

Design system ja komponentit

Laura Pekkanen

Opinnäytetyö
Joulukuu 2020
Tietojenkäsittely ja tietoliikenne
Insinööri (AMK), tieto- ja viestintätekniikka

Tekijä(t) Pekkanen, Laura	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Joulukuu 2020
	Sivumäärä 30	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Design system ja komponentit		
Tutkinto-ohjelma Tieto- ja viestintäteknikka		
Työn ohjaaja(t) Kari Niemi		
Toimeksiantaja(t) Into-Digital Oy		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli perehtyä erilaisiin design systeemeihin ja tutkia voisiko näitä hyödyntää toimeksiantajan WordPress -teemapohjan kehityksessä. Design systeemin keskiössä ovat uudelleen käytettävät komponentit, ja tavoitteena oli myös suunnitella ja toteuttaa teemapohjaan muutama komponentteja.</p> <p>Työ aloitettiin perehtymällä design systeemeihin ja miten niitä voisi opinnäytetyössä hyödyntää. Esille nousivat komponentit ja varsinkin atomic designin komponenttien eri tasot toivat uutta näkökulmaa komponenttien jaotteluun. Tämän jälkeen valittiin kolme design systeemiä vertailtavaksi teemapohjan jatkokehitystä varten. Valitut design systeemit olivat Material Design, GOV.Uk Design System ja Bootstrap. Näistä päädyttiin hyödyntämään jo aiemmin käytössä ollutta Bootstrapia.</p> <p>Teoriaosuudesta edettiin käytännön toteutukseen, jossa kartoitettiin verkkosivustojen yleisimpiä komponentteja. Näistä päädyttiin toteuttamaan komponentti kuva ja tekstialue. Komponenteille suunniteltiin kansiorakenne, jonka mukaan komponentille luotiin tiedostot. Tämän jälkeen tunnistettiin komponentin kentät ja näiden mukaan tehtiin rautalankamalli, joiden mukaan komponenttia alettiin toteuttamaan. Komponentti totutettiin WordPress demosivustolle Advanced Custom Fields lisäosan avulla ja sille koodattiin PHP-rakenne.</p> <p>Opinnäytetyön tuloksena komponenteille toteutettiin rakenne ja ohje komponenttien koostamiseen. Tuloksena saatiin myös demo, jonne luotiin yksi komponentti.</p> <p>Komponenttien lisäämiseksi teemaan voisi jatkokehittää oman työkalun tai pluginin.</p>		
Avainsanat (asiasanat) Design system, komponentit, WordPress		
Muut tiedot (Salassa pidettävät liitteet)		

Author(s) Pekkanen, Laura	Type of publication Bachelor's thesis	Date December 2020 Language of publication: Finnish
	Number of pages 30	Permission for web publication: x
Title of publication Design system and components Possible subtitle		
Degree programme Information Technology and Communication Technology		
Supervisor(s) Niemi, Kari		
Assigned by Into-Digital Oy		
Abstract <p>The purpose of this bachelor's thesis was to get familiar with design system and to investigate whether these could be utilized in the development of the client's WordPress theme template. Another goal was to identify the most common components used on websites and implement a couple of them.</p> <p>The work began with an introduction to what a design system is and how it could be utilized. Components became the focus in the research and the different levels that Atomic design defined for the components. Three design systems were chosen for comparison in theme template development. The selected design systems were Material Design, GOV.Uk Design System and Bootstrap. Bootstrap got chosen to be further used because of the familiarity.</p> <p>Practical implementation started after the theoretical research. Most common components of web pages were mapped and out of these one component was implemented. A folder structure was designed for the components, according to which files were created. The fields of the component were then identified, and a wireframe was created. The component was constructed on the WordPress demo site using the Advanced Custom Fields plugin.</p> <p>As a result of the thesis, the structure of the components was implemented and how the components should be assembled. The result was also a demo site where one component was made and where you can add components in the future.</p> <p>The client could further develop their own tool or plugin to add components to the theme.</p>		
Keywords/tags (subjects) Design system, components, WordPress		
Miscellaneous (Confidential information)		

Sisältö

Sanasto	3
1 Työn lähtökohdat	4
1.1 Toimeksiantaja	4
1.2 Taustaa	4
1.3 Tehtävä ja tavoitteet	5
2 Design System	5
2.1 Yleistä	5
2.2 Komponentit.....	8
2.3 Atomic design	9
2.4 Erilaisia Design Systemejä	10
2.4.1 Material Design.....	11
2.4.2 GOV.UK Design System.....	12
2.4.3 Bootstrap	12
2.4.4 Design systeemien vertailu.....	13
3 Työn toteutus	15
3.1 Taustaa	15
3.2 Teemapohjan rakenne	16
3.3 Komponentit.....	19
3.3.1 Komponenttien kartoittaminen	19
3.3.2 Komponentin suunnittelu.....	19
3.3.3 Komponentin ACF-asetukset	20
3.3.4 Komponentin PHP-rakenne	24
4 Tulokset ja johtopäätökset	27
4.1 Lopputulos.....	27
4.2 Jatkokehitys	27
4.3 Pohdinta	28
Lähteet	29

Kuviot

Kuvio 1. HubSpotin auditoinnissa löytyneet painikkeet (Muscato 2018).....	6
Kuvio 2. United States Web Design System painikkeiden määrittely (United States Web Design System: Components Button N.d.)	8
Kuvio 3. Atomic Design (Moreno Celta 2019)	10
Kuvio 4. WordPressin Twenty Twenty -teeman rakenne.....	17
Kuvio 5. Components kansion rakenne.....	18
Kuvio 6. Komponentin tyyli- ja JavaScript -tiedostojen linkityskansiot.	18
Kuvio 7. Rautalankamalli kuva ja tekstialue elementistä.....	20
Kuvio 8. Kuva ja tekstialue -elementin ACF kenttien asetukset.....	21
Kuvio 9. Komponentin lisääminen sivulle	22
Kuvio 10. Kuva ja tekstialue -elementti sivun hallinnassa	24
Kuvio 11. Komponentin muuttujien määrittely	25
Kuvio 12. Komponentin HTML-rakenne.....	26
Kuvio 13. Kuva ja tekstialue -elementti sivulla.....	26

Taulukot

Taulukko 1. Design systeemien vertailu.....	14
---	----

Sanasto

ACF	Advanced Custom Field. WordPressin lisäosa, jolla voidaan luoda omia elementtejä.
Bootstrap	CSS framework eli kirjasto joka sisältää CSS tyylejä.
CDN	Content Delivery Network eli sisällönjakeluverkko, joka on hajautettu ja koostuu välityspalvelimista. Tarkoitus on palvelujen tarjoaminen loppukäyttäjille mahdollisimman pienellä viiveellä.
CSS	Cascading Style Sheets. Tyylitiedosto, jolla kuvataan HTML tiedoston tyylejä.
HTML	Hyper Text Markup Language. Kuvauskieli, jolla rakennetaan nettisivuja.
Node.js	Avoimen lähdekoodin JavaScript runtime-ympäristö JavaScript-koodin suorittamiseen palvelimella.
NPM	Node Package Manager on avoimen lähdekoodin paketinhallinta työkalu Node.js:lle.
PHP	Palvelinohjelmointikieli, jolla pystyy tulostamaan HTML tiedostoja.
React	JavaScript kirjasto, jolla rakennetaan käyttöliittymiä.
Responsiivisuus	Vekkosivujen skaalautuvuus eri laitteille.
Sass	Syntactically Awesome Style Sheets on tyylitiedosto, joka täytyy koostaa CSS-tiedostoksi, jotta selain voi lukea sitä.
WordPress	Verkkosivujen sisällönhallintajärjestelmä

1 Työn lähtökohdat

1.1 Toimeksiantaja

Opinnäytetyön toimeksiantajana toimi jyvaskyläläinen digitaalisen markkinoinnin ja palveluiden tuottaja Into-Digital. Yritys on perustettu vuonna 2002 ja työllistää noin 15 henkilöä Helsingissä ja Jyväskylässä. Helsingissä sijaitsee yrityksen projektinjohto ja hallinto ja projektien tekninen toteutus on Jyväskylässä. Into-Digital on Ida Fram Group -konsernin tytäryhtiö.

Into-Digital toteuttaa verkkosivuja pääasiassa WordPress sisällönhallintajärjestelmällä, mutta myös esimerkiksi Drupalilla. Yrityksen osaamiseen kuuluu myös mobiilisovellukset ja display-mainokset.

1.2 Taustaa

Toimeksiantajalle tuli tarve yhtenäistää ja kehittää toimintatapojaan. Verkkosivujen kehittämisessä ei ollut selkeitä ohjeistuksia tai yrityksen sisäisiä standardeja ja informaatio ei kulkenut kehittäjien välillä. Verkkosivuilla samankaltaiset elementit saattoivat toimia hieman eri tavalla tai olla visuaalisesti erilaisia riippuen kehittäjästä.

Ratkaisuksi toimeksiantaja halusi perehtyä design systeemeihin ja tutkia olisiko näistä hyötyä olemassa olevan WordPress-teemapohjan kehityksessä. Design system sisältää uudelleen käytettäviä komponentteja ja niiden käyttö on tarkkaan määritelty (Anne, Mounter, Stanfield, Suarez & Saylor-Miller N.d.). Tärkeää oli määritellä komponentit ja koota ne komponenttikirjastoon, josta jokainen kehittäjä voi ottaa ne tarpeensa mukaan käyttöön. Myös uusien komponenttien lisääminen ja kirjaston päivittäminen oli tärkeää.

1.3 Tehtävä ja tavoitteet

Opinnäyte toteutettiin toiminnallisena kehittämistyönä. Sen tavoitteena oli perehtyä design systeemiin ja kartoittaa olisiko joku valmiista design systeemeistä toimiva ratkaisu toimeksiantajan teemapohjan kehittämisessä vai pitäisikö toimeksiantajan kehittää itse oma design system.

Design systeemien tutkimisen jälkeen tarkoituksena oli kartoittaa ja määritellä toimeksiantajan tarvitsemat komponentit ja koostaa niistä muutama valmiiksi. Komponenteille oli myös tarkoitus tehdä demosivu WordPressin päälle, jossa komponentteja voitaisiin demota. Tärkeässä osassa demosivustolla oli myös komponenttien dokumentaatio.

2 Design System

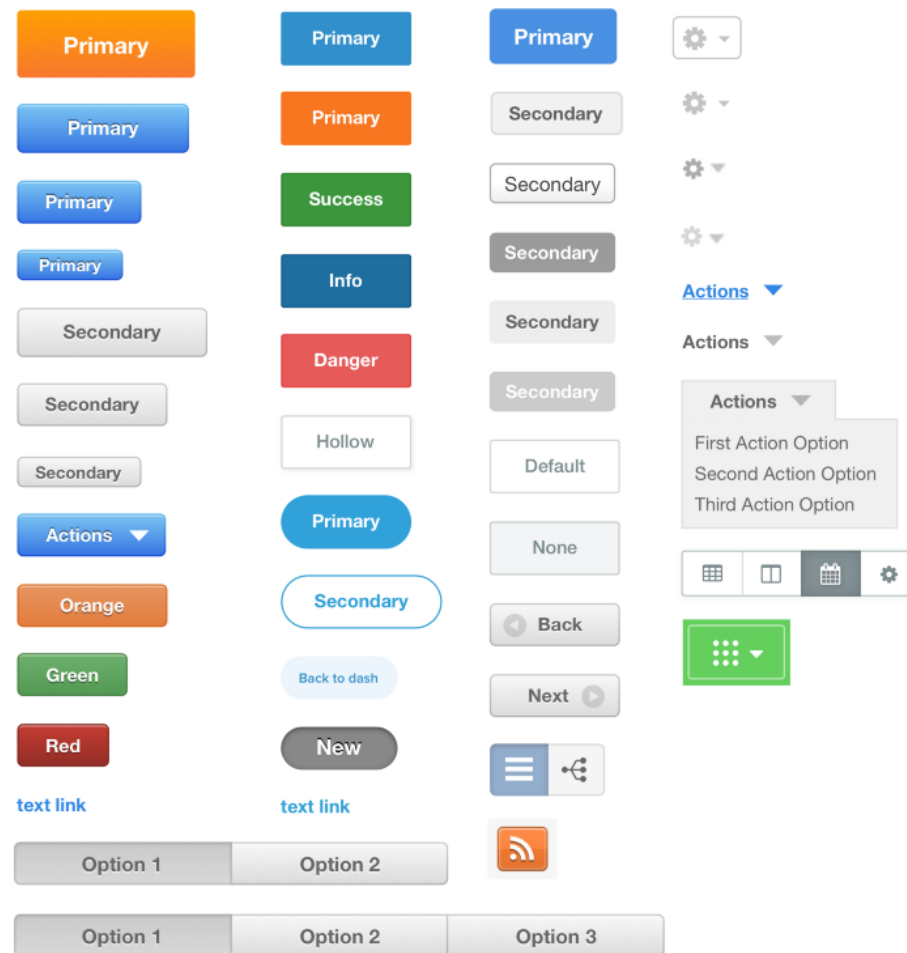
2.1 Yleistä

Web- ja ohjelmistokehityksessä projektien skaalautuessa yhä suuremmiksi, kasvaa yrityksen tarve yhtenäistää ja tehostaa toimintaansa. Usean henkilön tehdessä samaa projektia, saattavat sivuston samankaltaiset osiot näyttää tai toimia eri tavoin kommunikaation ja yhteisten sääntöjen puuttuessa. (Araújo 2018)

Hyvä esimerkki design systeemin kannattavuudesta on markkinoinnin automaatiojärjestelmä HubSpotilla. Heillä huomattiin, että heidän omat tuotteensa sisälsivät epä johdonmukaisuuksia muun muassa painikkeiden sijainnissa ja välilehtien ulkoasussa. HubSpot aloitti tuotteidensa kehittämisen kysymällä asiakkailtaan palautetta. Palautekyselyssä selvisi selkeä tarve yhtenäistää ja selkeyttää tuotteiden ulkoasua.

HubSpot aloitti prosessin auditoimalla kaikki aiemmin luodut komponenttinsa 10 vuoden ajalta ja laati näistä raportin. Raportissa kävi ilmi, että HubSpotilla oli käytössään muun muassa yli 100 erilaista harmaan sävyä, 6 erilaista pääpainiketta ja paljon

muita epäjohdonmukaisuuksia projekteistaan. Kuviossa 1 on visualisoitu kaikki painikkeet, jotka olivat HubSpotin käytössä. Tämän kaiken johdosta luotiin Canvas, HubSpotin oma design systeemi. (Muscato 2018)



Kuvio 1. HubSpotin auditoinnissa löytyneet painikkeet (Muscato 2018)

Design system tähtää ratkaisemaan ulkoasun epäjohdonmukaisuudet ja koodin turhan uudelleen kirjoittamisen tehostamalla, yhtenäistämällä ja skaalaamalla yrityksen toimintaa. Design system sisältää uudelleen käytettäviä komponentteja, joiden käyttö on tarkkaan määritetty. Kehittäjien ei tarvitse joka kerta rakentaa samankaltaisia elementtejä alusta asti, vaan he voivat käyttää näitä valmiita komponentteja, joita voi räätälöidä omaan tarkoitukseensa. Kehittämisestä tulee tehokkaampaa ja jälki pysyy yhtenäisenä, kun komponenttien toimintatapa on aina sama. (Araújo 2018)

Design system on siis laajempi kuin perinteiset brändioppaat tai graafiset ohjeistukset. Visuaalisten elementtien lisäksi se sisältää ohjeistukset muun muassa komponenttien, esimerkiksi lomakkeiden ja painikkeiden, toiminnasta ja responsiivisuudesta eri laitteille. Design systeemistä hyödytään vielä enemmän, jos yritys kehittää useampia digitaalisia palveluja, jolloin komponentteja voi käyttää uudelleen, eikä niitä tarvitse suunnitella ja toteuttaa joka kerta uudestaan. (Leppänen 2019)

Design system itsessään ei takaa sen toimivuutta. Design systeemin tulee olla hyvin dokumentoitu, jotta sitä osattaisiin käyttää oikein. Henkilöstöä tulee myös kouluttaa design systeemin käyttöön. (Mounter 2017)

Monella suurella yrityksellä on oma design systeeminsä. Esimerkiksi Audi on kirjannut design systeemiinsä visuaalisten ohjeiden lisäksi brändin ohjeistuksia (Audi Design System N.d.). USA:n julkinen hallinto on myös laatinut itselleen design systeemin. Tämä sisältää ohjeiston verkkosivujen rakentamiseen visuaaliset ja toiminnalliset ohjeet. Design system on tarkoitettu kaikille USA:n julkisen hallinnon virastoille ohjeistoksi verkkosivujen tekemiseen (U. S. Web Design System (USWDS) N.d.).

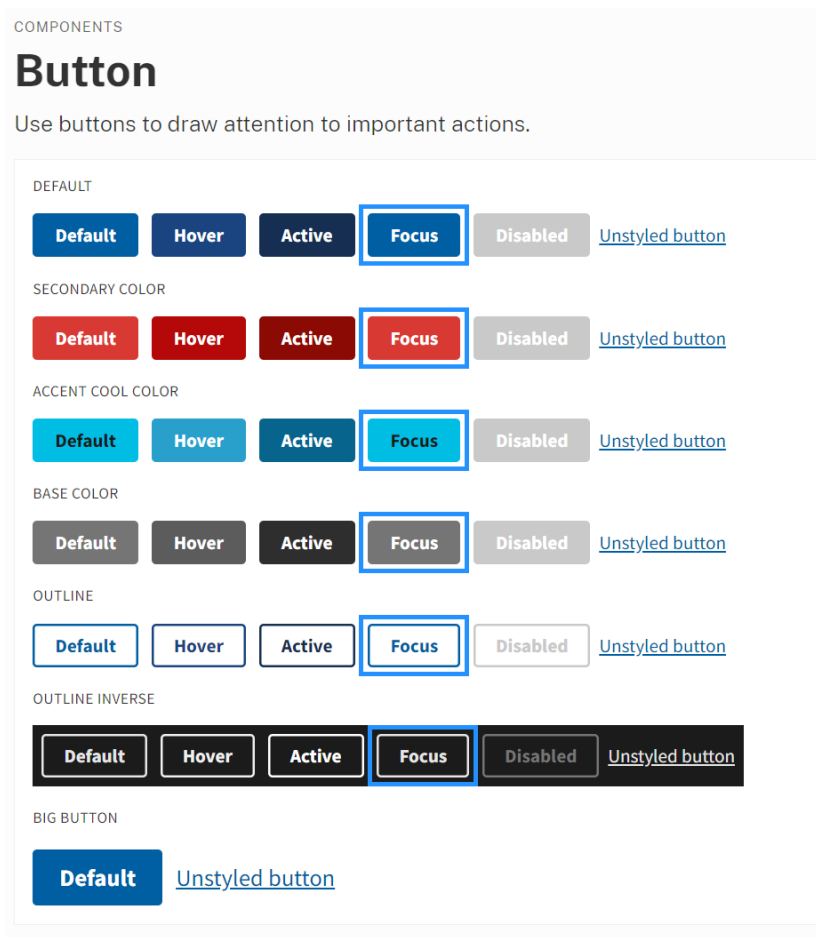
Google on myös kehittänyt oman design systeeminsä nimeltä Material Design. Tämä on avoimen lähdekoodin design system web ja Android projekteille. Google on perehtynyt käyttöliittymiin tieteen ja teknologian avulla. (Material Design N.d.)

Design systemit voivatkin olla hyvin erilaisia niiden laatijasta ja tarpeesta riippuen, kuten yllä olevista esimerkeistä huomataan. Moni yritys onkin jakanut design systeeminsä netissä, ja ne ovat kehittäjien vapaassa käytössä.

Design systeemit ovat kohdanneet myös kritiikkiä. Suunnittelijan näkökulmasta voi helposti tuntua siltä, että samoja komponentteja käyttäessä kaikki alkaa näyttämään samalta. Design systeemin tarkoitus ei olekaan olla valmis staattinen paketti, vaan ennemminkin ekosysteemi, jonka pohjalta tuotteita kehitetään. (Leppänen 2019)

2.2 Komponentit

Design systeemissä tärkeässä roolissa ovat komponentit. Ne ovat uudelleenkäytettäviä valmiita palikoita, joiden käyttö, rakenne ja tyylit ovat ennalta määriteltäviä. Esimerkiksi USA:n julkinen hallinto on tarkkaan määritellyt komponenttinsa ulkonäön, käytön ja toiminnan. Kuviossa 2 on nähtävissä, miten USWDS (The United States Web Design System) määrittelee painikkeiden visuaalisen ilmeen. Kuvioista voi huomata painikkeilla olevan muutama eri väri vaihtoehto ja jokaisen eri toiminnallisen vaiheen ulkoasu. Näitä ovat muun muassa, kun hiiri vie painikkeen päälle, painiketta painetaan ja kun painike on kohdistettuna. (U. S. Web Design System (USWDS) N.d.)



Kuvio 2. United States Web Design System painikkeiden määrittely (United States Web Design System: Components Button N.d.)

Sivulta löytyy myös ohjeet painikkeiden käyttöön, kuten missä tilanteissa painikkeita kannattaa käyttää ja mitä painike voi sisältää. Myös komponenttien koodit, HTML tagit ja luokkamääreet, löytyvät sivulta. Komponentit voivat olla myös isompia kokonaisuuksia, jotka sisältävät pienempiä palikoita. Esimerkiksi footer on myös komponentti ja se voi sisältää muita komponentteja, kuten painikkeita, lomakkeita, linkkejä ja niin edelleen.

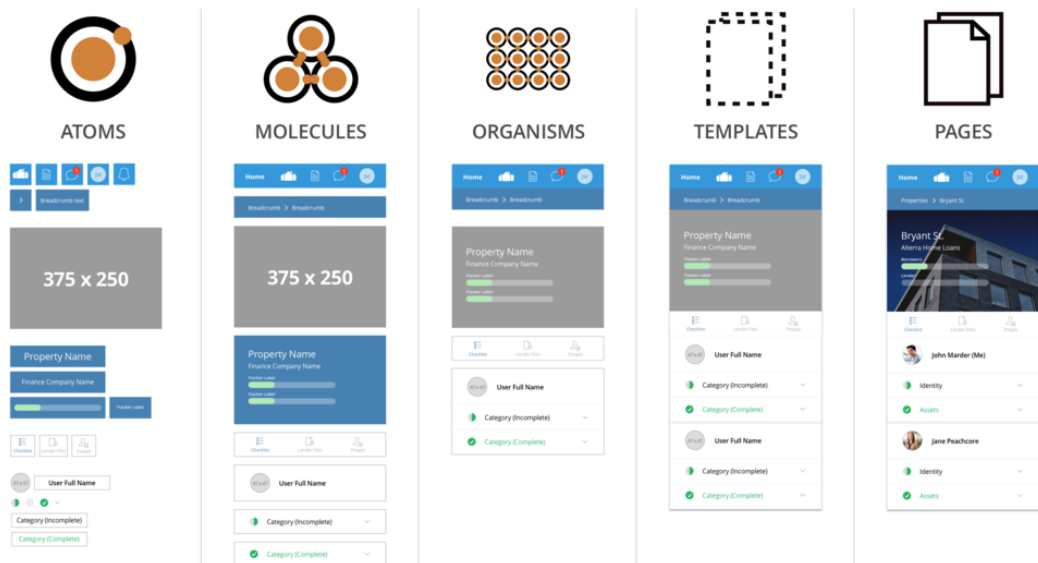
2.3 Atomic design

Atomic design on metodiikka, joka on suunniteltu design systeemien kehittämiseen. Atomic designissa komponentit jaetaan viiteen eri tasoon. Kuviossa 3 on havainnollistettu tasojen sisällöt ja rakenne. (Frost 2013)

Ensimmäinen taso ovat Atomit. Atomit ovat sivuston pienimpiä palasia ja niitä ei voi enää pilkkoa pienemmäksi. Yksinkertaisuudessaan ne ovat yksittäisiä HTML tageja. Esimerkkejä Atomeista ovat painikkeet, kuvat tai ikonit. Molekyylit ovat Atomeista koostuvia isompia kokonaisuuksia, esimerkiksi painikkeista ja ikoneista koostuva navigaatio tai lomake, jossa on kenttiä, otsakkeita ja painikkeita.

Kolmannella tasolla on ovat Organismit. Nämä ovat edellisiä isompia kokonaisuuksia ja koostuvat molekyyleistä. Organismit laajentavat molekyylin rakennetta yhdistelemällä niitä vielä isommiksi kokonaisuuksiksi, jotka ovat käyttöliittymän selkeitä osia. Esimerkki organismista on header, jossa voi olla esimerkiksi navigaatio, hakupalkki ja murupolku. Organismit taas koostavat yhdessä seuraavan tason kokonaisuuksia Templaatteja. Templaattit ovat sivupohjia, rautalankoja sivuista. Nämä hahmottavat asiakkaalle miltä sivu näyttää ja hahmottavat sivun kokonaisuutta. Viimeisellä tasolla on itse sivu. Sivun laajentaa Templaattia sisältäen esimerkisisältöä. (Frost 2013)

Atomic Design for Streamloan



Kuvio 3. Atomic Design (Moreno Celta 2019)

Atomic designin tarkoituksena on lähteä rakentamaan elementtejä pienemmästä suurempaan. Näin säilytetään projektien yhtenäisyys ja skaalautuvuus. Yhdistelemällä pienempiä kokonaisuuksia suuremmiksi, sivustojen kehittäminen nopeutuu. (Frost 2013)

2.4 Erilaisia Design Systemejä

Internet on pullollaan erilaisia design systeemejä. Opinnäytetyön kannalta onkin tärkeää tarkastella erilaisia design systeemejä, jotka sopisivat toimeksiantajan käyttötarkoitukseen mahdollisimman hyvin. Design systeemin tulee olla avointa lähdekoodia ja sitä pitää pystyä käyttämään kaupallisissa projekteissa. Myös käyttöönoton ja asentamisen olisi hyvä olla helppoa. Toimeksiantajalla on käytössään oma WordPress teemapohja ja design systeemin tulee myös olla yhteensopiva tämän kanssa.

Vertailtaviksi design syteemeiksi valittiin Material Design, GOV.UK Design System ja Bootstrap. Juuri nämä design systeemit valittiin, siksi että ne täyttävät lyhyen tutkimisen jälkeen yllä mainitut kriteerit ja jokainen design system on hyvin erilainen. Niiden käyttökohteet ja -tarkoitukset ovat erilaiset.

2.4.1 Material Design

Aiemmin jo mainittu Material Design on varmaankin yksi tunnetuimmista design syteemeistä. Se on Googlen 2014 kehittämä design systeemi, joka pohjautuu avoimeen lähdekoodiin. Design systeemiin on otettu inspiraatiota fyysisestä maailmasta. Sen design periaatteet pohjautuvat oikean maailman tekstuureihin, valoon ja varjoihin. (Material Design N.d.)

Monelle Material Designin komponentit ovat tulleet tutuiksi Android laitteista ja Googlen omista tuotteista. Ne ovat helposti tunnistettavia ja yhdistettävissä Googleen. Material Design sisältää ohjesäännöt lähes kaikkeen: typografiaan, gridiin, välityksiin, skaalautuvuuteen, komponentteihin, väreihin ja kuvastoon. Kaiken visuaalisen lisäksi, Material Design ohjeistaa suunnittelijoita luomaan harkittuja elementtejä, joilla on hierarkia ja tarkoitus. (Chapman C. N.d.)

Material Designilla voi kehittää Android, iOS ja Flutter sovelluksia mobiiliin tai tabletille. Verkkosivujen kehittäminen onnistuu myös JavaScript, React tai CSS elementeillä. Myös muita frameworkoja pystyy integroimaan Material Designin kanssa. Verkkosivukehitykseen on myös luotu Material Design Lite, joka on kevyempi ja ei vaadi välttämättä ollenkaan JavaScriptin käyttöä. (Material Design N.d.)

Material Designin ehdottomasti parhaita puolia ovat hyvä dokumentaatio ja Googlen tarjoama tuki. Google tarjoaa myös paljon muita työkaluja Material Designin ohessa, kuten design paketteja eri designalustoille, svg ikonikirjaston, Google Fontit ja muun muassa erilaisia värityökaluja (Material Design N.d.). Design systeemiä on myös helppo teemoittaa omaan käyttöönsä. Vaikka Material Designista onkin moneksi, on sen ensisijainen käyttötarkoitus Android laitteissa. Komponenttien rakenteeseen ja

toimintaan on vaikeampaa päästä käsiksi, koska näin ei ole tarkoituskaan. Siispä elementit tulevat kuitenkin näyttämään saman kaltaisilta, vaikka niiden tyylejä pyrkisi muokkaamaan erilaiseksi. (Chapman C. N.d.)

2.4.2 GOV.UK Design System

GOV.UK Design System on Iso-Britannian hallituksen laatima design systeemi, joka on vapaassa käytössä kaikille kehittäjille. Se on kuitenkin suunnattu nimensä mukaisesti Iso-Britannian julkisen hallinnon palveluiden toteuttamiseen. Kuka tahansa voi myös osallistua design systeemin kehitykseen ehdottamalla uusia komponentteja tai jopa itse kehittämällä niitä. Design tiimin on kuitenkin hyväksyttävä muutokset ennen kuin ne otetaan käyttöön. (Noakes & Hupe 2018; GOV.UK Design System N.d.)

Design system on jaoteltu komponentteihin, kuten painikkeisiin, taulukoihin ja varoitusteksteihin, ja tyyleihin, jotka sisältävät muun muassa fontit, värit ja typografian. Komponentit ovat joko HTML pohjaisia tai Nunjucks templaatteja. Nunjucks on JavaScript pohjainen mallinnuskieli, jolla voi luoda lohkotettuja komponentteja (Nunjucks N.d.). (GOV.UK Design System N.d.)

Koska GOV.UK Design System on tarkoitettu julkisen hallinnon käyttöön, on sillä myös tiettyjä rajoitteita ja vaatimuksia. Julkisen sektorin verkkopalveluilta vaaditaan saavutettavuutta Euroopan parlamentin ja neuvoston direktiivin, (EU) 2016/2102, mukaan (Euroopan parlamentin ja neuvoston direktiivi (EU) 2016/2102). Saavutettavuus tarkoittaa sitä, että mahdollisimman moni erilainen ihminen pystyy käyttämän palvelua mahdollisimman helposti (Tietoa saavutettavuudesta N.d.). GOV.UK Design System on saavutettava Web Content Accessibility Guidelines 2.1 -ohjeistuksen AA-tasolla (GOV.UK Design System N.d.).

2.4.3 Bootstrap

Bootstrap on vuonna 2010 alun perin Twitterille kehitetty ns. "Twitter Blueprint". Projekti kuitenkin laajeni, kun Twitter järjesti Hack Weekin, joka houkutteli monen

tasoisia kehittäjiä osallistumaan aikaisen Bootstrapin kehitykseen. Tämän innoittamana Bootstrap julkaistiin avoimena kaikille vuonna 2011. (About N.d.)

Bootstrap on maailman suosituin verkkosivujen kehitykseen tarkoitettu CSS framework, joka on suunnattu responsiiviseen mobiili ensin-kehitykseen. Se sisältää muun muassa responsiivisen gridin, Sass-muuttujia ja mixineitä, valmiita komponentteja ja ikonikirjaston. (Bootstrap (front-end framework) 2020; Bootstrap. N.d.)

Bootstrap ei varsinaisesti itsessään ole design systeemi, vaan enemmänkin rautalanka tai sivuston runko, vaikka se sisältääkin tyylejä ja samankaltaisia elementtejä kuin design systeemit. Siinä missä muut design systeemit ovat kehitetty tiettyä sovelusta tai yritystä varten, on Bootstrap kehitetty muille kehittäjille. Tämä jättää kehittäjille paljon vapauksia, joka käyttökohteen mukaan saattaa olla hyvä tai huono asia. (Barnard 2020; Coyier 2019)

2.4.4 Design systeemien vertailu

Taulukossa 1 on vertailtu aiemmin esiteltyä kolmea design systeemiä. Design systeemeissä on yhtäläisyyksiä, mutta myös eroavaisuuksia. Material Design on vertailtavista design systeemeistä laajin. Siinä missä muut ovat vain web käyttöön, Material Designia voi käyttää myös Android ja iOS laitteilla. Jokainen kolmesta design systeemistä käyttää MIT lisenssiä. MIT lisenssi tarkoittaa sitä, että design system on vapaassa käytössä myös kaupallisesti ja sen osia saa muokata ja kopioida haluamallaan tavalla (MIT-lisenssi 2020).

Kaikki design systeemit sisältävät erilaisia komponentteja. Taulukossa 1 on lueteltu missä muodossa komponentit ovat. Jokainen design system sisältää webiin tarkoitettuja HTML-komponentteja. Material Designissa on HTML-komponenttien lisäksi React-komponentteja ja myös muita ei webiin tarkoitettuja komponentteja. GOV.UK Design System sisältää myös Nunjucks-komponentteja. Bootstrapin komponentit ovat lähinnä HTML-komponentteja. Osa tarvitsee toimiakseen JavaScriptiä, mutta erilaisia kirjastoja ei tarvitse asentaa.

Jokainen design system sisältää myös tyylejä, kuten värit, typografian, layoutin ja gridin. Tärkein näistä ulkopuolisen kehittäjän näkökulmasta on varmasti grid eli sivuston asettelu tai palstoitus. Design systeemeissä tämä on tehty valmiiksi responsiiviseksi, eli kehittäjän ei tarvitse erikseen miettiä palstoitusta eri näyttökoille. Toimeksiantaja on aiemmin jo hyödyntänyt Bootstrapin gridiä teemapohjassaan, joten sen käyttö on tuttua jo ennestään.

Jokaisen design systeemin pystyy asentamaan npm:llä eli node package managerilla. Bootstrapin pystyy asentamaan monella muullakin paketinhallintatyökalulla, nämä on lueteltu taulukossa 1. Material Designin ja Bootstrapin saa käyttöönsä myös CDN:n avulla eli linkittämällä tiedoston sisällönjakeluverkosta projektiinsa. GOV.Uk Design Systeemissä ei ole tätä mahdollisuutta, mutta tarvittavat tiedostot voi ladata ja lisätä omaan projektiinsa. Tämä vaihtoehto on kuitenkin suppeampi kuin asentaminen paketinhallinnalla.

Taulukko 1. Design systeemien vertailu

Design System	Material Design	GOV.UK Design System	Bootstrap
Kehittäjä	Google	GOV.UK Design System team	Bootstrap Core Team
Julkaistu	2014	2018	2011
Viimeisin versio	Web: 7.0.0	3.8.1	4.5.2
Käyttökohteet	Android, iOS, Web	Web	Web
Lisenssi	MIT lisenssi	MIT lisenssi	MIT lisenssi

Ominaisuudet	<ul style="list-style-type: none"> • Android, iOS, Web (HTML ja React) ja Flutter komponentit • tyylit • äänet • animaatiot 	<ul style="list-style-type: none"> • HTML ja Nunjucks komponentit • tyylit • saavutettava (WCAG 2.1 level AA) 	<ul style="list-style-type: none"> • HTML komponentit • tyylit • ikonit
Asennus	npm, CDN	npm, lataa ja linkitä (suppeampi)	CDN ja paketinhallinta työkalut: npm, yarn, Rubygems, Composer ja NuGet
Samankaltaisia	<ul style="list-style-type: none"> • Carbon Design System • Fluent Design System 	<ul style="list-style-type: none"> • U. S. Web Design System • Australian Government Design System 	<ul style="list-style-type: none"> • Foundation • Pure.css
Plussat	<ul style="list-style-type: none"> • laaja tuki • dokumentaatio • käyttökohteiden monipuolisuus • useita design työkaluja 	<ul style="list-style-type: none"> • saavutettava • mahdollisuus osallistua kehitykseen 	<ul style="list-style-type: none"> • laaja tuki • muokattavuus
Miinukset	<ul style="list-style-type: none"> • helposti tunnistettava • kehitetty Googlelle 	<ul style="list-style-type: none"> • kehitetty Iso-Britannian julkiselle sektorille • Beta versio 	<ul style="list-style-type: none"> • vähemmän valmiista koodia

3 Työn toteutus

3.1 Taustaa

Toimeksiantajalla on käytössään itse kehittämänsä WordPress teemapohja. Teeman elementtejä muokataan aina sopivaksi projektin ja asiakkaan mukaan. Sivustot näyttävät siis hyvinkin erilaisilta ja ratkaisut ovat uniikkeja. Kuitenkin suurimmassa osassa projekteja komponenttien rakenne on hyvinkin samanlainen. Esimerkiksi monella sivustolla etusivulla on käytössä hero-elementti, joka sisältää ainakin kuvan ja yleensä

myös tekstiä. Sisältö-elementeistä yleisimpiä ovat kuva ja teksti vierekkäin ja teksti useassa palstassa.

Toimeksiantaja on koonnut omia räätälöityjä elementtejä asiakkaille Advanced Custom Fields -lisäosalla. Lisäosa mahdollistaa WordPressin hallintaan erilaisten kenttien lisäämisen. WordPress esitteli kaksi vuotta sitten versiossa 5.0 Gutenberg editorin, joka tunnetaan myös nimellä block editor (Pulkkinen 2018). Toimeksiantaja on tähän mennessä hyödyntänyt melko vähän block editoria, mutta nyt komponentit halutaan käyttöön tätä kautta.

Valmiiden design syteemien tutkimisen jälkeen huomattiin, että isot valmiit design systeemit ovat tehty hyvin paljon sen kehittäjää mielessä pitäen, eli niiden räätälöiminen ja muokkaaminen on vaikeampaa tai jopa mahdotonta. Muiden ylitse nousi selkeästi Bootstrap sen tuttuuden ja räätälöitävyyden vuoksi. Teemapohjassa on jo valmiiksi käytössä Bootstrapin grid ja muita komponentteja tai tyylejä voi tarpeen mukaan lisätä.

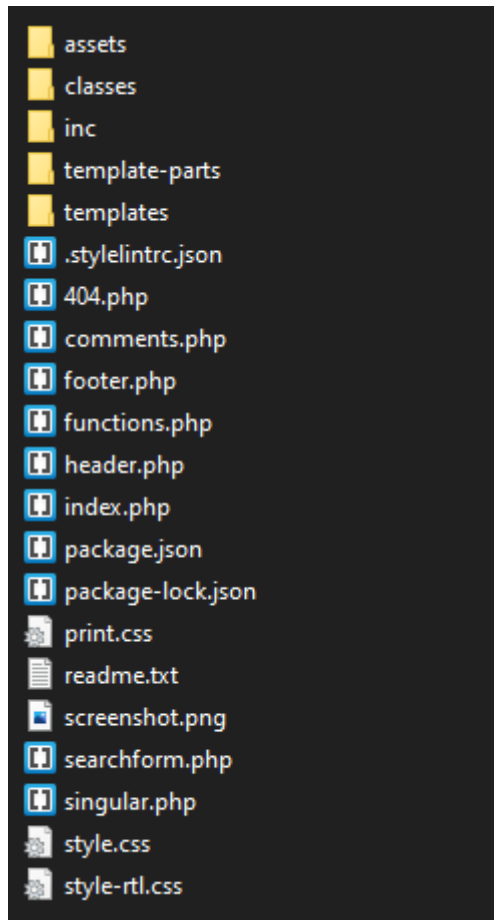
Komponenttien rungot on tarkoitus koota itse. Jokaiselle komponentille tehdään oma PHP-tiedosto. Tyylit ovat projektikohtaisia, joten näitä ei tehdä ennakoon. Kuitenkin jokaisella komponentilla on oma tyylitiedostonsa, joka vaikuttaa vain tietyn komponentin käyttäytymiseen. Jos komponentti käyttää JavaScriptiä tehdään tällekin oma tiedosto ja sisällytetään se päätiedostoon.

Demo aloitetaan WordPressin asennuksella ja toimeksiantajan teeman käyttöön-otolla. Komponentit tulee suunnitella ja niistä tehdään demoon esimerkit.

3.2 Teemapohjan rakenne

Kuviossa 4 on nähtävillä WordPressin oman teeman Twenty Twenty kansiorakenne. Teeman juuressa ovat päätemplaatti-tiedostot kuten header.php ja 404.php. Assets kansiota löytyvät kaikki tyylit, JavaScript tiedostot ja kuvat. Template-parts kansiossa ovat kaikki templaattien osatiedostot eli pienempiä palasia, jotka sisällytetään

muihin templaatteihin. Inc kansio sisältää funktioita WordPressin ydintoiminnallisuuksiin. (Organizing Theme Files N.d.)

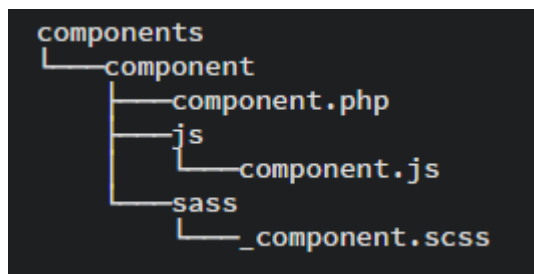


Kuvio 4. WordPressin Twenty Twenty -teeman rakenne

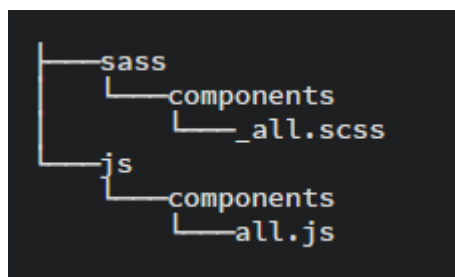
Komponenttien rakennetta ja käyttöä on tarkoitus yksinkertaistaa, joten niille luodaan teemaan oma kansio nimeltä components. Aiemmin itse luodut komponentit on luotu template-parts kansioon. Jokainen komponentti on myös oma kansionsa components kansion sisällä Komponentin kansio ja siihen liittyvät tiedostot nimetään komponentin nimellä. Tämä kansio sisältää komponentin PHP-tiedoston. Tyylit laiteetaan omaan sass kansioonsa ja myös JavaScriptille tehdään vastaavasti oma kansionsa. Kuviossa 5 on havainnollistettu komponenttien rakenne teemassa. ACF luokentilleen json tiedoston, jossa ovat kaikki kenttien tiedot kuten tyypit ja nimet. Tee-

mapohjassa näitä tiedostoja on säilytetty teeman juuresta löytyvästä acf-json kansio-
osta. Demoa rakennettaessa json tiedostot jätetään vielä tähän kansioon, mutta jat-
kossa ne voisi sisällyttää components kansion alle.

Komponentin tyyli- ja JavaScript tiedostot tulee vielä linkittää käyttöön. Komponent-
tijaattelua hyödyntäen teeman sass -kansion alle luodaan komponenteille oma com-
ponents -kansio. Kansioon luodaan tiedosto, jonne kaikkien komponenttien tyyli-
dostot linkitetään ja tämä tiedosto linkitetään päätyylitiedostoon. Myös JavaScriptille
tehdään vastaava kansiorakenne.



Kuvio 5. Components kansion rakenne



Kuvio 6. Komponentin tyyli- ja JavaScript -tiedostojen linkityskansiot.

Atomic designissa komponentti ajattelu lähti pienemmästä suurempaan ja pienim-
mät komponentit eli atomit olivat niinkin yksinkertaisia kuin yksittäisiä HTML tageja.
Teemaan koottavat komponentit pyritään pitämään organismitasolla. Teemassa voisi
kuitenkin hyödyntää atomic design ajattelua enemmänkin pilkkomalla tyylejä ja Ja-

vaScriptiä omiin tiedostoihinsa, joita voisi tarvittaessa linkittää käyttöön. Näitä tyy-
lejä voisi käyttää tietyllä HTML luokalla, jolloin esimerkiksi painikkeiden tyylejä ei tar-
vitse määritellä useaan kertaan.

3.3 Komponentit

3.3.1 Komponenttien kartoittaminen

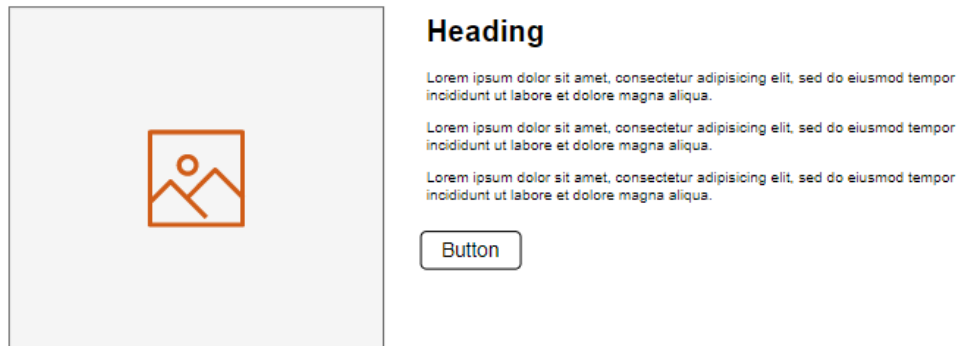
Useimmissa verkkosivustoissa käytettävät komponentit ovat rakenteeltaan hyvinkin
samanlaisia. Tavallisimmat elementit sivulla ovat header, footer ja itse sivuston si-
sältö. Header on sivustolla heti yläreunassa ja se sisältää yleensä sivuston logon, navi-
gaation, haun ja kirjautumisen. Footer on sivuston pohjalla ja sinne sijoitetaan tavalli-
sesti yrityksen yhteystiedot, linkit sosiaaliseen mediaan ja sivuston sivukartan.

Näiden lisäksi sisältöä voidaan esittää eri tavoin. Tavallisimpia sisällön esittämistä-
poja ovat otsikko ja teksti, kuva ja tekstialue, karuselli, video ja erilaiset sisältönotot.
Komponenteista päätettiin ottaa kehitykseen kuva ja tekstialue.

3.3.2 Komponentin suunnittelu

Komponenttien suunnittelu kannattaa aloittaa pilkkomalla elementti pienempiin
osiin. Kuviossa 7 on hahmoteltu rautalankamalli elementille kuva ja tekstialue. Ele-
mentit osat voidaan pilkkoa nyt ACF kentiksi. ACF kenttiä elementissä ovat kuva, ot-
sikko, teksti ja linkki. Hyvin usein myös kuvan ja tekstin puolta halutaan vaihtaa, joten
näiden lisäksi pitää tehdä kenttä, jolla kuvan puoli valitaan. Komponentin suunnitte-
lun jälkeen voidaan luoda sille tarvittavat tiedostot ja linkitykset aiemmin suunnitel-
lulla rakenteella, jotka on kuvattu kuvioissa 5 ja 6. Toimeksiantaja on jo aiemmin

tehty komponenttien rekisteröintiin tarvittavat funktiot ja kutsut, joten näitä ei tarvitse lähteä uudestaan kirjoittamaan.



Kuvio 7. Rautalankamalli kuva ja tekstialue elementistä

3.3.3 Komponentin ACF-asetukset

Kun komponentti on suunniteltu ja siinä olevat kentät on tunnistettu, voidaan alkaa tekemään ACF-kentän asetuksia. Kuviossa 8 näkyy kuva ja tekstialue elementin ACF-kenttien asetukset. Kentillä on nimiö, joka näkyy hallinnassa tietoja syöttäessä. Kentän nimen avulla päästään kenttään kiinni koodin puolella ja kentän tyyppi kertoo millaista data on. Asetusten pohjalla on kenttäryhmän sijainti, jossa määritellään missä elementtiä voi käyttää.

Elementti / Kuva- ja tekstialue

Järjestys	Nimiö	Nimi	Tyyppi
1	Kuvan puoli	image_side	Painikeryhmä
2	Otsikko	title	Teksti
3	Tekstialue	content	Wysiwyg-editori
4	Kuva	image	Kuva
5	Painike	button	Linkki

+ Lisää kenttä

Sijainti ▲

Säännöt

Tästä voit määrittää, missä muokkausnäkylässä tämä kenttäryhmä näytetään

Näytä tämä kenttäryhmä, jos

Lohko ▼

on sama kuin ▼

Kuva- ja tekstialue ▼

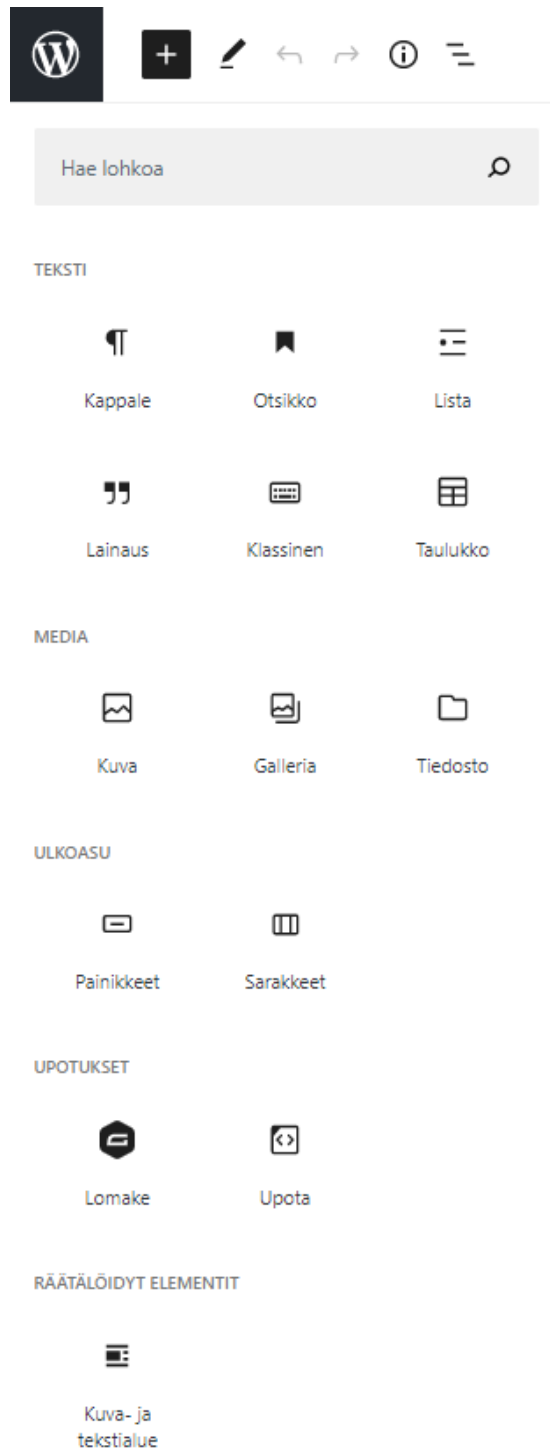
ja

tai

Lisää sääntöryhmä

Kuvio 8. Kuva ja tekstialue -elementin ACF kenttien asetukset

Elementti on helppo lisätä kuten muutkin Gutenbergin elementit sivun vasemmasta ylälaidasta plus merkistä. Kuviossa 9 on kuvattu mistä komponentin saa lisättyä. Toimeksiantaja on rekisteröinyt komponentit listautumaan omaan osioonsa nimeltä räätälöidyt elementit.

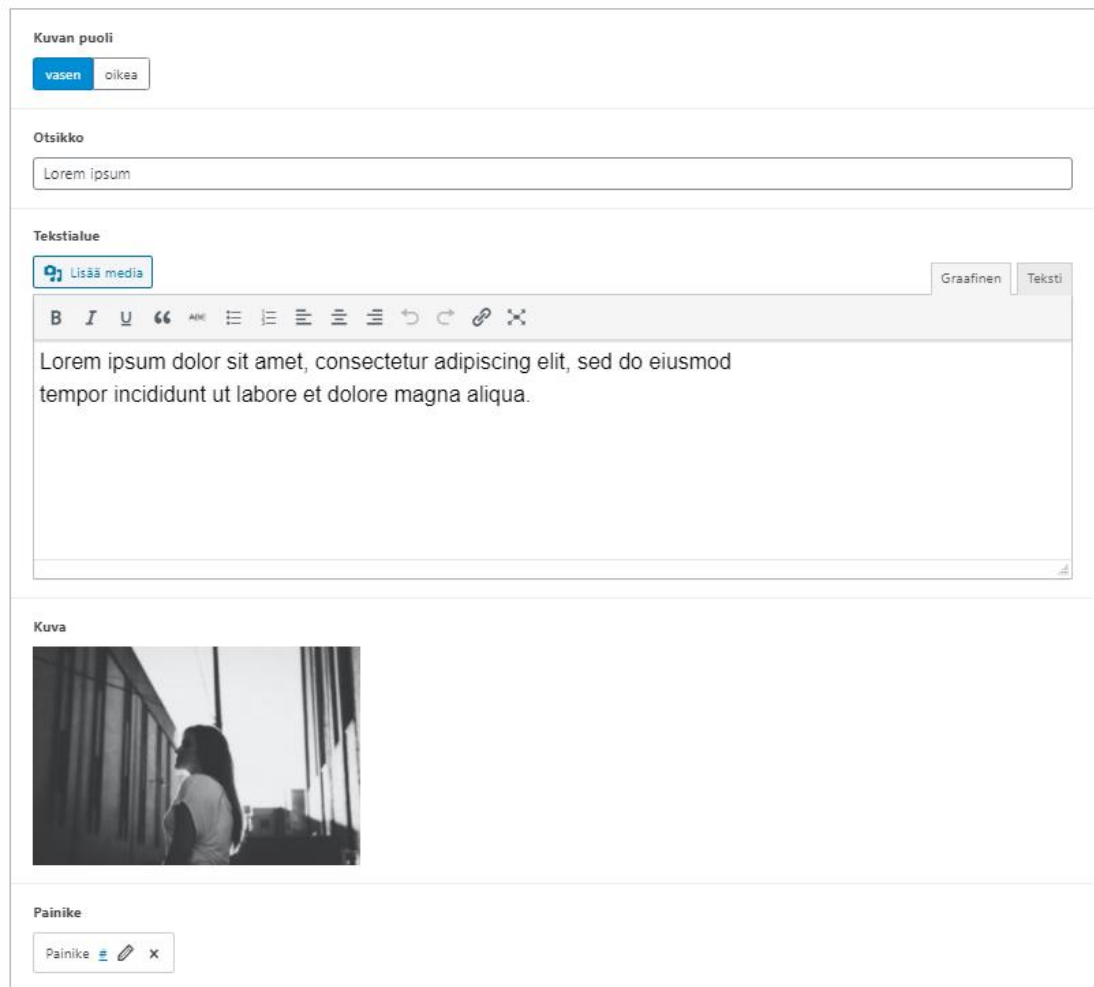


Kuvio 9. Komponentin lisääminen sivulle

Kuviossa 10 komponentti on lisätty sivun hallintaan ja siinä olevat kentät on täytetty. Tässä vaiheessa kenttien tyyppien eron näkee selkeästi. Esimerkiksi Kuvan puoli -kenttä on tyyppiltään Painikeryhmä, jolle voi itse ennalta määrittää halutut arvot.

Tässä tapauksessa arvoiksi on annettu left : vasen ja right : oikea. Kenttä palauttaa valitun vasemmanpuoleisen arvon, joka on string eli merkkijono. Kentän tyyppi on painikeryhmä ja sen arvot voi ennalta määrittää kenttäryhmän asetuksista. Myös otsikko palauttaa vain yhden arvon, joka on molemmissa muotoa string.

Tekstialue on tyypiltään Wysiwyg-editor, jolla käyttäjä pystyy muotoilemaan tekstiä haluamallaan tavallaan, kuten lihavoimaan, kursivoimaan ja tekemään listoja. Kenttä palauttaa stringin, jossa on mukana tekstin muotoilut HTML tageina. Kenttätyypit kuva ja linkki palauttavat arrayn eli taulukon. Kuva sisältää muun muassa kuvan osoitteen, nimen, vaihtoehtoisen tekstin ja kaikki saatavilla olevat kuvakoot. Linkki palauttaa vähemmän tietoja, vain linkin otsikon, osoitteen ja targetin eli mihin ikkunaan linkki avataan.



Kuvio 10. Kuva ja tekstialue -elementti sivun hallinnassa

3.3.4 Komponentin PHP-rakenne

Jotta komponentti saadaan vielä näkymän sivulla, tarvitsee sen PHP-tiedosto vielä tehdä. Aluksi pitää määrittää ACF-kenttien muuttujat. Kuviossa 11 ACF-kentät on haettu muuttujiin ACF:n omalla `get_field()`-funktioilla. Funktioon on vietävä parametrinä aiemmin hallinnassa asetettu ACF-kentän nimi. Kuvan osoite -muuttujaan (`$image_url`) haetaan kuvakooksi `large` eli isokuva tai jos tätä ei ole määritelty, niin haetaan muuttujaan alkuperäinen kuva.

```
$title = get_field('title');
$content = get_field('content');
$image = get_field('image');
$image_side = get_field('image_side');
$button = get_field('button');

$image_url = '';

if (isset($image['sizes']['large'])) {
    $image_url = $image['sizes']['large'];
} else {
    $image_url = $image['url'];
}
```

Kuvio 11. Komponentin muuttujien määrittely

Seuraavaksi voidaan rakentaa komponentin HTML-rakenne PHP-tiedostoon, johon PHP-muuttujat haetaan. Kuviossa 12 voi nähdä millainen rakenne komponentille on rakennettu. Komponentin ylimmän tason diville on annettu oma uniikki luokkansa, tässä tapauksessa image-and-text. Tälle luokalle voidaan määrittää omat tyylit, jolloin ne ovat käytössä vain tällä komponentilla. Koska käytössä on Bootstrap, komponentit koostetaan riveistä ja palstoista. Ensimmäiseen palstaan haetaan kuva. Tämä kääritään if-lauseeseen, jos kuvaa ei olekaan hallinnasta asetettu, niin mitään ei tulostu sivulle.

Toiseen palstaan tulostetaan otsikko, tekstialue ja painike. Myös nämä on kääritty if-lauseisiin, jotta tyhjiä HTML tageja ei tulostuisi. Painikkeen eri kentät on haettu taulukon indekseillä. Kuvan puolta vaihtaessa voidaan käyttää Bootstrapin order-luokkia. Tällä voidaan muokata HTML tagien järjestystä visuaalisesti. Tämän jälkeen komponentille tulisi määritellä tyylit, mutta komponentin tullessa vain demo käyttöön, riittää pelkän HTML-rakenteen tekeminen.

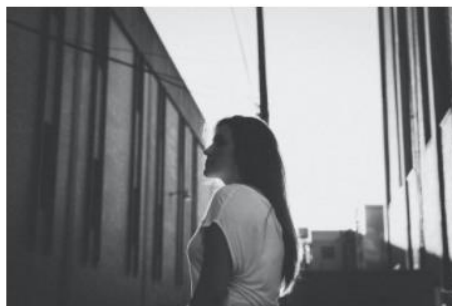
Tämän jälkeen voidaankin tarkastella valmista komponenttia itse sivulla, joka on nähtävillä kuviossa 13. Komponentti saa tyylit teemapohjasta, jossa on valmiiksi määritellyt tyylit esimerkiksi typografioille, palstoille ja niin edelleen.

```

<div class="container-fluid custom-element image-and-text">
  <div class="container">
    <div class="row">
      <div class="col-12 col-md-6 image-col <?php if($image_side ==
'left') echo 'order-md-1'; else echo 'order-md-2'; ?>">
        <?php if ($image_url) { ?>
          ">
          <?php } ?>
        </div>
      <div class="col-12 col-md-6 content-col <?php if($image_side
== 'left') echo 'order-md-2'; else echo 'order-md-1'; ?>">
        <?php
        if ($title) { ?>
          <h2 class="element-title"><?php echo $title ?></h2>
          <?php }
          if ($content) {
            echo $content;
          }
          if ($button) { ?>
            <a href="<?php echo $button['url']?>"<?php
            if($button['target']) echo
            'target='.$button['target'].' ?>><?php echo
            $button['title']?></a>
          <?php }
        ?>
      </div>
    </div>
  </div>
</div>

```

Kuvio 12. Komponentin HTML-rakenne



Lorem ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[Painike](#)

Kuvio 13. Kuva ja tekstialue -elementti sivulla

4 Tulokset ja johtopäätökset

4.1 Lopputulos

Opinnäytetyön tavoitteena oli perehtyä design systeemeihin ja tutkia miten näitä voisi hyödyntää toimeksiantajan teemapohjassa. Valmiista design systeemeistä ei päätetty ottaa sellaisenaan käyttöön, vaan Bootstrapia päätettiin hyödyntää enemmän kuin aiemmin.

Alkuperäisenä tavoitteena oli kartoittaa toimeksiantajan käyttämät komponentit ja luoda näistä muutama valmiiksi. Kuitenkin opinnäytetyön edetessä todettiin, että yhden komponentin tekeminen riitti esimerkiksi muille. Komponentin rakenteen suunnittelu sai tärkeämmän roolin opinnäytetyössä ja siihen keskittyminen oli tärkeämpää kuin useamman komponentin luominen.

Lopputuloksena opinnäytetyössä saatiin toteutettua rakenne, miten komponentit tulisi koostaa, yksi valmis komponentti sekä demo johon komponentit voidaan jatkossa koota.

4.2 Jatkokehitys

Teemapohjan kehityksessä ja varsinkin komponenttien käyttöönotossa ja muokattavuudessa on paljon varaa lisäkehitykselle. Komponenttien käyttöönotto tai koostaminen tapahtuu tällä hetkellä tiedostoja luomalla tai siirtämällä. Tähän voisi kehittää esimerkiksi jonkin työkalun tai pluginin, jolla voisi valita kyseiseen projektiin tarvittavat komponentit. Myös tyylien ja JavaScript-tiedostojen importaamisen voisi automatisoida, tällä hetkellä tämä pitää tehdä itse manuaalisesti.

Jatkokehityksessä voisi myös miettiä komponenttien tasoa. Voitaisiinko näitä jakaa vielä pienempiin osiin ja onko se kannattavaa? Myös tyylien osalta visuaalisesti samanlaiset toistettavat komponentit voisi koota yhteen paikkaan, jolloin nämä tyylit saisi helposti käyttöönsä luokilla.

4.3 Pohdinta

Opinnäytetyössä päästiin pääosin alussa asetettuihin tavoitteisiin. Design systeemin vertailussa ei löytynyt uutta design systeemiä käyttöön, mutta jo aiemmin käytössä ollut Bootstrapia päätettiin edelleen hyödyntää. Alussa myös pohdittiin sitä pitäisikö toimeksiantajan lähteä kehittämään omaa design systeemiä. Kokonaan omaa design systeemiä ei vielä lähdetty kehittämään, vaan teemapohjaa lähdettiin kehittämään komponenttien kautta.

Lopputulokseen toteutettiin vain yksi komponentti, vaikka alun perin näitä oli tarkoitus luoda useampi. Opinnäytetyön keskiöön nousi komponenttien rakenteen suunnittelu ja koostaminen. Useamman komponentin tekeminen olisi tuonut opinnäytetyöhän toistoa, joten komponenttien määrä jätettiin yhteen.

Lähteet

About. N.d. Yleistä tietoa Bootstrapista. Viitattu 27.8.2020.
<https://getbootstrap.com/docs/4.5/about/overview/>

Anne J., Mounter D., Stanfield R., Suarez M. & Saylor-Miller K. N. d. Design Systems Handbook. N.d. <https://www.designbetter.co/design-systems-handbook/introducing-design-systems>

Araújo J. 2018. Design Systems: benefits, challenges & solutions. Viitattu 14.1.2020.
<https://uxdesign.cc/design-systems-62f648c6dccb>

Audi Design System N.d. Viitattu 28.4.2020. <https://www.audi.com/ci/en/renewed-brand.HTML>

Barnard L. 2020. How to use wireframes with design systems. Viitattu 27.8.2020.
<https://www.freecodecamp.org/news/how-to-use-wireframes-with-design-systems-46f3040829b6/>

Bootstrap. N.d. Viitattu 27.8.2020. <https://getbootstrap.com/>

Bootstrap (front-end framework). 2020. Wikipedia artikkeli Bootstrapista. Viitattu 27.8.2020. [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

Chapman C. N.d. Why Use Material Design? Weighing the Pros and Cons. Viitattu 26.8.2020. <https://www.toptal.com/designers/ui/why-use-material-design>

Coyier C. 2019. Who Are Design Systems For?. Viitattu 27.8.2020. <https://css-tricks.com/who-are-design-systems-for/>

Euroopan parlamentin ja neuvoston direktiivi (EU) 2016/2102. Viitattu 8.9.2020.
<https://eur-lex.europa.eu/legal-content/FI/TXT/HTML/?uri=CELEX:32016L2102&from=FI>

Frost B. 2013. Atomic Design. Viitattu 15.4.2020.
<https://bradfrost.com/blog/post/atomic-web-design/>

GOV.UK Design System. N.d. Viitattu 31.8.2020. <https://design-system.service.gov.uk/>

Leppänen N. 2019. Mikä on design system ja mitä hyötyä siitä on bisnekselleni?. Viitattu 14.1.2020. <https://lamia.fi/blog/mika-on-design-system>

Material Design. N.d. Viitattu 28.4.2020. <https://material.io/>

MIT-lisenssi. 2020. Wikipedia-artikkeli MIT-lisenssistä. Viitattu 8.9.2020. <https://fi.wikipedia.org/wiki/MIT-lisenssi>

Moreno Celta R. 2019. How Atomic Design improves development structures. Viitattu 27.1.2020. <https://blog.prototypr.io/how-atomic-design-improves-development-structures-1eb4d617c7d6>

Mounter D. 2017. Design Systems, when and how much? — Diana Mounter. Luento Design systeemeistä. Lataaja: Netlify. Viitattu 28.1.2020. https://www.youtube.com/watch?v=Hx02SaL_IH0

Muscato M. 2018. How building a design system empowers your team to focus on people — not pixels. Viitattu 14.1.2020. <https://product.hubspot.com/blog/how-building-a-design-system-empowers-your-team-to-focus-on-people-not-pixels>

Noakes A. & Hupe A. 2018. Introducing the GOV.UK Design System. Viitattu 2.9.2020. <https://gds.blog.gov.uk/2018/06/22/introducing-the-gov-uk-design-system/>

Nunjucks. N.d. Viitattu 2.9.2020. <https://mozilla.github.io/nunjucks/>

Organizing Theme Files. N.d. Artikkele WordPressin teemojen kansiorakenteesta. Viitattu 14.11.2020. <https://developer.wordpress.org/themes/basics/organizing-theme-files/>

Pulkkinen A. 2018. Wordpress 5 ja Gutenberg – Selviytymisopas. Viitattu 14.11.2020. <https://www.zoner.fi/wordpress-5-ja-gutenberg/>

Tietoa saavutettavuudesta. N.d. Viitattu 8.9.2020. <https://www.saavutettavuusvaatimukset.fi/tietoa-saavutettavuudesta/>

U. S. Web Design System (USWDS) N.d. Viitattu 14.1.2020. <https://designsystem.digital.gov/>