 25.7.2019

Aloitin toisen linjan järjestelmäkuvien uudelleenpiirtämisen myös, sillä siinä oli myös ongelmia vanhojen muotoilujen kanssa. En kopioinut tietoja suoraan vanhoista versioista, vaan tein suurimman osan käsin, jotta vanhat muotoilut eivät kopioituisi uuteen. Tämä vaikutti aluksi hitaammalta tavalta, kun olin tätä ennen vain kopioinut vastaavan elementin toisesta ja korjannut muotoilun vasta myöhemmin. Tulin kuitenkin tulokseen, että tämä uusi tapa on nopeampi, sillä muotoilua ei pidä jälkikäteen korjata. Uskon, että edellisten muotoilujen ongelmat johtuivat valitusta temasta, jonka yhteensovittaminen ei toiminut uuden teeman kanssa.

Perjantai 26.7.2019

Saldovapaa

Viikkoanalyysi – Saavutettavuus värien avulla

Viikko meni taas Vision parissa järjestelmäkuvien teossa. Selvittelin myös, millä ohjelmalla voisin aloittaa tehtaan alueen kartan PDF-kuvan muokkaamisen, jotta saisin aloitettua uuden projektin. Tämän lisäksi perehdyin enemmän käyttöliittymäsuunnitteluun, koska mietin väri- ja kuviovaihtoehtoja järjestelmäkuvuihin.

Kaikki ihmiset eivät erota värejä tai sävyjä toisistaan kunnolla tai eivät ollenkaan. Suunniteltaessa ulkoasua niin kaavioille kuin websivustoille, on hyvä ottaa myös heidät huomioon. Osa ihmisistä ei erota esimerkiksi punaisen ja vihreän sävyjä toisistaan, joten on hyvä olla käyttämättä tällaisia värejä välittömässä läheisyydessä.

Väripalettia valittaessa on tärkeää kiinnittää huomiota värien ja sävyjen kontrasteihin. Tehokas tapa on suunnitella alusta asti mustavalkoisena ja laittaa värit vasta lopuksi. Toinen keino kontrastien tarkistamiseen on käyttää kuvan päällä mustavalko-filtteriä, jolloin näkee miten värien sävyt erottuvat toisistaan kontrasteina. (Tuchkov, I. 2018, viitattu 25.7.2019.) Värien lisäksi kannattaa käyttää myös kuviointeja, merkintöjä ja symboleita erottamaan objektit toisistaan.

Järjestelmäkuvia piirtäessäni kiinnitin huomiota etenkin siihen, että päällekkäiset elementit erottuvat toisistaan tarpeeksi. Käytin myös objekteja, jotka kuvaavat sitä mitä ne esittävät. Esimerkiksi palvelimilla on tietokoneen kotelon kuva ja työasemilla monitorin symboli. Vaikka värejä ei erottaisikaan toisistaan, voi näistä silti päätellä mitä ne esittävät. Niiden yhteyteen on toki myös kirjoitettu nimet, osoitteet ja yhteydet.

4.7 Viikko 7 – Järjestelmäkaavio

Maanantai 29.7.2019

Saldovapaa

Tiistai 30.7.2019

Parin saldovapaan jälkeen tuntui virkistävältä palata takaisin töihin. Päivän tavoitteeksi asetin edellisen viikon tehtävien jatkamisen eli piirtämisen järjestelmäkuvia ajan tasalle. Tämän päivän aikana ei tapahtunut mitään tavallisesta poikkeavaa.

Keskiviikko 31.7.2019

Järjestelmäkartta-projektissa pitäisi keksiä tapa, jolla muokata PDF-kuvaa ilman, että menettää kuvan skaalautuvuutta, sillä kartta pitää tulostaa A0-kokoisena. Itse Adoben Acrobat Reader DC -ohjelmalla ei voi kuin kommentoida ja merkitä kuvaan symboleita, mutta koska tiedosto on niin iso, ei koneeni jaksanut pyörittää näitä toimintoja ilman, että ohjelma jäätyy. Lisenssi Adoben viralliseen ohjelmaan on varmaan liian kallis kesätyöntekijälle annettavaksi, joten täytyy kysyä lupaa asentaa ilmaisohjelma, joka ei vaatisi lisenssiä. Tämä pitää kysyä ensin esimieheltä ja sitten myös Service Deskiltä, joka asentaisi ohjelman koneelleni, koska en omista Admin-oikeuksia.

Huomasin järjestelmäkuvien mittasuhteiden olevan ristiriidassa A0-kokoiseen paperiin nähden, joten kuvat vaativat uudelleen skaalaamisen. Suoraan koko piirustusta pienennettäessä kaikki objektit ja yhteysviivat menevät sotkuun, joten täytyy manuaalisesti määritellä ensin taustan koko ja asetella objektit uudelleen paikoilleen.

Torstai 1.8.2019

Määritin sivun tarkalleen A0-kokoiseksi eli 1189mm x 841 mm ja annoin objektien ja viivojen mennä sekaisin, sillä saan ne kyllä selvitettyä sitä mukaan, kun alan niitä asettelemaan uudelleen. Tässä menikin hetki, kun tarkastin kuvioiden yhteysviivat, nimet ja paikat, jotta ne ovat varmasti oikeilla paikoilla. Tarkistelin, joka välissä myös, miten kuva asettuu tulostuksen esikatseluun, jotta tiedän, että ollaan menossa oikeaan suuntaan.

Iltapäivästä alkoi näyttää jo hyvältä, joten aloin vielä muokkaamaan myös fonttien ja kuvioiden kokoa ja mittasuhteita. Voisin vielä tulostaa esikatselun vuoksi koetulostuksen molempien linjojen tämänhetkisistä järjestelmäkuvista, jotta näkisin miten kaikki ovat asettuneet isolla skaalalla ja

tarvitseeko jotain vielä hienosäätää tai muuttaa kokonaan. 2-linja on nyt hyvällä mallilla ja 1-linjan vastaava pitää saada vielä samalle viivalle.

Iltapäivällä sain aloitettua 1-linjan skaalaamisen, jossa päädyin kopioimaan 2-linjan kuvan ja piirtämään siihen suoraan 1-linjan vastaavat objektit, jotta kuvista tulisi varmasti samanlaiset mittasuhteiltaan ja ulkoasuiltaan.

Perjantai 2.8.2019

Jatkoin 1-linjan järjestelmäkaavion puhtaaksi piirtämistä. Mittasuhteiden kanssa oli hieman ongelmia, kun molempien linjojen kuvissa ei ole sama määrä objekteja keskenään. Kuvista haluan kuitenkin mahdollisimman yhtenevät keskenään, jotta niitä voisi tarkastella rinnakkain.

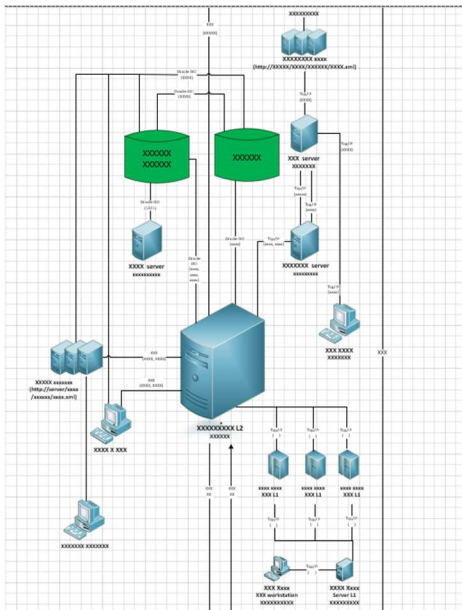
Viikkoanalyysi – Järjestelmäkaavio

Viikon aikana ei tapahtunut juuri mitään, mikä olisi poikennut tavanomaisesta. Päivät koostuivat järjestelmäkaavioiden piirtämisestä, eli objektien ja tekstien siirtelystä ja muokkaamisesta. Suurehkoja ongelmiakaan ei tullut vastaan, mutta joka kerta oppii aina hieman lisää Visio-ohjelman ominaisuuksista ja olen nyt nopeampi käyttämään sitä.

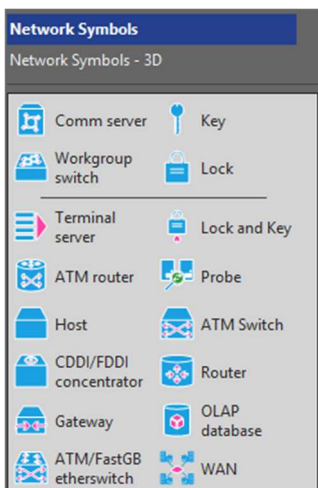
Silmät ovat kehittyneet tunnistamaan nopeammin esimerkiksi yhteysviivojen suoruuden ja tarkistamaan ovatko objektit oikeassa linjassa keskenään. Enää ei mene paljon aikaa siihen, että etsii tarvittavan ominaisuuden Vision käyttöliittymästä, kun ennen saattoi joutua turvautumaan hakukoneeseen sen etsimiseen.

Järjestelmäkaavioita on fyysisiä sekä loogisia. Fyysinen kaavio tehdään esimerkiksi verkkokomponenttien järjestyksestä, kun taas looginen kaavio kuvantaa datan kulkemista verkossa. Loogisissa kaavioissa kuvataan siis yleensä aliverkkoja, verkkolaitteita ja reitityksiä. (Lucidchart. 2020, viitattu 31.3.2020.)

Työstämäni kaavio on looginen kaavio. Tässä tapauksessa kuvataan esimerkiksi mitkä järjestelmät toimivat milläkin tasolla, sekä mikä tietokantayhteys, IP-osoite tai palvelin on kyseessä. Kaavio auttaa selkeyttämään mitkä ovat minkäkin tason vastuulla. Kuvakkeina käytetään symboleita, joista selviää muun muassa onko kyseessä tietokanta, palvelin tai työasema.



Kuvio 10 Mallikuva järjestelmäkaaviosta



Kuvio 11 Verko symboleita

Järjestelmäkuvien piirtämiseen käytän Microsoftin Visio-ohjelmaa, joka kuuluu Microsoftin Office-tuoteperheeseen.

4.8 Viikko 8 – ODBC

Maanantai 5.8.2019

Viimeistelin 1-linjan järjestelmäkaaviota. Päätin tehdä vihdoin tavoitteeksi asettamani koetulostuksen, jota mietin jo viime viikolla. Niistä pitäisi nyt nähdä, mitä muutettavaa niissä on. Kävi myös pieni vahinko, kun määrittelin tulostuksen kokoa, eli unohdin tarkistaa, että koko on

määritetty A0-kokoiseksi, joten tuloksena tulostin vahingossa A4-kokoisia kuvia, joten piti käydä uusi matka sulaton toimiston kopiohuoneessa, jossa tulostin sijaitsee.

Koetuloksessa selvisi, että kuvat eivät siltikään vielä koon puolesta ole identtiset, vaikka olinkin kopioinut toisen pohjaksi. Tämän selvittelyä pitää jatkaa seuraavana työpäivänä.

Tiistai 6.8.2019

Tajusin eilisen ongelman syyn, joka johtui siis siitä, kun Vision asetuksista laittaa päälle Auto Size-toiminnon. Kyseinen toiminto muuttaa kuvan koon piirustuksen sisällön mukaan, eikä asettamani kokomäärittelyt olleet enää voimassa. Olin luullut näitä vain tulostuksen omiksi toiminnoiksi, sillä tämä näkymä tuli, kun aloin tulostamaan ja luulin, että se koskee vain tulostusesikatselua eikä koko dokumenttia. Ei siis auttanut muu kuin tehdä taas kuvat uusiksi ja määritellä nyt tarkkaan, että dokumentti on A0-kokoinen ja Auto Size-toiminto ei ole päällä.

Piirustuksen kokoaminen onnistuu jo aika nopeasti ja vaivattomasti, kun sen on aloittanut jo muutamaan kertaan. Mittasuhteita mietin kuitenkin nyt tarkasti, jotta tätä ei tarvitsisi tehdä enää uudestaan. Pohja oli erikokoinen kuin edellisissä, joten tein monta asiaa manuaalisesti ja kopioin vain tekstit vanhoista kuvista parhaan tuloksen saavuttamiseksi.

Keskiviikko 7.8.2019

Eilinen meni 1-linjan piirtämisessä, joten tämän päivän tavoitteena oli ainakin aloittaa 2-linjan käsittely. Tarkistelin tällä kertaa moneen otteeseen, että koko on varmasti sama kuin 1-linjassa, jotta edelliset ongelmat eivät toistuisi. Päivän aikana ei tapahtunut mitään poikkeavaa, josta voisi kirjoittaa.

Torstai 8.8.2019

Päivän tavoitteena oli jatkaa siitä mihin edellisenä päivänä jäin. Sain 2-linjan järjestelmäkuvan piirtämisen niin hyvälle mallille, että huomenna voin aloittaa molempien linjojen kuvien hienosäädöt ja tulostaa ne molemmat.

Perjantai 9.8.2019

Tämän päivän tavoitteeksi valitsin järjestelmäkaavioiden viimeistelyn sekä mahdollisesti niiden tulostamisen, jotta voisin kysyä niihin vielä kommentteja.

Aamupäivä meni järjestelmäkuvien hienosäätöjä tehtäessä sekä varmistaessa, että kuvat ovat mahdollisimman yhtenevät keskenään. On yllättävää, miten molemmista kaavioista löytyy aina hieman jotain muokattavaa.

Iltapäivällä sain tutkittavaksi uunien mittaustuloksia kerääviä raportteja, joiden koodista pitäisi löytää kohta, jossa määritellään kuvaajien ToolTip-ominaisuudet. ToolTip-ominaisuus halutaan kokonaan pois, sillä se hidastaa kuvaajan lataamista liikaa etenkin, kun kuvaajaan halutaan suuria määriä dataa. Raakatietoja päivittyy tietokantaan kahden sekunnin välein, joten jos etsitään viikon tai kuukauden ajalta arvoja, tuloksia tulee niin suuria määriä, että kuvaajan lataaminen kestää niin pitkään, ettei raportti edes välttämättä lataudu ja jos latautuu, kuvaaja menee niin täyteen dataa, ettei siitä saa selvääkään.

Raportti on toteutettu ASP.net-tekniikalla ja sen kuvaajat on tehty käyttämällä ZedGraph-luokkakirjastoa. Löysin ZedGraphin dokumentaation, joten kävin sitä läpi ja yritin etsiä kohdan, jossa ToolTip-ominaisuus määritellään tai muuta kohtaa, josta voisi olla apua.

Viikkoanalyysi – ODBC

Alkuviikko meni aika pitkälti samoissa merkeissä kuin edellinen viikko. Mieleen jäi tulostusongelmat, joihin törmäsin muutamaan otteeseen. Tulostin on tarkoitettu suurien kuvien tulostamiseen, joten sen käyttö eroaa tulostimista, joita olen ennen käyttänyt. Ongelmien myötä olen oppinut tämän tulostimen ominaisuuksia, joten jatkossa tulostaminen onnistunee vaivatta.

Kesän aikana olen käyttänyt projekteissa enemmän ODBC-yhteyttä, mutta joissain tapauksissa olen käyttänyt myös OCI-yhteyden muodostamista. ODBC on lyhenne sanoista "Open Database Connectivity" ja se on ohjelmointirajapinta, joka mahdollistaa sovelluksen pääsemään käsiksi tietokantaan (Collins C, 2017, viitattu 20.11.2019). Tietokantojen käsittelyyn käytän ohjelmia Toad for Oracle ja Microsoft SQL Server.

4.9 Viikko 9 – Työturvallisuus

Maanantai 12.8.2019

Tutkin ZedGraphin dokumentaatiota ja etsin sieltä sopivia luokkia ja niiden nimiavaruuksia, jotka voisivat liittyä taulukon ToolTipseihin tai muuhun siihen liittyvään. Tutkin myös tarkemmin raporttien koodia, mutta ainakaan vielä en löytänyt sieltä kohtia, joista ongelma johtuu. Siellä on laitettu jo rajaus, jolla ei voi hakea raakatietoja kuukauden ajalta, mutta ToolTipseihin liittyvää kohtaa en löytänyt. ASP.net-tekniikkaa pitää vielä vähän kerrata ja katsoa viime vuoden projektia, jossa käytin sitä jonkin verran. Tarkastelen raportin koodia etäyhteyden kautta, jossa Visual Studio -ohjelman sijaan on vain Notepad++, joten Asp.net-koodin tarkastelu eroaa hieman siitä kuin mihin olen tottunut.

Tiistai 13.8.2019

Tämä päivä meni myös ZedGraphin ja raporttien tutkimiseen. En ole vielä löytänyt kohtaa, jota etsin, mutta epäilen, että se sijaitsee ZedGraphin ominaisuuksissa, joihin pitäisi päästä käsiksi joltain muuta kautta tai sitten osata kutsua koodiin ZedGraphin luokkia. En uskalla kokeilla mitään ennen kuin olen varma, mitä mistäkin kohdasta tapahtuu.

Keskiviikko 14.8.2019

Jatkoin samoilla linjoilla kuin eilen ja tarkastelin myös hieman järjestelmäkuvia. Ei mitään muuta raportoitavaa.

Torstai 15.8.2019

Tarkistin aamupäivällä järjestelmäkuvien tietoja. ASP-raporteista löytyi kohta, joka piti kommentoida pois, jotta saadaan ToolTipsit pois. Työtehtävän minulle antanut työkaveri oli löytänyt tämän tutkiessaan myös koodia, ja sain hoitaa kommentoinnin kaikkiin muihinkin raporteihin. Kohta sijaitsi yhden silmukan sisällä, jossa koordinaatit luodaan kuvaajan käyrälle, enkä ollut huomannut sen sisällä ollutta kohtaa, jossa puhutaan tageista. Kohdan ylle oli kommentoitu teksti "Create tag for each point" ja se määrittä ToolTipsien sisällön.

```

' Creating points for ImageMap
' Create tag for each point
For Each curve As CurveItem In myPane.CurveList
    c = curve.Points
    curveName = curve.Label.Text
    For Each pt As PointPair In c
        pt.Tag = curveName & "=" & Math.Round(pt.Y, 3) & vbCrLf & XDate.XLDateToDateTime(pt.X)
    Next
Next

```

Kuvio 12 Tooltip-ominaisuus

Kommentoin ensimmäisen silmukan sisällä olleen silmukan kokonaan pois raporteista, jolloin raportit toimivat kuten pitääkin. Tämän ansiosta latautuminen sujuu hieman sulavammin ja keveämmin.

Viikkoanalyysi – Työturvallisuus

Outokummulla työskennellessä turvallisuus on yksi tärkeimmistä asioista, jota painotetaan heti taloon astuessa. Vaikka kaikki työntekijät eivät olekaan töissä tuotannon puolella, on tehdasalue silti riskialtis paikka, jos on varmaton. Alueella on tarkat turvallisuussäännökset joka paikassa. Perusturvallisuuden lisäksi tietoturvallisuudesta pidetään hyvää huolta.

Outokummun kaiken toiminnan tavoitteena on nolla tapaturmaa (Outokumpu Stainless Oy. 2019, viitattu 16.3.2020). Kaikki pienimmätkin tapaturmat tutkitaan tarkoin ja siten kehitetään toimintaa turvallisemmaksi. Työturvallisuutta tuetaan kouluttamalla kaikki työntekijät vähintäänkin tehdasalueen yhteisellä turvallisuuskoulutuksella. Koulutusten lisäksi jokaisella osastolla on SBO-kierrokset ja velvollisuudet tehdä turvallisuushavainnoja. SBO-kierroksella jokainen käy läpi työpisteeseensä liittyvät riskit ja voidaan pohtia voiko asioita tehdä turvallisemmiksi. Turvahavainnot voivat olla niin pieniä kuin suuriakin, mutta myös positiivisia. Esimerkiksi jos joku asia on hoidettu oikein hyvin turvallisuuden kannalta, voi siitä antaa positiivisen turvahavainnon. Nämä tuovat positiivista suhtautumista turvallisuuteen, sillä tiukkoja säädöksiä ei ole tehty ketään ärsyttääkseen vaan suojelemaan.

Osastojen sisäisten turvallisuusohjeiden noudattamisen lisäksi tulee noudattaa myös tehdasalueen sisäisiä liikennesääntöjä ja nopeutusrajoituksia. Esimerkiksi omalla työmatkallani parkkipaikalta työpisteelleni on useampikin paikka, missä tulee olla valppaana etenkin liikenteen vuoksi. Alueella kulkee suuria ajoneuvoja, joilla on monen tuhannen kilon lasteja, eikä kaikilla ole ihmiskuskia vaan

osa kulkee automaattisesti. On myös hoksattava seurata alueella olevia hälyttimiä, jotka ilmaisevat esimerkiksi häkävuotoja.

Koska olen ollut töissä aiemmin tuotannon puolella, turvallisuussäännöksiä omaksuminen ei ole ollut hankalaa. Silti se yllätti hieman, kun tällaisessa toimistotyössäkin pitää tehdä SBO-kierroksia ja turvahavainnoja, sillä helposti mieltää, että vaaratilanteita tulee vain tuotannossa laitteiden parissa. Aluksi oli hieman hankalaa löytää oman työpisteen vaaratilanteita aiheuttavia seikkoja, mutta kyllä niitä lopuksi löytyi, jolloin pystyin tekemään vaaditut havainnot ja raportin kierroksesta.

4.10 Viikko 10 – Tietoturvallisuus

Maanantai 19.8.2019

Aamupäivä meni nettiyhteysongelmien korjaantumisen sekä ajastetun Windows 10 päivityksen valmistumisen odottamiseen. Ruokatunnin jälkeen kävin tulostamassa tämänhetkiset versiot järjestelmäkaavioista, jotta saisin viimeiset mielipiteet niistä, jotta pääsen viimeistelemään viimeisimmät versiot. Tulostaminen ei onnistunut tänäänkään ilman kimmelluksia, sillä toinen kuva tulostui A4-kokoisena ja kopioita tuli turhan monta samasta kuvasta, mutta ehdin onneksi keskeyttämään tuloksen, ettei kopioita tullut enempää. Kävin vielä ennen uutta tulostamista tarkistamassa tietokoneella, että kopioita molemmista tulee vain yksi ja että koot ovat varmasti A0-kokoisia. Tulostaminen onnistui nyt hyvin ja kuvat näyttivät hyviltä.

Uusia järjestelmäkuvia tarkastelivat esimieheni lisäksi yksi tiimiläisistä. Suuria symboleita haluttiin pienemmiksi ja tekstejä isommiksi, jotta kuva olisi selkeämpi. Myös muutamia yhteysvirheitä ja vanhentuneita laitteita tulisi vaihtaa ja päivittää kuvaan.

Palvelinsymbolit muutetaan laatikoiksi, joissa on isolla teksti ja pienellä palvelimen symboli laatikon reunassa. 1-tasoa kasvatetaan, jotta siitä erottuu paremmin uudet palvelimet, jotka lisätään kuviin. 3- ja 4-tason mittasuhteita muutetaan hieman, jotta symbolit ovat oikeissa paikoissa tasojen mukaisesti. Aloitan heti huomenna näiden muutosten tekemisen.

Tiistai 20.8.2019

Aloitin aamulla järjestelmäkaavioihin halutut tyylilliset muutokset ja sisällölliset korjaukset ja päivitykset. Testailin muutamia eri kokoja laatikoille ja nyt kaavio näyttää mielestäni selkeämmältä.

Iltapäivällä sain ohjelmointitehtävän, jossa sivun elementtiin haluttiin hover-efektillä aktivoituva info-ikkuna. Ensinnäkin piti saada pieni info-kuvake elementin oikeaan reunaan vasemman sijasta, joka ei veisi ylimääräistä tilaa mistään suunnasta. Sitten pitää määritellä pop-up -kuva, joka ilmestyy näytölle, kun info-kuvakkeen päälle viedään hiiri. Pop-up -kuvan asettelu pitää myös määritellä niin, ettei se mene tiettyjen elementtien päälle ja varmistaa, että se tulee elementtien eteen, eikä jää niiden taakse. Koska tätä ominaisuutta mietitään vielä, teen sen vain yhdelle sivulle. Jos ominaisuus halutaan ottaa käyttöön, on se helppo kopioida, kun kommentoin tekemäni koodin tarpeeksi selkeästi, jolloin ei haittaa, vaikka en olekaan enää töissä.

Keskiviikko 21.8.2019

Jatkoin järjestelmäkuvien muutosten tekemistä. Nyt aloitin 2-linjan muokkaamisen, kun eilen keskityin 1-linjaan. Järjestelmäkuvat ovat nyt hyvällä mallilla ja uskon, että saan ne valmiiksi ennen työsopimuksen päättymistä.

Torstai 22.8.2019

Sain sähköpostia liittyen PDF-kuvan editointiin, jossa suositeltiin ilmaisohjelman sijaan käyttämään standardina olevaa Adoben virallista PDF-editointiohjelmia. Työsopimusta minulla on enää viikko jäljellä, joten en usko, että saan ohjelmaa tarpeeksi ajoissa, jotta ehtisin toteuttamaan karttaprojektin.

Perjantai 23.8.2019

Tarkistin opinnäytetyön materiaaleja ensi viikkoa varten, kun esimies tarkistaa alkuviikosta sisällön, ettei se sisällä mitään arkaluontoista tietoa mikä pitää joko poistaa tai sensuroida.

Hioin myös vielä järjestelmäkuvia, jotka viimeistelen ensi viikolla loppuun ennen kuin työsopimukseni loppuu. Illalla on tiedossa päättäjäisjuhla, johon osallistuu melkein koko tiimi.

Viikkoanalyysi – Tietoturvaluisuus

Yleisen työturvaluisuuden lisäksi Outokummulla on tiukka linja myös tietoturvaluuuteen. Tietoturvaluuutta pidetään yllä kieltämällä kaikki kuvaaminen, videointi ja tallentaminen ilman erillistä lupaa. Tietokoneisiin ei saa asentaa omia ohjelmia ja muistitikun saa asentaa vasta, kun on tehty virustarkastus. (Outokumpu Stainless Oy, 2017, viitattu 16.3.2020.)

Tiukat tietoturvaluuusohejet antavat hieman haasteita opinnäytetyötä ajatellen. Monet käsiteltävät asiat ovat salassapitosopimusten ja liikesalaisuuksien alaisia, joten pitää olla tarkkana, mitä kaikkea voi kirjoittaa julkiseksi. Koodeihin viitatessa on kuitenkin helppo poimia yleisesti käytettäviä funktioita paljastamatta liikaa prosesseista ja arkaluontoiset asiat voi sensuroida.

Sensuroin tässä opinnäytetyössä näkyviä asioita esimerkiksi muuttamalla muuttujien nimet ja tietokantoihin liittyvät nimet, jotta niitä ei voisi yhdistää mihinkään. Käytän teoriaan viitatessa lähteinä yrityksen julkisia lähteitä sekä esimiehen luvalla joitain poimintoja sisäisestä intrasta tai sitten suullisia tietoja ja ohjeita työkavereilta. Opinnäytetyön valmistuessa tarkistutan sen vielä esimiehellä, ettei mukaan pääse mitään arkaluontoista mitä ei saisi julkisesti näyttää.

5 POHDINTA

Näiden kymmenen viikon aikana kehityin mielestäni merkittävästi, kun verrataan aikaisempaa tietotaitoani ohjelmoinnissa ja työelämässä. Työtehtävät erosivat toisistaan välillä paljonkin, jolloin tuli kehittyä monella eri saralla, joka tuki ammatillista kehittymistäni. Suurin kehitys tapahtui mielestäni PHP:n ja tietokantojen kanssa.

Käytin paljon eri ohjelmia ja alustoja, joista useimmin käytin Notepad++, Microsoft Office Visio, Toad for Oracle, Microsoft SQL Server.

Haluan tulevaisuudessa kehittyä dokumentoinnissa, vuorovaikutustaidoissa sekä ajanhallinnassa. Vuorovaikutustaidot kehittyivät kesän ajan jonkin verran. Työkavereiden kanssa kommunikointi helpottui, kun heihin tutustui kahvihuoneessa sekä ruokatuntien aikana. Viikoittaiset tiimipalaverit auttoivat myös kehittymään esimerkiksi omien työtehtävien vaiheiden ja ominaisuuksien kertomisesta, vaikka joskus termit saattoivatkin mennä vähän sekaisin. Minun pitäisi silti uskaltaa kysyä enemmän apua ongelmatilanteissa, eikä yrittää viimeiseen asti itse etsiä tietoa. Siitäkään ei aina ole haittaa, sillä silloin asiasta voi myös oppia, mutta usein olisi tehokkaampaa kysyä joltain, jolla on asiasta enemmän kokemusta.

Suurin tavoitteeni on kuitenkin kehittyä ohjelmoinnissa ja oppia ainakin muutama tekniikka niin hyvin, että voin hyvällä omallatunnolla sanoa ne osaavani. PHP, HTML, CSS:n, eri SQL-kielet ja JavaScript-tekniikat ovat tämän työpaikan ansiosta jo hyvällä pohjalla, joiden syvempää opiskelua on helppo jatkaa. Tuli myös tutustuttua enemmän C#-kieleen sekä asp.net-tekniikoihin, joista omaan nyt joitain perusteita, joita osaan käyttää, jos tulee tarvetta.

LÄHTEET

Asial Corporation, 2016. Purpose and Usage. Viitattu 19.7.2020,
<https://jpgraph.net/download/manuals/chunkhtml/index.html>

Auer, E. 2019. Systemisuunnittelija, Outokumpu Stainless Oy. Keskustelu 4.7.2019.

Boskovic, T. 2018. CSS Position: Relative vs Position Absolute. Viitattu 3.3.2020,
<https://dzone.com/articles/css-position-relative-vs-position-absolute>

Collins C. 2017. History of ODBC. Viitattu 20.11.2019,
<https://ccollins.wordpress.com/2007/06/03/history-of-odbc/>

Hayes, D. 2019. Rubber duck debugging: The psychology of how it works? Viitattu 24.7.2019,
<https://www.thoughtfulcode.com/rubber-duck-debugging-psychology/>

Lewis D. 2011. Starting to use ODBC with PHP. Viitattu 28.7.2019,
<https://www.ibm.com/developerworks/library/os-php-odbc/index.html>

Lucidchart. What is a Network Diagram. Viitattu 31.3.2020,
https://www.lucidchart.com/pages/network-diagram#section_3

Melcher S. 2015. CSS wind speed/direction. Viitattu 17.6.2019,
<https://codepen.io/xixao/pen/OPovyN>

Outokumpu Stainless Oy. 2019. Työturvallisuus Tornion tehdasalueella. Viitattu 16.3.2020,
<https://www.outokumpu.com/-/media/files/locations/tornio/outokummun-tornion-tehtaiden-turvallisuus-ja-ymparistokoulutus-2019.pdf?revision=022eb8b0-d65e-477a-82bb-8daffbce15bd&modified=20190208105833>

Outokumpu Oyj, 2017. Viitattu 27.8.2019. Sisäinen intra.

Outokumpu Oyj, 2019. Tietoa Outokummusta. Viitattu 27.8.2019, <https://www.outokumpu.com/fi-fi/about-outokumpu>

PHP Documentation Group. 2019. Str_replace. Viitattu 8.7.2019, https://www.php.net/str_replace

Tuchkov, I. 2018. Color blindness: how to design an accessible user interface. Viitattu 25.7.2019, <https://uxdesign.cc/color-blindness-in-user-interfaces-66c27331b858>

Vihavainen A. & Luukkainen M. Viitattu 9.3.2020, <https://tietokantojen-perusteet.github.io/>

W3resource. 2019. PHP: number_format() function. Viitattu 8.7.2019, https://www.w3resource.com/php/function-reference/number_format.php

W3Schools. 2020. CSS transform Property. Viitattu 3.3.2020, https://www.w3schools.com/cssref/css3_pr_transform.asp