



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Tiina Ojala

Tehokkuutta ja yhtenäisyyttä suunnittelujärjestelmän avulla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

29.11.2018

Tekijä Otsikko	Tiina Ojala Tehokkuutta ja yhtenäisyyttä suunnittelujärjestelmän avulla
Sivumäärä Aika	31 sivua 29.11.2018
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Ohjelmistotuotanto
Ohjaajat	Director Consulting Services Markus Miettula Yliopettaja Erja Nikunen
<p>Insinööriyössä perehdyttiin Design Systemeihin eli suunnittelujärjestelmiin teorian ja esimerkkien kautta. Työn keskeisenä tavoitteena oli osoittaa, että suunnittelujärjestelmä olisi hyödyllinen työkalu julkishallinnon organisaatiolle, joka tarjoaa useita palveluita sekä kansalaisille että julkishallinnon työntekijöille. Palveluiden kehittäminen on jatkuvassa kasvusuunnassa digitalisoitumisen myötä, ja suunnittelujärjestelmä toisi monia etuja järjestelmien käyttöliittymäsuunnitteluun ja -toteutukseen.</p> <p>Työssä tarkasteltiin viittä eri suunnittelujärjestelmää, joiden avulla havainnollistettiin sisältöjen muovautuvuus yrityksen, liiketoiminnan ja projektien tarpeiden mukaan. Työssä esitettiin, kuinka suunnittelujärjestelmä tehostaa sekä yhtenäistää toimintaa ja tukee niin suunnittelijoita, toteuttajia kuin myös palveluiden tarjoajaa. Esimerkkejä ei-toivotuista esiintymistä, joita suunnittelujärjestelmän avulla pystytään minimoimaan, kerättiin tutkimalla julkishallinnon organisaatiolle toteutettuja järjestelmiä kehitysvaiheessa olevan järjestelmän aktiivisen seuraamisen lisäksi.</p> <p>Insinööriyön lopputuloksena syntyi perustelujen lisäksi jatkokehittävissä oleva suunnittelujärjestelmän pohja, joka luotiin kehitysvaiheessa olevan järjestelmän tarpeiden mukaan. Ohjeistot luotiin kyseisen järjestelmän kehityksessä ilmenneiden kompastuskivien pohjalta, mutta ohjeistoja on helppo lähteä muovaamaan koko organisaation kattaviksi järjestelmäriippumattomiksi ohjeistoiksi. Työssä esitettiin myös jatkokehitystoimia, joiden myötä koko organisaation kattavasta suunnittelujärjestelmästä saisi miellyttävän käyttöä, ja se olisi helposti jalkautettavissa projektista toiseen.</p>	
Avainsanat	Design System, brändi, käyttöliittymä, käyttöliittymäsuunnittelu, suunnittelujärjestelmä, UI

Author Title	Tiina Ojala Efficiency and Consistency by Design System
Number of Pages Date	31 pages 29 November 2018
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Instructors	Markus Miettula, Director Consulting Services Erja Nikunen, Principal Lecturer
<p>This Bachelor's Thesis focused on design systems by theory and examples. The key objective was to indicate that the design system would be a useful tool for the public administration organization that provides many services for both citizens and employees of public administration. Due to digitalization, the number of different kinds of services growth continuously and the design system would provide many benefits for a user interface design and implementation.</p> <p>The five different design systems were used to illustrate the conformability of the contents according to the needs of the company, business and projects. The thesis states that the design system will enhance and unify activities and supports designers, developers as well as service provider. Unwanted occurrences can be minimized by the design system and the examples of some were collected by monitoring the system that is still in development, and by investigating systems that have been already implemented for the same public administration organization.</p> <p>As a result of this Bachelor's Thesis the base of the design system was created and it is ready for further development. The base of the design system was created according to needs of the system that is still in development. These guidelines are easy to edit to serve the whole organization by being system independents. The thesis also included suggestions about further development activities that would help to create a better design system for the whole organization which can easily use from project to another.</p>	
Keywords	Design System, brand, UI, user interface, user interface design

Sisällys

Käsitteet

1	Johdanto	1
2	Suunnittelujärjestelmä – monta vastaajaa, monta määritelmää	3
2.1	Yrityksen, liiketoiminnan ja projektien tarpeiden mukainen	5
2.2	Sisältö	11
3	Kohti tehokkaampaa käyttöliittymäsuunnittelua ja -toteutusta	13
3.1	Nykyiset ohjeistukset Venlassa	15
3.2	Ohjeistukset jo toteutetuissa järjestelmissä	17
3.3	Inventointivaihe	18
4	Suunnittelujärjestelmä Oikeusrekisterikeskukselle	21
5	Yhteenveto ja pohdinta	27
	Lähteet	30

Käsitteet

Atomisuunnittelu

Engl. *atomic design*. Menetelmä, jossa suunnittelujärjestelmän sisältö jaetaan atomeihin, molekyyliin ja organismeihin.

High-fidelity wireframe

Korkean tarkkuustason rautalankamalli, jossa lopputulos on hyvin oikean näköistä. Rautalankamallissa käytetään tekstityylejä, kuvia ja värejä sellaisina, kuin ne ovat käyttöliittymällekkin määrä toteuttaa.

Käyttöliittymäkonsepti

Dokumentaatio, jossa on kuvattu rautalankamallien ja tekstimuotoisten selvitysten avulla palvelun keskeiset ominaisuudet ja toiminnallisuudet.

Low-fidelity wireframe

Matalan tarkkuustason rautalankamalli. Karkea hahmotelma, jossa on hyvin vähän yksityiskohtia. Sivun eri osioita havainnollistetaan laatikoin, ja yksityiskohtaisempi sisältö jätetään kuvaavamatta.

Middle-fidelity wireframe

Myös *medium-fidelity wireframe*. Rautalankamalli, jonka tarkkuustaso asettuu matalan ja korkean väliin.

Prototyypointi

Toteutettavan järjestelmän konkretisointi ja visualisointi vastaamaan todennäköistä. Voidaan toteuttaa joko paperin ja kynän avulla, tai luoda interaktiivinen prototyyppi tähän tarkoitettu ohjelmistolla.

Rautalankamalli

Kuva, joka havainnollistaa sivun eri osioiden sijoittumisen. Pääpaino sisällössä ja toiminnallisuuksissa. Rautalankamallien tarkkuustasot jaetaan matalaan, keskitasoon ja korkeaan.

Suunnittelumallikirjasto

Engl. *pattern library*. Koostuu uudelleen käytettävistä komponenteista ja elementeistä, joista käyttöliittymä rakennetaan.

Suunnitteluohjeisto

Engl. *design guideline* tai *style guide*. Ohjeisto, johon on kuvattu käyttöliittymän tyyliasioita, ja joiden avulla saadaan vahvistettua yhtenäistä brändiä. Näihin lukeutuvat muun muassa typografiaohjeet sekä värikartat.

Suunnitteluvelka

Engl. *design debt*. Velka, joka kohdistuu käyttöliittymätoteutukseen. Johtuu usein epäyhtenäisistä tyyleistä ja käytännöistä, jotka eivät ole helposti uudelleen hyödynnettävissä. Näitä usein kiireessä tehtyjä nopeita toteutuksia joudutaan myöhemmin parantamaan.

Tekninen velka

Kehitystiimien kesken myös purkkatoteutus. Järjestelmän arkkitehtuurissa ja toteutustyössä valitaan ratkaisu, jonka tiedetään olevan kiireessä nopein mahdollinen. Kiireessä tehtyä ratkaisua joudutaan myöhemmin parantamaan.

1 Johdanto

Noin kymmenen vuotta sitten visuaalinen ilme ja brändiajattelu olivat vielä hieman vieraita käsitteitä Suomen julkishallinnossa, kuten Eeva Korvola kertoi opinnäytetyössään vuonna 2010 [1]. Nämä käsitteet miellettiin usein yksityisen sektorin yrityksille, joilla on selkeä kilpailuasettelu muiden toimijoiden kanssa ja jossa vahva sekä erottuva brändi on suuri etu. Julkinen sektori toimii yhteiskunnassamme hyvin eri tavalla ja eri alueella, eikä samanlaista kilpailuasettelu ole. Jokaisella valtiomme ministeriöllä on oma vastuualueensa, eikä heidän tarvitse kilpailla muiden toimijoiden kanssa. Lopputulos on molemmissa sektoreissa kuitenkin sama: palveluita tuotetaan asiakkaiden käyttöön, joten loppukäyttäjiä tulisi arvostaa panostamalla muun muassa järjestelmien käytettävyyteen ja miellyttävään ulkoasuun riippumatta siitä, ovatko järjestelmien käyttäjät kansalaisia vai julkishallintoon kuuluvia työntekijöitä.

Visuaalinen ilme on osa brändiä, ja se luo punaisen langan sille, miten asiat esitetään organisaatiosta ulospäin ja millainen on organisaation tavoiteviesti [1, s. 11-13]. Visuaalisen ilmeen määrittely auttaa paljon viime vuosina kovin julkisuudessa olleen digitalisoitumisen aikakaudellakin, sillä myös ministeriöiden toiminnat siirtyvät edelleen yhä enemmän verkkoalustoille – silti tietysti myös ei-digitaalista materiaalia tarvitaan. Visuaalisen ilmeen luominen luo yhtenäisen linjan materiaaleille niiden fyysisestä sijainnista ja käyttötarkoituksesta riippumatta.

Nykyään, alussa mainitun kilpailuasettelu puutteesta huolimatta, visuaalisen ilmeen ymmärretään tehostavan ja selkeyttävän toimintaa myös julkishallinnon puolella. Kuluneiden lähivuosien aikana ministeriöt ovatkin päivittäneet visuaalisia ilmeitään. Visuaaliseen ilmeeseen kohdistuneen positiivisen muutoksen lisäksi järjestelmien käytettävyyteen ja ulkoasuun on alettu panostaa yhä enemmän ja sen nähdään olevan kasvava trendi.

Otetaan visuaalisen ilmeen päivittämisen esimerkkinä Ulkoministeriö, jonka tehtävänä on edustaa Suomea maailmalla: tammikuussa 2018 Ulkoministeriö julkaisi uuden logonsa lisäksi graafisen ohjeiston [2], jonka monista ohjeistuksista esimerkiksi valittakoon lehti- ja rekryilmoitukset. Kyseisten ilmoitusten tekijä näkee suoraan ohjeistosta, minkälainen ilmoituksen tulee olla niin visuaalisesti kuin sisällöllisestikin.

Graafisen ohjeiston typografiaisuus kertoo tekijälle, minkälaista fonttia ja tyyliä hänen tulee ilmoituksissa käyttää, ja värikartta kertoo käytettävien värien koodit. Ohjeisto kertoo esimerkkikuvineen myös sen, miten Ulkoministeriön Internet-sivujen ja Intranetin sisältö tulee asetella. Juuri tällaiset ohjeistukset voidaan lukea osaksi laajempaa kokonaisuutta, Design Systemiä, johon perehdymme tässä työssä. Design System, jota kutsumme tästä eteenpäin suunnittelujärjestelmäksi, pitää sisällään sellaiset ennalta määritellyt standardit, periaatteet ja komponentit, jotka ovat käyttövalmiita sopiviin tarkoituksiin [3]. Suunnittelujärjestelmän avulla käyttöliittymien suunnittelu- ja toteutustyö tehostuu, ja monia ei-tervetulleita seurauksia pystytään minimoimaan.

Tämän insinööriyön tarkoituksena on perehtyä suunnittelujärjestelmään ja toteuttaa tästä pohja Oikeusrekisterikeskuksen käyttöön. Oikeusrekisterikeskus on yksi Oikeusministeriön hallinnonalan virastoista, jonka tehtäviin kuuluu muun muassa käyttäjäturvallisten ja toimintaa tukevien oikeushallinnon tietojärjestelmien hankkiminen ja kehittäminen [4, s. 3]. Työn tilaajana toimii CGI Suomen Oikeusministeriölle palveluita tuottava tiimi, joka on osa Valtionhallinnon ja liikenteen yksikköä.

Suunnittelujärjestelmää työstetään osana Talous- ja velkaneuvonnalle kehitettävää järjestelmää, Venlaa. Vuoden 2019 alusta alkaen yksityishenkilöiden avustaminen taloudellisissa asioissa siirtyy kunnilta ja aluehallintovirastoilta oikeusaputoimistoille, jotka kuuluvat Oikeusministeriön Oikeushallinto-osaston Oikeusapu- ja ulosottoyksikköön [5; 6]. Venla tulee olemaan oikeusaputoimistojen yhteinen työkalu, jolla hoidetaan yksityishenkilöiden talous- ja velkaneuvontaan kuuluvia asioita. Tämän kehitysvaiheessa olevan järjestelmän aktiivinen seuranta antaa välittömän palautteen siitä, mitä hyvää ja mitä kehitettävää nykyisissä ohjeistuksissa on. Määrittelydokumentaatiot ja käyttöliittymäohjeistukset määrittävät järjestelmiin toteutettavat kokonaisuudet, ja mikäli ohjeistukset ovat puutteelliset, saattaa määrittelijöiden, suunnittelijoiden, toteuttajien sekä myös tilaajan näkemys lopputuloksesta erota.

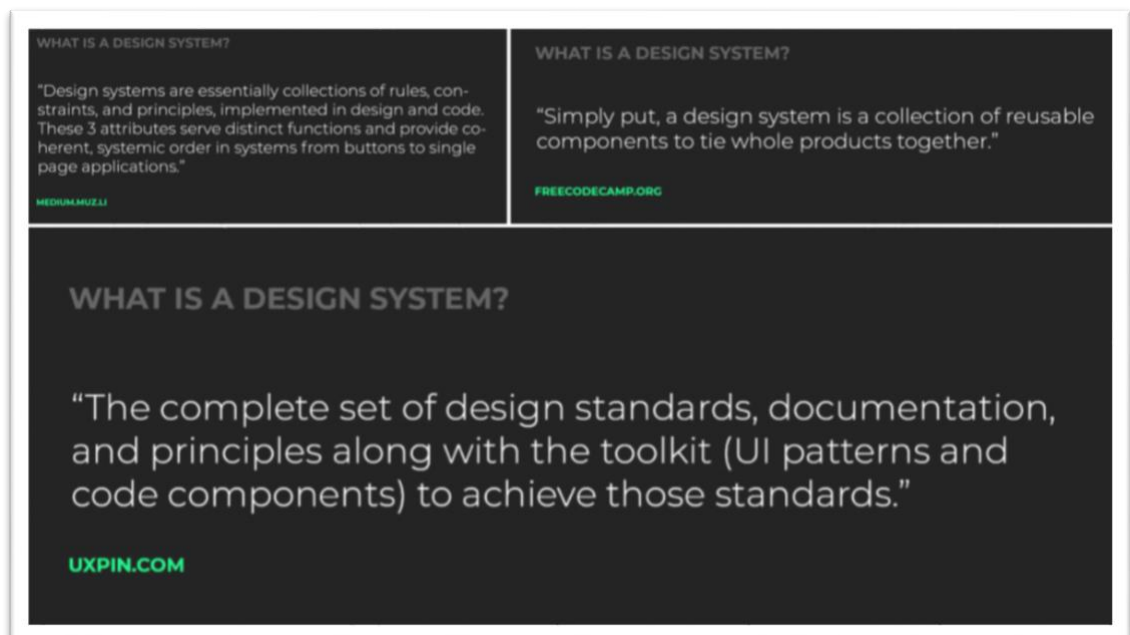
Osana tätä insinööriyötä läpikäydään myös muiden Oikeusministeriölle toteutettujen ja jo tuotannossa olevien järjestelmien käyttöliittymiä koskevia dokumentaatioita ja toteutuksia. Järjestelmien pohjalta tehdään inventointi käytössä olevista

komponenteista, esimerkiksi samaa tehtävää suorittavista painikkeista ja niiden erilaisista toteutustavoista.

Työn keskeisenä tavoitteena on osoittaa, kuinka suunnittelujärjestelmä olisi koko organisaation kattavana hyödyllinen työkalu, sillä se palvelisi niin Oikeusrekisterikeskusta kuin myös sen järjestelmätoimittajia ja siellä olevia eri rooleja. Kun koko organisaation kattavasta suunnittelujärjestelmästä tehdään mahdollisimman geneerinen Oikeusrekisterikeskuksen toimintatapoja ja brändiä mukailleen on sen pohjalta yksittäisiä järjestelmiä mahdollista lähteä kehittämään tiettyine ominaispiirteineen, esimerkiksi eri värein ja eri tyyppisin tekstikentin.

2 Suunnittelujärjestelmä – monta vastaajaa, monta määritelmää

Kuten Gary Simonkin [7] toteaa YouTube-videossaan, Internetistä löytyy monia erilaisia määritelmiä suunnittelujärjestelmälle. Näistä osa saattaa olla hieman suurpiirteisiä, joten selkeää käsitystä ei välttämättä saa heti ensimmäisellä kerralla asiaan tutustuessa. Kuvaan 1 on koottu Simonin videolla esiintyviä erilaisia määritelmiä.



Kuva 1. Suunnittelujärjestelmän erilaisia määritelmiä [7].

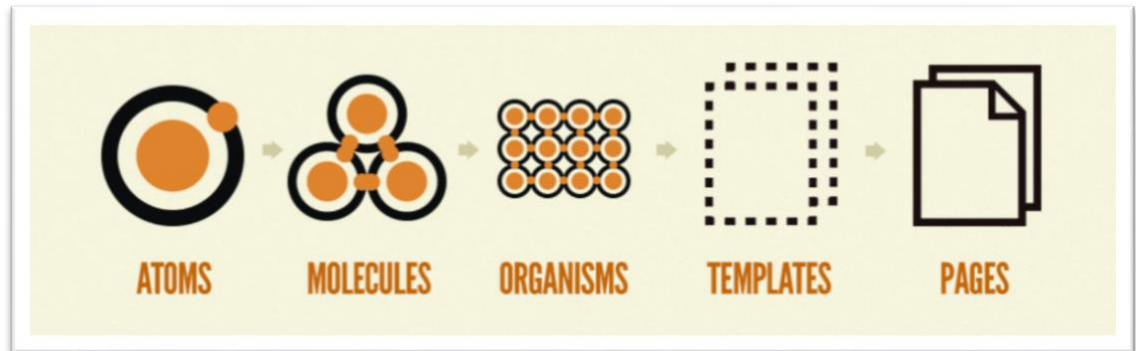
Lähdeaineistojen pohjalta voidaan havaita, että määritelmä riippuu paljon vastaajasta. Kholmatovakin [8, s. 18] mainitsee kirjassaan, ettei kyseiselle järjestelmälle ole muodostunut standardia määritelmää. Yksinkertaisimmillaan määritelmä on yksi tiivis lause, kuten kuvassa 1 sivuston freecodecamp.org määritelmä on: Suunnittelujärjestelmä on kokoelma uudelleenkäytettäviä komponentteja, jotka sitovat tuotteet yhteen. Toisinaan määritelmä voi olla usean lauseen mittainen, eikä sitä ole lähdetty tiivistämään.

Simonin oman määritelmän mukaan suunnittelujärjestelmä on kokoelma erilaisia ohjeita ja periaatteita, jotka ovat tarkoituksenmukaisia juuri kyseiselle liiketoiminnalle ja joiden tarkoitus on auttaa yritystä oppimaan ja kasvamaan yhtenäisen brändin vahvistamisen lisäksi. Laitisen [3] mukaan suunnittelujärjestelmä pitää sisällään sellaiset ennalta määritellyt standardit, periaatteet ja komponentit, jotka ovat käyttövalmiita sopiviin tarkoituksiin. Kholmatovan [8, s. 18] mukaan koko tämän järjestelmän tarkoitus on digitaalisten tuotteiden tavoitteiden saavuttaminen ennalta määriteltyjen mallien ja yhteisten käytäntöjen avulla. Tässä mallit ovat erilaisia komponentteja, jotka ovat pieniä itsenäisiä osia, joita yhdistellään käyttöliittymällä. Yhteiset käytännöt pitävät sisällään sen, miten yrityksessä päätetään luoda, jakaa ja käyttää kyseisiä malleja.

Kuten edellä mainituissa määritelmissäkin huomattiin, suunnittelujärjestelmän yhteydessä törmätään useisiin erilaisiin käsitteisiin. Suunnitteluohjeistot ja mallikirjastot kuuluvat erilaisten suunnitteludokumentaatioiden joukkoon. Suunnitteluohjeisto pitää sisällään niitä Simonin mainitsemia ohjeita, joiden avulla saadaan vahvistettua yhtenäistä brändiä. Näihin lukeutuvat muun muassa typografiaohjeet sekä värikartat. Mallikirjasto koostuu aiemmin mainituista uudelleen käytettävistä komponenteista ja elementeistä, joista käyttöliittymä rakennetaan. Suunnittelujärjestelmästä, -ohjeistosta ja -mallikirjastosta saatetaan silloin tällöin puhua tarkoittaen samaa asiaa, mutta ne ovat kuitenkin toisistaan poikkeavia. Suunnittelujärjestelmä on se, joka kokoaa yhteen kaikki erilaiset suunnitteludokumentaatiot. [8; 9.]

Sen lisäksi, että suunnittelujärjestelmälle on useita eri määritelmiä, myös sen toteuttamiseen on olemassa erilaisia menetelmiä. Hierarkkinen menetelmä koostuu nimensä mukaisesti eri tasoista, jotka toimivat yhdessä. Yksi suosituimmista tällaisista on *atomic design* eli atomisuunnittelu, jossa suunnittelujärjestelmän sisältö jaetaan atomeihin, molekyyliin ja organismeihin. Atomit ovat pieniä yksittäisiä sivuston osia,

joita yhdistämällä saadaan aikaiseksi hieman monimutkaisempia elementtejä, molekyylejä. Organismit ovat isompia kokonaisuuksia, jotka koostuvat molekyyleistä ja/tai useammasta atomista. Edellä mainittuja yhdistelemällä syntyy malleja, ja malleja yhdistelemällä saadaan luotua kokonaisia sivuja. [8, s. 251; 10.] Eri tasot on esitetty kuvassa 2.



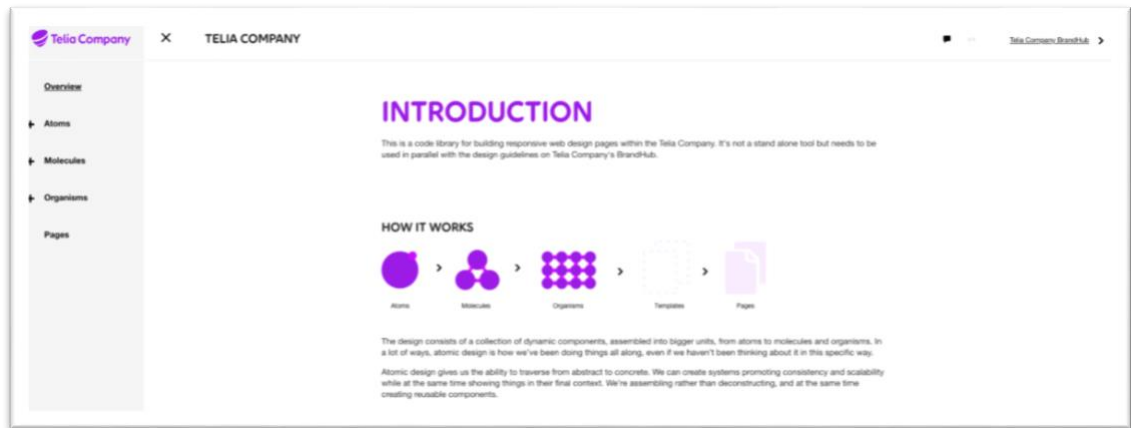
Kuva 2. Atomisuunnittelun eri tasot, joita ovat atomit, molekyylit, organismit, mallit ja sivut [10].

Komponenttiperustaisessa menetelmässä sisältöä ei pilkota yhtä pieniin osiin mitä atomisuunnittelussa. Komponenttiperustaisessa suunnittelujärjestelmässä sisältöön voi kuulua komponenttien lisäksi myös isompiakin kokonaisuuksia.

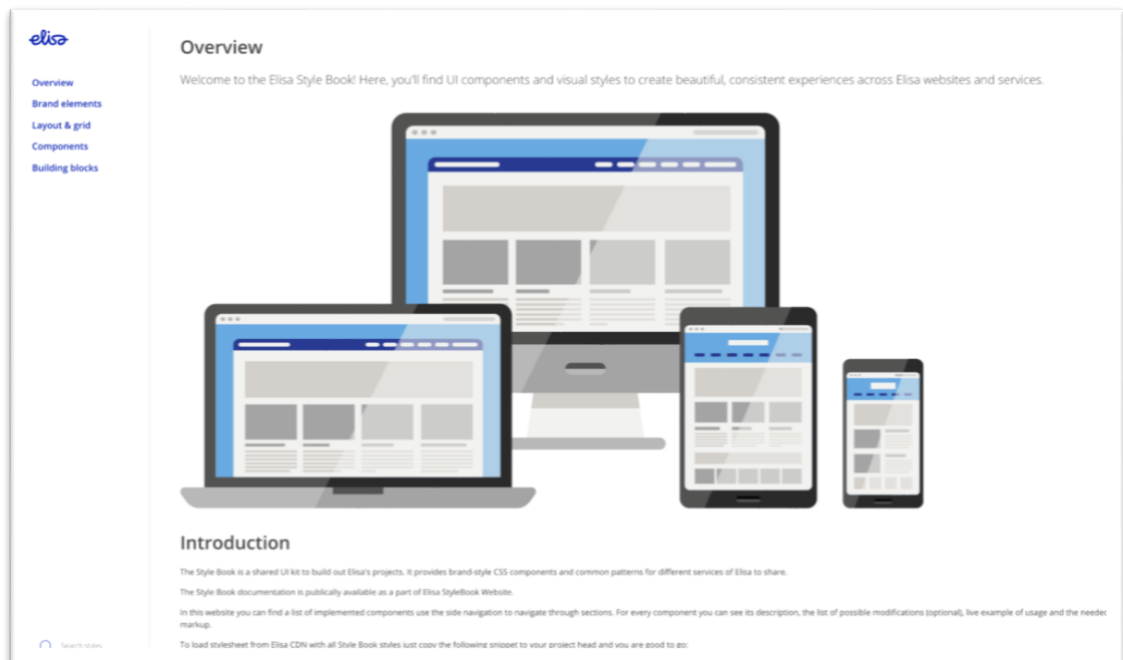
2.1 Yrityksen, liiketoiminnan ja projektien tarpeiden mukainen

Eri yritysten suunnittelujärjestelmät voivat olla keskenään hyvinkin erilaisia. Kyseessä on niin laaja ja jatkuvasti muuttuva järjestelmä, että voisi sanoa olevan lähes mahdotonta löytää kaksi täysin identtistä versiota. Suunnittelujärjestelmä tulee rakentaa yrityksen, liiketoiminnan ja projektien tarpeita tukevaksi. Joillain yrityksillä kyseinen järjestelmä voi olla koostettu täysin eri tavalla muihin yrityksiin verrattuna, vaikka kyseessä olisi sama toimiala, samat asiakkaat ja samankaltaiset tuotteet. Otetaan esimerkkeinä Telia ja Elisa, jotka ovat kaikille suomalaisille tuttuja puhelinoperaattoreita, mutta joiden suunnittelujärjestelmät eroavat silti joissain määrin toisistaan.

Tarkastelemalla kuvissa 3 ja 4 esitettyjä etusivuja voidaan havaita, kuinka näiden kahden suunnittelujärjestelmän välillä on jo sisällön jakamisessa eri osioihin eroavaisuuksia.



Kuva 3. Puhelinoperaattori Telian suunnittelujärjestelmän etusivu. Telia Company on ottanut suunnitteluun atomi-molekyyli-organismi -lähestymistavan. [11.]

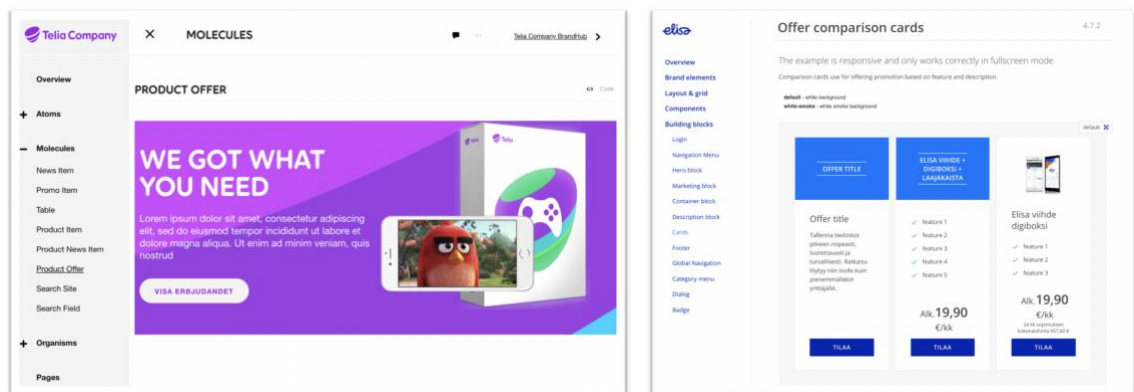


Kuva 4. Puhelinoperaattori Elisan suunnittelujärjestelmän etusivu. Elisa on nimennyt suunnittelujärjestelmänsä Elisa Style Bookiksi. Elisa on jaotellut sisällön brändielementteihin, aseteluihin, komponentteihin ja isompien kokonaisuuksien rakentamiseen. [12.]

Telia on ottanut lähestymistavakseen atomisuunnittelun, joka on esimerkki suunnittelujärjestelmän hierarkkisesta luokittelusta. Atomit ovat pieniä yksittäisiä sivuston osia, kuten painikkeita ja typografiaa. Molekyylit koostuvat atomeista ja määrittävät esimerkiksi, millainen hakutoiminto on tyyliltään ja mitä atomeita se pitää sisällään. Organismit taas koostuvat molekyyleistä ja/tai atomeista ja määrittävät ohjeet

isommille kokonaisuuksille, kuten esimerkiksi ylä- ja alatunnisteille. Elisa on jaotellut sisällön komponenttiperustaisesti, ja se koostuu brändielementeistä, asetteluohjeista, komponenteista ja isommista kokonaisuuksista. Brändielementteihin kuuluvat muun muassa värit ja logo. Asetteluohjeet kertovat, kuinka sivujen sisältö tulee asetella eri kokoiset näytöt huomioon ottaen. Komponentteihin kuuluvat muun muassa painikkeet, lomakkeet sekä huomautus-/varoituseroitukset, ja isompiin kokonaisuuksiin sisältyy ylä- ja alatunnisteiden lisäksi esimerkiksi navigaatiot.

Eri lähestymistavan lisäksi löytyy muitakin eroavaisuuksia, joista seuraavaksi mainittakoon muutama. Elisa on esimerkiksi keskittynyt huomattavasti enemmän asettelua koskeviin ohjeisiin, jonka huomaa jo etusivujen sisältöä silmäillessä: Elisalla on erikseen *Layout & grid* -sivu, jonka alta löytyy yhdeksän eri alisivua ohjeineen, kun vastaavasti Teliällä *Grid* on yksi *Atoms*-sivun alisivuista. Kuvaan 5 on tuotu vertailuksi molempien puhelinoperaattoreiden ohjeistus koskien tarjouksen esittämistä sivustollaan. Kuvasta voimme havaita, että Telia ei ole käyttänyt sanallista selitystä kuvan yhteydessä, kun taas Elisa on kertonut käyttötarkoituksen, käyttäytymisen eri kokoisilla näytöillä ja näiden lisäksi havainnollistanut esimerkein, mitä tietoja tulee antaa ja minkälaisin asetteluin. Elisalla on tähän myös erikseen vaalea- ja tummapohjainen tyyli, kun taas Teliällä on vain yksi vaihtoehto.



Kuva 5. Telian ja Elisan ohjeistukset koskien tarjouksien esittämistä sivustoillaan. Erot esitystavassa havaittavissa. [11; 12.]

Käsitellään vielä kaksi esimerkkiä erilaisista suunnittelujärjestelmistä. Ensimmäinen näistä on pilvipohjainen verkkokauppa-alusta Shopify, joka on nimennyt oman suunnittelujärjestelmänsä Polarikseksi [13]. Polariksen tarkoitus on auttaa verkkokaupan perustajia kehittämään ja tarjoamaan hyviä käyttäjäkokemuksia asiakkailleen. Shopify

käsittelee ohjeita neljän eri kategorian alla. Kategoriat ovat sisältö, suunnittelu, komponentit ja mallit. Kunkin kategorian alla on paljon havainnollistavia kuvia ja lähtökohtana on myös kertoa, mitä ei kannata tehdä sen sijaan, että keskityttäisiin ainoastaan siihen, mikä olisi ideaalia. Näin verkkokaupan perustaja ymmärtää paremmin suunnittelujärjestelmässä esitetyt ohjeet, etenkin, jos hänellä ei ole suunnittelupohjaista taustaa. Yksi esimerkki tällaisesta *do and do not*-ohjeesta on kuvassa 6, jossa käyttäjille avataan esimerkein tekstin ja taustavärien kontrastin merkitys.

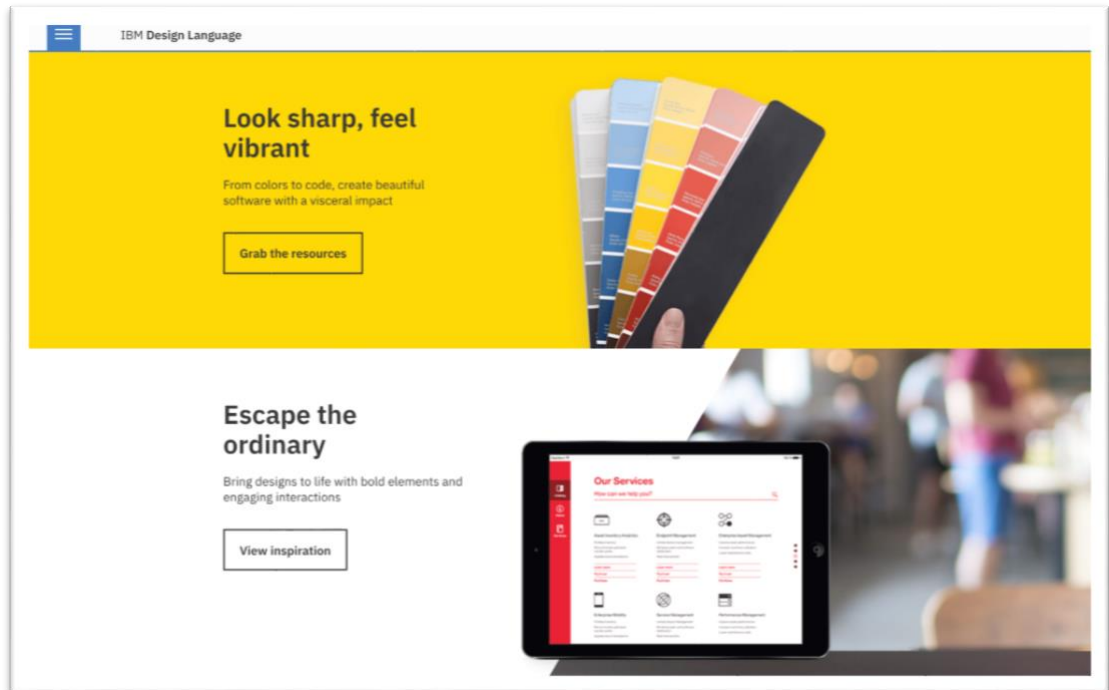


Kuva 6. Shopify:n Polariksesta poimittu kuva esimerkistä, jossa käyttäjälle havainnollistetaan taustavärien ja tekstin kontrastin merkitys esimerkkikuvien kera. Käyttäjälle kerrotaan, mitä kannattaa tehdä ja mitä ei kannata tehdä. [13.]

Sisältö-osiossa Shopify tarjoaa verkkokaupan perustajille vinkkejä muun muassa siitä, miten kirjoitettua kieltä olisi suositeltavaa käyttää. Suunnitteluosiossa käsitellään visuaalisuuteen liittyviä asioita, kuten värejä, typografiaa, sijoittelua ja erilaisia diagrammeja. Komponentteihin on kerätty valmiita elementtejä, joita ovat muun muassa

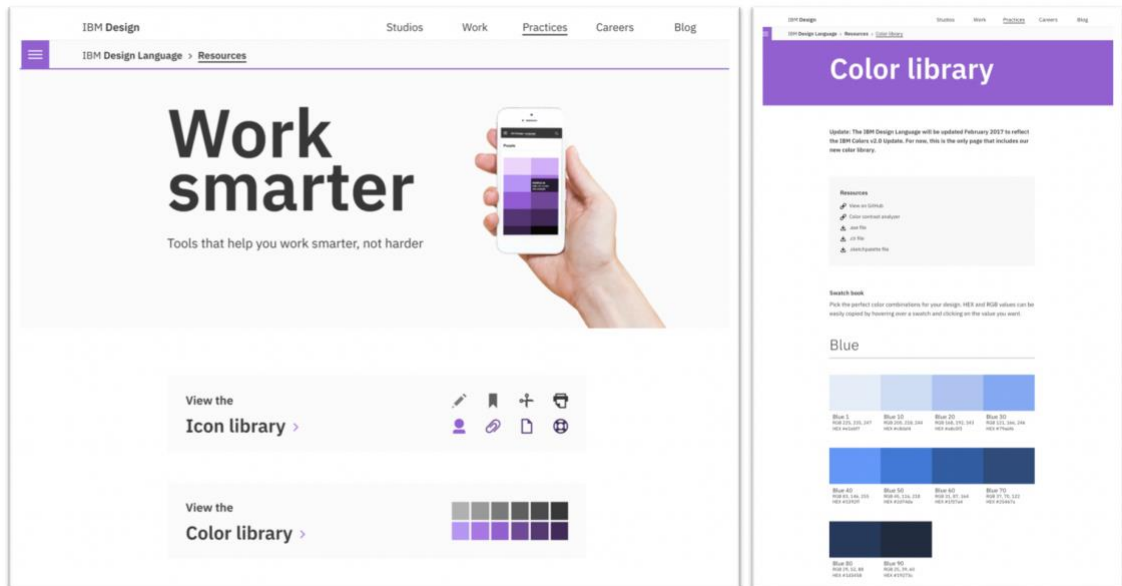
lomakkeet, taulukot ja huomautus-/varoitusikkunat. Malleissa on kerrottu sivuston perusrakenteesta ja tarjottu käyttöön valmiita malleja eri asetteluiin.

Teknologiayritys IBM mainostaa suunnittelujärjestelmänsä olevan elävä kieli, jonka sanasto on suunnittelua varten [14]. IBM Design Language on rakennettu hyvin eri tavalla aiempiin esimerkkeihin verrattuna. Kuvasta 7 voidaan havaita, kuinka jo etusivu on hyvin poikkeava verrattuna esimerkiksi Telian tai Elisan vastaaviin.



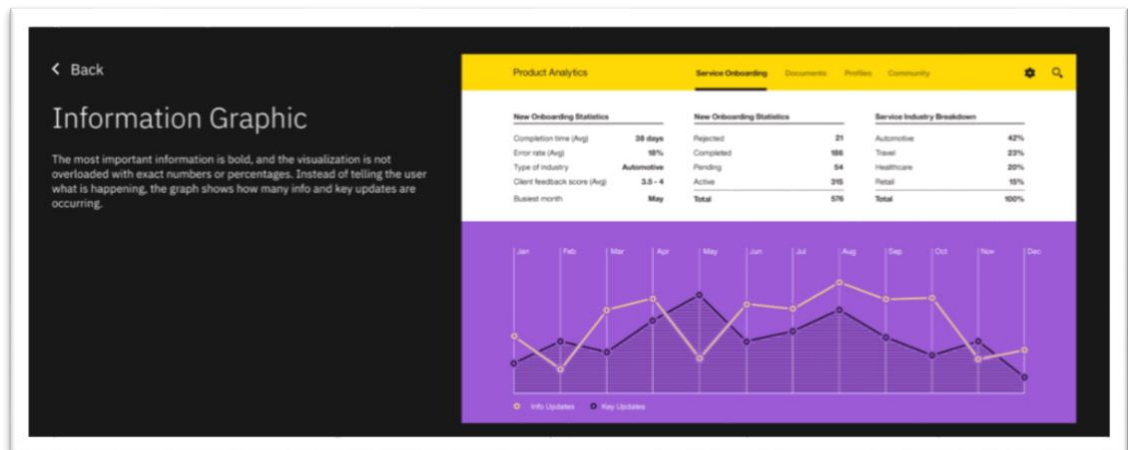
Kuva 7. Kuvankaappaus osasta IBM:n Design Languagen etusivusta. Suunnittelujärjestelmä on rakennettu hyvin eri ideologialla verrattuna aiemmin esitettyihin esimerkkeihin. [14.]

IBM Design Languagessa ohjeet on viety ikoni-, väri ja animaatiokirjastojen alle kuvassa 8 esitetyllä tavalla. Animaatiokirjaston alta löytyvät muun muassa jo aiemmissä esimerkeissä tutuksi tulleet navigaatiot sekä huomautus-/varoitusikkunat.



Kuva 8. Kuvankaappaus IBM Design Language -suunnittelujärjestelmän *Look Sharp, feel vibrant* -osiosta ja sen alta löytyvästä värikirjastosta. [14.]

IBM Design Language poikkeaa aiemmin läpikäydyistä suunnittelujärjestelmäesimerkeistä myös siten, että se sisältää hyvin suppeasti komponenttikohtaisia ohjeita. Sen sijaan IBM on keskittynyt enemmän artikkelityyppiseen ohjeistamiseen ja yhtenä osana on perehdytys hyvän käyttäjäkokemuksen tuottamiseen. IBM Design Language tarjoaa myös inspiraatioita kuvien ja vinkkien kera, joista yksi esimerkki on nähtävillä kuvassa 9.



Kuva 9. IBM Design Language tarjoaa myös inspiraatioita. Kuvassa infografiikan yksi kolmesta kuvaesimerkistä, jonka yhteydessä on annettu vinkkejä vastaaviin esitustapoihin. [14.]

Jokainen edellä mainituista suunnittelujärjestelmistä on yhtä oikea, vaikka eroja on niin lähestymis- ja esitystavassa kuin yksityiskohtaisuudessaakin, sillä jokainen näistä on luotu vastaamaan yrityksen ja sen tiimien omia tarpeita. Kuten aiemmin on jo mainittu, suunnittelujärjestelmä tulee rakentaa yritystä, liiketoimintaa ja projekteja tukevaksi, mutta sillä on myös tapana muokkautua automaattisesti muun muassa tiimikulttuurin ja tuotteiden/palveluiden mukaan. Tiimejä on hyvin erilaisia, ja se, mikä toimii yhden tiimin kohdalla, ei välttämättä palvelekaan toisen tiimin työskentelytapoja. Suunnittelujärjestelmässä määritetyt ohjeet voivat olla todella tarkkoja ja yksityiskohtaisia, löyhiä tai jotain tältä väliltä. Joihinkin tuotteisiin/palveluihin sopii modulaarisempi suunnittelujärjestelmä, jossa eri osioita voidaan yhdistellä ja muokata vielä ennen niiden viemistä toteutukseen, kun taas joidenkin kohdalla voi olla perusteet sille, että osiot ovat jo ennalta määritelty tarkkaan. Tällöin toteutusvaiheeseen siirtyminen on nopeampaa, mutta sen hintana on ketterän työskentelyn ja muutosmahdollisuuksien minimoituminen. [8, s. 135-162.]

2.2 Sisältö

Kuten on jo todettu, suunnittelujärjestelmissä on lähes poikkeuksetta eroavaisuuksia, eikä näin yleispätevää ohjetta sisältöönkään ole. Edellisessä luvussa mainittiin, että jokainen suunnittelujärjestelmä on yhtä oikea eroavaisuuksista huolimatta. Näin voidaan myös todeta, että jokainen suunnittelujärjestelmä voi pitää sisällään lähes mitä tahansa. Sisältö koostuu kaikesta siitä, minkä dokumentoinnista yritys kokee olevan apua suunnittelu- ja sitä kautta käyttöliittymien toteutustyössä.

Jokaiselle on jo oletettavasti muodostunut jonkinasteinen käsitys siitä, mitä kaikkea suunnittelujärjestelmä voi pitää sisällään, jos ajatellaan yleisesti käyttöliittymiä ja elementtejä, joista ne voivat koostua. Järjestelmiä on erilaisia ja eri tarkoituksiin toteutettuja. Toiset voivat olla isojakin kokonaisuuksia ja koostua useammasta eri sivusta ja eri toiminnoista, jolloin on selvää, että käyttöliittymäelementtejä on useita. Tällöin myös suunnittelujärjestelmän sisältö ja ohjeistuksien tarve kasvaa.

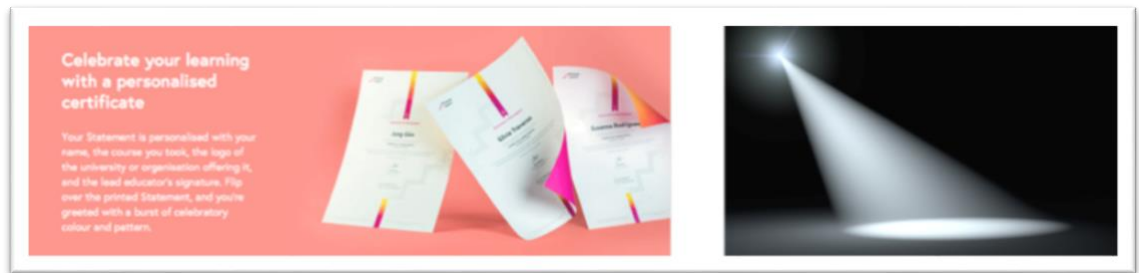
Kuvaan 10 on listattu sisältöesimerkkejä jaoteltuina tyyli- ja suunnitteluohjeisiin sekä malli- ja komponenttikirjastoon. Suunnittelujärjestelmät eivät välttämättä sisällä jokaista

näistä esitetyistä esimerkeistä, ja jotkut voivat sisältää ohjeita, joita kuvaan 10 ei ole listattu.

Tyyli- ja suunnitteluohjeet	Malli- ja komponenttikirjasto													
<p>Usein staattisia ohjeita, jotka kuvaavat miltä järjestelmä näyttää ja tuntuu. Kutsutaan myös brändiohjeiksi.</p> <ul style="list-style-type: none"> Typografia Väripaletti Logo Sijoittelu Välilytykset Ääni ja kieli 	<p>Uudelleenkäytettäviä käyttöliittymäelementtejä, joilla jokaisella on jokin tehtävä.</p> <table border="0"> <tr> <td>Painikkeet</td> <td>Listat</td> </tr> <tr> <td>Lomakkeet</td> <td>Virheilmoitukset</td> </tr> <tr> <td>Ylätunniste</td> <td>Tekstikentät</td> </tr> <tr> <td>Alatunniste</td> <td>Checkbox</td> </tr> <tr> <td>Navigaatio</td> <td>Radiobutton</td> </tr> <tr> <td>Taulukot</td> <td>Valintalista</td> </tr> </table>		Painikkeet	Listat	Lomakkeet	Virheilmoitukset	Ylätunniste	Tekstikentät	Alatunniste	Checkbox	Navigaatio	Radiobutton	Taulukot	Valintalista
Painikkeet	Listat													
Lomakkeet	Virheilmoitukset													
Ylätunniste	Tekstikentät													
Alatunniste	Checkbox													
Navigaatio	Radiobutton													
Taulukot	Valintalista													

Kuva 10. Sisältöesimerkkejä, joita suunnittelujärjestelmät voivat pitää sisällään. Jaoteltu tyyli- ja suunnitteluohjeisiin sekä malli- ja komponenttikirjastoon.

Suunnittelujärjestelmässä jokainen komponentti on nimetty. Nimet ja termit ehkäisevät väärinymmärryksiä, ja etenkin tilanteissa, jossa kehitystyöhön osallistuvilla ei ole yhteistä äidinkieltä, on ennalta määriteltyjen nimien ja termien tärkeyttä mahdoton korostaa liikaa. Ennalta sovittujen nimien tulee kuvata komponentteja ja niiden käyttötarkoituksia, mutta tämän lisäksi nimien olisi hyvä olla sellaisia, että ne jäävät mieleen ja jokainen kehitystyössä mukana oleva hyväksyy ne käyttämällä niitä. [8, s. 109-120.] Kuvassa 11 on hyvä esimerkki persoonallisesta ja mieleenpainuvasta nimestä. Kyseessä on mainoselementti, joka olisi sellaisenaan kuvaava nimi, mutta tämän sijaan kyseisestä elementistä käytetään nimeä *Spotlight*, eli valokeila. Mainoselementin tarkoitus on kiinnittää käyttäjän huomio, ja asettamalla tällainen sisältö valokeilaan, saadaan tavoite toteutettua.



Kuva 11. Esimerkki mieleenpainuvasta elementin nimestä. Kuvassa on mainoselementti, jonka tarkoitus on kiinnittää käyttäjän huomio. Mainoselementti-nimen sijaan elementistä käytetään nimeä *Spotlight*, eli valokeila. [8, s. 112.]

3 Kohti tehokkaampaa käyttöliittymäsuunnittelua ja -toteutusta

Niemistö [15] kertoo Kauppalehden blogissa teknisestä velasta, jonka voisi lähes poikkeuksetta ajatella kuuluvan jokaiseen ohjelmistokehitysprojektiin. Usein tilanne, jossa teknistä velkaa otetaan, on tiukka aikataulu järjestelmän tuotantoon toimittamisessa. Tällöin järjestelmän arkkitehtuurissa ja toteutustyössä tehdään ratkaisuja, joiden tiedetään olevan siinä hetkessä nopeimpia. Yleensä nämä ratkaisut johtavat kuitenkin viimeistään jatkokehityksessä tilanteeseen, jossa ennen uusien toiminnallisuuksien toteuttamista joudutaan parantamaan näitä aiemmin tehtyjä nopeahkoja ratkaisuja. Tekninen velka on tälle hieno termi, mutta kehitystiimeissä tällaisista nopeista ratkaisuista saatetaan käyttää myös termiä ”purkkatoteutus”. Teknistä velkaa käsitellään tarkemmin tietojärjestelmätyön ammattilaiset yhdistävän Sytyke Ry:n sivuilla Jaana Lehojärven kandidaattitutkielmaan viittaavassa blogikirjoituksessa *Tekninen velka – tunnusta, tunnista ja rajoita* [16].

Teknisen velan kaltainen velka voidaan hyvin ajatella kuuluvan myös käyttöliittymä- ja käytettävyyssuunnitteluun sekä käyttöliittymien toteutukseen. Tätä voidaan kutsua suomalaisittain termillä suunnittelovelka, englanniksi *design debt*. Vaikkakin nämä asiat ovat usein hyvin pieniä järjestelmän muihin toteutuksiin verrattuna, vaikuttaa silti kokonaistehokkuuteen se, jos ratkaisuja joudutaan myöhemmin muuttamaan. Käytettävyyttä ja visuaalista ulkoasua voi aina päivittää, mutta mikäli punainen lanka puuttuu alusta alkaen, voi velkaa syntyä helpommin. Suunnittelovelka johtuu usein epäyhtenäisistä tyyleistä ja käytännöistä, jotka eivät ole helposti uudelleen hyödynnettävissä [17].

Suunnittelujärjestelmän avulla käyttöliittymien suunnittelu- ja toteutusvaiheen velkaan pystytään vaikuttamaan pienentävästi ja tämän lisäksi myös nopeutettua eri toimintoja. Standardisoitujen komponenttien myötä suunnitteluun ei tarvitse keskittyä yhtä paljon kuin aikaisemmin, ja aikaa jää enemmän itse toteutustyöhön ja jatkokehitykseen. Prototyyppi, eli toteutettavan järjestelmän konkretisointi ja visualisointi vastaamaan todenmukaista myös nopeutuu, sillä ennalta suunniteltuja komponentteja on helppo yhdistellä Lego-palikkoiden tavoin.

Suunnittelujärjestelmän myötä myös perehdytyksen tehokkuus kasvaa. Uusi toteuttaja tai suunnittelija pystyy perehtymään organisaation visuaaliseen runkoon suunnittelujärjestelmän avulla. Suunnittelujärjestelmä tarjoaa myös sen edun, että järjestelmästä toiseen siirtyvälle suunnittelijalle tai toteuttajalle ohjeistot ja termit ovat jo ennalta tuttuja.

Ketterän työskentelyn myötä isommissa järjestelmissä on riskinä se, että samaa tehtävää suorittavat komponentit toteutetaan eri tavoin eri näytöillä. Tämä vaikuttaa negatiivisesti järjestelmän käyttäjäkokemukseen ja laskee toteutustyön tehokkuutta, kun kertaalleen toteutettuja komponentteja ei käytetä uudelleen.

Suunnittelujärjestelmän avulla kaikkia edellä mainittuja seurauksia pystytään minimoimaan. Kun eri käyttöliittymäkomponentit kuvataan riittävän tarkalla tasolla ja tarjotaan pohja komponenttien koodille, pystytään yhtenäisen linjan säilymistä parantamaan. Tämä selkeyttää myös käyttöliittymän suunnitteluvaihetta, kun jokainen suunnittelija käyttää saman näköisiä, toteuttajillekin tuttuja komponentteja. Suunnittelujärjestelmä muodostaa myös ikään kuin yhteisen kielen suunnittelijoille, toteuttajille ja suunnittelujärjestelmän omistajalle. [17.]

Suunnittelujärjestelmä on jatkuvasti kehittyvä ja päivitettävä ohjeisto. Sen voidaan ajatella elävän ja hengittävän yrityksen kasvamisen ja kehityksen mukaan, ja se luo yhtenäisen linjan ja punaisen langan käyttöliittymien suunnitteluun ja toteutukseen. [3.] Tehokkuus kasvaa ja resursseja voidaan keskittää laajemmalle. Lisäksi järjestelmien ulkoasullisella laadulla on mahdollisuus parantua, kun pohjalla on määritykset sille, miten tietyt asiat tulee esittää. Pidettäessä suunnittelujärjestelmästä hyvää huolta se antaa sen, mitä siltä odotetaan.

3.1 Nykyiset ohjeistukset Venlassa

Suunnittelujärjestelmän luomisessa ensiaskel on tarkastella nykyisiä toimintatapoja ja ohjeistuksia. Kehitysvaiheessa oleva järjestelmä antaa reaaliaikaista palautetta siitä, mitkä asiat ovat hyvällä tasolla ja mitkä koituvat kompastuskiviksi.

Parhailtaan kehitettävän operatiivisen järjestelmän, Venlan, käyttöliittymää koskevat ohjeistukset ovat kehittyneet toteutuksen edetessä, mutta tarve selkeämmälle, yhtenäisemmälle ja jokaista osapuolta tukevalle ohjeistolle on edelleen. Paikallisen osaamisen lisäksi kehitystiimiä on kasvatettu kansainvälisillä resursseilla ja työskentelyssä käytetään suomen kielen sijaan englantia. Yhteisen äidinkielen puuttuminen antaa lisäperusteen suunnittelujärjestelmän käyttöönottoon, sillä tämän avulla kommunikoinnissa esiintyviä vääринymmärryksiä pystytään minimoimaan.

Kesän 2018 alussa Venlan käyttöliittymädokumentaatioita on lähdetty laajentamaan suunnitteluohjeistolla, johon on kuvattu käyttöliittymäkomponenttien tyylejä ja käyttäytymisiä esimerkkien avulla. Suunnitteluohjeistoa on täydennetty tarpeen esiintyessä. Tarve on saattanut tulla esimerkiksi määrittelyn yhteydessä havaitusta epäselvyydestä, mutta joskus tarve on huomattu vasta silloin, kun kompastuminen on jo tapahtunut. Tällöin ohjeistukset on saatettu antaa pikaisesti suullisesti tai pikaviestintävälineitä käyttäen ja niiden dokumentointi on saattanut unohtua. Suunnittelujärjestelmässä tavoitteena on pyrkiä ennaltaehkäisemään kompastumisia luomalla ja parantamalla ohjeistoja ennen kuin isompi kivi tulee vastaan. Tehokkainta on, jos suunnittelujärjestelmän avulla pystytään olemaan jatkuvasti askel edellä.

Venlan käyttöliittymätoteutuksessa käytetään Angular Materialia [18], joka sisältää Google Material Designin komponentteja Angular-pohjaisiin toteutuksiin. Material on avoimen lähdekoodin järjestelmä, joka tarjoaa työkaluja käyttöliittymätoteutuksiin, ja itseasiassa, se on myös suunnittelujärjestelmä. Materialissa olevia komponentteja käytetään Venlassa kuitenkin harvoin sellaisinaan ja niihin tehdään lähes poikkeuksetta muutoksia muiden määrittelyjen yhteydessä, joihin lukeutuu muun muassa tyylin soveltuvan asettelun tarkennus.

Venlassa on käytetty tähän asti melko vähän yhtenäisiä nimiä eri komponenteille. Nimeämiset ovat painottuneet lähinnä painikkeisiin. Kuten jo aiemmin mainittiin,

tällaisessa tilanteessa, jossa osapuolilla ei ole yhteistä äidinkieltä, ennalta sovitut nimet ja termit ehkäisevät väärinymmärryksiä ja helpottavat sekä nopeuttavat niin suunnittelijoiden kuin toteuttajienkin työtä.

Rautalankamallit voidaan jakaa tarkkuustason mukaan kahteen ryhmään, joista käytetään termejä *low-fidelity wireframes*, eli matalan tarkkuustason rautalankamallit, ja *high-fidelity wireframes*, eli korkean tarkkuustason rautalankamallit. Näiden väliin asettuvien suunnitelmien kohdalla saatetaan silloin tällöin käyttää omaa termiä, *middle-fidelity wireframes* tai *medium-fidelity wireframes*. Matalan tarkkuustason rautalankamallit ovat nimensä mukaisesti vain karkeita hahmotelmia, joissa on hyvin vähän yksityiskohtia. Sivun eri osioita havainnollistetaan laatikoin, ja yksityiskohtaisempi sisältö jätetään kuvaavamatta. Korkean tarkkuustason rautalankamalleissa lopputulos on hyvin oikean näköistä, ja näissä käytetään tekstityylejä, kuvia ja värejä sellaisina, kuin ne ovat käyttöliittymälleen määrää toteuttaa. [19.]

Venlan käyttöliittymäsuunnitelmat ovat kuuluneet alusta asti pääosin matalan ja korkean tarkkuustason välimaastoon, mutta ajoittain on saattanut esiintyä myös korkean tarkkuuden hahmotelmia jopa siten, että nämä ovat näyttäneet toteutuksesta otetuilta kuvankaappauksilta. Rautalankamallien tarkoitus ei ole kuvata tarkkaan sitä, miltä järjestelmä tulee visuaalisesti näyttämään, vaan tarkoitus on keskittyä sisältöön ja toiminnallisuuksiin. Käyttöliittymien tarkka, yksityiskohtainen ja oikean näköinen suunnittelu vie suunnittelijoilta paljon aikaa ja sen lisäksi kyseisten suunnitelmien läpikäynti voi harhauttaa ajatukset pois toiminnallisuuksista ja ideoinnista. [19.] Kun jokainen osapuoli hyväksyy suunnittelujärjestelmän työskentelyn pohjaksi ja sisäistää siellä esitetyt ohjeistot, pystytään jo matalammankin tarkkuustason rautalankamallin pohjalta hahmottamaan lopullisen toteutuksen visuaalinen ulkonäkö.

Venlan suunnitelmat ovat noudattaneet muutoin rautalankamallien peruspiirteitä, mutta näistä poiketen suunnitelmiin on lisätty värejä harmaan eri sävyjen sijaan. Ketterän kehityksen ja värien käytön myötä käyttöliittymäsuunnitelmissa on havaittu eroavaisuuksia, ja tämä on saanut aikaan epäselvyyksiä toteuttajienkin keskuudessa. Epäselvyydet ovat syntyneet, jos esimerkiksi jo toteutettuun näyttöön haluttu lisätoiminnallisuus on suunniteltu alkuperäiseen rautalankamalliin tai jos uuden näytön rautalankamallissa on käytetty vanhalla tyylillä olevaa pohjaa, eikä kummassakaan tapauksessa värityksiä ole muistettu muuttaa uuden, sovitun tyylin mukaisiksi.

3.2 Ohjeistukset jo toteutetuissa järjestelmissä

Kun samalle organisaatiolle toteutettuja järjestelmiä on useita, on järkevää ottaa nämäkin tarkasteluun. Näin saadaan hyvä kokonaiskuva ohjeistuksien eri luontitavoista, yksityiskohtaisuuksista ja siitä, viitataan joidenkin projektien osalta suoraan jonkun toisen järjestelmän ohjeistuksiin vai onko kunkin projektin kohdalla lähdetty liikkeelle puhtaalta paperilta. Tässä luvussa tarkastellaan kahta järjestelmää, joita kutsuttakoon nimillä A ja B. Myös näiden järjestelmien pohjana käytetään Materialia, aivan kuten edellisessä luvussa läpikäydyssä Venlassa. Järjestelmien A ja B käyttöliittymiä koskevat ohjeistukset eroavat niin toisistaan kuin myös Venlasta.

Järjestelmässä A käyttöliittymätoteutuksen pohjana ovat olleet matalan tarkkuustason rautalankamallit, joissa on käytetty väreinä ainoastaan mustaa sekä valkoista ja sisältöosista on tehty otsikkotasoisia laatikoita ottamatta kantaa yksityiskohtaisempiin sisältöihin. Käytössä olevat teknologiat, Angular ja Material, ovat tämänkin järjestelmän osalta asettaneet tietyt suuntaviivat ulkoasulle. Tarkempia suunnitelmia ja esiin nousseita muutoksia ei ole päivitetty kyseisen järjestelmän omiin käyttöliittymädokumentaatioihin, vaan nämä on kuvattu toteuttajille työpyynnön ohessa. Järjestelmän A käyttöliittymäparannuksissa on hyödynnetty järjestelmässä C luotua käyttöliittymäohjeistusta, joka tarjoaa tukea sähköisten asiointipalveluiden käyttöliittymien suunnitteluun ja saavutettavuuden [20] noudattamiseen.

Järjestelmälle B on toteutettu käyttöliittymäkonsepti, jossa on kuvattu rautalankamallien avulla palvelun keskeiset ominaisuudet ja toiminnallisuudet. Konseptissa muistutetaan, että rautalankamallit ovat vain suuntaa antavia ja niiden sisältökin on viitteellinen. Järjestelmään on luotu konseptoinnin lisäksi visuaalisen ohjeiston dokumentaatio, joka on laadittu tukemaan tämän järjestelmän suunnittelua ja kehitystä. Visuaalinen ohjeisto pitää sisällään sivuston rakennekuvauksen, johon kuuluvat ylä- ja alatunniste, sivunavigaatio sekä sisältöosio. Tämän lisäksi dokumentaatioon on kerätty väri- ja fonttiohjeistuksia kuvan 12 mukaisesti.

Elementti	Tausta	Teksti
Ylätunniste	#004477	#e2e2e2
Vasemman laidan navigaatio	white (#ffffff)	#000000; 60% läpäisevyys
Vasemman laidan navigaatio, kaksitasoinen	#f5f5f5	
Vasemman laidan navigaatio, aktiivinen linkki	#e0e0e0	#212121
Vasemman laidan navigaatio, muutosten ilmais	#004477	#####
Sisältöosa	#fafafa	#212121
Linkit; käymätön	transparent	#004477
Linkit; käyty		#00afef

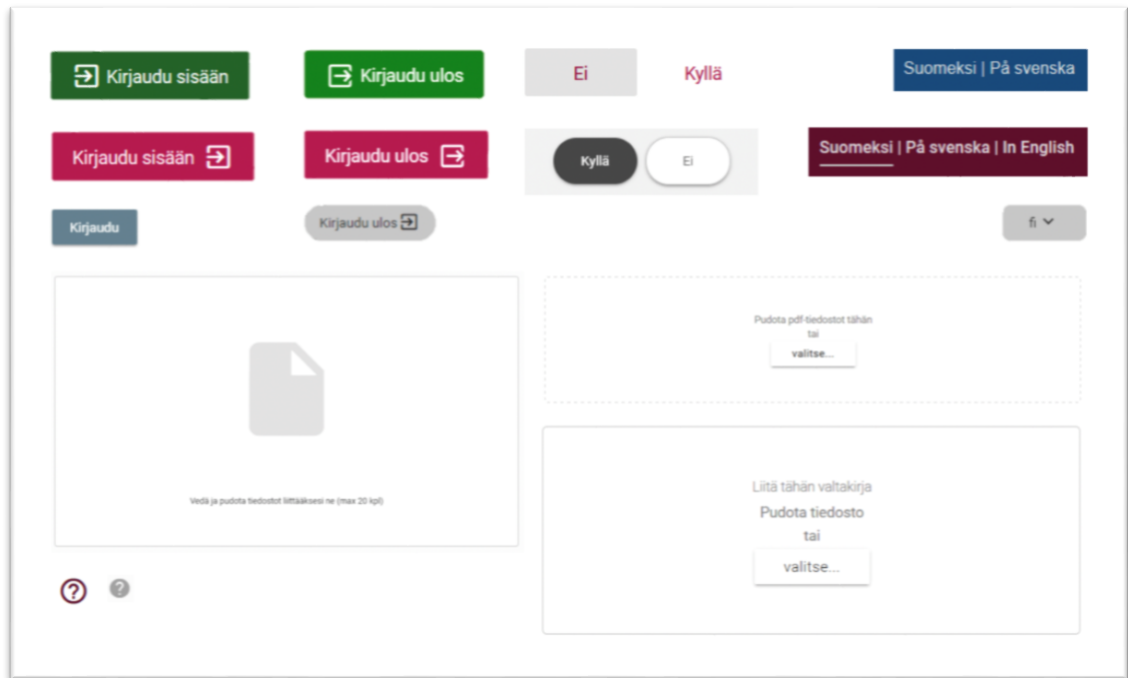
Tekstityyppi	Koko
Ylätunniste	
Valikkoikoni	35px
h1	24px
span	14px
Korttikomponentti	
otsikko	20px
Oletuspainike/-nappi	
ikonit	22px

Kuva 12. Oikeusministeriölle toteutetun järjestelmän visuaalisesta ohjeistosta poimittuja ohjeita koskien värejä ja fontteja.

3.3 Inventointivaihe

Suunnittelujärjestelmän tarkoitus on toimia yhteisenä kielenä jokaiselle osapuolelle projektista, tiimistä ja roolista riippumatta. Jotta tällainen yhtenäinen suunnittelujärjestelmä saadaan luotua, on tärkeää hahmottaa kokonaiskuva siitä, kuinka montaa eri tyyliä on tähän asti käytetty eri järjestelmien eri komponenteille. Kokonaiskuvan hahmottamiseksi suoritetaan inventointi. Inventoinnin tarkoitus on koota yhteen käyttöliittymien visuaaliset piirteet ja elementit. Inventointivaihe voi olla isojen järjestelmien kohdalla aikaa vievää, mutta kuitenkin erittäin hyödyllistä. Tämä vaihe antaa usein viimeisen perustelun sille, miksi suunnittelujärjestelmästä todella olisi hyötyä. Inventointi paljastaa usein sen, että samaa tehtävää suorittavia komponentteja on toteutettu eri tavalla, eikä uudelleenkäytettävyyden tarjoamia etuja ole hyödynnetty. [17; 8, s. 185-189.]

Kuvaan 13 on poimittu Venlasta sekä järjestelmistä A ja B muutamia esimerkkejä samoja tehtäviä suorittavista komponenteista, joiden toteutuksessa on silmin havaittavia eroja. Tunnusomaisia värejä ei lasketa näiksi eroiksi, sillä niiden avulla on tarkoitus erottaa järjestelmät toisistaan.



Kuva 13. Venlasta sekä järjestelmästä A ja B inventoinnin yhteydessä poimittuja käyttöliittymien osia. Muun muassa samoja toimintoja suorittavissa painikkeissa on eroavaisuuksia, mikä kertoo siitä, ettei aiemmin toteutettuja painikkeita ole uudelleenkäytetty.

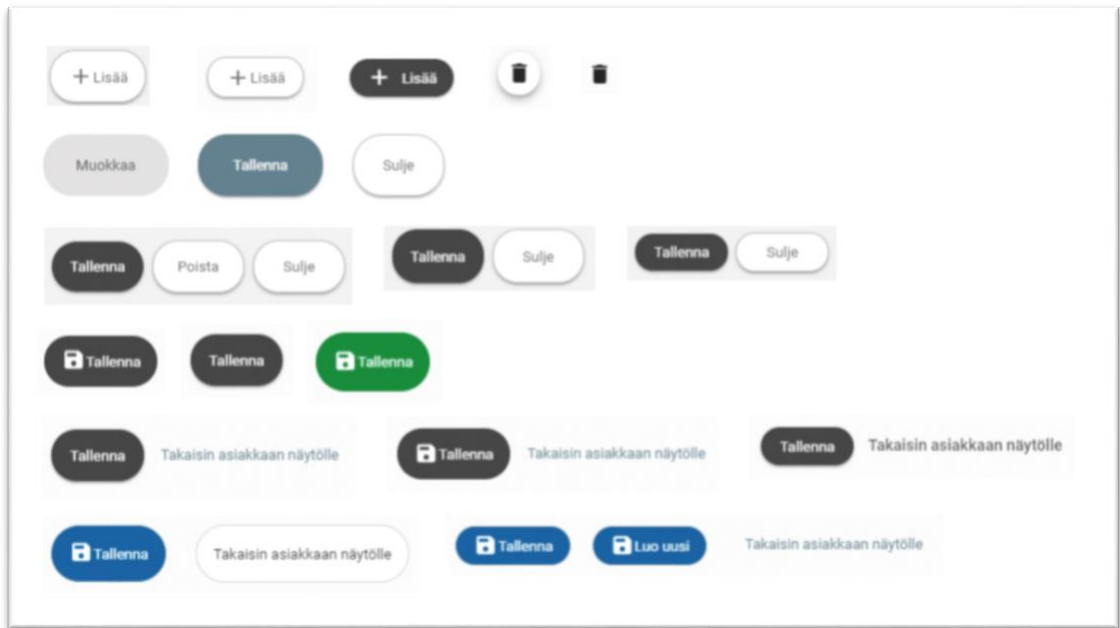
Sisäänkirjautumispainikkeet ovat kahdessa järjestelmässä lähes samanlaiset, ja erona on ainoastaan tekstien ja ikonien eri järjestykset. Kolmannessa, Venlassa, vastaava painike on täysin erilainen, mutta mainittakoon, että kyseisessä järjestelmässä on erillinen kirjautumisikkuna, kun taas järjestelmässä A ja B sisäänkirjautuminen tapahtuu ylätunnisteen kautta. Myös uloskirjautumispainikkeissa tekstit ja ikonit ovat eri järjestyksessä, mutta tämän lisäksi Venlassa olevan painikkeen muotokin on eri. Kaikissa järjestelmissä uloskirjautumispainike sijaitsee samassa paikassa ylätunnisteessa.

Kun mikä tahansa järjestelmä pyytää käyttäjää vahvistamaan toimintansa *kyllä*- tai *ei*-valinnalla, olisi näissä hyvä noudattaa aina samaa järjestystä, samoin kun kielivalinnoissa. Kielivalintojen järjestykset ovatkin esimerkkijärjestelmissä samat, mutta toteutustavoissa on silti eroja: Venlan kielivalinta on toteutettu alavetovalikolla, kun taas järjestelmissä A ja B nämä ovat rinnakkain olevia valintoja. Järjestelmässä A valitun kielen merkiksi tulee alleviivaus, kun taas järjestelmässä B valittu kieli ei eroa tyyliltään vaihtoehtoisesta kielestä.

Kuvaan on tuotu myös isompi elementti, liitetiedoston lisääminen. Jokaisessa esimerkkijärjestelmässä liitetiedostolle varattu pudotusalue on toteutettu eri tavalla niin tekstien, kehyksien kuin senkin osalta, avataanko liitteiden selaaminen omalta laitteelta pudotusalueen sisältä vai sen ulkopuolelta.

Sen lisäksi, että inventointi suoritetaan monelle järjestelmälle ja tämän pohjalta vertaillaan, mitä yhteistä ja mitä eroa toteutuksissa on, voi silloin tällöin olla järkevää tarkastella järjestelmiä yksistäänkin. Kuten luvussa 3 mainittiin, ketterän työskentelyn myötä isommissa järjestelmissä riskinä on se, että samaa tehtävää suorittavat komponentit toteutetaan eri tavoin eri näytöillä. Tämä on seurausta siitä, ettei komponenteille ole luotu globaalia tyyliä. Tämä saa aikaan ei-toivottuja seurauksia käyttäjäkokemukseen sen lisäksi, että toteutuksen tehokkuuden ei voida olettaa olevan maksimissaan. Tällaisessa tilanteessa tullaan ottaneeksi myös luvussa 3 mainittua suunnitteluvälikä, sillä toteutuksia joudutaan myöhemmin muuttamaan, jotta saavutetaan Nielsenin heuristiikan [21] kohta *Yhteneväisyys ja standardit*. Järjestelmissä tulisi käyttää yhteneväisiä komponentteja niin värien kuin muotojenkin osalta ja aina samoihin tarkoituksiin.

Kuvaan 14 on koottu inventoinnin tuloksena Venlassa esiintyviä painikkeita. Inventointi paljastaa, että ketterästi päivitetyn ulkoasun lomassa samoja toimintoja suorittaviin painikkeisiin ei ole luotu globaalia tyyliä. Tämä on johtanut siihen, että esimerkiksi *Tallenna*-painikkeella on seitsemän erilaista versiota niin kokonsa, värinsä kuin ikoniensakin osalta.

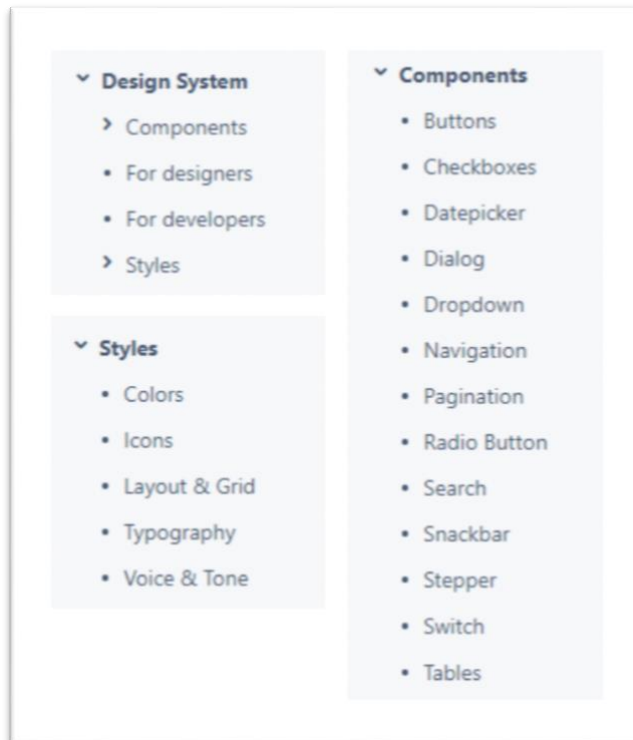


Kuva 14. Venlasta poimittuja painikkeita. Painikkeet on toteutettu yksilöinä globaalien tyylien sijaan, ja sen vuoksi järjestelmässä esiintyy muun muassa seitsemän erilaista *Tallenna*-painiketta.

4 Suunnittelujärjestelmä Oikeusrekisterikeskukselle

Oikeusrekisterikeskukselle luotu pohja suunnittelujärjestelmästä on toteutettu tiimin yhteiseen dokumentaatioarkistoon, Atlasianin tarjoamaan Confluence-sisältöyhteistyöohjelmistoon. Ohjeistot on toteutettu englanniksi, sillä Venlassa ja useissa muissakin hankkeissa kehitystiimeihin kuuluu kansainvälisiä resursseja paikallisen osaamisen lisäksi.

Nopeasti kehittyvässä isossa järjestelmässä, kuten Venlassa, kehitystiimi koostuu useasta osapuolesta. Oikeusministeriö ja Oikeusrekisterikeskus ovat isoja organisaatioita, ja järjestelmiä on useita. Tästä johtuen suunnittelujärjestelmän tulisi palvella mahdollisimman hyvin monia eri henkilöitä ja rooleja. Kuvassa 15 on esitettyä suunnittelujärjestelmän sisältöpohja, joka on laajennettavissa tarvittavine lisäkohtineen.



Kuva 15. Suunnittelujärjestelmän sisältö, joka elää ja kehittyy muun toiminnan ohessa.

Kuten jo todettu, suunnittelujärjestelmä on elävä, hengittävä työkalu, ja se kehittyy käsi kädessä muun toiminnan kanssa. Tällä hetkellä suunnittelujärjestelmä on jaettu komponenttikirjastoon ja tyyliohjeisiin, ja näiden lisäksi sisältöön kuuluu kaksi irrallaan olevaa ohjeistoa.

Käyttöliittymäsuunnittelijoiden työpanoksen tarve on suuri, joten osaavia henkilöitä tarvitaan. Ajoittain määrittelijätkin pääsevät esittämään ideoitaan käyttöliittymäsuunnitelmien muodossa, joten suunnitteluperiaatteet on hyvä tuoda esille. Kuvassa 16 on suunnittelijoille luotu erillinen ohjeisto, jossa on kuvattuna käyttöliittymäperiaatteita sekä suunnitteluprosessi. Näiden lisäksi suunnittelijoita varten on luotu ohjeet rautalankamallien luomiseen. Toteuttajille on vastaava, hieman suppeampi ohjeisto, jossa on teknisten yleisohjeiden lisäksi kerrottu muun muassa, mitä rautalankamallien eri merkinnät tarkoittavat.

For designers

- For a start
- UI principles
- Design process
- The way to create UI designs

For a start

1. Install requirement designing tools.
2. Get familiar with UI principles and the way to create UI designs.
3. Get familiar with [Angular Material](#) components.

UI principles

- **Consistency and standards**
Unified style supports usability and provide trustworthy feeling. Addition in usability, learnability improve when similar elements have consistent look and function in similar way. This minimize the user's memory load and they will learn faster how to use system.
- **User control and freedom**
Provide clearly marked "emergency exit" to leave the unwanted state, if function has choose by mistake.
- **Error prevention**
Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
- **Error and forgiveness**
When error occur, design must minimize the negative impact. Provide to user way to reverse them if possible. Use plain language, precisely indicate the problem and constructively suggest a solution.
- **Help**
If it is necessary, provide help. List concrete steps for example, avoid long and complex sentences.

Design process

```

graph LR
    A[Investigate background] --> B[Analyse background with customer]
    B --> C[Create wireframe and add brand color]
    C --> D[The revision of the wireframe with for example end users, stakeholders, and project team]
    D --> E[Receive approval for the design]
    E --> F[Implementation starts]
    F --> G[Communicate with developers]
  
```

The way to create UI designs

- Use components from [Angular Material](#). Select components that serves best
- Add additional information for developers if needed, make clear which functions are new if use former base and make clear which functions are purpose to implement later. Add info for every text field and area which type they are. See different types from [here](#)
 - Purpose of colors:
 - Purple - Additional information
 - Red - Implement later
 - Blue - New thing
- Use unified, middle-fidelity wireframe style. Do not make wireframes look real!
- If wireframe has selection that opens new section/field, please make sure the section/field that appearing is indented. Example with checkboxes:

Kuva 16. Suunnittelijoille luotu erillinen ohjeisto, jossa on kuvattuna käyttöliittymäperiaatteita sekä suunnitteluprosessi.

Kuvassa 17 on suunnittelujärjestelmän *Voice & Tone* -osio. Tässä osiossa kerrotaan, miksi järjestelmän viestien sisältöön tulee kiinnittää huomiota, mitä tulee välttää ja minkälaista kieltä on hyvä käyttää. Tätä osiota jatkokehittämällä jokaiselle muodostuu yhä parempi käsitys siitä, mitä ja miten järjestelmien tulee käyttäjilleen viestiä.

Voice & Tone

When system takes care of communication, it is important to concentrate on what say and how - the same voice all the time, but tone changes. Provide for users the information they need and show that you care.

Voice - the way of communication, part of brand.

- concise
- direct
- clear and easy to understand
- official

Tone - depending on the context and situation. Think user's feeling when they encounter message.

Avoid	Use
Slang and buzzwords	Active voice
Humor	Positive language
Unnecessary and redundant information	Concise and direct message
Long, complex sentences	
Technical language	

Style of error messages

- Give the user clear instructions on how to correct the error - do not be ambiguous
- Be short and meaningful
- Avoid uppercase text

Examples:

- ✓ Username and/or password does not match
- ✗ Sorry, but you cannot login to the application

Kuva 17. Suunnittelujärjestelmän *Voice & Tone* -osio, jossa kerrotaan järjestelmässä käytettävän kielen merkitys ja miten sitä tulee käyttää.

Tyyliohjeiden alta löytyy myös muun muassa ikoneja koskevat ohjeistukset. Yhdellä ikonilla tulee olla yksi, ainoa tehtävä. Ajoittain voi olla haastavaa löytää käyttötarkoitukseen sopivia ikoneita, etenkin, jos konteksti on hyvin lähellä jotain toista. Ylläpidettäessä ikonikirjastoa (kuva 18) pystytään nopeasti katsoa, onko suunniteltu ikoni otettu käyttöön jo jossain muussa yhteydessä. Ikonikirjaston yhteydessä on annettu ohjeet toteutusta varten.













Icons

Icons provide visual context and better usability. All of the icons can be found [here](#). Follow the implementation instructions from the same page. If you need additional icons, use [Material icons](#) in your project.

Rules for icons. It is mentioned on the specific row, if exceptions:

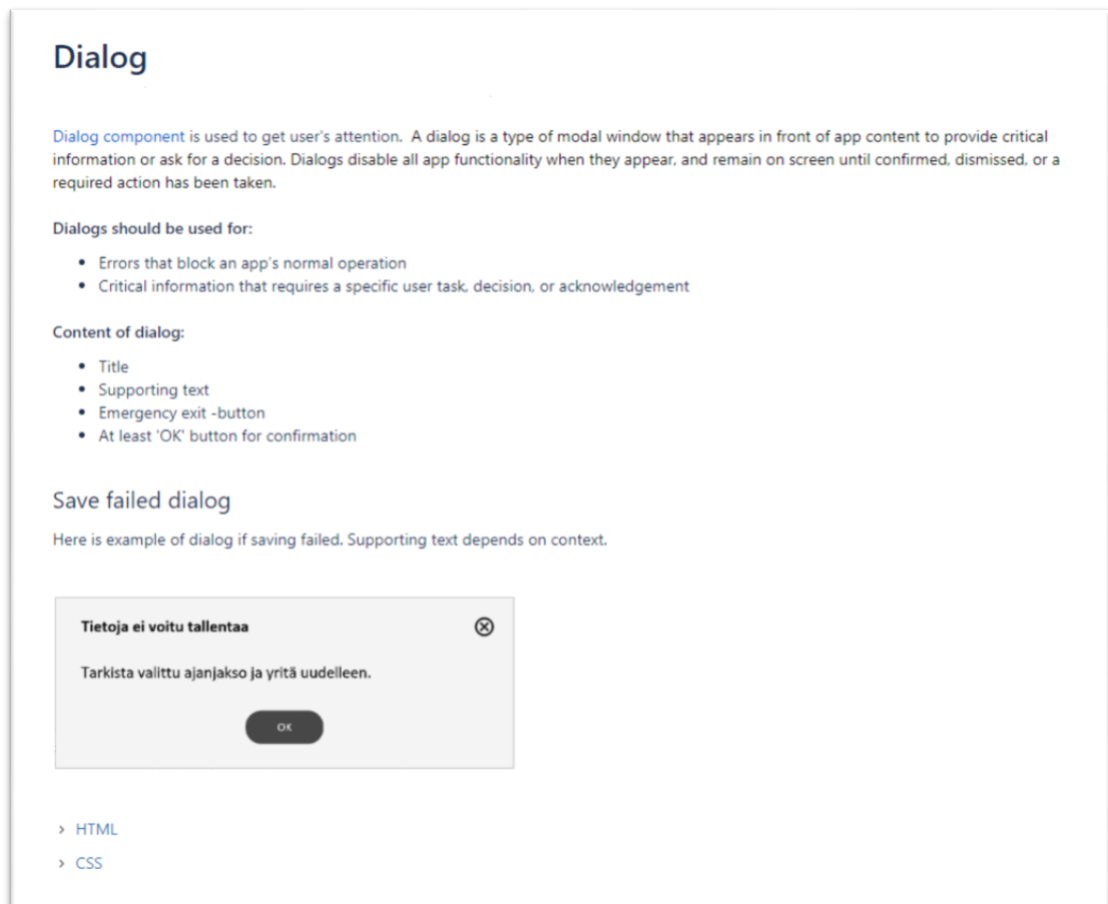
- Size: 18dp
- Spacing to other elements: 12dp
- Color: black

General icons

Usage	Icon/text	Name/file	Usage	Icon/text	Name/file
Back to home page		baseline_home_black_18dp	Save button		baseline_save_black_18dp
Search		baseline_search_18dp	Edit content		baseline_edit_black_18dp
Sign out		baseline_exit_to_app_18dp	Delete content		baseline_delete_black_18dp
System help		baseline_help_outline_18dp	Exit from modal		outline_cancel_black_18dp
Page content help		baseline_help_18dp	Edit process address		baseline_edit_location_black_18dp
Client card, name		baseline_person_black_18dp	Add new row		baseline-add_circle-18dp Background color: #02912f

Kuva 18. Ikonikirjastosta pystyy nopeasti katsomaan, onko suunniteltu ikoni jo käytössä jossain muussa yhteydessä. Samassa yhteydessä on annettu toteutuksessa huomioon otettavia ohjeita.

Komponenttikirjaston alta löytyy muun muassa *Dialog*-komponenttia koskevat ohjeistukset, jotka on nähtävillä kuvassa 19. Tässä on kerrottu muun muassa, missä yhteydessä kyseistä komponenttia voidaan käyttää ja mistä se koostuu. Näiden lisäksi on annettu esimerkki *Voice & Tone* -ohjeistusta noudattavasta ilmoituksesta, jonka käyttäjä saa, jos tallennus epäonnistuu virheellisten tietojen takia. Esimerkin yhteydessä tarjotaan valmis koodi komponentin käyttöön.



Kuva 19. *Dialog*-komponenttia koskevat ohjeistukset. Käyttötarkoituksen ja sisällön lisäksi on annettu esimerkki tallennusvirheen ilmaantuessa esitetystä ilmoituksesta, ja tälle on tarjottu toteutusta varten koodi.

Edellä mainittujen ohjeistuksien lisäksi suunnittelujärjestelmään oli myös tarve luoda erityyppisille tekstikentille määrytykset. Venlasta muodostetaan monisivuinen tulosteiden joukko, joka koostuu monen eri näytön tiedoista. Jotta tuloste on aina saman pituinen syötettyjen tietojen laajuudesta riippumatta ja jotta käyttäjän on mahdollista oppia osuukseen sijoittuminen tulosteelle, tulee tekstikenttien merkkimääriä rajoittaa käyttöliittymällä. Koska eri asiayhteyksissä on tarve eripituisille selvityksille, on tarpeen määrittellä erilaisia tekstikenttätyyppisiä, jotka ovat nähtävissä kuvassa 20. Määrytyksissä olevia kenttätyyppien nimiä käytetään jatkossa käyttöliittymäsuunnitelmien yhteydessä.

Field types

- Text fields and areas
- Number fields

Text fields and areas

Table with different text field and text area types and their properties

Name	Type	Chars	Scalable	Calculator	Usage example
Small text	Text field	50			First name, last name, address, e-mail, written dates
Medium text	Text field	100			Two small text data in one field (name & address etc.)
Small description	Text area	255	x	x	
Medium description	Text area	400	x	x	
Large description	Text area	1000	x	x	
Extra-large description	Text area	4000	x	x	

Table with specific fields:

Usage	Type	Chars	Scalable	Calculator
SSN	Text field	11		
Business ID	Text field	9		

Kuva 20. Määritellyt tekstikenttätyypit, jotka parantavat käytettävyyttä sekä mahdollistavat tulosteen vakiosivutuksen.

5 Yhteenveto ja pohdinta

Insinööriyössä perehdyttiin suunnittelujärjestelmiin. Työssä selvitettiin, mikä suunnittelujärjestelmä on ja mikä sen arvo on järjestelmien käyttöliittymien suunnittelu- ja toteutustyössä. Työllä haluttiin osoittaa Oikeusrekisterikeskukselle, että suunnittelujärjestelmä olisi koko organisaation kattavana hyödyllinen työkalu. Suunnittelujärjestelmästä luotiin myös pohja kehitysvaiheessa olevan Venla-järjestelmän tarpeiden pohjalta.

Nopeasti kehittyvässä isossa järjestelmässä, jossa kehitystiimi koostuu useammasta käyttöliittymäsuunnittelijasta, määrittelijästä ja toteuttajasta, on suunnittelujärjestelmän luonnille perusteet. Perustelut tällaisen käyttöönoton hyödyllisyydestä kuvattiin työssä siten, että niiden pohjalta on helppo nähdä hyödyt niin suunnittelijoiden, toteuttajien kuin

myös Oikeusrekisterikeskuksen näkökulmasta. Oikeusrekisterikeskus on iso organisaatio, ja he kehittävät toimintaansa jatkuvasti tarjoamalla operatiivisten järjestelmien lisäksi sähköisiä asiointipalveluita kansalaisten käyttöön. Jotta palvelut olisivat yhtenäisiä ja tunnistettavissa riippumatta järjestelmätoteuttajasta, on koko organisaation kattaville ohjeistuksille paikkansa.

Perehtymisvaiheen jälkeen lähdettiin tekemään taustatyötä tutkimalla kehitysvaiheessa olevan Venlan sekä jo tuotannossa olevien järjestelmien käyttöliittymiä koskevia ohjeistuksia ja toteutuksia. Toteutuksien pohjalta kerättiin havainnollistavaa materiaalia sellaisista ei-toivotuista esiintymistä, joita suunnittelujärjestelmän avulla pystytään minimoimaan. Tämän vaiheen jälkeen Venlan dokumentaatioihin rakennettiin suunnittelujärjestelmän pohja, jonne lähdettiin luomaan ohjeistuksia sen pohjalta, mitkä ovat olleet niitä suurimpia kompastuskiviä Venlan kohdalla niin suunnittelun kuin myös toteutuksenkin osalta. Vaikka luodut ohjeistot ovat osittain Venla-kohtaisia, sisältävät ne paljon yleisiä periaatteita, jotka ovat sellaisenaan hyödynnettävissä myös muihin järjestelmiin.

Suunnittelujärjestelmästä on pidettävä jatkuvasti huolta, jotta se palvelee parhaalla mahdollisella tavalla jokaista osapuolta. Suunnittelujärjestelmä elää ja kasvaa yrityksen ja liiketoiminnan kanssa käsi kädessä. Jotta sitä olisi miellyttävä käyttää ja se olisi helposti jalkautettavissa projektista ja roolista toiseen, olisi Atlassianin tarjoaman Confluence-sisältöyhteistyöohjelmiston sijaan selainpohjainen, luvussa 2 käytyjen esimerkkien kaltainen toteutustapa kenties järkevin. Toteuttajat ovat tottuneita Internet-ohjeiden käyttäjiä, ja suunnittelijat inspiroituvat visuaalisuudesta. Vaikka perinteisten dokumentaatiotyökalujen käyttö on joissain tehtävissä perusteltua, haluttaessa ottaa askel kohti nykyaikaa, saattaisi moderni ratkaisu olla monelle miellyttävämpi käyttää, ja se nostaisi varmasti entisestään julkishallinnon organisaation arvoa.

Koko organisaation kattavan suunnittelujärjestelmän luomisessa olisi hyvä, jos sen suunnitteluun ja kehittämiseen osallistuisi useampi henkilö. Suunnittelujärjestelmän kohdalla on tärkeää, että jokainen hyväksyy sen sisältämät ohjeistot ja on kykeneväinen noudattamaan niitä. Tästä syystä olisi erittäin hyödyllistä, että järjestelmäriippumatonta suunnittelujärjestelmää luodessa pidettäisiin työpajoja, joihin voisi osallistua eri järjestelmien suunnittelijoita, määrittelijöitä ja toteuttajia Oikeusrekisterikeskuksen eri

vastuuhenkilöiden lisäksi. Näin olisi useampi eri näkökulma ja ohjeistot olisivat yhteisesti sovittuja, jolloin niiden noudattaminenkin olisi luontevampaa.

Lähteet

- 1 Korvola, Eeva. 2010. Brändi- ja ulkoasuprosessi julkishallinnon organisaatiossa. Opinnäytetyö. Metropolia Ammattikorkeakoulu. Theseus-tietokanta.
- 2 Ulkoministeriön Graafinen Ohjeisto. 2018. Verkkoaineisto. <<https://mediabank.finland.fi/l/knqtg6bz8Vvb>>. 12.1.2018. Muokattu 16.1.2018. Luettu 10.9.2018.
- 3 Laitinen, Niko. 2018. Design System, avain skaalautuvaan designiin. Verkkoaineisto. <<https://www.nitor.com/fi/uutiset-ja-blogi/design-system-avain-skaalautuvaan-designiin>>. 1.3.2018. Luettu 28.8.2018.
- 4 Oikeusrekisterikeskuksen työjärjestys. 2017. Verkkoaineisto. <https://www.oikeusrekisterikeskus.fi/material/attachments/ork/oikeusrekisterikeskus/T3Weoflcv/tyojarjestys_voimassa_1.3.2017_alk_allek.docx.pdf>. Luettu 21.9.2018.
- 5 Talous- ja velkaneuvonta siirtyy oikeusaputoimistoille. 2017. Verkkoaineisto. <<https://oikeus.fi/oikeusapu/fi/index/tiedotteet/2017/11/talous-javelkaneuvontasiirtyyoikeusaputoimistoille.html>>. Luettu 23.9.2018.
- 6 Oikeusministeriön oikeushallinto-osasto. Verkkoaineisto. <<https://oikeusministerio.fi/oikeushallinto-osasto>>. Luettu 5.10.2018.
- 7 Simon, Gary. 2018. What is a Design System? Design Systems 101 for Designers. Verkkoaineisto. <<https://www.youtube.com/watch?v=wc5krC28ynQ>>. Katsottu 12.8.2018.
- 8 Kholmatova, Alla. 2017. Design Systems – A practical guide to creating design languages for digital products. E-kirja. Saksa: Smashing Media AG.
- 9 Braga, Gabi. 2017. Pattern Library, Style guides, Design Systems. Do you need one? Verkkoaineisto. <<https://blog.prototypr.io/pattern-library-style-guides-design-systems-do-you-need-one-b7857af0f255>>. Luettu 24.11.2018.
- 10 Frost, Brad. 2013. Atomic Design. Verkkoaineisto. <<http://bradfrost.com/blog/post/atomic-web-design/>>. Luettu 22.11.2018.
- 11 Elisa Style Book. Verkkoaineisto. <<http://stylebook.elisa.fi/#!/>>. Katsottu 1.10.2018.
- 12 Digital Style Telia Company. Verkkoaineisto. <<http://digitalstyle.teliacompany.com/>>. Katsottu 1.10.2018.

- 13 Polaris Design System. Verkkoaineisto. <<https://polaris.shopify.com/>>. Versio 2.12.1. Katsottu 3.10.2018.
- 14 IBM Design Language. Verkkoaineisto. <<https://www.ibm.com/design/language/>>. Katsottu 3.10.2018.
- 15 Niemistö, Tero. 2016. Tekninen velka - yrityksesi suurin digitalisaation jarru? Verkkoaineisto. <<https://blog.kauppalehti.fi/digiarjessa/tekninen-velka-yrityksesi-suurin-digitalisaation-jarru>>. 27.12.2016. Luettu 24.9.2018.
- 16 Tekninen velka – tunnusta, tunnista ja rajoita. 2017. Verkkoaineisto. <<http://www.sytyke.org/tapetilla/tekninen-velka-tunnusta-tunnista-ja-rajoita/>>. 12.2.2017. Luettu 24.9.2018.
- 17 Suarez, Marco; Anne Jina; Saylor-Miller Katie; Mounter Diana & Stanfield Roy. Design Systems Handbook. Verkkoaineisto. <<https://www.designbetter.co/design-systems-handbook>>. Luettu 24.9.2018.
- 18 Angular Material – Material Design component for Angular. Verkkoaineisto. <<https://material.angular.io/>>. Luettu 24.6.2018.
- 19 Myhill, Chris. How detailed should wireframes be? A guide to wireframe fidelity. 2017. Verkkoaineisto. <<https://justuxdesign.com/blog/wireframe-fidelity>>. Luettu 10.11.2018.
- 20 Saavutettavuus. Verkkoaineisto. <<https://vm.fi/saavutettavuusdirektiivi>>. Luettu 8.11.2018.
- 21 Käyttötuotteen heuristinen arviointi. Verkkoaineisto. <http://mlab.uiah.fi/polut/Design/tyokalu_heuristinen_arvio.html>. Luettu 8.11.2018.