



# Datan visualisointi laitosylläpidon avuksi

Helmi Laakkonen

Opinnäytetyö, AMK

Joulukuu 2023

Tieto- ja viestintätekniikan tutkinto-ohjelma

**Laakkonen, Helmi**

## **Datan visualisointi laitosylläpidon avuksi**

Jyväskylä: Jyväskylän ammattikorkeakoulu. Joulukuu 2023, 39 sivua.

Tieto- ja viestintäteknikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

### **Tiivistelmä**

Opinnäytetyön toimeksiantajalle Elomatic Oyille tuli ajankohtaiseksi laajentaa GroundWork-sovellusta. GroundWork on teollisuuden kunnossapidon ja omaisuudenhallinnan selainpohjainen sovellus. Sovellus sisältää paljon laitosylläpitoon liittyvää dataa, jota opinnäytetyössä jatkojalostettiin sovelluksen käyttäjille. Opinnäytetyön tavoitteeksi muodostuikin selvittää mitä teollisuuden kunnossapitoon liittyvää dataa olisi hyödyllistä näyttää ja mikä olisi siihen parhain tapa.

Opinnäytetyö toteutettiin tutkimuksellisenä kehitystyönä. Teoriaosuudessa keskityttiin datan visualisoinnin perusteisiin. Datan visualisoinnin käytänteitä hyödynnettiin moduulin suunnittelussa, jotta tuloksista saatiin mahdollisimman ajankohtainen.

Opinnäytetyön tuloksena toteutettiin uusi moduuli sovellukseen. Moduuli koostuu kahdesta välilehdestä sekä useista erillisistä komponenteista. Moduulin avulla laitoksen ohjaaja pystyy näkemään ajankohtaista tietoa laitoksistaan ja tekemään johtopäätöksiä laitoksen tilasta. Sovellus toteutettiin Laravel-sovelluskehikseen, jossa moduulin komponentit toteutettiin Livewirellä. Toteutuksen kaavioissa käytettiin AnyChart JavaScript-kirjastoa.

Opinnäytetyön jatkokehityksenä on pilotointi asiakkaan kanssa. Tarkoituksena on kerätä moduulin toimivuudesta palautetta, jotta moduulia voidaan muokata käyttäjäystävällisemmäksi ja hyödyllisemmäksi.

### **Avainsanat (asiasanat)**

Kunnossapito, Datan visualisointi, Laravel

### **Muut tiedot (salassa pidettävät liitteet)**

-

**Laakkonen, Helmi**

### **Utilizing data visualization for a facility maintenance program**

Jyväskylä: JAMK University of Applied Sciences, December 2023, 39 pages.

Degree Programme in Information and Communication Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

### **Abstract**

Expanding of the GroundWork application became relevant for Elomatic Oy, the client of the thesis. GroundWork is a browser-based application for industrial maintenance and asset management. The application contains a substantial amount data related to plant maintenance, which was further refined for the application's users in the thesis. The goal of the thesis was to show which industrial maintenance-related data is useful and what would be the best way of visualizing it.

The theory part focused on the basics of data visualization. The conventions of data visualization were utilized in the design of the module to make the results as up to date as possible.

As a result of the thesis, a new module application was implemented. The module consists of two tabs and several separate components. With the help of the module, the supervisor of the facility can see current information about his facilities and is able to make conclusions about the status of the facility. The application was implemented using the Laravel framework, where the module's components are made by using Livewire. The AnyChart JavaScript library was used in the diagrams of the implementation.

The further development of the thesis is to pilot with the customer. The purpose is to collect feedback on the functionality of the module, so that the module can be improved and made more useful for the customer over time.

### **Keywords/tags (subjects)**

Facility maintenance, Data visualization, Laravel

### **Miscellaneous (Confidential information)**

-

## Sisältö

<b>1</b>	<b>Johdanto .....</b>	<b>3</b>
1.1	Tausta .....	3
1.2	Elomatic ja GroundWork .....	3
1.3	Aihe ja tavoitteet .....	4
1.4	Tutkimusmenetelmät .....	4
1.4.1	Tutkimuksellinen kehittämistoiminta .....	4
1.4.2	Tutkimuksellisen kehittämistoiminnan hyödyntäminen opinnäytetyössä .....	5
1.5	Eettisyys ja luotettavuus .....	5
<b>2</b>	<b>Datan visualisoinnin perusta .....</b>	<b>5</b>
2.1	Kunnossapito .....	5
2.2	Data visualisointiin .....	6
2.3	Kaaviot ja taulukot .....	7
2.3.1	Taulukot .....	8
2.3.2	Pylväskaaviot .....	9
2.3.3	Ympyrä- ja rengaskaaviot .....	10
2.3.4	Viivakaavio ja sen muunnelmät .....	13
2.4	Visuaalinen suunnittelu .....	15
2.4.1	Tekstien merkitys .....	16
2.4.2	Värien merkitys .....	17
2.4.3	Tilan käyttö .....	17
2.4.4	Datan asettelu ja yhtenäistäminen .....	18
2.5	Design Process .....	19
<b>3</b>	<b>Toteutus .....</b>	<b>21</b>
3.1	Suunnittelu .....	21
3.1.1	Käyttäjätarinat .....	21
3.1.2	Ideointi .....	21
3.1.3	Prototyyppi .....	23
3.2	Ohjelmointi .....	24
3.2.1	Käytetyt teknologiat .....	24
3.2.2	Backend .....	24
3.2.3	Frontend .....	31

<b>4 Tulokset</b> .....	<b>34</b>
<b>5 Pohdinta</b> .....	<b>36</b>
<b>Lähteet</b> .....	<b>38</b>

## **Kuviot**

Kuvio 1. GroundWork-sovelluksen etusivu .....	3
Kuvio 2. Tyypillinen sekä harhaanjohtava pylväskaavio .....	9
Kuvio 3. Tyypillinen sekä liian täytetty ympyräkaavio .....	11
Kuvio 4. Pylväsdigrammi, jonka luvuista ei saa selvää.....	12
Kuvio 5. Tyypillinen rengaskaavio .....	13
Kuvio 6. Tyypillinen viivakaavio .....	14
Kuvio 7. Tyypillinen summaviivakaavio.....	15
Kuvio 8. Suunnitteluprosessin vaiheet.....	19
Kuvio 9. Karkea luonnos.....	22
Kuvio 10. Figmaassa suunniteltu versio analysoinnin välilehdestä .....	23
Kuvio 11. Yksinkertainen controller .....	25
Kuvio 12. TaskList-komponentin mount-funktio .....	26
Kuvio 13. Tapahtumalokille suunniteltu tietokantataulu .....	27
Kuvio 14. Osa tapahtumalokin suodatustoimintaa .....	28
Kuvio 15. Datan sovittaminen summaviivakaavioon .....	29
Kuvio 16. Kalenterin tapahtuma .....	31
Kuvio 17. Ensimmäinen hiotumpi versio ulkonäöstä.....	32
Kuvio 18. Aikaa vievien laitteiden -komponentin ensimmäinen ja viimeisin toteutus .....	33
Kuvio 19. Vikaantuvien laitteiden komponentti .....	33
Kuvio 20. Tapahtumalokin päivitetty ulkonäkö .....	34
Kuvio 21. Ensimmäisen välilehden viimeisin toteutus.....	35
Kuvio 22. Toisen välilehden viimeisin toteutus .....	36

## **Taulukot**

Taulukko 1. Vuoden 2023 keskilämpötilat Jyväskylässä .....	8
Taulukko 2. Livewire komponentit .....	25

# 1 Johdanto

## 1.1 Tausta

Informaatiotekniikoiden kehittyessä dataa kerätään aina vain suurenevissa määrin. Dataa pommitetaan jokaiselta alueelta, kuten rahoitusosalta, myynnistä ja asiakaspalvelusta. Laajan datan lukeminen taulukosta on vaikeaa, jolloin datan sisältämä informaatio katoaa.

Datan visualisoiminen ei ole uusi keksintö. William Playfair julkaisi useat nykypäivänä käytetyt kaaviot jo vuonna 1786 (Yau 2013). Datan visualisointi on vielä nykypäivänäkin puhuttu aihe. Datan analysoinnilla ja visualisoimisella voidaan valjastaa data kilpailukyiseksi. Visualisoinnin avulla voidaan tuottaa tärkeitä oivalluksia datasta, joita muuten ei huomattaisi.

## 1.2 Elomatic ja GroundWork

Opinnäytetyö toteutettiin Elomatic Oy:lle. Elomatic on suomalainen yli 50 vuoden ikäinen suunnittelu- ja konsultointiyritys, jolla on tällä hetkellä yhteensä 1300 työntekijää useassa maassa. Elomatic tarjoaa liudan erilaista osaamista, mukaan lukien: suunnittelua, teknistä konsultaatiota, tuote- ja palvelukehitystä sekä ohjelmistokehitystä. (Elomatic n.d.)

GroundWork

KIRJAUDU
VALITSE KIELI

---

### Näe tulevaisuus jo tänään

Kaikki tarvitsemasi tiedot yhdeltä data-alustalta omaisuuden, kunnossapidon ja laitosten hallintaan

Groundwork on visuaalinen ja käyttäjystävällinen data-alusta palvelu. Palvelun avulla suunnittelet, ohjaat kunnossapitoa ja hallitset omaisuuttasi koko sen elinkaaren ajan.



Kuvio 1. GroundWork-sovelluksen etusivu

Opinnäytetyö Elomaticin sisällä toteutettiin selainpohjaiseen sovellukseen nimeltä GroundWork. GroundWork on teollisuuden ylläpitoon tarkoitettu kunnossapidon sekä omaisuudenhallinnan sovellus. Sydäntoiminnallisuuksia sovelluksessa on erilaisten tehtävien kirjaaminen sekä omaisuudenhallinnassa olevat laitelistaukset.

### **1.3 Aihe ja tavoitteet**

Koska GroundWork sisältää paljon dataa laitossylläpidosta, heräsi halu hyödyntää tätä dataa muutenkin. Jatkojalostamalla dataa kohdekäyttäjien hyödyksi, voitaisiin antaa kohdekäyttäjille lisää työkaluja tehdä älykkäitä ratkaisuja. Tämä myös tuottaisi hyötyä GroundWorkin haluttavuudelle; käyttämällä sovelluksen muita osia, saa myös käyttöönsä data-analyysiä.

Opinnäytetyön ydintavoitteeksi muodostui uuden moduulin toteuttaminen laitossylläpidon datasta. Tehtäväksi tuli selvittää millainen data olisi hyödyllistä kohdekäyttäjille, sekä mikä olisi parhain tapa visualisoida sitä. Opinnäytetyöhön sisältyi siis datan visualisoinnin kirjallisuuteen tutustuminen, moduulin suunnittelu sekä moduulin toteutus.

### **1.4 Tutkimusmenetelmät**

#### **1.4.1 Tutkimuksellinen kehittämistoiminta**

Toikan ja Rantasen (2009) mukaan työelämään kuuluu isona osana kehittämistoiminta. Alan asiantuntemus voidaan osoittaa sen kehittämistyöllä. (Mts. 9.) Kehittämistoiminnan prosessiin kuuluu lähtökohdan määrittely. Yleensä siihen liittyy tutkimusongelma, johon etsitään ratkaisua tai tavoite, jota halutaan edistää kehitystoiminnalla. Lähtökohdan perustelu auttaa rajaamaan tavoitteita ja toteutusta. (Mts. 57.)

Tutkimuksellisessa kehittämistoiminnassa kerätään tutkimustietoa. Tietoperusta sisältää tutkimukselle olennaista tietoa. Tietoperustan tarkoituksena on laajentaa tutkijan osaamista uusista näkökulmista. Tavoitteena on kartoittaa tutkimukseen jo olemassa olevaa tietoa. Tietoperustan avulla tietoa sovelletaan käytännön työhön. (Mts. 18-21.)

### 1.4.2 Tutkimuksellisen kehittämistoiminnan hyödyntäminen opinnäytetyössä

Opinnäytetyö toteutettiin tutkimuksellisena kehittämistoimintana. Opinnäytetyön lähtökohdaksi muodostui selvitystyö, joka myös toteutetaan konkreettisesti. Lähtökohdan määrittelyn jälkeen kerättiin tietoperustaa, jota hyödynnetään opinnäytetyön toteutuksessa. Aineistoa kerättiin suurimmaksi osaksi aiheeseen liittyvistä kirjoista, jotta tietoperusta olisi mahdollisimman luotettavaa.

Opinnäytetyöhön toteutukseen liittyi myös sen suunnittelu. Suunnittelussa otettiin huomioon tietoperustassa esiin tulleet yleiset käytänteet datan visualisoinnissa. Käytänteitä sovellettiin toteutuksessa. Koska opinnäytetyön lopputulos näkyy pitkälti vain visuaalisena tuotoksena,

## 1.5 Eettisyys ja luotettavuus

Toikka ja Rantanen (2009, 121) kuvaavat luotettavuutta yhdeksi tärkeimmäksi tunnusmerkiksi tieteelliselle tiedolle ja korostavat kehittäjän velvollisuutta pyrkiä luotettaviin tuloksiin. Kehittämistyössä luotettavuus nähdään myös käyttökelpoisuutena, eli tutkimustieto on hyödyllistä. Tutkimustyössä luotettavuus määrittyy käytetyistä tutkimusmenetelmistä, prosessista sekä saaduista tutkimustuloksista. (Toikka & Rantanen 2009, 121.)

Opinnäytetyön eettisyyteen pätee myös Jyväskylän ammattikorkeakoulun eettiset periaatteet. Niissä edellytetään myös käyttämään hyvää tieteellistä käytäntöä. Eettisiin periaatteisiin lukeutuu käytettyjen lähteiden merkitseminen ja vilpittömän toiminta. Vilppi sisältää sepittämisen, vääristelyn, plagioinnin ja anastamisen. (Eettiset periaatteet 2018.)

## 2 Datan visualisoinnin perusta

### 2.1 Kunnossapito

Laitosylläpitoon lukeutuu sen kunnossapito. Kunnossapito on laaja käsite, jonka ydintavoitteena on varmistaa koneiden, laitteiden ja rakennusten kunto. Kunnossapitoon liittyy konkreettisten toimien ohella myös ajattelutapa. (Kunnossapidon käsitteet n.d.)

Kunnossapidon teknisillä toiminnoilla on kaksi eri toisiaan täydentävää toimintaa: korjaava ja ennakkoiva. Korjaava kunnossapito sisältää toimenpiteitä, jotka suoritetaan vasta, kun kohde on vi-



kaantunut. Näitä ovat esimerkiksi osittais- tai kokonaisvika, joka estää kohdetta toimimasta oikein tai koko kohteen käytöstä poisto. Korjaukseen liittyy myös erilaisia toimintatavoitteita. Tyypillinen korjaus on kohteen toimintakyvyn palautus, joko kohteen vian korjaamisella tai kohteen vaihtamisella vastaavaan tuotteeseen. Väliaikaisella korjauksella voidaan vähentää hetkellisestä vikaantumisesta johtuvaa käyttökatoa, joka ei ole kuitenkaan pysyvä korjaus pitkällä aikavälillä. Tällöin kohdetta voidaan korjata uudelleen pysyvästi. Tämä on parantavaa korjausta, jonka tarkoituksena on estää tulevaisuudessa toistuva vika. (Korjaava kunnossapito n.d.)

Ennakoiva kunnossapito sisältää kunnonvalvonta- ja huoltotoimenpiteet. Nämä toimenpiteet ovat ennalta laadittuja, joten niitä suoritetaan ennen kuin kohteeseen esiintyy vika. Jaksotetut huollot ovat jatkuvaa ylläpitämistä. Kunnonvalvonnassa kohdetta tarkkaillaan, jotta mahdollinen vikaantuminen voidaan estää, ennen kuin kohde lakkaa toimimasta. Kunnonvalvonnassa voidaan myös asettaa suureita, joita tarkkaillaan hälytysrajojen avulla. (Kunnossapidon käsitteet n.d.)

Ennakoivalla kunnossapidolla on selkeitä hyötyjä. Kunnonvalvonnalla sekä huoltojaksoilla pystytään estämään käyttökatoja, jotka vaikuttavat kustannuksiin, kun vioittunut kohde ei ole toiminnassa. Huolto lisää toimitusvarmuutta ja laadun ylläpitoa, joka taas vaikuttaa asiakastyytyväisyyteen. Ennakoivalla kunnossapidolla ehkäistään myös turvallisuusriskejä, jotka voivat aiheutua kohteen vikaantumisesta. Vaikka ennakoiva kunnossapito on hyvin olennaista toimintaa, aina sitä ei ole kannattavaa tehdä. Itse kunnossapidosta syntyy jo henkilöstökustannuksia, joten on syytä miettiä, milloin se on kustannustehokkainta. Pienissä kohteissa, jotka eivät aiheuta käyttökatoja tai laitoksen seisokkia, korjaava kunnossapito on yleinen menettely. (Kunnossapidon kustannukset n.d.)

## **2.2 Data visualisointiin**

Data on jokaisen visualisoinnin pohja. Data sisältää tietoa ja faktoja. Mitä parempi ymmärrys on datalähteeseen, sitä vahvempi on pohja hyvälle datagrafiikalle. Pelkällä hyvällä visualisoinnilla ei pääse pitkälle, jos ei ole dataa, joka tukee sitä. Datan visualisointi on monipuolinen prosessi, johon datan ymmärtäminen liittyy yhtenä tärkeimpänä vaiheena. (Yau 2013, XI.)

Yleensä datan tarkoituksena on ratkaista ongelmia ja kipukohtia. Datan visualisoinnissa ensimmäisenä tulisi miettiä kenelle suunnitellaan. Se määrittää millaista dataa tullaan keräämään ja käyttämään. (Yuk & Diamond 2014, 85.)

Usein datan visualisoinnissa käytetty data liittyy KPI:hin (Yuk & Diamond 2014, 89). KPI on englanninkielinen lyhenne sanoista Key Performance Indicator. Suomeksi sitä sanotaan suorituskykymitariksi. KPI-lukuja voi olla useita ja niiden avulla seurataan säännöllisesti suorituskykyä ja edistymistä. Ne kertovat, kuinka tehokkaasti tavoitteita saavutetaan. (Mikä on KPI? 2021.) Näiden lukujen löytäminen on tärkeä, mutta hankala prosessi (Yuk & Diamond 2014, 90).

Tutkiva ja selittävä analyysi ovat kaksi erilaista tapaa jäsenellä dataa. Tutkivassa analyysissä jokainen lukija joutuu käyttämään aikaa datan muuttamiseksi tiedoksi (Knaflic 2015, 33). Tutkivilla ratkaisuilla pyritään luomaan työkalu, joka tarjoaa käyttäjälle tavan datan visuaaliseen tutkimiseen. Se tarkoittaa sitä, että lukijat tekevät itse analyysin ja pyrkivät löytämään asioita, joita he pitävät merkittävänä tai kiinnostavana. Tämä lisää lukijan kuormitusta. (Kirk 2012, 53.)

Selittävässä analyysissä data on valmiiksi prosessoitu sellaiseen muotoon, josta lukijan on nopea ymmärtää sen tarkoitus. Tällöin suunnittelijan tulee tehdä laajempi työ ja luoda selkeä kuva mielenkiintoisista tarinoista ja analyyseista datasta valmiiksi lukijalle, jolloin datan lukeminen on nopeampaa verrattuna tutkivaan analyysin dataan. (Kirk 2012, 53.)

## **2.3 Kaaviot ja taulukot**

Mobiilisovellusten ja sosiaalisen median suosion nousun myötä hetkessä tuotetun datan määrä on valtava. Miljoonia rivejä sisältävän datan tarkastelu yksinään on lähes mahdotonta. Tämän takia useat yritykset tarvitsevat jonkinlaisen datan visualisoinnin käyttöä ymmärtääkseen dataa. (Yuk & Diamond 2014, 2.)

Yau (2013) vertaa kirjassaan datan visualisointia kirjoittamiseen; visualisointi on yksi tapa esittää dataa, joka on myös ote todellisesta maailmasta, samalla tavalla kuin kirjoitusta voidaan käyttää erilaisten tarinoiden kertomiseen. Tyypillisimmät visualisoinnit, kuten esimerkiksi viiva- ja pylväsdiagrammit, pystytään tuottamaan helposti sekä lukemaan nopeasti, mikä tekee niistä tehokkaita välineitä datan tutkimisen avuksi (Yau 2013, 141.)

Kaaviot ja taulukot vuorovaikuttavat ihmiseen eri tavoin. Taulukot vuorovaikuttavat verbaaliseen, eli sanalliseen järjestelmään. Taulukon dataa on mahdollista verrata vain kohta kohdalta. Kaaviot ovat taas vuorovaikutuksessa visuaaliseen järjestelmään, joka on nopeampi prosessoimaan kaikenlaista informaatiota. Tämä tarkoittaa sitä, että huolellisesti suunnitellusta kaaviosta saatava tieto menee perille nopeammin kuin huolellisesti suunnitellusta taulukosta. (Knafllic 2015, 49.)

Oikean kaavion valinta voi kuitenkin tuottaa vaikeuksia. Asiaan perehtymättömät sekä aloittelijat usein käyttävät kaavioita, joilla data saadaan näyttämään vain kauniilta, mutta ne eivät ole erityisen käyttökelpoisia. Monille kaavioille on kuitenkin hyvin tiettyjä käyttötarkoituksia. (Yuk & Diamond 2014, 43.) Teoriaosuudessa tullaan käymään opinnäytetyön kannalta oleelliset kaaviot.

### 2.3.1 Taulukot

Taulukoiden prosessointi tapahtuu verbaalisessa järjestelmässä, eli niitä luetaan. Ne soveltuvat hyvin sellaiseen dataan, jossa lukijat haluavat kiinnittää huomion eri riveihin ja sarakkeisiin saadakseen haluamansa tiedon. Taulukon avulla on myös usein helpompaa havainnollistaa sellaista dataa, jossa on useita eri mittayksiköitä, sillä tyypillisesti kaavioissa käytetään yhtä mittayksikköä per akseli. (Knafllic 2015, 47.)

Taulukko 1. Vuoden 2023 keskilämpötilat Jyväskylässä (Ilmatieteenlaitos n.d.)

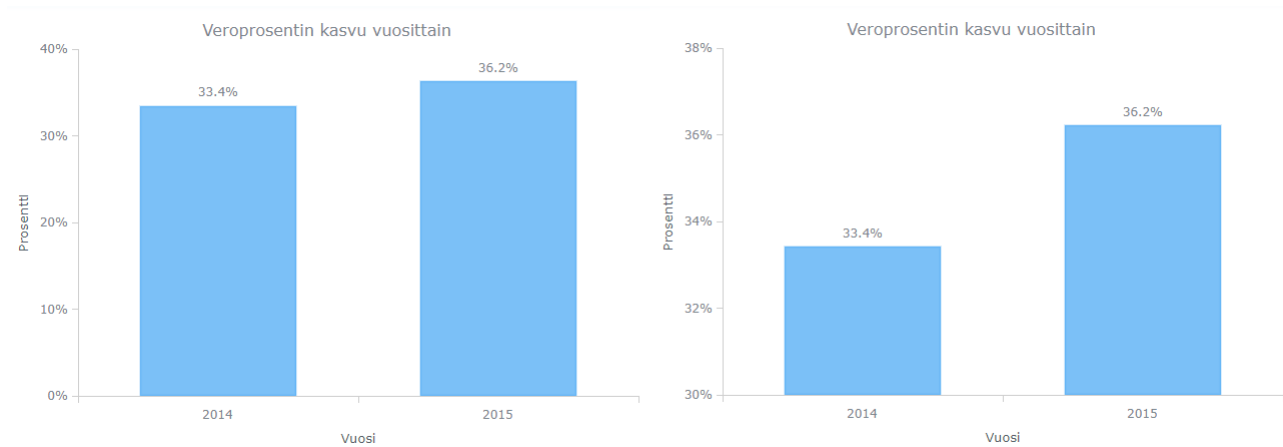
Havaintoasema	Vuosi	Kuukausi	Keskilämpötila [°C]
Jyväskylä lentoasema	2023	01	-3.4
Jyväskylä lentoasema	2023	02	-4.8
Jyväskylä lentoasema	2023	03	-4.7
Jyväskylä lentoasema	2023	04	2.5
Jyväskylä lentoasema	2023	05	10
Jyväskylä lentoasema	2023	06	15.4
Jyväskylä lentoasema	2023	07	15.9
Jyväskylä lentoasema	2023	08	16
Jyväskylä lentoasema	2023	09	13
Jyväskylä lentoasema	2023	10	1.1

Koska taulukon lukeminen on kuormittavaa, voidaan siihen käyttää erilaisia tehosteita sen luetavuuden helpottamiseksi. Taulukon rivit voidaan järjestää esimerkiksi suuresta pienempään. Suu-

rimman ja pienimmän arvon voi esimerkiksi värittää, jotta ne erottuvat. Koko taulukko voidaan myös värjätä lämpökartta-tyylisesti kuten taulukossa 1 näkyy, jolloin värin kylläisyys osoittaa, kuinka lähellä arvo on maksimi- tai minimiarvoa. (Knaflic 2015, 48.)

### 2.3.2 Pylväskaaviot

Pylväskaaviot ovat Yukin ja Diamondin (2014) mukaan erinomaisia vertailuiden käyttämiseen visualisoinnissa. Pylväskaavion palkit nousevat pystysuoraan. Toinen yleinen pylväskaavion kaltainen kaavio on palkkikaavio, jossa palkit ovat vaakasuorassa. (Yuk & Diamond 2014, 44.)



Kuvio 2. Tyypillinen sekä harhaanjohtava pylväskaavio

Pylväskaavio on yleisin tapa näyttää luokiteltua dataa. Kuvion 2 mukaisesti jokainen pylväs edustaa omaa luokkaansa. Mitä pidempi pylväs on, sitä suurempi on sen arvo. Pylvään pituus ei kuitenkaan välttämättä tarkoita, että arvo olisi parempi tai huonompi, vaan se riippuu datasta ja mihin tarkoitukseen se on suunniteltu. Yau (2013) vertaakin pylväskaaviota visuaaliseksi listaksi, jossa listan numerot ovat muunnettu omiksi pylväiksi. (Yau 2013, 143.)

Pylväskaavion tarkoituksena on vertailla pylväiden pituuseroja. Kaikki pylväät lähtevät samasta alkuviihvasta kuvion 2 mukaisesti. Näin eri palkkien koot ja suhteelliset pituuserot on helppo havaita. Kuviossa tulee myös ilmi, kuinka vaihtamalla alkuviihan arvoa saadaan luotua vääristetty mielikuva palkkien arvoista. Palkkien suhteellinen ero näyttää nyt suuremmalta, kun palkkien koko pituutta ei näytetä. Tämä johtaa lukijaa harhaan ja vääristelee dataa, joka ei ole eettistä. (Knaflic 2015, 58.)

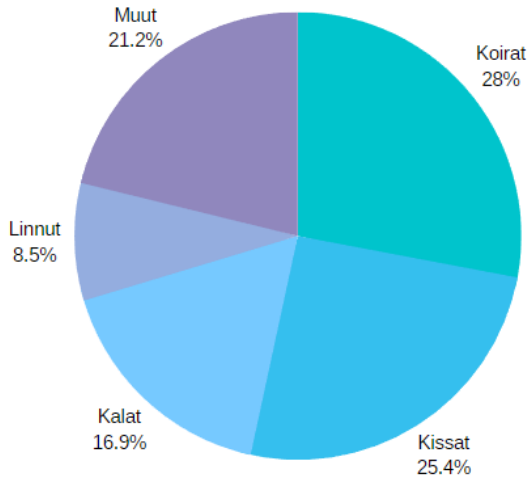
Knaflicin (2015) mukaan pylväskaavioita usein vältellään, koska ne ovat hyvin yleisesti käytettyjä. Hänen mielestään juuri sen takia niitä kannattaisikin hyödyntää, koska kaaviotyyppi on jo entuudestaan tuttu. Näin lukijan ei tarvitse opetella itse kaavion lukemista, vaan pystyy keskittymään olennaiseen, eli mitä informaatiota kaavio välittää. (Knaflic 2015, 56.)

### **2.3.3 Ympyrä- ja rengaskaaviot**

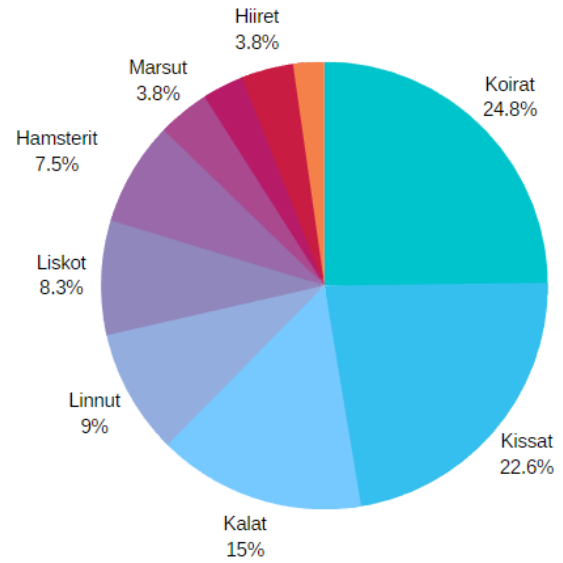
Ympyräkaavio on kaavio, joka on ympyrä ja sen pinta-ala jakautuu siivuihin. Siivun tarkoituksena on näyttää sen suhteellinen osuus täydestä ympyrästä. Täysi ympyrä on sata prosenttia kokonaisuudesta, eli kaikkien siivujen summa. Kokonaisuutta pitäisi pystyä tulkita pinta-alan avulla, mitkä siivut ympyrästä vievät eniten ja vähiten pinta-alaa. Ympyräkaaviossa siivun kulma auttaa myös sen hahmottamisessa. (Yuk & Diamond 2014, 45; Yau 2013, 98.)

Ympyräkaavion käyttö on jo pitkään ollut kiistanalaista. Monien asiantuntijoiden mielestä ympyräkaaviossa on vaikeampi hahmottaa siivujen eroavaisuuksia ja mahdotonta havaita trendejä, eli kehityssuuntaa. Sanotaan myös, että ihmissilmällä on vaikea arvioida pinta-aloja, johon koko ympyräkaavion käyttö perustuu. (Yuk & Diamond 2014, 44.) Vaikka ympyräkaaviosta on vaikea saada selville tarkkaa arvoa, voi siitä silti saada yleisen kuvan tilanteesta, kun siivuja ei ole monta (Yau 2013, 98).

Yleisimmät lemmikkieläimet

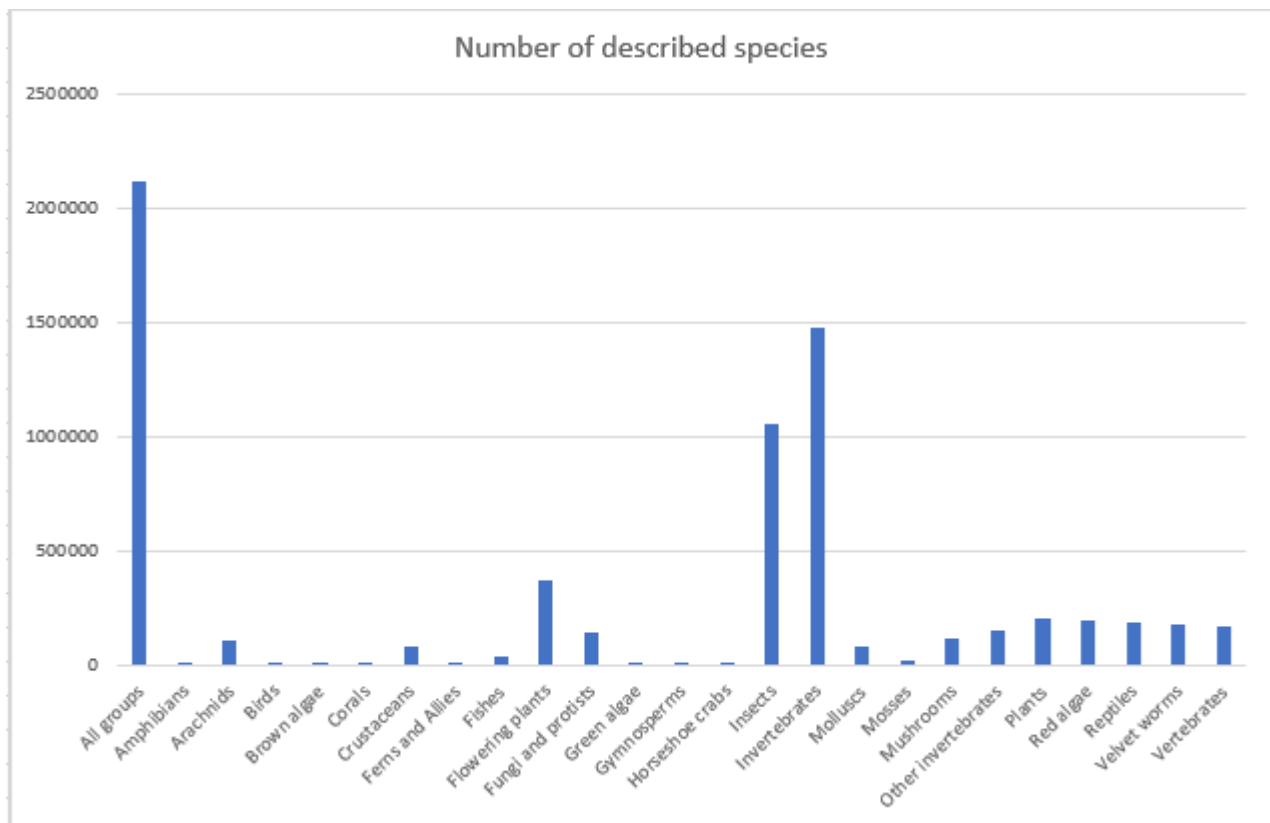


Yleisimmät lemmikkieläimet



Kuvio 3. Tyypillinen sekä liian täytetty ympyräkaavio

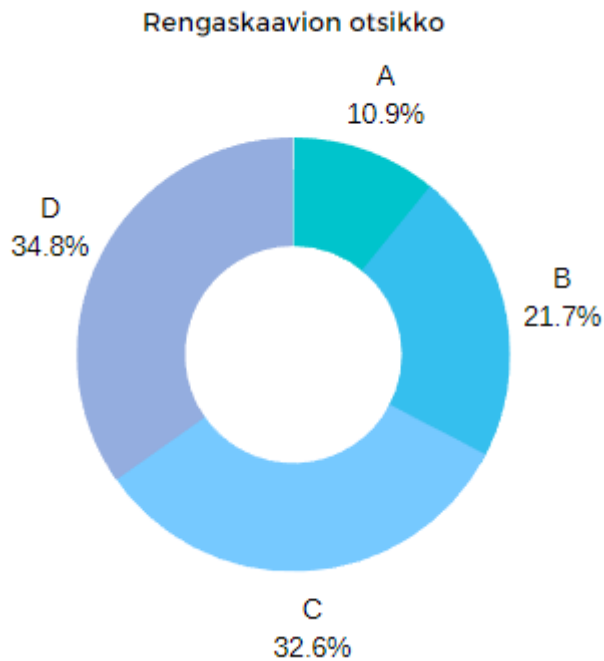
Yukin ja Diamondin (2014) mukaan ympyräkaavioita on kuitenkin mahdollista käyttää tehokkaasti, kunhan välttää sen suunnittelussa yleisimmät kompastuskivet, jotka vaikuttavat negatiivisesti sen luettavuuteen. Ympyräkaavion siivujen määrä kannattaa rajoittaa alle viiteen. Näin ympyräkaavio pysyy siistinä ja paremmin luettavana, sekä useiden pienien siivujen määrä ei häiritse käyttäjää kaavion pääasialta, kuten kuviossa 3 näkyy. (Yuk & Diamond 2014, 45; Yau 2013, 98.)



Kuvio 4. Pylväsdigrammi, jonka luvuista ei saa selvää (Eläinlajit 2022)

Kuviossa 4 on vaikea lukea osaa pylväistä. Käyttämällä samaa dataa ympyräkaaviossa data voidaan visualisoida alueen perusteella. Ympyräkaavion alueiden vertailu on hyödyllisempi, kun arvojen välillä on eksponentiaalisia eroja. Tilanteissa, joissa datan arvot eroavat suuresti, on vaikea havainnollistaa pylväskaaviolla. Osa pylväistä on niin tynkiä, että ne eivät juurikaan näy. Tosin ympyräkaavioita käyttäessä arvoja ei pystytä visualisoimaan täysin tarkasti, mutta siitä saa silti yleisen kuvan. (Yau 2013, 210.)

Rengaskaavio on hyvin samankaltainen ympyräkaavion kanssa, ja usein sitä mielletään ympyräkaavion läheiseksi serkuksi. Rengaskaavion havainnointi tapahtuu kuitenkin sen kaaren pituuden vertailulla. Renkaan keskiosan kulmat ovat poistettu, joten vertailuun niitä ei pysty käyttämään. (Yau 2013, 98.)



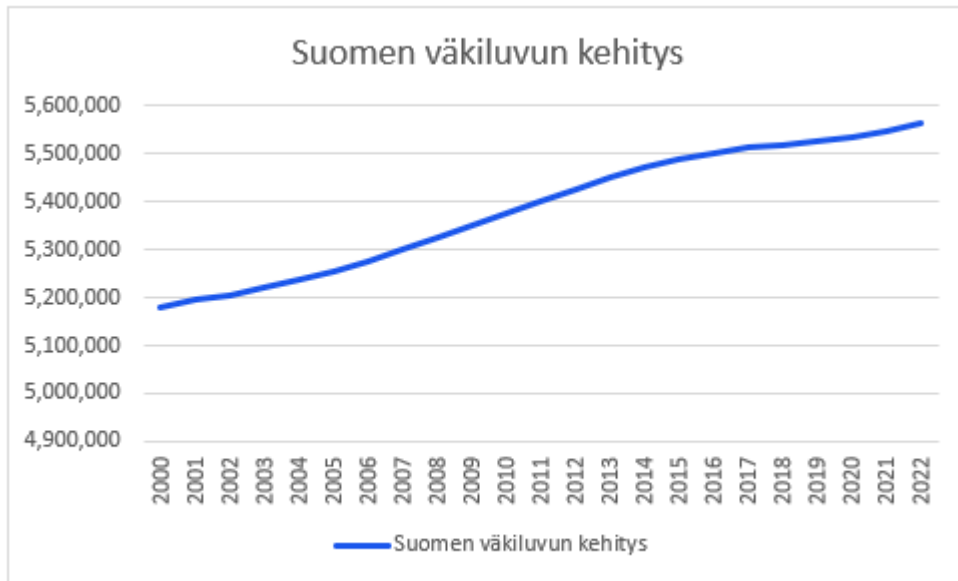
Kuvio 5. Tyypillinen rengaskaavio

Samaan tapaan kuin ympyräkaavio, on rengaskaaviosta vaikeampi hahmottaa tarkkoja lukuja verrattuna esimerkiksi pylväskaavioon. Kuten kuviossa 5 näkyy kaksi hyvin lähellä olevaa arvoa, on vaikea kertoa, kumpi niistä on isompi. Tätä voidaan selkiyttää laittamalla täsmentävät tekstit kuvion mukaisesti. (Knaflic 2015, 69.)

#### 2.3.4 Viivakaavio ja sen muunnelmat

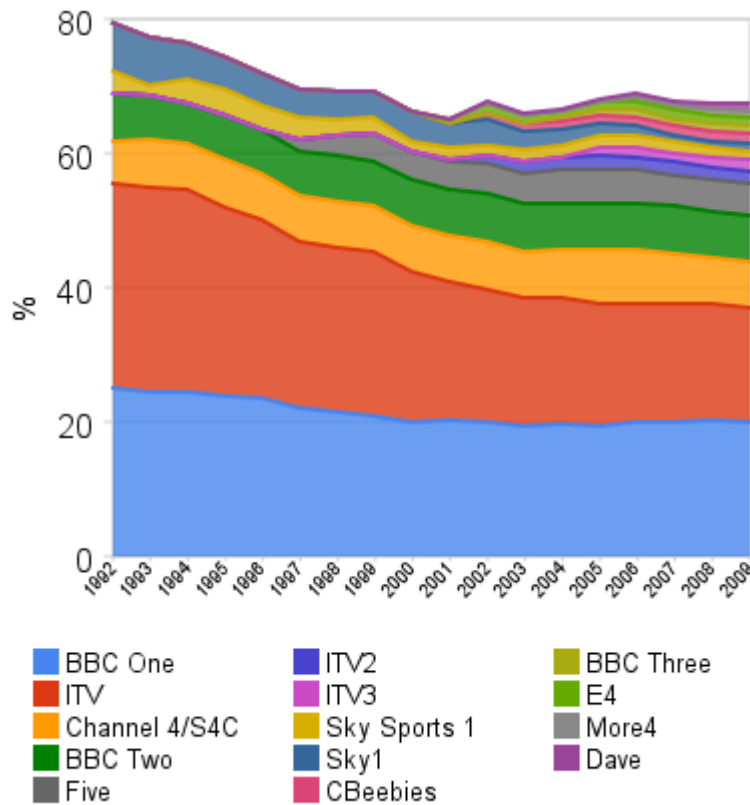
Viivakaaviota käytetään yleensä kuvaamaan jatkuvaa dataa. Koska data on jatkuvaa, se yleensä liittyy johonkin aikayksikköön. Pisteet ovat yhdistetty viivalla toisiinsa, joka implikoi yhteyttä pisteiden välillä. (Knaflic 2015, 52.) Pisteet jätetään kuitenkin harvoin näkyviin, jolloin vain viiva näkyy (Yau 2011, 118). Samassa kaaviossa voi olla useita viivoja, jolloin yksi viiva edustaa yhtä kategoriala (Knaflic 2015, 52).





Kuvio 6. Tyypillinen viivakaavio (Tilastokeskus 2023)

Viivakaavioilla seurataan usein trendejä. Tämä tarkoittaa katsomista yksittäisten pisteiden ulkopuolelle, jotta saadaan kokonaiskuva. Näin viivakaaviosta on mahdollista saada nopeasti tietoa siitä, laskeeko vai lisääntyykö jokin, kuten kuviosta 6 ilmenee. (Yau 2011, 92.)



Kuvio 7. Tyypillinen summaviivakaavio (Summaviivakaavio 2009)

Summaviivakaavio on viivakaavio, jossa viivat ovat pinottu toistensa päälle niin, että jokaisen viivan aloitusviiva on aikaisemman viivan kohdalla. Näin korkeimman viivan kohta toimii kokonaismääränä, eli summana kaikista viivoista, kuten kuviossa 7 näkyy. Tämän avulla on nopeaa nähdä kokonaismäärä sekä yksittäisten viivojen arvo ja niiden panos kokonaisuuteen. (Yi n.d.)

## 2.4 Visuaalinen suunnittelu

Suunnittelu on harvoin selkeä ja yksisuuntainen prosessi. Suunnittelussa joutuu usein ottamaan huomioon monia asioita. Kirkin (2012) mielestä on oltava avoin muutoksille ja ideoille, joita saatetaan tulla vasta myöhemmissä vaiheissa. Hän pitää suunnittelua iteratiivisena prosessina, jolloin on välillä otettava peruutusaskelia. Kirk kuvaakin suunnitteluprosessia parhaan tien etsimisenä suunnitteluvaihtojen miinakentän läpi. (Kirk 2012, 37.)

Datan visualisoinnin pitäisi myös kertoa selkeä tarina. Tässä tapauksessa tarinalla tarkoitetaan, että datalla on jokin tarkoitus, yleensä esimerkiksi tavoitteiden saavutus tai ongelman ratkaisu.

(Yuk & Diamond 2014, 90.) Knafllic (2015, 28) listaa kirjassaan myös tarinan kertomisen yhtenä tärkeimpänä oppina visualisoinnin suunnitteluun.

#### 2.4.1 Tekstien merkitys

Datan visualisointi ei tarkoita pelkästään vain kuvioita, vaan siihen liittyy myös ympärille muodostuvat tekstit. Teksti täydentää katseltavaa kaaviota ja parantaa sen luettavuutta. Hyödyllisin tapa lisätä tekstiä kaavioihin on lisätä siihen kaavion ja akseleiden otsikot. Nämä ovat hyvin tärkeä osa kaaviota, sillä ne kertovat lukijalle, mitä kaaviossa on. Kaavion otsikko on tyypillisesti myös ensimmäinen asia, jonka lukija näkee. (Yuk & Diamond 2014, 43.)

Tekstin lisääminen kaavioon estää usein sen väärintulkintaa. Jokainen tulkitsee näkyvän omalla tavallaan. Selkeän tekstin lisääminen visuaaliseen kaavioon auttaa myös lukijaa, sillä lukijan ei tarvitse etsiä ja prosessoida tietoa pelkästään kaaviosta. Se nopeuttaa myös kaavion tulkintaa, kun lukijalla on jo idea otsikon avulla, mitä kaaviossa tullaan esittämään. (Yuk & Diamond 2014, 63.) Yau (2013) perustelee tekstien lisäämistä myös sillä, että ilman tekstiä on vaikea välillä edes tulkita mitä kaavion data ilmaisee. Ilman tekstiä yhteys kaavioon ja taustalla olevan datan välillä katkeaa. Yau (2013, 206.)

Yuk ja Diamond (2014) kehottavat välttämään turhan tekstin lisäämistä. Teksti kannattaa jättää lyhyeksi, jotta se pysyy hyödyllisenä, koska usein tilaa on käytettävissä vain rajoitetusti. Jokaisen tekstin tulisi täydentää kaaviota. Jos tilaa ei ole hirveästi, on mahdollista käyttää vain yksinkertaisia sanoja tai yksittäisiä kirjaimia. Kun teksti on lyhyt ja selkeä, lukijan on helppo ymmärtää sen yhdellä silmäyksellä. (Yuk & Diamond 2014, 55.)

Kaikkea ei kuitenkaan tarvitse visualisoida. Jos dataa on hyvin vähän, voi harkita datan näyttämistä sellaisenaan. Datan visualisoinnin tarkoituksena on ymmärtää yhteyksiä ja ilmentyviä trendejä. Jos niitä ei ole, ei ole tarpeellista lähteä visualisoimaan vain muutamaa arvoa. (Yau 2013, 248.) Knafllic (2015, 45) ehdottaa tällöin yksinkertaisen tekstin lisäämistä, joka selittää kyseisen arvon.

### 2.4.2 Värien merkitys

Huolellisesti laadittu värikartta voi antaa datalle paljon kontekstia. Värien avulla voi visualisoida paljon informaatiota, koska ne eivät ole riippuvaisia sijainnista tai koosta. Eri väreillä yleensä erotellaan ryhmiteltyä dataa, jolloin yksi väri tarkoittaa yhtä ryhmää. Numeerisessa datassa on yleensä kahtiajako väreille; positiiviset ja negatiiviset arvot havainnollistetaan eri väreillä. Värin kylläisyys on myös tärkeä elementti, jolla voidaan ilmaista eroavaisuuksia datassa. Eri värejä sekä värin kylläisyyttä hyödyntäen on käytössä laaja valikoima visualisoinnin hyödyntämiseen. (Yau 2013, 103.)

Värien valinta on hyvin tärkeää. Kirkkaat värit vievät yleensä ensimmäisenä huomion, jolloin vähemmän värikkäät jäävät takasijalle tai joskus jopa lukematta. Kirkkaita värejä hyödyntäen voidaan ohjata lukijan katsetta kiinnittämään huomiota tärkeisiin kohtiin. (Yuk & Diamond 2014, 43.) Väreillä korostaminen auttaa myös lukijaa vahvistamaan kaavion sanoman. Huono värivalinta voi taas johtaa väärintulkintaan. (Yau 2013, 221.)

Datan visualisoinnissa punainen, keltainen ja vihreä ovat hyvin yleisesti käytettyjä ja niihin liittyy automaattisesti mielikuvia. Punainen kuvastaa, että jokin on pielessä ja keltainen tarkoittaa, että jokin on menossa huonompaan suuntaan. Vihreä taas indikoi, että kaikki on kunnossa. Tämän takia Yukin ja Diamondin (2014) mielestä näitä värejä tulisi käyttää vain ylempänä mainituissa tilanteissa, kun näytetään ilmoituksia datasta. (Yuk & Diamond 2014, 61.)

### 2.4.3 Tilan käyttö

Yleisesti kaaviossa palkkien tulisi olla suhteellisesti laajempia kuin palkkien valkoiset välit. On tärkeää välttää palkkien liiallista leveyttä, jolloin lukija voi erehtyä vertaamaan alueita pituuksien sijasta. (Knaflic 2015, 58.)

Datan visualisoinnissa täytyy ottaa myös huomioon datan ja kaavioiden välissä olevat tilat. Jos kaikki on tiivistetty hyvin lähelle toisiaan, huonontaa se luettavuutta ja kaavio näyttää mahdollisesti sekavalta. Kaaviossa eri osioiden välille kannattaa jättää tilaa, jotta datan jäsentelystä tulee helpompaa. Useiden kaavioiden välille on myös kannattavaa jättää tilaa, jotta lukija pystyy keskittymään kerrallaan yhteen kaavion ja paloittelemaan luetun. (Yau 2013, 217.)

Knafllic (2015) vertaa kirjassaan tyhjän tilan jättämistä kaavoihin puheessa pidettäviin taukoihin. Ei ole luonnollista olla pitämättä mitään taukoja. Tyhjän tilan käyttö strategisesti auttaa kiinnittämään huomion sivun tärkeisiin osiin, jossa data sijaitsee. Hänen nyrkkisääntönänsä on, että reunat jätetään tyhjäksi, jolloin visuaalinen osuus ei levity koko sivun laajuudelle. Muissa paikoissa on myös hyvä miettiä, kuinka tyhjällä tilalla pystyy painottamaan tärkeää kaaviota. Jos kaavio on erityisen tärkeä, voi sen mahdollisesti myös jättää ainoaksi asiaksi koko sivulle. (Knafllic 2015, 86.)

#### **2.4.4 Datan asettelu ja yhtenäistäminen**

On tutkitusti todettu, että ihmiset lukevat tekstejä ja kuvia tietyn kaavan mukaisesti. Kun asiat näyttävät samanlaisilta, ne myös koetaan samantyyppisinä. On luonnollista olettaa, että samankaltaiset muodot, värit ja muut asiat liittyvät toisiinsa. Tämä auttaa kaavion lukijaa tekemään yhteyksiä dataan ja ryhmittelemään niitä. (Yuk & Diamond 2014, 69–71.)

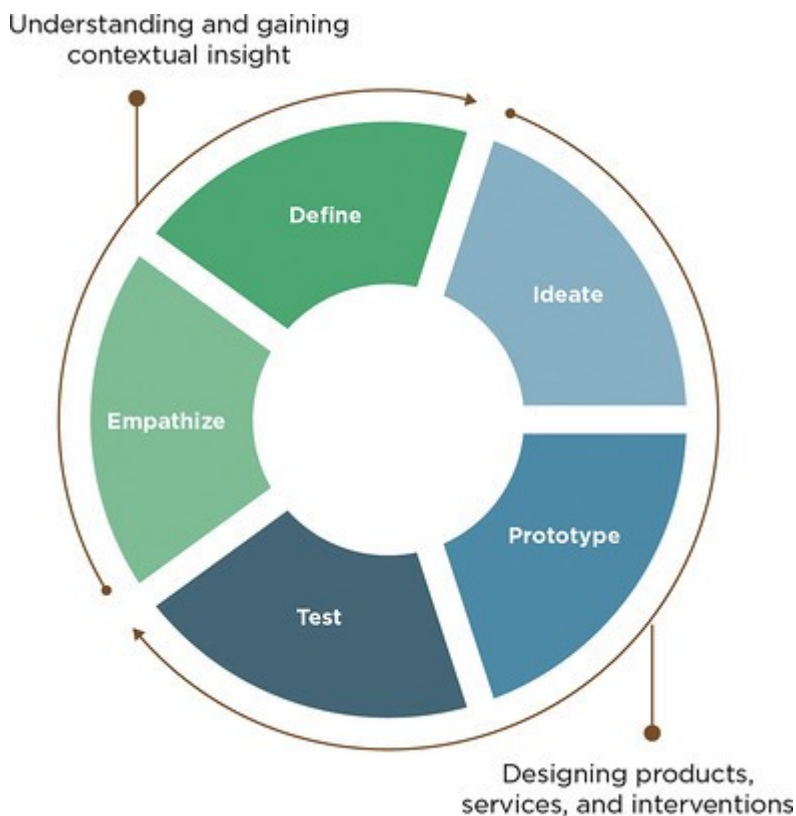
Tämän takia kannattaa yleisesti hyödyntää ”Z” tai ”F” -kaavan mukaisia suunnitelmia, eli ensimmäiseksi kannattaa laittaa kaikista merkittävintä tieto. (Knafllic 2015, 62.) Tämä kaava toimii myös kaaviossa. Kaavan mukaan yleinen käytäntö on laittaa kaavion otsikko ylös ja seuraavaksi vasta visuaaliset elementit. Hyvin järjestelty kaavio vaikuttaa sen lukunopeuteen ja auttaa vähentämään lukijan kuormaa. (Yuk & Diamond 2014, 71.)

Luokitellun kaavion suunnittelussa, esimerkiksi pylväskaaviossa, on tärkeää ottaa huomioon, kuinka eri kategoriat on järjestelty. Numeerisen datan järjestelyssä on luonnollista järjestää data siten, että data lähtee pienimmästä suurimpaan, tai toisinpäin. Jos kategorioilla ei ole mitään selkeää järjestystä, jota hyödyntää, voi datan järjestää esimerkiksi tärkeimmästä vähemmän merkittävään. (Knafllic 2015, 62.)

Datan visualisoinnissa yhtenä päätarkoituksena on antaa lukijan tehdä vertailuja datapisteiden välillä. Datan visualisointi kannattaa, vaikka datan arvot olisivat yhtä suuria. Tämä antaa silti lukijalle mahdollisuuden tehdä vertailun ja muodostaa siitä johtopäätökset. Monet yleiset kaaviot, kuten pylväs-, viiva- ja pistekaaviot, ovat suunniteltu juuri vertailujen tekemisen avuksi. Niillä saadaan abstrakti data yksinkertaisiin geometrisiin muotoihin, jolloin vertailuun pystyy käyttämään pituutta, paikkaa, suuntaa tai aluetta. Tekemällä pieniä muutoksia näihin voidaan parantaa kaavion luettavuutta. (Yau 2013, 208.)

## 2.5 Design Process

Design process, eli suomeksi suunnitteluprosessi, on iteratiivinen menetelmä, joka tarjoaa ratkaisupohjaisen lähestymistavan luovaan prosessiin. Prosessin tarkoituksena on ohjata ja selkiyttää monimutkaisia tavoitteita jakamalla ne pienemmiksi paloiksi. Tavoitteena on vähentää riippuvuutta luovuuteen ja inspiraation avulla työskentelyyn, jotta tuotteliaisuus pysyy tasaisena. Prosessilla voidaan ehkäistä konseptien liiallista ajautumista väärään suuntaan, sillä prosessi sisältää vaiheita niiden estämiseen. (Drysdale n.d.)



Kuvio 8. Suunnitteluprosessin vaiheet (Suunnitteluprosessi 2022)

Lähteistä riippuen suunnitteluprosessiin sisältyy eri määrä vaiheita tai vaiheita kutsutaan eri termeillä, koska se ei ole yksinkertainen lista, vaan työkalu. Sisältö ja ydinajatus pysyy kuitenkin yhtenäisenä. (Phan 2022.) Kuviossa 8 näkyy yleisimmät vaiheet. Prosessin tarkoituksena on edetä vaihe kerrallaan, kunnes työ on valmis. Sitä myös voidaan toistaa useita kertoja. Prosessia ei ole tarkoitettu täysin lineaariseksi, vaan aina on mahdollista palata aikaisempaan vaiheeseen, jos työssä

ilmenee epäselvyyksiä tai ongelmia. Tämä auttaa tilanteissa, jossa jumitutaan tai asiaa joudutaan miettimään uudelleen, kun vaiheessa tulee ilmi uusia asioita. (Drysdale n.d.)

Ensimmäinen vaihe suunnitteluprosessissa on eläytyminen. Tässä vaiheessa tunnistetaan ja niin sanotusti eläydytään kohdekäyttäjiin. Tarkoituksena on saada ymmärrys käyttäjään ja heidän tarpeisiinsa. (Phan 2022.)

Toinen vaihe, määrittely, keskittyy ongelmien, tavoitteiden ja rajoitusten tarkentamiseen. Kun ensimmäisessä vaiheessa on tunnistettu tulevat käyttäjät, nyt tarkoituksena on rajata heidän tarpeistaan työlle merkitsevimmät kohdat. Vastataan kysymyksiin, kuten mitä heidän ongelmansa ovat tai kuinka he käsittelevät niitä tällä hetkellä. (Phan 2022.)

Kolmannessa vaiheessa tärkeintä on ideointi. Kehitetään ideoita, haastetaan oletuksia ja lopulta luodaan ratkaisun ongelmaan, joka on määritelty aiemmissa vaiheissa. Tutkitaan, kuinka kerätty tieto ja informaatio saattavat vaikuttaa suunnitelmaan. (Phan 2022.)

Neljännessä vaiheessa muodostetaan jonkinlainen prototyyppi ideointivaiheesta syntyneistä suunnitelmista. Prototyypillä tarkoitetaan yleensä ensimmäistä mallia suunnitelmasta. Prototyypin toteutuksen muoto vaihtelee projektin mukaan. Yleensä prototyyppi on jollain suunnitteluohjelmistolla toteutettu ratkaisu. (Drysdale n.d.)

Viidennessä vaiheessa olennaisinta on testaus ja palautteen saanti. Aikaisemmassa vaiheessa tehtyä tuotosta arvioidaan laajemmalla yleisöllä, joko sisäisellä tiimillä tai jopa asiakkailla. Suunnitteluprosessi ei siis ole, että ensimmäisellä kerralla suunnittelu olisi jo valmista, eikä virheitä olisi. Joskus ympäristö ja lähtötiedot muuttuvat, jolloin isomman ryhmän on helpompi havaita ongelmakohdat. (Drysdale n.d.) Sekä Drysdale (n.d) että Phan (2022) suosittelevat palautteen huomiointamista ja sen avulla uudelleen suunnittelemista. Palautteen avulla voidaan siirtyä takaisin ideointivaiheeseen, jossa voidaan jatkaa suunnittelua tai tehdä muutoksia palautteen perusteella. On hyödyllistä, kun suunnitteluun tehdään jatkuvasti parannuksia ja päivityksiä kohdekäyttäjien palautteen pohjalta. Näin ohjelma pysyy ajan tasalla. (Phan 2022.)

## 3 Toteutus

### 3.1 Suunnittelu

Opinnäytetyön suunnittelussa hyödynnettiin suunnitteluprosessin tarjoamaa metodologiaa. Suunnitteluprosessi oli iteratiivista. Opinnäytetyön toteutuksessa pidettiin useita katselmointeja suunnittelun aikana, joista kerättiin palautetta. Palaute huomioitiin, jolloin yleensä siirryttiin takaisin ideointivaiheeseen miettimään ja suunnittelemaan asiaa uudelleen.

Ajatusten keräämiseen ja suunnitteluun käytettiin Miro-ohjelmaa. Miro on verkkopohjainen suunnitteluohjelma tiimeille, jonka avulla pystyy tekemään monipuolista ryhmätyötä eri ominaisuuksilla (Miro 2023).

#### 3.1.1 Käyttäjätarinat

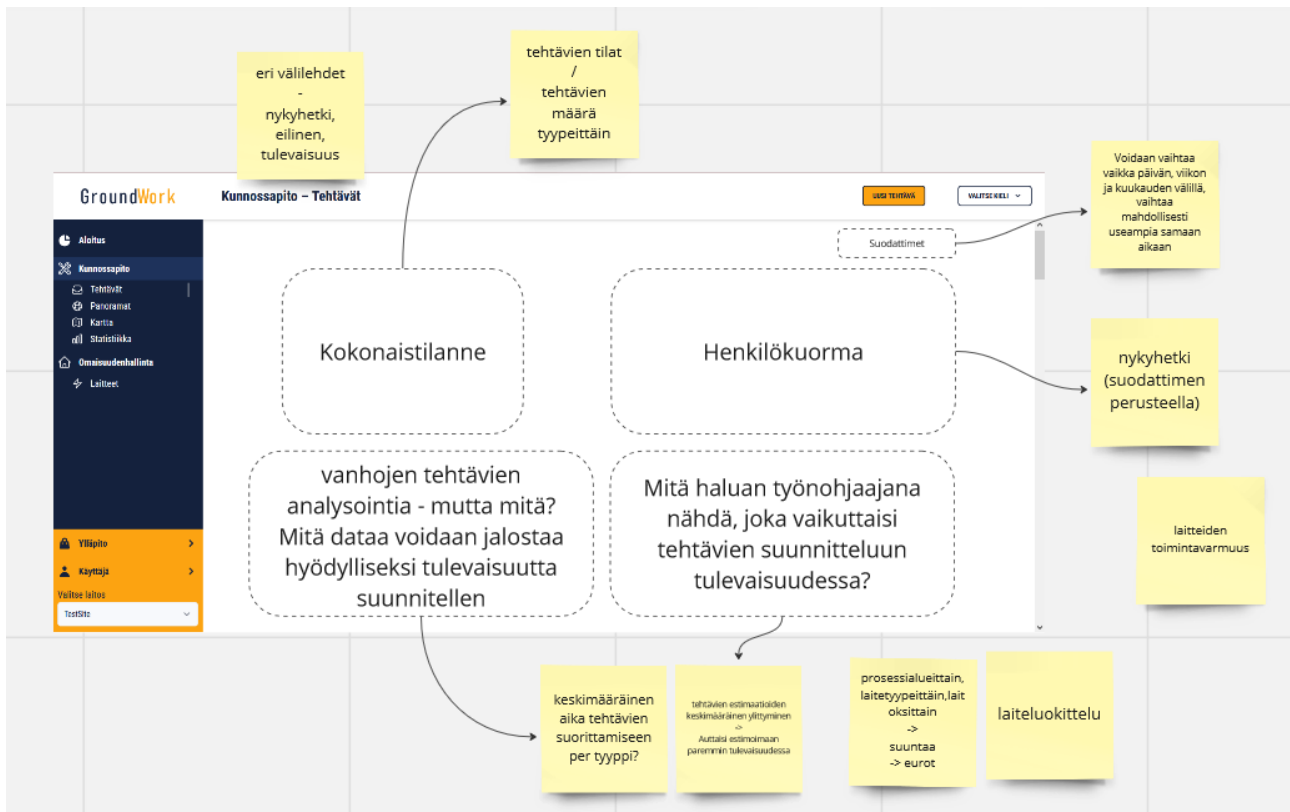
Kun opinnäytetyön aihe ja kohde oli selvillä, sitä ruvettiin määrittelemään tarkemmin. Koska toteutettava moduuli tulee jo valmiiksi olemassa olevaan ohjelmaan, ei itse kohdekäyttäjää tarvinnut pohtia tai tunnistaa. Suunnittelussa tulikin miettiä, mitä nykyiset käyttäjät haluisivat nähdä datastaan, jotka hyödyttäisivät heitä.

Suunniteltavan moduulin käyttöön liittyviä asioita tunnistettiin ja rajattiin tekemällä käyttäjätarinoita kohdekäyttäjistä. Käyttäjätarina on kuvaus ohjelmiston ominaisuudesta, joka on kirjoitettu loppukäyttäjän tai asiakkaan näkökulmasta. Käyttäjätarinan perusmuoto on: [käyttäjänä], [haluan] x, [jotta voin] y. (Rehkopf n.d.) Esimerkkinä moduuliin keksittiin ”Työnohjaajana, haluan tietää mitä minun pitää seuraavaksi tehdä, jotta aloitan tekemään oikeaa asiaa”.

#### 3.1.2 Ideointi

Käyttäjätarinoiden jälkeen aloitettiin hahmottamaan niiden pohjalta muodostuvia KPI-lukuja, sekä muita laitosylläpitoon liittyviä arvoja, joita voitaisiin visualisoida. Samalla perehdyttiin ohjelman muista moduuleista saatavaan ylläpitodataan ja hahmoteltiin myös niistä soveltuvaa dataa. Jos soveltuvaa dataa ei niistä löytyisi, vaihtoehtona oli myös suunnitella muiden moduulien muokkausta, jotta saadaan haluttua dataa.





Kuvio 9. Karkea luonnos

Ideoinnin aloituksessa hyödynnettiin Yukin ja Diamondin (2014) esittelemää Business Intelligence Dashboard Formula -kaavaa, jotta suunnittelussa saatiin käyntiin. Kaavassa käyttäjän vaatimukset käännetään neliosaiseksi kaavioksi, jotka vastaavat seuraaviin kysymyksiin:

- Mikä on nykyinen tilanteesi?
- Kuinka päädyit tänne?
- Mihin päädyt?
- Mitä voit muuttaa saavuttaaksesi tai ylittääksesi tavoitteesi?

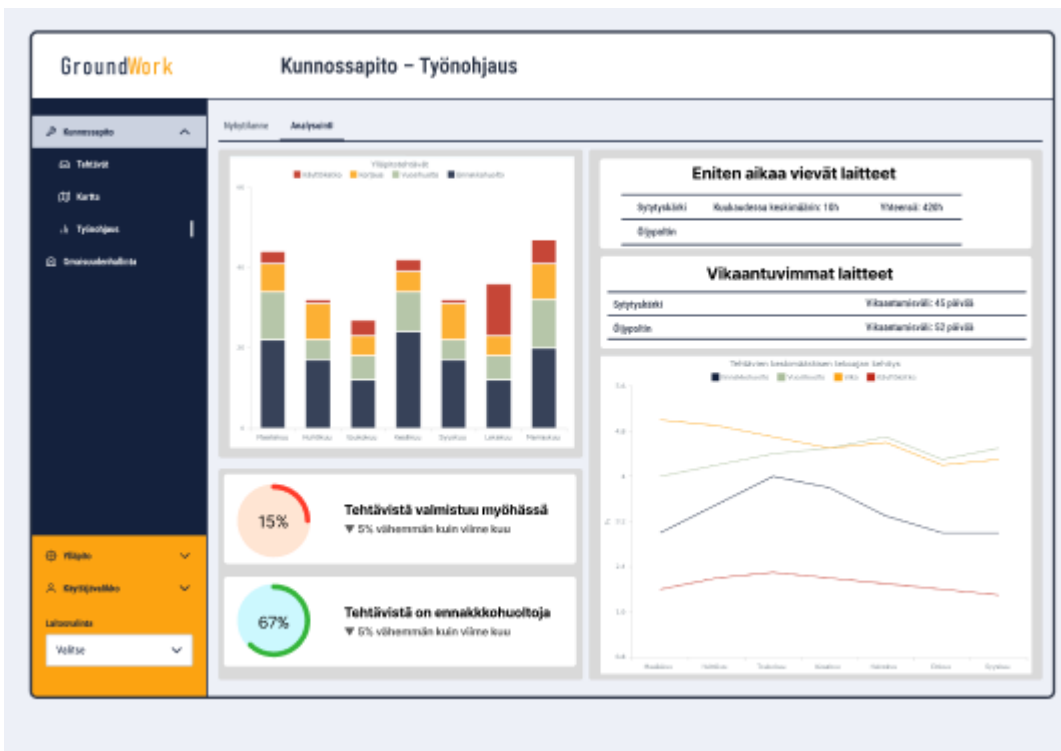
(Yuk & Diamond 2014, 83.) Kuviossa 9 näkyy yksi ensimmäisistä karkeista luonnoksista Miro ohjelmassa, joka on täytetty useilla ideoilla. Pohjana toimi kuvankaappaus tyhjästä GroundWorkin sivusta, johon kirjoitettiin Mirolla tekstiä ja kuvia. Kuvion muistilappuja lisättiin vähitellen, kun ideoita syntyi tai ideoita tuli esille opinnäytetyön katselmoinnissa. Niitä kerättiin niin paljon kuin vain tuli ideoita. Vasta myöhemmin poimittiin parhaimmat ideat suunnittelua varten.

Hyvin alussa tuli jo selväksi, että kaikki kaaviot eivät tulisi mahtumaan yhdelle sivulle, koska sivua ei haluttu ahtaa täyteen. Osa ideoista erosi toisistaan myös sen verran, että loogisinta olisi järjes-

tellä ne eri sivuille, jotta datan tarina pysyisi johdonmukaisena. Näin päädyttiin jo alkuvaiheessa kahden välilehden ajatukseen. Ensimmäinen välilehti kuvastaisi lähinnä nykyhetkeä ja toinen sisältäisi enemmän data-analytiikkaa. Tässä vaiheessa keskityttiin kuitenkin vain suurin piirtein ideoiden järjestykseen.

### 3.1.3 Prototyyppi

Kun koettiin, että ideoita oli jo sen verran, että niitä voisi alkaa visualisoimaan paremmin, siirryttiin Figmaan tekemään suunnitelmia ideoiden pohjalta. Figma on suunnittelutyökalu, jonka avulla voi luoda lähes mitä tahansa; kokonaisista sivuista ja sovelluksista logoihin (Twarog 2021).



Kuvio 10. Figmassa suunniteltu versio analysoinnin välilehdestä

Kuviossa 10 näkyy yksi Figmassa suunnitelluista versioista. Suunnitelmaa päivitettiin useaan kertaan saadun palautteen myötä.

## 3.2 Ohjelmointi

Opinnäytetyötä lähdettiin toteuttamaan ensimmäisenä eniten back endin puolelta. Front end jätettiin tarkoituksella hyvin karkeaksi, kunnes back end oli suurimmalta osin tehty. Visuaalinen ilme haluttiin jättää viimeiseksi, sillä siinä tulisi olemaan eniten hiottavaa.

### 3.2.1 Käytetyt teknologiat

Opinnäytetyön toteutus tehtiin Laravel-sovelluskehikseen. Laravel on full stack selainpohjainen sovelluskehys, eli se pystyy hoitamaan front endin sekä back endin. Sovelluskehys luo pohjan ohjelmalle ja tarjoaa valmiita toiminnallisuuksia, joita ei itse tarvitse tehdä alusta alkaen. Laravelin back end perustuu PHP-kieleen ja front endissä käytetään JavaScriptiä. (Laravel n.d.)

Toteutuksen komponenteissa hyödynnettiin myös Livewire-sovelluskehystä. Livewire toimii Laravelin päällä antaen sille lisää ominaisuuksia. Livewiren ydintoimintona on luoda uudelleen käytettäviä komponentteja, jotka pystyvät esimerkiksi kommunikoimaan back endin kanssa ilman sivun lataamista uudelleen.

Kaavioiden piirtoon käytettiin JavaScriptiin pohjautuvaa AnyChart-kirjastoa. AnyChart on ollut kehitteillä jo vuodesta 2003, mutta sitä päivitetään vielä jatkuvasti. AnyChartin tarjoaa valmiiksi paljon ominaisuuksia, joten sen käyttöönotto ja kaavion upottaminen valmiille sivulle on yksinkertaista. (AnyChart n.d.)

### 3.2.2 Backend

Hyväksytyjen prototyyppien jälkeen aloitettiin toteutus. Pohja koodille aloitettiin luomalla development nimisestä haarasta uusi haara, johon koodi toteutetaan.

```

<?php
namespace App\Http\Controllers;

class DataVisualizationController extends Controller
{
    public function index()
    {
        return view('data_visualization.index');
    }
}

```

Kuvio 11. Yksinkertainen controller

Ensimmäisenä luotiin Laravelille tyypillinen controller sekä sille kuuluva blade.php. Tämä tulee toimimaan koko sivun pohjana, johon upotetaan muut komponentit. Kuvio 11 näkyy, kuinka yksinkertaisena controller toimii; se palauttaa vain näkymän. Kun pohja oli testattu, toteutettiin siihen suunniteltu välilehden-toiminto, jotta komponentit voidaan sijoittaa suunniteltuun kohtaan. Muuta varsinaista koodia ei tule pohjalle.

Taulukko 2. Livewire komponentit

Komponentin nimi	Välilehti
Calendar	Analysointi
EventLog	Nykyhetki
LateTaskCount	Analysointi
MostFailingEquipment	Analysointi
MostTimeConsumingEquipment	Analysointi
StackedAreaChart	Analysointi
TaskDonut	Nykyhetki
TaskList	Nykyhetki
TaskPercentage	Analysointi

Seuraavaksi luotiin taulukkoon 2 listatut Livewire-komponenttien perustat. Koska komponentteja on monta, aloitettiin niiden toteutus ensimmäiseltä välilehdeltä. Suunnitelmien mukaan useampi komponentti tällä välilehdellä on yksinkertaisempi toteuttaa, kuin analysointivälilehdellä. Kun ensimmäisen välilehden komponentit ovat suurin piirtein toteutettu, siirryttäisiin seuraavaan välilehteen. Seuraavaksi tullaan käymään komponenttien toteutukset yksitellen.

## TaskDonut

Ensimmäisen välilehden ainoa visualisointi on kyseisen viikon tehtävien statuksen visualisointi rengaskaavion muodossa. Backendin puolelta moduuli on suhteellisen yksinkertainen. Koska data ei tule muuttumaan missään vaiheessa, kaikki data voidaan hakea mount-funktiossa ja sen jälkeen palauttaa frontendiin.

Rengaskaavioon tulee dataa kolmesta eri hausta. Ensimmäiseksi haetaan kyseisellä viikolla suljetut tehtävät. Toisella haulla haetaan kaikki tehtävät, jotka ovat "in progress" tilassa, eli niitä ollaan parhaillaan tekemässä. Kolmantena hakuna haetaan kaikki avoimet tehtävät, jotka ovat joko "open" tai "to do" -tilassa. Tässä haussa tuli ottaa huomioon myös, että on mahdollista olla tehtäviä, jotka ovat avoimia, mutta ne ovat ajastettu alkamaan vasta kyseisen viikon jälkeen. Tällaiset tehtävät tuli siis suodattaa hausta pois.

## TaskList

Seuraava komponentti nykyhetkivälilehdellä on myöhässä olevien sekä kriittisten tehtävien listaus. Vaikka listaukset ovat erilaisia, oli silti loogista toteuttaa ne saman komponentin alle, sillä ne ovat suhteellisen yksinkertaisia staattisia listauksia.

```
public function mount()
{
    $getTasks = Task::where('end_time', '<', (new DateTime())->format('Y-m-d'))->whereNull('done')->get();
    $this->lateTasks = $getTasks->sortBy('end_time');

    $criticality = $this->getTaskPriorities();
    $this->criticalTasks = Task::whereIn('priority', $criticality)->whereNull('done')->get();
}
```

### Kuvio 12. TaskList-komponentin mount-funktio

Molempien tietokantahaut olivat yksinkertaisia, kuten kuviosta 12 näkyy. Myöhästyneissä tehtävissä tuli ottaa huomioon tehtävän suunniteltu lopetuspäivämäärä ja verrata sitä nykyiseen päivämäärään. Molemmissa oli otettava huomioon myös vain tehtävät, jotka ovat tekemättä. Tämä ilmenee molemmissa haussa, kun haetaan tyhjää "done" -saraketta, eli kyseistä tehtävää ei ole vielä tehty, koska sillä ei ole suorituspäivämäärää.

Kriittisten tehtävien haussa käytettiin apufunktiota, jotta saataisiin haettua vain tärkeiksi merkityt tehtävät. Apufunktio hakee tehtävien config-tiedostosta eri prioriteetit ja palauttaa listana vain korkeimmat prioriteetit. GroundWorkissa on tällä hetkellä 4 eri prioriteetin tasoa, joista kaksi korkeinta on ”korkea” ja ”kriittinen”. Ainoastaan nämä tasot tulee näyttää korkean prioriteetin listalla.

## EventLog

Eventlog, eli tapahtumaloki, vaatii enemmän teknistä suunnittelua. Tapahtumalokin tarkoituksena on kerätä tietokantatauluun GroundWorkissa asiakkaan käyttäjien tapahtuvista päivityksistä, liisäyksistä, poistoista ja kaikesta muusta. Tämän avulla käyttäjä näkisi, mitä on viime aikoina GroundWorkissa tapahtunut ja mahdollisesti reagoida sen mukaan. Tapahtumalokin oletusaikaväliksi asetettiin nykyinen viikko.

EventLog	
id	integer
created_by	integer
customer_site	integer
event_type	text
detail	text
targetable_id	integer
targetable_type	integer
created_at	datetime

Kuvio 13. Tapahtumalokille suunniteltu tietokantataulu

Ensimmäisenä suunniteltiin tapahtumalokiin tarvittava tietokantataulu migraatiolla. Tapahtumaloki suunniteltiin keräämään tehtävissä ja laitteissa tapahtuvia muutoksia sekä niiden luontea ja että poistoja. Myöhemmin on mahdollista tulla tarve kerätä myös muita muutoksia, joten taulu täytyi suunnitella kasvutarve huomioiden. Eloquentin polymorfinen relaatio mahdollisti eri taulujen yhdistämisen käytännöllisesti. Tätä varten täytyi luoda yksi id-sarake, joka osoittaa laitteen tai

tehtävän pääavainta. Toinen luotava sarake on tyyppisarake, johon tallennetaan mallin luokka, kuten kuviossa 13 näkyy. Näin tiedetään mille mallille id kuuluu. Tämä mahdollistaa käyttöliittymän puolella joko laite- tai tehtäväkortin avaamisen oikein, kun käyttäjä klikkaa lokiriviä.

Seuraavaksi tapahtumalokille täytyi luoda toiminnallisuudet, jossa lokidata kerätään. Toiminnallisuudet toteutettiin Observer-luokkiin. Observer luokat kuuntelevat Eloquent tapahtumia. Tehtäville ja laitteille luotiin funktiot: created, updated ja deleted.

```
->when($this->filterUser, function ($query, $filterUser) {
    $query->where('event_log.created_by', 'LIKE', "%$filterUser%");
})
->when($this->filterEvent, function ($query, $filterEvent) {
    $query->where('event_log.event_type', 'LIKE', "%$filterEvent%");
})
```

Kuvio 14. Osa tapahtumalokin suodatustoimintaa

Tapahtumalokille toteutettiin myös suodatus, jotta halutessaan käyttäjä pystyy näkemään aikaisempia tapahtumia, mitä oletusaikaväli antaa nähdä. Suodatustoimintoon lisättiin myös käyttäjien sekä tapahtumien tyyppien mukaan suodatus. Kuviossa 14 näkyy osa suodatustoiminnon toteutuksesta backendissä. Suodatustoiminto on osana hakua render-funktiossa, jotka aktivoituvat, kun käyttäjä hyödyntää suodatusta. Kun muuttujat eivät ole enää tyhjiä, haussa otetaan huomioon when-metodit.

### TasksPercentage

Analysointivälilehden moduulin tarkoitus on luoda tehtävien myöhästymisten sekä korjaustehtävien prosentuaalisen määrän. Vaikka laskettavat arvot ovat toisiinsa verrattuna erilaisia, päätettiin yhdistää samaan moduuliin niiden yksinkertaisuuden takia. Myös ulkomuoto tulee olemaan samannäköiset molemmissa, joten niiden päivittäminen olisi helpompaa, kun molemmat sijaitsevat samassa tiedostossa.

Korjaustehtävien prosenttien laskennassa otetaan huomioon kaikki suljetut tehtävät, josta lasketaan, kuinka monta prosenttia niistä on korjaustehtäviä. Oletusaikaväli on yksi kuukausi, josta tehtävät haetaan. Jotta saadaan tietää, onko korjaustehtävien määrä kasvanut tai vähentynyt, lasketaan myös samalla logiikalla aikaisempi yhden kuukauden jakso. Tästä näytetään käyttöliittymästä vain erotus, jotta käyttäjä saa tietää tehtävien kehityssuunnan.

Tehtävien myöhästymisprosentti lasketaan pitkälti samalla logiikalla. Myöhästyneet tehtävät saadaan, kun verrataan tehtävän valmistumispäivää sen suunniteltuun valmistumispäivään. Tätä luku vertaillaan kaikkiin tehtyihin tehtäviin, jotta saadaan prosenttiluku.

### StackedAreaChart

Tehtäviin käytetyn ajan summaviivakaavion tarkoituksena on visualisoida laitokseen käytetty aika. Käytetty aika jaetaan eri tehtävätyyppien mukaan. Käytetty aika lasketaan suoritettujen tehtävien summasta jokaista eri tehtävätyyppiä kohden. Näin kohdekäyttäjä pystyy näkemään erityyppisiin tehtäviin käytetyn ajan ja kuinka ne ovat kehittyneet. Tällöin voidaan havaita muutokset tehtävissä.

```
chartData.forEach(type => {
  var data = anychart.data.set(type);
  var x = data.mapAs({x: 'x', value: 'value', name: 'type'});
  var c = chart.splineArea(x);
  c.name(type[0]['type']); //For legend formatting
  setupSeries(c, type[0]['type'])
});
```

Kuvio 15. Datan sovittaminen summaviivakaavioon

Tehtävien haku tietokannasta tehtiin hakemalla tehtävätyyppi kerrallaan, josta laskettiin jokaiselle kuukaudelle summa silmukassa. Saadusta datasta muodostettiin taulukko tehtävätyypeittäin, jossa tehtävätyypin taulukko sisältää jokaiselle kuukaudelle luodun olion. Front endissä data pitää käydä uudelleen silmukassa tehtävätyypeittäin kuvion 15 mukaisesti, jotta AnyChart-kirjasto osaa luoda datan oikein summaviivakaaviossa.



### **MostTimeConsumingEquipment**

Eniten aikaa vievien laitteiden -moduulin tarkoituksena on tuoda käyttäjän huomioon laitteet, jotka vievät eniten aikaa kuukaudesta. Tämän ideana on korostaa niitä laitteita, jotka vaativat eniten aikaa. Korostettavien laitteiden määräksi valittiin 5 laitetta, jolloin muihin laitteisiin käytetty aika näytettäisiin vain summana, jotta rengaskaaviossa pystyy vertaamaan kuinka paljon korostettavat laitteet vievät aikaa kaikista muista laitteista.

### **MostFailingEquipment**

Eniten vikaantuvien laitteiden -moduulin tarkoituksena on nostaa esiin laitteita eniten aikaa vievien laitteiden tavoin. Moduulin ideana on korostaa laitteita, joiden korjausväli on pieni, eli ne vikaantuvat useammin. Vikaantuvia laitteita myös ryhmiteltiin laitetyypeittäin, sekä prosessialueittain.

Yksittäisen laitteen vikaantumisväli lasketaan laitteen luontihetkestä asti tehtyjen tehtävien välissä olevien päivien perusteella. Näistä päivistä lasketaan keskiarvo lisäämällä ne yhteen ja jakamalla ne tehtävien määrällä. Laskentaan tuli myös ottaa huomioon vain sellaiset tehtävät, jotka ovat määritelty korjaustehtäviksi, eli ennakoivaa huoltoa sekä muita vastaavantyyppisiä tehtäviä ei oteta huomioon. Laitetyypeittäin ja prosessialueittain laskenta oli hyvin samanlainen kuin yksittäisten laitteen laskennassa. Laitetyypeissä otettiin huomioon kaikki ne laitteet, jotka olivat samantyyppisiä. Sama prosessialueittain.

### **Calendar**

Kalenteri on ainoa komponentti, joka on vaikutuksessa muihin komponentteihin. Sen avulla pystyy vaihtamaan muiden komponenttien käyttämää aikaväliä. Oletusaikaväli analysointivälilehdellä on aina yksi kuukausi, paitsi tehtävien summaviivakaaviossa, jossa se on kuusi kuukautta.

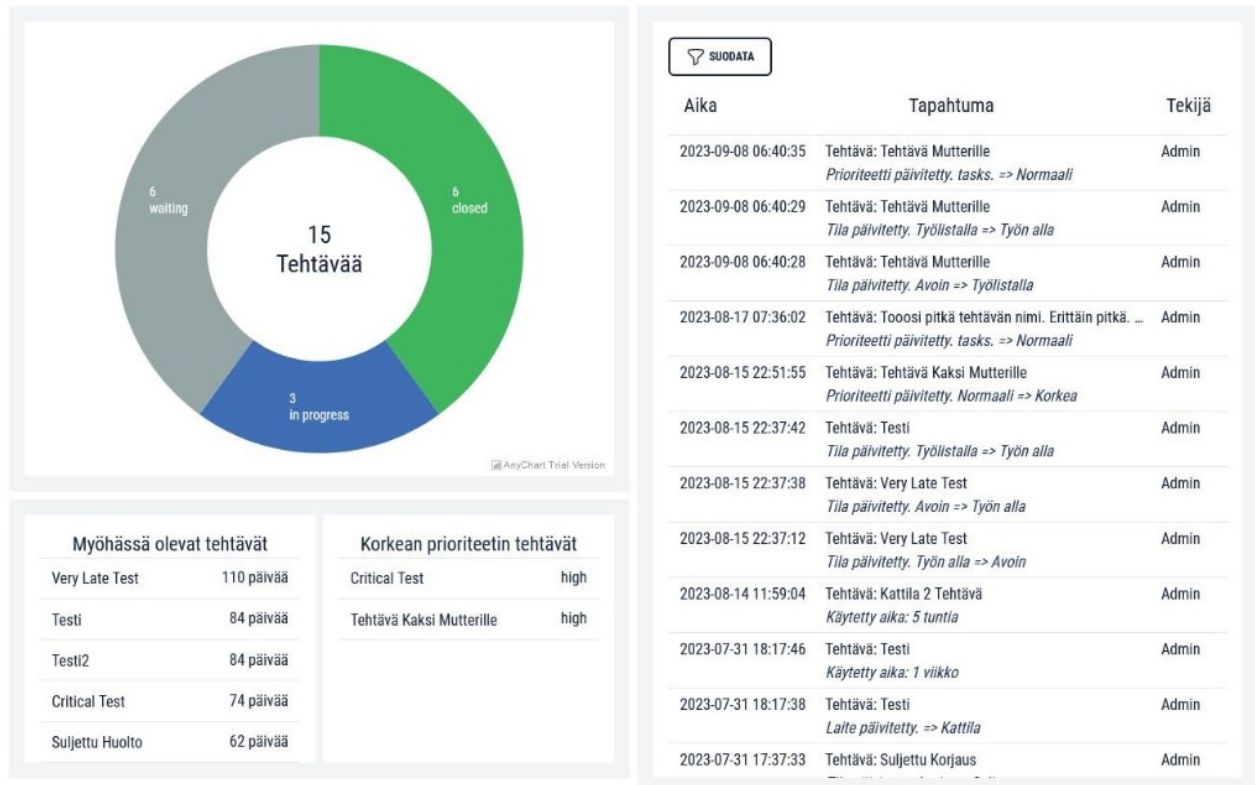
```
public function changeCalendarDates()  
{  
    $this->emit( event: 'CalendarDatesChanged', $this->startDate, $this->endDate);  
}
```

Kuvio 16. Kalenterin tapahtuma

Kalenterin oma toiminnallisuus on yksinkertainen. Komponentissa kuunnellaan, kun käyttäjä vaihtaa päivämääriä käyttöliittymässä, jolloin otetaan ylös valitut päivämäärät. Sen jälkeen Livewiren toiminnallisuuksia hyödyntäen komponentti lähettää uuden tapahtuman päivämäärien kanssa kuvion 16 mukaan edelleen muille komponenteille, jossa kuunnellaan tätä tapahtumaa. Näin toiset komponentit tietävät tehdä haut uudelleen. Ainoastaan laitteen vikaantumisvälin moduulissa ei kuunnella tapahtumaa, koska laskenta tapahtuu ottamalla huomioon koko laitteen historia.

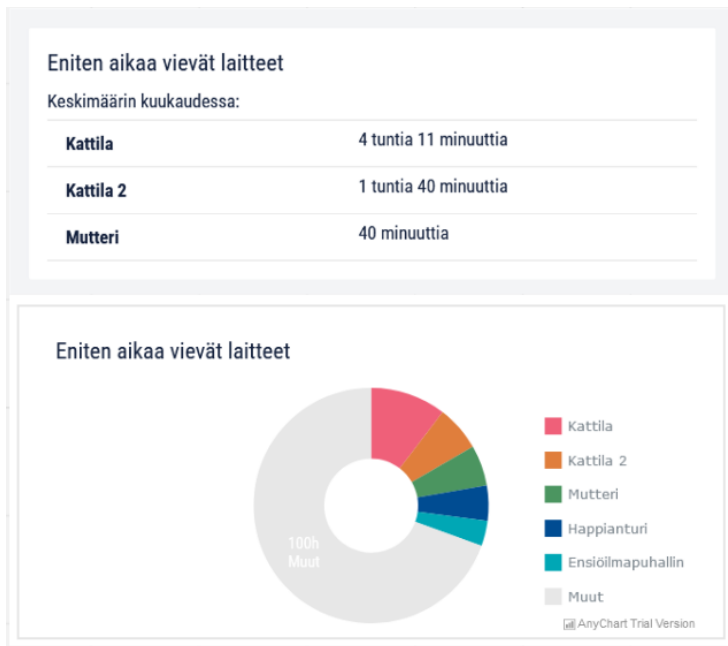
### 3.2.3 Frontend

Ulkonäköön ja käytettävyyteen ryhdyttiin keskittymään, kun kaikki perustoiminnot oli saatu toimintaan. Näin komponentteja pystyttiin työstämään samanaikaisesti ja varmistamaan komponenttien yhtenäisyys ulkonäöllisesti. Ulkonäön hiomista tehtiin iteratiivisesti, jonka välissä kerättiin mielipiteitä sekä parannusehdotuksia, jotta suunta pysyisi oikeana.



Kuvio 17. Ensimmäinen hiotumpi versio ulkonäöstä

Ensimmäisenä tyylittelynä komponenttien ympärille tehtiin harmahtava laatikko kuvion 17 mukaisesti, jotta komponentit erottuvat taustasta, joka on valkoinen. Alkuperäisessä suunnitelmassa oli vaihtaa koko tausta harmaaksi, jotta valkoiset komponentit erottuisivat taustasta. Tähän ei kuitenkaan päädytty, koska muualla GroundWorkissa on valkoinen tausta, jolloin sivu olisi erottunut muista liikaa.




Kuvio 18. Aikaa vievien laitteiden -komponentin ensimmäinen ja viimeisin toteutus

Eniten aikaa vievien laitteiden koko toiminnallisuus muuttui, kun ulkonäköä hiottiin. Alun perin laitteet olivat listassa kuvion 18 mukaisesti, jossa myös laskettiin keskimääräinen aika laitetta kohden. Viimeisimmässä versiossa listaus muutettiin rengaskaavioksi kuvaamaan visuaalisesti halutun aikajakson välillä eniten aikaa vieviä laitteita. Viimeisessä versiossa ei siis laskettu enää keskimääräistä aikaa laitteille, vaan näytettiin kulutettu aika aikajaksolla.



Kuvio 19. Vikaantuvien laitteiden komponentti

Vikaantuvien laitteiden komponentissa ensimmäisessä versiossa vikaantuvimmat laitteet olivat samanlaisessa listassa kuin eniten aikaa vievät laitteet. Listaus muutettiin kuitenkin laatikoiksi kuvion 19 mukaisesti, jotta vikaantuvimmat laitteet kiinnittäisivät enemmän huomioita, kuin normaali listaus. Näin myös sivulla päästiin eroon toistuvista listauksista muuttamalla ulkonäköä.



Aika	Tapahtuma	Tekijä
<b>07.11.2023</b>		
09:01:49	<b>Tehtävä:</b> Kattila 2 Tehtävä <i>Tekijä päivitetty. =&gt; Admin</i>	Admin
<b>06.11.2023</b>		
10:46:05	<b>Tehtävä:</b> Tehty korjaus syyskuussa <i>Tehtävä luotu</i>	Admin
10:44:27	<b>Tehtävä:</b> tehty korjaus <i>Raportoitu aika päivitetty. =&gt; 360</i>	Admin
10:43:52	<b>Tehtävä:</b> tehty korjaus <i>Tehtävä luotu</i>	Admin
08:59:35	<b>Tehtävä:</b> Kattila 2 Tehtävä <i>Raportoitu aika päivitetty. 300 =&gt; 480</i>	Admin

Näytetään 16 - 20 Yhteensä 20 tulosta.

<
1
2
>

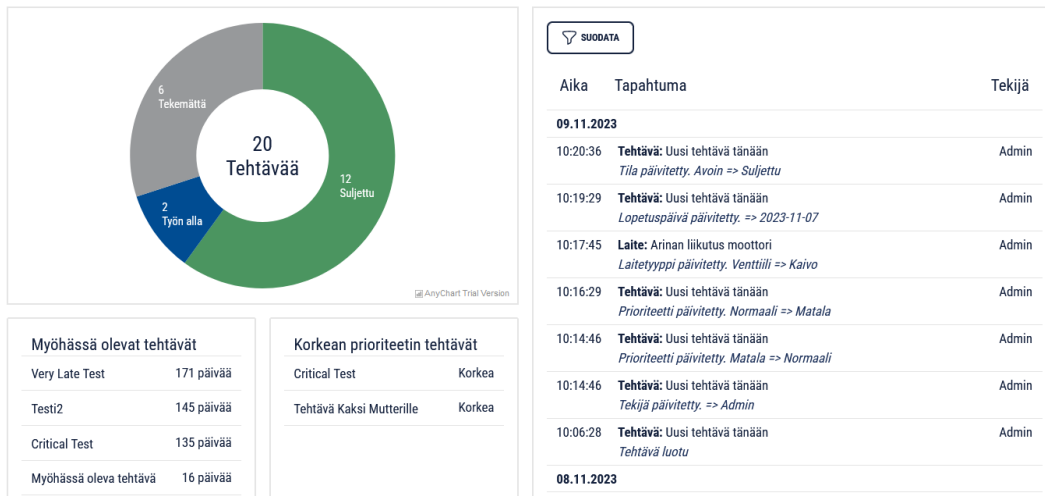
Kuvio 20. Tapahtumalokin päivitetty ulkonäkö

Tapahtumalokin ulkonäköä päivitettiin kuvion 20 mukaan. Päivämäärä siirrettiin omalle riville, joka näytetään ainoastaan, kun tapahtumien päivämäärä vaihtuu. Lokiriville jätettiin vain kellonaika, jotta rivi olisi selkeämpi ja luettavampi.

## 4 Tulokset

Opinnäytetyön tuloksena syntyi GroundWork-ohjelmaan uusi toiminnallisuus. Toiminnallisuudessa on kaksi välilehteä. Komponentit jaettiin välilehdille niiden tarkoitusten mukaan.

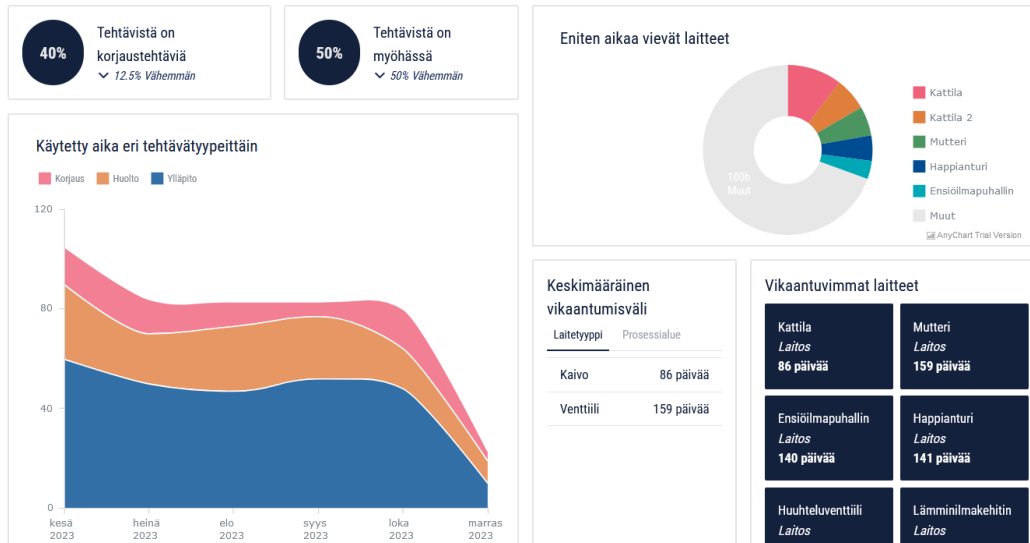
## Nykyinen viikko



Kuvio 21. Ensimmäisen välilehden viimeisin toteutus

Ensimmäisen välilehden myötä kohdekäyttäjä pystyy näkemään laitoksen tapahtumia sekä tehtävien nykytilanteen. Välilehdellä myös korostetaan myöhässä olevia ja kriittisiä tehtäviä, jotta kohdekäyttäjä pystyy näitä halutessaan priorisoimaan. Välilehden tarkoituksena on antaa nopealla vilkaisulla käyttäjälle kaikki tärkeimmät tiedot menossa olevasta viikosta. Näin käyttäjä pystyy seuraamaan, kuinka viikko on edennyt ja mitä on vielä jäljellä.

19.10.2023 – 19.11.2023



Kuvio 22. Toisen välilehden viimeisin toteutus

Toisella välilehdellä kohdekäyttäjä pystyy näkemään analyttistä dataa, joka kerätään Ground-Workin muista toiminnallisuuksista. Tällä välilehdellä käyttäjä pystyy muuttamaan haluttua tarkasteluajankohtaa, jolloin komponentit hakevat datansa sen mukaisesti. Välilehden komponenteilla käyttäjän huomiota kiinnitetään ongelmisiin laitteisiin, sekä laitoksessa käytettyyn aikaan.

## 5 Pohdinta

Datan visualisointi on nimensä mukaisesti pitkälti pelkkää visuaalista tekemistä. Päällepäin näkyy vain visuaalinen tuotos, vaikka kyseessä onkin pitkä prosessi. Ulkonäkö on aina subjektiivista, joten yhtä ainoaa ratkaisua siihen ei ole. On vain hyviä käytänteitä sekä ohjeita, kuinka visualisoida dataa. Tämän takia Yau (2013, 45) viittaa kirjassaan siihen, että datan visualisoinnilla on aina oma tarkoituksensa, joten niitä tulisi arvioida sen mukaan, kuinka hyvin haluttu tavoite saavutetaan. Datan visualisointi oli opinnäytetyön tekijälle myös uusia asia, joten aiempaa kokemusta ei ollut, joka olisi auttanut opinnäytetyössä. Tämän takia opinnäytetyön visuaalisessa toteutuksessa yritettiin hyödyntää opinnäytetyöhön kerättyä tutkimusmateriaalia sekä saatua palautetta.

Datan visualisointi on kuitenkin vain yksi osa analyttistä prosessia. Iso osa alkusuunnittelusta liitettiin laitossylläpidon piiriin, josta opinnäytetyön tekijällä ei ollut paljon aikaisempaa tietoa tai koke-

musta. KPI ja muiden lukujen keksiminen näiden pohjalta oli työlästä ja tuotti hankaluuksia. Näiden selvittämisessä onneksi koko tiimi auttoi. Suunnitelmien esittäminen edisti opinnäytetyötä palautteen ansiosta, jota oli hyvä hyödyntää selvityksessä ja uudelleen suunnittelussa. Vaikka työtä tehtiin lähinnä yksin, avun saaminen auttoi opinnäytetyön edistymiseen. Katselmointien myötä opinnäytetyön suunta pysyi haluttuna.

Jatkokehityksen ideoita on jo. Kaikkea pientä toiminnallisuutta ja käyttäjäystävällisyyttä voisi aina lisätä. Esimerkiksi ensimmäisen välilehden rengaskaaviota voisi hiirellä klikata, jolloin siitä aukeaisi lista niistä tehtävistä, jotka kuuluvat klikattuun kaareen. Toisella välilehdellä rengaskaaviota klikkaamalla kaaresta voisi aueta laitekortti tai kun kyseessä on yhteenlaskettu aika, aukeaisi lista laitteista, johon on laskettu aika laitteittain.

Ennen opinnäytetyön käyttöönottoa, voisi sitä vielä pilotoida asiakkaille ja kerätä palautetta. Näin opinnäytetyöstä saataisiin konkreettista palautetta kohdekäyttäjiltä, jotta voitaisiin varmistaa ja lisätä opinnäytetyön tuotoksen käytettävyyttä. Tämän takia toteutuksessa käytettiin modulaarisuutta hyväksi, jotta yksittäisiä moduuleja olisi tulevaisuudessa helppo lisätä tai poistaa. Tämä lisää toteutetun työn ylläpidettävyyttä. Mahdollisesti käyttäjä pystyisi tulevaisuudessa itse valitsemaan haluamansa moduulit. Modulaarisuudessa toinen hyöty on, että toteutettuja komponentteja olisi mahdollista upottaa itsekseen sellaisenaan muualle GroundWorkiin ja ne toimisivat heti.



## Lähteet

Anychart. N.d. Anychart JavaScript-kirjaston dokumentaation sivu. Viitattu 27.11.2023. <https://www.anychart.com/>.

Drysdale, J. N.d. Design Process. Artikkelin suunnitteluprosessin hyödyntämisestä. Viitattu 23.10.2023. <https://proximityschool.com/learn/design-process/>.

Eettiset periaatteet. 2018. Jyväskylän ammattikorkeakoulun eettiset periaatteet. Viitattu 30.10.2023. <https://jamkstudent.sharepoint.com/sites/Eettiset-periaatteet-Elmo/SitePages/Etusivu.aspx>.

Elomatic. N.d. Elomaticin tietosivu. Viitattu 31.10.2023. <https://www.elomatic.com/fi/tietoa-meista/>.

Eläinlajit. 2022. Number of described species. Lista eläinlajeista. Viitattu 1.12.2023. <https://ourworldindata.org/how-many-species-are-there>.

Ilmatieteenlaitos. N.d. Havaintojen lataus. Ilmatieteen laitoksen sivu, jossa käyttäjä pystyy lataamaan säähän liittyviä havaintoja. Viitattu 1.12.2023. <https://www.ilmatieteenlaitos.fi/havaintojen-lataus>.

Kirk, A. 2012. Data Visualization: a Successful Design Process. Birmingham: Packt Pub. Viitattu 19.9.2023. <https://janet.finna.fi>.

Knaflic, C. N. 2015. Storytelling with data: A data visualization guide for business professionals. Hoboken: John Wiley & Sons. Viitattu 19.9.2023. <https://janet.finna.fi>.

Korjaava kunnossapito. N.d. Opetushallituksen laatima ohjeistus kunnossapidon käsitteisiin sekä määritelmiin. Viitattu 19.11.2023. [http://www03.edu.fi/oppimateriaalit/kunnossapito/perusteet\\_2-2\\_korjaus.html](http://www03.edu.fi/oppimateriaalit/kunnossapito/perusteet_2-2_korjaus.html).

Kunnossapidon kustannukset. N.d. Opetushallituksen laatima ohjeistus kunnossapidon käsitteisiin sekä määritelmiin. Viitattu 19.10.2023. [http://www03.edu.fi/oppimateriaalit/kunnossapito/perusteet\\_3-2\\_kunnossapidon\\_tuotot\\_ja\\_kustannukset.html](http://www03.edu.fi/oppimateriaalit/kunnossapito/perusteet_3-2_kunnossapidon_tuotot_ja_kustannukset.html).

Kunnossapidon perusteet. N.d. Opetushallituksen laatima ohjeistus kunnossapidon käsitteisiin sekä määritelmiin. Viitattu 9.11.2023. [http://www03.edu.fi/oppimateriaalit/kunnossapito/perusteet\\_2-1\\_kunnossapidon\\_kasitteet\\_ja\\_maaritelmat.html](http://www03.edu.fi/oppimateriaalit/kunnossapito/perusteet_2-1_kunnossapidon_kasitteet_ja_maaritelmat.html).

Laravel. N.d. Laravelin dokumentaatio. Viitattu 24.10.2023. <https://laravel.com/docs/10.x>.

Mikä on KPI? 2021. Mikä on KPI ja mitä sillä voi mitata? Artikkelin Sampo Consulting -sivulla. Viitattu 21.10.2023. <https://sampoconsulting.com/mika-on-kpi/>.

Miro. 2023. Artikkeleli Miron neuvontasivulla. Viitattu 25.10.2023. <https://help.miro.com/hc/en-us/articles/360017730533-What-is-Miro->.

Phan, L. L. 2022. The design process and 5 reasons why it matters. Artikkeleli suunnitteluprosessin vaiheista Zeplin-sivulla. Viitattu. 23.10.2023. <https://blog.zeplin.io/the-design-process-and-5-reasons-why-it-matters>.

Rehkopf, M. N.d. User stories with examples and a template. Artikkeleli käyttäjätarinoista Atlassian sivulla. Viitattu 23.10.2023. <https://www.atlassian.com/agile/project-management/user-stories>.

Summaviivakaavio. 2009. Havainnollistava kuva summaviivakaaviosta. Viitattu 21.10.2023. [https://commons.wikimedia.org/wiki/File:Viewing\\_share\\_1992\\_-\\_2009\\_of\\_uk\\_channels\\_above\\_1\\_percent\\_area\\_aggregation.png](https://commons.wikimedia.org/wiki/File:Viewing_share_1992_-_2009_of_uk_channels_above_1_percent_area_aggregation.png).

Suunnitteluprosessi. 2022. Rosenfeld Median havainnollistava kuva suunnitteluprosessin vaiheista. Viitattu 22.10.2023. <https://www.flickr.com/photos/rosenfeldmedia/52596859716/in/photostream/>.

Tilastokeskus. 2023. Väestörakenne. Suomen väkiluku Excel-tiedostona tilastokeskuksen sivuilla. Viitattu 15.11.2023. <https://stat.fi/tilasto/vaerak>.

Toikko, T. & Rantanen, T. 2009. Tutkimuksellinen kehittämistoiminta: näkökulmia kehittämisssessiin, osallistamiseen ja tiedontuotantoon. Tampere: Tampereen Yliopistopaino. 3. korjattu painos. Viitattu 30.10.2023. <https://urn.fi/URN:ISBN:978-951-44-7732-4>.

Twarog, A. 2021. What is Figma? A Design Crash Course [2021 Tutorial]. Artikkeleli Figmasta Free Code Campin sivuilla. Viitattu 25.10.2023. <https://www.freecodecamp.org/news/figma-crash-course/>.

Yau, N. 2011. Visualize This: The FlowingData Guide to Design, Visualization, and Statistics. Indianapolis: John Wiley & Sons. Viitattu 19.9.2023. <https://janet.finna.fi>.

Yau, N. 2013. Data Points: Visualization That Means Something. Indianapolis: John Wiley & Sons. Viitattu 19.9.2023. <https://janet.finna.fi>.

Yi, M. N.d. A Complete Guide to Area Charts. Artikkeleli chartio-sivulla. Viitattu 20.10.2023. <https://chartio.com/learn/charts/area-chart-complete-guide/>.

Yuk, M. & Diamond, S. 2014. Data visualization for dummies. Hoboken: John Wiley & Sons. Viitattu 19.9.2023. <https://janet.finna.fi>.