



Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

KASPERI PRUUKKA
MATIAS OJAKOSKI

Käyttöliittymän uudistaminen web- komponenttipohjaisella kirjastolla

Opinnäytetyö

TIETOJENKÄSITTELY

2021

Tekijä(t) Pruukka, Kasper Ojakoski, Matias	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 4/2021
	Sivumäärä 72 sivua	Julkaisun kieli Suomi
Julkaisun nimi Käyttöliittymän uudistaminen web-komponenttipohjaisella kirjastolla		
Tutkinto-ohjelma Tietojenkäsittely		
<p>Tiivistelmä</p> <p>Tässä opinnäytetyössä tutustuttiin verkkosivujen ja sovellusten käytettävyyden ja käyttöliittymän suunnitteluun, sekä kehitettiin Visma Tampuuri Oy:n tuotteesta löytyvää QR-skannerin käytettävyyttä hyödyntämällä modernia web-teknologiaa.</p> <p>Opinnäytetyössä kehitystyö käyttöliittymän käytettävyyden parantamiseksi tehtiin ja suunniteltiin Visma Tampuurin kehitystiimin arkkitehtien ja ohjelmistokehittäjien kanssa. Käytettävyyden parantamisen toteutuksessa käytettiin useita moderneja web-kehityksen tekniikoita ja työkaluja, kuten esimerkiksi JavaScript, TypeScript, LitElement, Rollup, ja Postman. Opinnäytetyön versionhallinnassa käytettiin Mercurialia yhdessä TortoiseHg:n graafisen käyttöliittymän kanssa.</p> <p>Työn lopputuloksena saavutettiin Visma Tampuuri Oy:n tuotteesta löytyvään avaintenhallinnan allekirjoitusalueen QR-skannerin paranneltu käytettävyys. Parannelun käytettävyyden lisäksi lopputuloksena oli uusi QR-skanneri komponentti. Lisäksi opinnäytetyöhön liittyvä uuden teknologian pilotointi sujui onnistuneesti.</p>		
<p><u>Asiasanat</u></p> Web-kehitys, web-komponentti, JavaScript, TypeScript, Rollup, LitElement, UI, UX		

Author(s) Last name, First name Pruukka, Kasper Ojakoski, Matias	Type of Publication Bachelor's thesis	Date 4/2021
	Number of pages 72 pages	Language of publication: Finnish
Title of publication User Interface Development with a Web-component library		
Degree programme Business Information Systems		
Abstract <p>In this thesis, the goal was the development of user experience of a QR-scanner feature in one of Visma Tampuuri Oy's products as well as learning about UX/UI design and development in the context of websites and applications.</p> <p>The development of a better user experience in this thesis was developed and designed in cooperation with Visma Tampuuri product development team's architects and software developers. In the process of creating a better user experience multiple different modern web-development technologies and tools were used, for example JavaScript, TypeScript, LitElement, Rollup and Postman. The version control system used during this thesis was Mercurial with TortoiseHg graphical user interface.</p> <p>The end result of this thesis was a better user experience in the QR-scanner and a new QR-scanner component was developed during this thesis. In addition, the piloting of the used new technology was successful.</p>		
<u>Key words</u> Web-development, web-component, JavaScript, TypeScript, Rollup, LitElement, UI, UX		

SISÄLLYS

1 JOHDANTO	6
2 KÄYTTÖKOKEMUKSEN PARANTAMINEN	8
2.1 Nykykatsaus käyttökokemuksesta.....	8
2.2 Mikä on UX?	8
2.3 Mitä UX-suunnittelija tekee?	9
2.4 Toimintamenetelmät.....	10
2.4.1 Rautalankamalli	10
2.4.2 Kuvakäsikirjoitus	11
2.4.3 Käyttäjäprofiilit (Persoonat)	12
2.4.4 Tutkimus	13
2.4.5 Testaus	14
2.5 Kehityskohteet ja niiden ratkaisut	14
3 KÄYTTÖLIITTYMÄN VISUAALINEN SUUNNITTELU	16
3.1 Käyttöliittymän visuaalisen suunnittelun perusteita.....	16
3.1.1 Tasapaino	16
3.1.2 Yhteneväisyys	17
3.1.3 Tilan käyttö	17
3.1.4 Hierarkia	18
3.1.5 Vallitsevat elementit	18
3.1.6 Värimaailma.....	18
3.1.7 Jatkuvuus.....	19
3.1.8 Typografia.....	19
3.2 Yleisiä käyttöliittymästä löytyviä elementtejä	20
3.2.1 Käyttäjän syöte.....	20
3.2.2 Navigaation komponentit.....	21
3.2.3 Informatiiviset komponentit.....	21
3.3 Sovellusten ja verkkosivujen käyttöliittymän suunnittelu.....	22
3.3.1 Sovellusten käyttöliittymän suunnittelu.....	22
3.3.2 Erityyppisistä sovelluksista.....	23
3.3.3 Verkkosivujen käyttöliittymän suunnittelu.....	26
3.3.4 Erityyppisistä verkkosivuista	26
3.3.5 Esimerkkejä verkkosivujen käyttöliittymistä.....	29
3.4 Suunnittelun työkaluja.....	30
3.4.1 Figma	30
3.4.2 HTML ja CSS	31
3.5 Käyttöliittymän suunnittelun tekniikoita ja metodeja	31

3.5.1 Kultainen leikkaus	31
3.5.2 Parallel design	32
3.6 Käyttöliittymän suunnittelun virheitä.....	33
4 TUTKIMUSONGELMA	35
4.1 Opinnäytetyössä käsiteltävän alueen määrittely ja rajaaminen.....	35
4.2 Allekirjoitusalue	35
4.3 Tutkimusongelma.....	41
4.4 Tavoitetila.....	42
4.5 Nordic Cool 4	42
4.6 Nordic Cool 4 Kirjaston esittely.....	44
4.6.1 Sivun päänäkymä	45
4.6.2 Hakusivun päänäkymä	46
4.6.3 Taulukot	46
4.6.4 Ilmoitusviesti & Sivutus	47
4.6.5 Linkkipolku.....	48
4.6.6 Välilehdet.....	48
4.6.7 Ikonit	49
5 KONKRETIA	52
5.1 Opinnäytetyössä käytettävät työkalut.....	52
5.2 Projektin aloittaminen	57
5.2.1 Target .NET Framework 2.0 nosto 3.1 versioon	57
5.2.2 QR-Skannerin päivitys uusimpaan versioon.....	58
5.2.3 TypeScriptin käyttöönotto	59
5.3 Selvitys NC4 komponenttien käyttöönotosta	60
5.4 Web-komponenttien luominen ja käyttöönotto.....	62
5.4.1 Kamerakomponentin korjaaminen.....	63
5.4.2 LitElement.....	65
5.4.3 Rollup käyttöönotto	66
6 PROJEKTIN LOPPUTULOS	68
7 POHDINTA	VIRHE. KIRJANMERKKIÄ EI OLE MÄÄRITETTY.

LÄHTEET

1 JOHDANTO

Tässä opinnäytetyössä käsittelemme käytettävyyden ja käyttöliittymän suunnittelun eri teemoja, kuten millä perusteella verkkosivujen ja sovellusten käyttöliittymän ulkoasua ja toimivuutta suunnitellaan ja kehitetään, sekä minkälaisia eri muotoja ja värejä käytetään mistäkin syystä. Lisäksi opinnäytetyössä tutustumme moderniin web-komponenttitekologiaan Visma Tampuuri -nimisen yrityksen näkökulmasta, toteuttaen käytettävyyden parantamista Visma Tampuurin tuotteeseen, sekä käyden läpi kyseiseen tuotteeseen tehtävästä käyttöliittymän uudistuksesta, käyttäen sitä varten yrityksessä suunniteltuja ja kehitettyjä web-komponentteja.

Opinnäytetyön aihe valittiin sillä perusteella, että kummallakin tekijällä on henkilökohtaista mielenkiintoa aihetta kohtaan ja se on jatkuvasti ajankohtainen tietotekniikan alalla. Kaupallisten verkkosivujen ja sovellusten visuaalinen ulkoasu sekä käytettävyys ovat ratkaisevia tekijöitä niiden menestyksen kannalta ja usea yritys, jotka eivät ole kyseiseen aiheeseen tarpeeksi perehtyneet ovat joko havainneet tai tulevat havaitsemaan, että aiheen laiminlyönti aiheuttaa ongelmia. Visuaalisesti miellyttävän käyttöliittymän ulkoasun ja käytettävyyden kehittämättä jättäminen voi johtaa vähe-neviin käyttäjämääriin ja tuotteen suosion laskuun, sillä nyky maailmassa arvostetaan enemmän ja enemmän nopeaa, sulavaa ja luotettavaa tuotteen toimintaa.

Visma Tampuurilla ollaan juuri siirtymässä uuden käyttöliittymän käyttöönottoon, mikä tarkoittaa Tampuurin tuotteiden kohdalla uutta käyttöliittymän ulkoasua, sekä niissä käytettävään teknologiaan tehtäviä muutoksia.

Käyttöliittymän ulkoasuun tehtävät muutokset perustuvat Visma-konsernin Nordic Cool 4:ään, joka on kehitetty edeltäjäksi aikaisemmin käytössä olleelle Nordic Cool 3:lle. Käyttöliittymän ulkoasun uudet muutokset liittyvät lähinnä tilan ja värien käyttöön, mutta tässä tapauksessa Tampuurin tuotteisiin tehtävät muutokset ovat paljon laajempia, sillä osa Tampuurin tuotteista ovat ulkoasultaan vanhentuneita, eikä niitä välttämättä tunnistaisi uudempien tuotteiden kanssa verrattuna saman yrityksen tuotoksiksi, joten ne ovat visuaalisen uudistuksen tarpeessa.

Käyttöliittymän uudistaminen tuo mukanaan myös uusia teknisiä muutoksia Tampuurin tuotteissa käytettävän teknologian puolelle. Yleisen tuotteisiin tehtävän kehitystyön rinnalla on kehitetty web-komponenttikirjastoa, jonka käyttöönottoon ollaan tämän opinnäytetyön myötä siirtymässä. Tämän lisäksi teknisiä uudistuksia tuo JavaScriptin kanssa käytettävä TypeScript, web-komponenttien ja JavaScript moduulien ”paketoimiseen” käytettävä rollup, sekä MVC pohjaiset projektit.

Tämän opinnäytetyön tavoitteena on tutustua käyttöliittymän suunnittelun teoriaan, niin visuaalisen suunnittelun kuin käytettävyyden puolesta. Käyttöliittymän suunnittelun teorian ymmärtäminen ja osaaminen on hyödyllinen taito Front End-kehittäjille, joten aiheesta oppiminen on hyödyttävä kumpaakin opinnäytetyön tekijää.

Lisäksi tavoitteenamme on oppia Visma Tampuurin Nordic Cool 4 web-komponenttikirjaston sisällöstä eli web-komponenteista, niiden käyttötarkoituksesta, käytännön hyödyistä ja niihin liittyvistä teknisistä vaatimuksista.

Tutkimus ja tavoitteet kohdistuvat Visma Tampuurin tuotteesta löytyvään Avaintenhallinnan allekirjoitusalueen kamerakomponenttiin ja siihen liittyvään käytettävyyden ongelmaan, jonka ratkaisemiseksi joko sovellamme Visma Tampuurin Nordic Cool 4:n valmiiksi tehtyjä web-komponentteja tai vaihtoehtoisesti sovellamme niissä käytettävää teknologiaa, jolloin lopputuloksena olisi uusi web-komponentti. Riippumatta siitä, kummalla tavalla ongelmaa lähdemme tässä opinnäytetyössä ratkaisemaan, tavoitteenamme on tuotteen käytettävyyden parantaminen.

Tätä opinnäytetyötä on mahdollista hyödyntää ohjeena, mikäli lukijalla on tarve uudistaa jossakin projektissa sovelluksen tai verkkosivun käyttöliittymä web-komponenttipohjaisella kirjastolla ja tarpeena on ottaa käyttöön joko yksi tai useampi teknologia, mitä tässä opinnäytetyössä käsitellään tai mikäli lukija on kiinnostunut käyttöliittymän suunnittelun teoriasta.

2 KÄYTTÖKOKEMUKSEN PARANTAMINEN

2.1 Nykykatsaus käyttökokemuksesta

Nykyajan digitalisoitumisesta aiheutuva tarve rakentaa moderneja sekä käyttäjäystävällisiä ohjelmistoja lisääntyy. Käyttäjät vaativat tuotteiden olevan helppokäyttöisiä, jotta aikaa ei kuluisi niiden toiminnallisuuden tutkimiseen ja opiskeluun (Collapick n.d). Tästä syystä ohjelmistoalalla kilpailu paremmasta teknologiasta ja tiedon järkevästä visualisoinnista kasvattaa rooliaan tärkeysasemassa loppukäyttäjien ja yritysten keskuudessa. Kilpailu ohjelmistojen laadinnasta nostaa tuotteiden laatutasoa ja käyttäjäkokemusten perusteella voidaan laatia standardeja, jotka voivat nopeuttaa ja helpottaa loppukäyttäjien toimintaa.

Käyttäjät kokevat, että palvelun käyttöönotto sekä itse käyttö tulee olla helppoa ja nopeaa. Heitä yritetään houkuttaa palaamaan sivustoille hyvän käyttökokemuksen ansiosta. Käyttökokemuksen edellä kulkijana voidaan pitää ohjelmistoyritys Applea. Yritys palkkaa vain henkilöitä, jotka näkevät ja ajattelevat asioita hyvän käyttökokemuksen perusteella. Onnistuvia yrityksiä luodaan pohtimalla asioita käyttäjän näkökulmasta. (Olander 2015.)

2.2 Mikä on UX?

Käyttökokemuksesta voidaan käyttää lyhennettä UX eli User experience, joka tarkoittaa kokonaisvaltaista kokemusta käyttäjän ja tuotteen välillä. UX perustuu aina käyttäjän havainnointiin ja tuntemuksiin tuotteen käytöstä. Käyttökokemukseen vaikuttaa pääasiassa tuotteen visuaalinen ulkonäkö ja sen suorituskykyisyys. (Virtanen 2020.)

Käyttöliittymäsuunnittelulla tavoitellaan tehokkuutta palvelun toimintaan. Suunnitteluun otetaan mukaan käyttäjän näkökulma ja tapa toimia. Pääperiaatteena on saada aikaan sovellus, jota olisi mahdollisimman helppo käyttää. Parhaaseen lopputulokseen päästään analysoimalla dataa oikein sekä hyödyntämällä sitä asiakaslähtöisesti.

Huonon suorituskyvyn omaava sovellus voi luoda käyttäjälle turvattoman tunteen ja hän saattaa pahimmassa tapauksessa lopettaa tuotteen käyttämisen kokonaan. Vastakohtaisesti esimerkiksi yrityksen liiketoimintaa kasvattava ja työn tekemistä helpottava sovellus luo varmuudentunnetta sekä perusteita käyttää tuotetta vaikkapa tietoturvallisuuden näkökulmasta katsottuna.

Hyvin suunnitellun sekä toteutetun sovelluksen käyttämistä ei edes huomaa. Heikosti suunniteltu sovellus korostuu käyttäjien tekemien virheiden ja ongelmien kautta huomattavan selkeästi (Collapick n.d). Tuotteen tulee aina täyttää käyttäjän tarpeet. Tällä voidaan tarkoittaa esimerkiksi sen käyttöönottamista ja sen asentamista.

2.3 Mitä UX-suunnittelija tekee?

Suunnittelijan työn tarkoitus on keskittyä miettimään asioita käyttäjän näkökulmasta. Käyttäjälle tärkeää on käyttöliittymän selkeys, visuaalisuus, intuitiiviset asiat, käyttämisen vaivattomuus ja helppous. Pää tarkoituksena ja tavoitteena on toteuttaa kokonaisvaltaisesti miellyttävä kokemus ja lopussa tapahtuvasta transaktion suorittamisesta on tarkoituksena jäädä hyvä mieli käyttäjälle. (Virtanen 2020.)

UX-suunnittelijan vaiheittaiset toimenpiteet ovat rautalankamallien laatimiset, kuväkirjoitukset, erilaiset käyttäjäprofiilien luonnit ja lopuksi parhaan mahdollisen optimoinnin tuloksen saavuttaminen testaamisella (Virtanen 2020). Esimerkki tilanteesta, jolloin tulisi ryhtyä yllä mainittuihin toimenpiteisiin voisi olla kirjautumisloMAKE, joka sisältää tarpeettomia kenttiä ja on ulkoasultaan epäselkeä.

Suunnittelijan tulee olla valmis laatimaan käyttökelpoisia sekä helposti omaksuttavia tuotteita. He työskentelevät useasti osana isompaa tuotetiimiä. Tärkeä osa työstä on puolustaa loppukäyttäjää. Tyypilliset työtehtävät ovat verkkosivustojen, mobiilisovellusten ja ohjelmistojen suunnittelut. Visuaalinen suunnittelu ei kuitenkaan ole UX-suunnittelijan vastuulla. (White 2021.)

Välttämättömät ja vaaditut luonteenpiirteet suunnittelijalta ovat sopeutumiskyky, viestintäkyky, empatiakyky, ongelmaratkaisukyky sekä ryhmätyöhön kykenevä. On tärkeä osata työskennellä tehokkaasti tiimin, asiakkaiden sekä loppukäyttäjien kanssa. Suunnittelijan tulee olla perillä myös yrityksen tavoitteista ja soveltaa niitä kehityksen kanssa. (White 2021.)

2.4 Toimintamenetelmät

Tässä luvussa tulemme kertomaan vaiheista, joita UX-suunnittelija tekee työssään. Asiaa on helpompi ymmärtää ja selventää, kun jaetaan töitä vaiheiksi. Tulemme myös kertomaan henkilökohtaisia kokemuksista ja vertailen asioita.

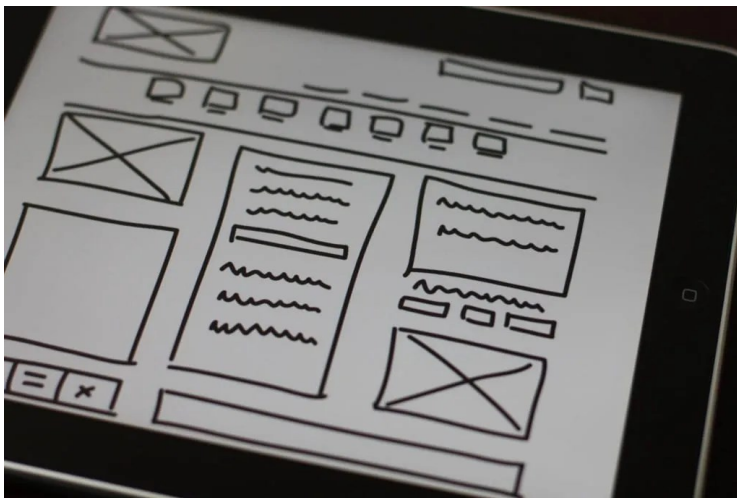
2.4.1 Rautalankamalli

Rautalankamallin (kuva 1) laatiminen on ensimmäinen askel, jonka suunnittelija ottaa suunnitellessaan uutta sovellusta tai nettisivua. Tämä malli on yksinkertaistettu pohja, joka sisältää sivustolla olevat elementit. Rautalankamallissa ei käytetä värejä, muotoiluja tai grafiikkaa. Mallin päätarkoituksena on vain esittää toiminnot, elementtien sijoittelut sekä niiden väliset vuorovaikutukset. Laatimalla tämänkaltaisen pohja helpotetaan suunnittelun etenemistä ja siihen on vielä yksinkertaisempaa tehdä muutoksia, kuin visuaalisesti panostettuun tuotokseen. Näin voimme huomata ongelmat aikaisessa vaiheessa ja säästää näin aikaa ja rahaa. (Parveen 2019.)

Rautalankamallit voidaan erotella niiden laatutasojen perusteilla. Heikkolaatuiset mallit ovat usein paperille käsin piirrettyjä hahmotuksia ja ne ovat yleensä staattisia eli eivät interaktiivisia. Parempilaatuisen mallin tekeminen vie paljon aikaa ja se on paljon yksityiskohtaisempi.

Teemme mielellämme tietokoneella malleja, koska silloin niistä tulee yleisesti hyvälaatuisia ja tarkkoja pohjia, joista on helppo jatkaa suunnittelua. Hyvä ohjelma rautalankamallien suunnitteluun on esimerkiksi Figma. Pidämme kyseisestä ohjelmasta,

koska sen käyttöliittymä on yksinkertainen ja lisäksi se on käytettävissä myös ilmaiseksi.



Kuva 1. Tässä näkyy laadultaan heikompi rautalankamalli. (Creative Commons n.d).

2.4.2 Kuvakäsikirjoitus

On sanomattakin selvää, että työyhteisömme kesken voimme kommunikoida esimerkiksi rautalankamalleilla. Tämä on ymmärrettävää, koska tämä on meille tuttu asia. On kuitenkin otettava huomioon ulkopuoliset ihmiset, joille tuotetta teemme. Kuvakäsikirjoitukset ovat tällaiseen tilanteeseen hyvä havainnollistamisen keino.

Kuvakäsikirjoitus (kuva 2) voidaan ajatella sekventaalisena taiteena, jossa kuvista muodostetaan tarina. Kuvilla voi parhaiten havainnollistaa termejä tai käsitteitä joutumatta laatimaan pitkästyttäviä virkkeitä. Kuvakäsikirjoitus sopii hyvin UX-suunnitteluun, koska se on iteratiivista ja tiimipohjaista suunnittelua. Näin kaikki voivat tuoda oman mielipiteensä esiin, eikä turhiin yksityiskohtiin paneuduta liikaa. (Crothers 2011.)

Kuvakäsikirjoituksessa on tiettyjä tärkeitä näkökohtia. Määritetään käyttäjän käyttöympäristö, esimerkiksi koulun tilat. Seuraavaksi mietitään tarkkaan ottaen henkilöiden persoonat ja tunteet. Näiden pohjalta mietitään ongelmaa, johon haetaan ratkaisua. On huomioitava koko tapahtumaketju eli tarinan alku, keskiosa ja sen loppuminen. (Sali 2020.)



Kuva 2. Leikkaus kuvakäsikirjoituksesta. (Morovián 2019.)

2.4.3 Käyttäjäprofiilit (Persoonat)

UX-suunnittelija tekee käyttäjäprofiileja edustamaan erilaisia mahdollisia todennukaisia käyttäjätyyppejä. Tämä luo todellisia mielikuvia tulevien käyttäjien tarpeista, käyttäytymisestä ja tavoitteista. Tässä voidaan ottaa huomioon muun muassa eri ikäluokkien väliset käyttäytymiseroavaisuudet. (Dam & Siang 2020.)

Käyttäjäprofiilit tekevät työstä vähemmän monimutkaisemman ja auttavat luomaan hyvän käyttökokemuksen tietyille kohderyhmille. Profiilit eivät kuvaa yksittäisiä ihmisiä, vaikka dataa kerätään todellisten tietojen perusteella. Ne luovat inhimillisen kosketuksen tutkimukseen, joka sisältäisi pelkkiä faktoja. (Dam & Siang 2020.)

Persoonia voidaan muodostaa neljästä eri perspektiivistä. Perspektiivit auttavat persoonan tekemisessä, kun voidaan kirjoittaa eri näkökulmista. Ensimmäinen näkökulma on tavoitteisiin tähtäävä persoona. Sen tarkoituksena on tutkia prosessi, jota tuleva käyttäjä haluaisi käyttää saavuttaakseen tavoitteensa. (Dam & Siang 2020.)

Seuraava perspektiivi on luoda persoona roolipohjaisesti. Tämä keskittyy myös käyttäytymiseen ja ne ovat massiivisesti datapainotteisia sekä peräisin luotettavista lähteistä. Tässä tarkoituksena on keskittyä käyttäjän rooliin organisaatiossa. (Dam & Siang 2020.)

Kolmantena näkökulmana on sitouttava persoona. Tässä ajatuksena on luoda käyttäjän renderöinti persoonien avulla. Siinä tutkitaan käyttäjän tunteita, heidän taustojaan ja

psykologisia mietteitä. Näkökulma korostaa, että kuinka tarinat voivat sitoutua ja herättää persoonat elämään. (Dam & Siang 2020.)

Viimeinen näkökulma on hieman erilainen. Tämä keskittyy kuvitteellisiin persoonoihin. Tässä korostuvat suunnittelijoiden oletukset ja aiemmat vuorovaikutukset henkilöiden kanssa. Nämä ovat täysin fiktiivisiä ja niiden ei tule olettaa toimivan oppaana tuotteiden kehityksessä. (Dam & Siang 2020.)

2.4.4 Tutkimus

Mitä tarkoittaa UX-Research eli suomeksi UX-tutkimus? Tutkimus sisältää laajasti erilaisia metodeja sisällön sekä syvyyden lisäämiseen suunnitteluprosessissa. UX-suunnittelijat käyttävät hyväksi tiedemiesten ja markkinatutkijoiden tekniikoita. Jotkut näistä ovat aivan omaa luokkaansa UX-maailmassa. Päätavoitteena on tutkia asioita loppukäyttäjän näkökulmasta. (Editorial Team 2018.)

Tutkimus jaetaan kahteen eri osa-alueeseen. Ensin prosessiin kuuluu datan kerääminen ja sen jälkeen datan hyödyntäminen käyttökokemuksen parantamiseksi. Datan kerääminen alkaa tarpeiden ja tavoitteiden selvittämisestä. Tehdään kyselyitä, pidetään haastatteluja ja pohditaan tulevaa. Tutkitaan asioita määrällisestä näkökulmasta, esimerkiksi käyttäjän klikkaamat toistokerrat, jonkin painikkeen suhteen. (Editorial Team 2018.)

Määrällisen näkökulman lisäksi otetaan myös huomioon laadullinen näkökulma, joka on niin sanottu pehmeämpi puoli. Tässä korostuu erityisesti haastattelut, joista saadut vastaukset esimerkiksi tilanteisiin, kun pohditaan että miksi käyttäjä ei nähnyt CTA-painiketta. Call-To-Action eli CTA-painike tarkoittaa sivustolla esiintyvää painiketta, lomaketta, banneria tai muuta elementtiä, joka ohjaa käyttäjää tekemään halutun toiminnon. Tällainen voisi olla nettisivuilla esimerkiksi ”Katso palvelut” -painike. (Editorial Team 2018; Kosola n.d)

2.4.5 Testaus

Tutkimusvaiheen jälkeen on seuraavaksi vuorossa testaaminen. On olemassa kaksi eri käyttäjätestauksen päätyyppiä. Näitä ovat valvotut ja valvomattomat testit. Valvotussa testissä tutkija analysoi käyttäjän toimintoja reaaliajassa. Tässä annetaan käyttäjälle tyypillisesti kysymyksiä, joihin hän antaa vastauksia ja tekee laadittuja tehtäviä. Hyvänä puolena valvotuissa testeissä on esimerkiksi käyttäjän reaktioihin perustuvat vastaukset. Testin järjestäminen voi tulla kuitenkin kalliiksi, kun mietitään siihen vaadittavia resursseja. (Tromm 2020.)

Valvomattomat testit eroavat käyttäjien läsnäolon puuttumisella. He voivat tehdä tehtäviään esimerkiksi kotoa käsin. Tämä on huomattavasti helpompaa ja halvempaa järjestää. Näin voidaan osallistujia testaamiseen vaikkapa toiselta puolelta maailmaa. (Tromm 2020.)

Käyttäjätestaus on prosessi, jossa todelliset käyttäjät testaavat jotain prototyyppiä esimerkiksi nettisivustoja. Käyttäjän selaamisen yhteydessä annetaan tehtäviä ja kysymyksiä, joita hänen tulee suorittaa. Samalla otetaan videomateriaalia talteen suorituksesta, jotta sitä voidaan myöhemmin käyttää analysointiin. (Tromm 2020.)

Valitettavasti usein käyttäjätestaaminen jää kehitystiimiin työksi. Tässä huonona puolena ovat rajalliset mielipiteet sekä vain sisäinen näkökulma asioihin. Olisi tärkeää saada palautetta ja kannanottoa myös tiimin ulkopuolelta. Vaikeuksia ulkoisiin näkemyksiin tuottaa salassapitovelvollisuus. Onkin tyypillistä, että haluttu palaute saadaan liian myöhään, kun sovellus on jo tuotannossa. (Valagroup 2018.)

2.5 Kehityskohteet ja niiden ratkaisut

Käyttökokemuksen parantamisessa on tärkeää ottaa huomioon web palveluihin sekä mobiilisovelluksiin muodostuneet käytännöt (Collapick n.d). Mobiilisovellusta käyttäneet tuntevat varmasti kolmen päällekkäisen viivan ikonista avautuvan valikon, jota sovelluskehittäjät kutsuvat hampurilaisvalikoksi. Tämänkaltaiset jo perinteeksi muodostuneet käytännöt ovat kehittäjän hyvä omaksua myös omalle tuotteelle. Näin vältetään turhat väärinymmärrykset ja niistä johtuvat huonot käyttökokemukset.

Yksi keskeisimmistä asioista, joka vaikuttaa käyttökokemukseen on johdonmukaisuus (Collapick n.d). Tuotteessa käytettävät kuvakkeet tulevat olla ennalta-arvattavia, kuten esimerkiksi suurennuslasikuvake, joka yleisesti tunnetaan hakutyökaluun viittaavana. Verkossa on saatavilla monipuolisesti kolmannen osapuolen sovelluksia, jotka ovat suuntautuneet tarjoamaan korkealaatuisia sekä johdonmukaisia kuvakkeita myös ilmaiseen käyttöön.

Johdonmukaisuuteen liittyvät myös visuaalisesti käyttäjälle näkyvä värimaailma. Suunnitteluvaiheessa tulee kiinnittää huomiota rajoittaviin tekijöihin, kuten värisokeisiin ihmisiin. Nykyään myös näkövammaisille suunnitellut käyttöliittymät ovat yleistyessä. (Uxpin n.d)

Kuvia ja videoiden käyttöä tuotteissa suositellaan vahvasti, koska ne lisäävät houkuttelevuutta ja saattavat opastaa paremmin sisällön ymmärtämisessä. Kuvista ja videoista tulee optimoida suorituskykyisiä ja responsiivisiä. Erityisesti mobiililaitteiden näkökulmasta on tärkeä huomioida niiden mitat suhteutettuna tuotteen muuhun sisältöön (Uxpin n.d).

Mikäli ohjelmiston kehitysvaiheessa unohdetaan käyttökokemus tai siihen panostetaan liian myöhään, saattaa olla, ettei tuotteessa ole yhtä ainoaa kriittistä ominaisuutta. Huono käyttökokemus voi vaikeuttaa ohjelmiston käyttöönottoa. Näihin ei voi markkinoinnilla tai hyvällä käyttöliittymällä vaikuttaa. Usein myös haalitaan liikaa ominaisuuksia, jolloin valmistuminen venyy ja käyttökokemus jää vanhanaikaiseksi. (Tapio n.d)

3 KÄYTTÖLIITTYMÄN VISUAALINEN SUUNNITTELU

Käyttöliittymän visuaalinen suunnittelu on jonkin ohjelman, sovelluksen tai verkkosivun käyttöliittymän ulkonäön suunnittelua. Sen avulla toteutetaan verkkosivulle tai sovellukselle tietynlainen ulkonäkö määrittelemällä esimerkiksi sovelluksen tai verkkosivun värimaailma, erilaisia muotoja ja elementtien järjestys, minkä mukaan sisältö näkyy käyttäjälle.

3.1 Käyttöliittymän visuaalisen suunnittelun perusteita

Käyttöliittymän visuaaliseen suunnitteluun liittyy useita hyväksi todettuja perusteita, joita seuraamalla pyritään määrittelemään, suunnittelemaan ja toteuttamaan miellyttävä ja käyttäjäystävällinen käyttöliittymä verkkosivulle tai sovellukselle. Useat näistä liittyvät vahvasti toisiinsa, joten jokaisen ymmärtäminen on yhtä tärkeää. Jokainen peruste on luonteeltaan joustava ja jättää suunnittelijalle tulkinnan varaa suunnittelua varten.

3.1.1 Tasapaino

Verkkosivulle tai sovellukselle käyttöliittymän ulkonäköä suunniteltaessa tulee tähdätä tietynlaiseen tasapainoon siinä käytettävien elementtien ja värimaailman kanssa. Suuria, näytön keskellä olevia elementtejä voidaan tasapainottaa lisäämällä pienempiä elementtejä sivun reunoille. Näille pienemmille elementeille voidaan myös määrittää huomattavasti tummempi väri kuin suuremmille, mikäli suurempien elementtien väri on kevyempi. Tämä johtuu siitä, että tummat värit vaikuttavat painavammilta kuin vaaleammat värit, joten pienelle elementille annettaessa tummempi väri saadaan siitä “painavampi”, luoden tasapainoa.

Liian paljon tai liian vähän saattaa johtaa epämiellyttävän näköiseen ja/tai epäkäytännölliseen käyttöliittymään. Huomioi, että tasapainolla ei tarkoiteta symmetriaa.

Tasapainolla on siis suuri vaikutus useaan muuhun suunnittelun kannalta huomioitavaan asiaan ja on suurin vaikutus:

- Värimaailmaan, erityisesti siihen, kuinka paljon värejä voidaan käyttää värিতasapainon säilyttämiseksi ja mitkä värit sopivat hyvin yhteen
- Verkkosivun tai sovelluksen yhteneväisyyteen
- Tilan käyttöön
- Elementtien hierarkiaan.

3.1.2 Yhteneväisyys

Hyvä käyttöliittymä on värimaailmaltaan ja yleiseltä ulkomuodoltaan yhteneväinen. Tällä tarkoitetaan sitä, että kaikki sivulta löytyvät elementit, logot tai muut ulkonäköön vaikuttavat tekijät näyttävät siltä, että ne kuuluvat yhteen. Mikäli visuaalinen yhteneväisyys ei toteudu kunnolla, voi käyttäjälle tulla epävarma olo siitä, toimiiko sivu tarkoitetulla tavalla vai onko se vielä keskeneräinen. Yhteneväisyyden saavuttamiseksi on käyttöliittymän oltava myös hyvin tasapainoinen.

3.1.3 Tilan käyttö

Tilan käyttö on mahdollisesti yksi tärkeimmistä perusteista ymmärtää. Sisällöltään liian tiiviiksi pakattu verkkosivu tai sovellus on käyttäjälle vaikea lukea ja mikäli näytölle ilmestyy liian paljon sisältöä käyttäjälle sisäistettäväksi voi se johtaa epämuakkaan käyttäjäkokemukseen. Tämä taas voi johtaa siihen, että käyttäjä vaihtaa toiseen sovellukseen tai verkkosivuun. Suotavaa myös olisi, että verkkosivulla tai sovelluksella on myös tyhjää tilaa, jonka avulla voidaan erotella sisältöä toisistaan sivuston tai sovelluksen luettavuuden helpottamiseksi. Lisäksi on huomioitava, saavutetaanko käytetyllä ja tyhjäksi jäävällä tilalla sopiva visuaalinen tasapaino.

3.1.4 Hierarkia

Elementtien hierarkialla määritellään näytöllä näkyvien elementtien tärkeys värien, muotojen ja kokojen kautta. Usein tärkeimmät elementit ovat mahdollisimman lähällä näytöllä, jotta käyttäjä huomaisi ne välittömästi niiden avauduttua näytölle. Useimmiten myös tärkeät elementit saavat korostetummat muodot ja huomattavasti kirkkaammat värit, kuin vähemmän tärkeät elementit. Tämän avulla käyttäjää voidaan siis ohjastaa haluttuun paikkaan, parantaen käyttäjäkokemusta. (White 2017.)

Elementtien hierarkiaan vaikuttaa niin tasapaino, tilan käyttö ja värimaailma, joten niitä on hyvä pohtia hierarkiaa suunniteltaessa.

3.1.5 Vallitsevat elementit

Elementtien vallitsevuus on huomioitava tietynlaisia verkkosivuja tai sovelluksia tehtäessä, yleensä keskittyen yhteen tai muutamaaan verkkosivun toiminnan kannalta keskeiseen elementtiin, antaen muille elementeille toissijaisen roolin. Tämä saavutetaan määrittelemällä elementeille niiden koko, muoto, väri ja sijainti niiden tärkeyden mukaan. Lopputuloksena on käyttäjälle selvä näkymä esimerkiksi siitä, mikä vaihtoehto käyttäjällä on valittuna. (Tervo 2016.)

3.1.6 Värimaailma

Sovelluksen tai verkkosivun värimaailma on myös yksi tärkeimmistä käyttöliittymään visuaalisen suunnittelun perusteista ymmärtää ja osata. Väreillä on tutkitusti psykologisia vaikutuksia ihmisiin ja suunnittelijalle on tärkeä tiedostaa tämä. Värimaailma on melkein poikkeuksetta yhteydessä sovelluksen tai verkkosivun omistavan yrityksen brändiin, joten on tärkeää, että ne ovat yhteneviä. Erinomaisesti toteutettu värimaailma on yksi niistä tekijöistä, mikä auttaa käyttäjää tunnistamaan käyttävänsä tietyn yrityksen sovellusta tai verkkosivua näkemättä yrityksen nimeä tai logoa. (deBara 2019.)

Värien avulla voidaan myös viestiä käyttäjällä, mitä toimintoja kullakin elementillä verkkosivussa tai sovelluksessa on.

- Punainen väri on usein yhdistetty lämpimiin ja mukaviin tunteisiin, mutta sillä voidaan myös viestiä negatiivisia asioita, kuten virheilmoituksia tai muita kriittisiä huomiota vaativia asioita.
- Keltaisella värillä usein symbolisoidaan iloa, onnellisuutta ja valoa ja se luetaan punaisen ja oranssin kanssa lämpimiin väreihin. Keltainen väri voi myös viestiä varoituksesta tai muusta huomioitavasta asiasta, mutta se ei ilmoita silti epäonnistumisesta, kuten punainen väri.
- Virheällä värillä usein viitataan luonnollisuuteen tai luontoon yleisesti. Vihreä väri liitetään myös onnistuneeseen tai valmiiseen tapahtumaan.
- Sininen väri usein yhdistetään turvallisuuden tunteeseen ja luottamukseen. Usein sovellukset, joilla pystyy hoitamaan verkkomaksuja tai muita vastaavia toimintoja, käyttävät sinistä väriä käyttöliittymässään.

3.1.7 Jatkuvuus

Jatkuvuus on usein tärkeämpi ymmärtää verkkosivuja tehdessä, mutta joissakin sovelluksissa on myös hyvä huomioida jatkuvuuden tärkeys. Jatkuvuudella tarkoitetaan sitä, että verkkosivu tai sovellus jatkuu sulavasti käyttäjän selatessa sivua alaspäin. Mikäli sivulla on esimerkiksi liian suuria tyhjiä välejä, saattaa käyttäjä olla huomaamatta niiden jälkeen tulevaa sisältöä. Tämä taas johtaa siihen, että käyttäjältä jää huomaamatta osa sivun sisällöstä. On siis tärkeää varmistaa käyttöliittymän sulava jatkuvuus, jotta käyttäjältä ei jää huomaamatta osaa sisällöstä tai käyttäjä ei menetä mielenkiintoa sisältöä kohtaan.

3.1.8 Typografia

Typografialla on myös tärkeä osa käyttöliittymän suunnittelussa. Tekstimuodossa oleva sisältö on pidettävä mahdollisimman selvänä, jotta kaikki mahdolliset käyttäjät ymmärtävät sivuston sisällön. Myös verkkosivussa tai sovelluksessa käytettävä fontti

on valittava käyttötarkoituksen mukaan, sillä liian erikoiset fontit saattavat johtaa epämiellyttäviin käyttäjäkokemuksiin. Erityisen tärkeä toteuttaa onnistuneesti verkkokäyttöön ja sivuissa sekä sovelluksissa, jotka sisältävät ohjeistusta johonkin asiaan.

3.2 Yleisiä käyttöliittymästä löytyviä elementtejä

3.2.1 Käyttäjän syöte

Checkboxit ovat neliön tai ympyrän muotoisia laatikoita, joita painettaessa yleensä valitaan kyseinen asia etsittäväksi, suodatettavaksi, hyväksyttäväksi tai poistettavaksi. Jokaiselta verkkosivulta löytyy **nappeja**, joita painaa. Usein nappi on suorakulmion muotoinen, sisältäen tekstiä ja/tai ikonin kertoen mitä nappi tekee. Napit voivat myös olla vain ikoneita, jolloin tavoiteltavaa olisi käyttää jotain hyvin tunnettua ikonia visuaalisesti kertomaan mitä nappi tekee.

Dropdown nappi on nappi, joka avaa dropdown valikon, mistä löytyy lisää nappeja, joita painaa.

Toggle-tyyppiset napit yleensä laittavat verkkosivussa tai sovelluksessa jonkin toiminnon päälle tai pois tai valitsevat jonkin asian käytettäväksi tai poistettavaksi käytöstä.

Useissa verkkosivuissa ja sovelluksissa on **tekstikenttiä**, joihin käyttäjä voi kirjoittaa sille tekstikentälle olennaista tekstiä, kuten esimerkiksi käyttäjänimen tai salasanan kirjoittaminen kirjautumisen tekstikenttään.

Date Picker on pieni interaktiivinen kalenteri, josta voi valita vuoden, kuukauden ja päivän vaivattomasti.

3.2.2 Navigaation komponentit

Hakukenttiä löytää melkein jokaiselta verkkosivulta. Niiden avulla pystytään etsimään avainsanojen tai lauseiden avulla sivustolta sisältöä. Hakukentän vieressä on usein suurennuslasin tai tekstiä sisältävä nappi, joka suorittaa haun hakukenttään kirjoitettujen merkkien perusteella.

Breadcrumb-toiminnon avulla käyttäjä pystyy tarkistamaan polun sille sivulle, jolla hän sillä hetkellä on. Tämä auttaa käyttäjää hahmottamaan mitä kautta käyttäjä on päässyt sivulle.

Liukusäätimen avulla käyttäjä pystyy säätämään jonkin asian arvoa, kuten esimerkiksi äänenvoimakkuutta. Usein Liukusäätimessä on jokin toiminto, joka ilmoittaa sillä hetkellä olevan arvon tarkasti.

Sivituksen avulla saadaan hallittua, paljonko sisältöä näytetään yhdellä sivulla. Tämä helpottaa sisällön etsimistä ja järjestelyä, sekä suodattamista.

Tagit viittaavat usein johonkin sisältöä sisältävään kategoriaan, mitä verkkosivulta tai sovelluksesta löytyy. Tagien avulla pystytään helpottamaan sisällön etsimistä ja suodattamista.

Ikonit ovat usein yksinkertaistettuja ja selviä, joiden avulla pystytään hallitsemaan käyttöliittymän tilankäyttöä. Ikonit on onnistuneesti toteutettu vain silloin, kun käyttäjä ymmärtää niitä katsomalla mitä niistä painamalla tapahtuu.

3.2.3 Informatiiviset komponentit

Tooltipit usein neuvovat käyttäjää jonkin napin tai muun toiminnon toiminnallisuudesta tai käyttötarkoituksesta, sisältäen lyhyen ohjeen miten kyseistä toimintoa tulee käyttää tai täyttää.

Progress Bar-elementit kertovat käyttäjän etenemisestä esimerkiksi jonkin lomakkeen täytössä ja mikä vaihe lomakkeen täytöstä on sillä hetkellä käsiteltävänä. Usein nämä eivät ole interaktiivisia, mutta joissakin sovelluksissa tai verkkosivuissa niistä voi siirtyä haluamaansa kohtaan.

Ilmoitukset yleensä kertovat käyttäjälle, mikäli jotain uutta tapahtuu. Ilmoitusten tarkoituksena on siis kertoa käyttäjälle, mikäli jokin prosessi on edennyt seuraavaan vaiheeseen, onnistunut tai epäonnistunut. Esimerkiksi jos käyttäjä saa uuden viestin tai käyttäjän täyttämät tiedot ovat puutteellisia eikä prosessi etene.

Message Boxit sisältävät jonkin viestin tai ohjeen ja useimmissa tapauksissa tämän tyyppisen laatikon piilottaminen vaatii viestin tai ohjeen kuittamista.

Pop-up modal usein lisää tummentavan filterin muulle sivulle, ohjaten käyttäjän huomion pop-upin sisältöön. Käytetään esimerkiksi joidenkin sovellusten ja verkkosivujen sisäänkirjautumista varten.

3.3 Sovellusten ja verkkosivujen käyttöliittymän suunnittelu

3.3.1 Sovellusten käyttöliittymän suunnittelu

Sovellusten käyttöliittymän suunnitteluun vaikuttaa pääosin sovelluksen tyyppi ja käyttötarkoitus, mutta myös sovelluksen tuottajan tai tilaajan yrityksen brändiin yhdistetty teema huomioidaan suunnittelussa.

Joissakin sovelluksissa sovelluksen käyttötarkoitus rajoittaa mahdollisuuksia käyttöliittymän suunnitteluun esimerkiksi siksi, että sovelluksen on tarkoitus viedä näytöltä mahdollisimman vähän tilaa tai itse laite, jolla sovellusta on tarkoitus käyttää, on pieni.

Sovellusten käyttöliittymän suunnittelu eroaa verkkosivujen suunnittelusta usein sovelluksen käyttötarkoituksesta ja rakenteesta johtuen. Esimerkiksi jos tarkoituksena on suunnitella sovellukselle, joka soittaa musiikkia, visuaalisesti miellyttävä ja toiminnallinen käyttöliittymä, käyttöliittymän suunnittelussa keskitytään vain musiikinsoittimelle olennaisiin toimintoihin.

Usein tuotettavasta sovelluksesta on jo tehty internetistä löytyvä verkkosivu, jolloin käyttöliittymää suunniteltaessa on hyvä mukailla selaimesta löytyvän verkkosivun esimerkkiä. Tämänlaisissa tapauksissa haasteena on suunnitella sovelluksen käyttöliittymä sillä tavalla, että se vastaa mahdollisimman paljon jo olemassa olevan verkkosivun käyttöliittymää. Usein tämä on mahdotonta toteuttaa täydellisesti, joten tavoitteena on sisältää verkkosivun keskeiset toiminnot sovelluksen käyttöliittymään.

3.3.2 Erityyppisistä sovelluksista

Sovelluksia löytyy monenlaisia, useisiin eri käyttötarkoituksiin. Jotkin sovellukset nopeuttavat ja helpottavat jonkin arkipäiväisen asian tekemistä, toiset toimivat eräänlaisena työkaluna, kuten laskimena tai kamerana. Suosituimpia sovelluksia ovat sosiaalisen median sovellukset ja viihdeteollisuuden sovellukset, kuten pelit sekä elokuvien ja sarjojen suoratoistoa tarjoavat sovellukset. Kaikkiin näihin on suunniteltu käyttöliittymä ja useimmissa tapauksissa tavoitteena on ollut tuottaa käyttöliittymästä selkeä ja helppokäyttöinen. Esimerkkejä tämän tyyppisistä sovelluksista ovat Uber, Wolt, Foodora ja Tinder.

Jokapäiväiset sovellukset eli ns. Lifestyle-sovellukset.

- Urheilu ja Fitness sovellukset
- Seuranhakuun tarkoitetut sovellukset
- Dieetti ja ruokailu sovellukset
- Matkailuun tarkoitetut sovellukset

Esimerkkejä tämän tyyppisistä sovelluksista ovat Uber, Wolt, Foodora ja Tinder.

Sosiaalisen median sovellukset ovat tällä hetkellä suosituimpia sekä koukuttavimpia sovelluksia, joita käytetään maailmanlaajuisesti. Yksi syy tähän suosioon löytyy erittäin hyvin tutkituista ja suunnitelluista sovellusten käyttöliittymään tehdyistä ratkaisuista, joilla on taattu mahdollisimman koukuttava käyttäjäkokemus. Useimmiten tämä ilmenee käyttöliittymässä päättymättömänä janana, joka syöttää jatkuvasti uutta sisältöä käyttäjälle, mikäli käyttäjä seuraa tarpeeksi montaa käyttäjää ja/tai sivua kyseessä olevassa sosiaalisen median sovelluksessa. Tämän lisäksi sisältö on sijoitettu keskelle näyttöä ja siitä on tehty mahdollisimman paljon tilaa vievä, jättäen vain vähän tilaa tekstille.

Sosiaalisen median sovellukset voidaan jakaa kahteen eri luokkaan:

- Sovellukset, joissa on mahdollista lisätä kontaktiksi täysin tuntematon ihminen, tietämättä heistä mitään tietoja lisätäksesi heidät kontakteihisi
- Sovellukset, joissa on pakko tietää jotain tietoa toisesta käyttäjästä ennestään, kuten käyttäjänimi tai puhelinnumero, jotta heidät voi lisätä kontaktiksi.

Esimerkkejä sosiaalisen median sovelluksista ovat Facebook, Instagram, Snapchat, Whatsapp, Signal, Telegram ja Microsoft Teams.

Tietokoneiden ja mobiililaitteiden yleistyttyä pelit sekä erilaiset viihde sovellukset ovat saavuttaneet suuren suosion todella nopeasti. Useat viihdeteollisuuden sovellusten käyttöliittymät jakavat monia samoja piirteitä sosiaalisen median sovellusten käyttöliittymien kanssa.

Peleistä varsinkin monipeleinä pelattavat pelit sisältävät myös jonkin verran sosiaalisen median sovelluksien käyttöliittymistä löytyviä piirteitä. Pelien käyttöliittymät ovat usein huomattavasti monimutkaisempia, kuin esimerkiksi sosiaalisesta mediasta löytyvät käyttöliittymät. Esimerkkejä näistä sovelluksista ovat Netflix, Youtube, Twitch ja Spotify.

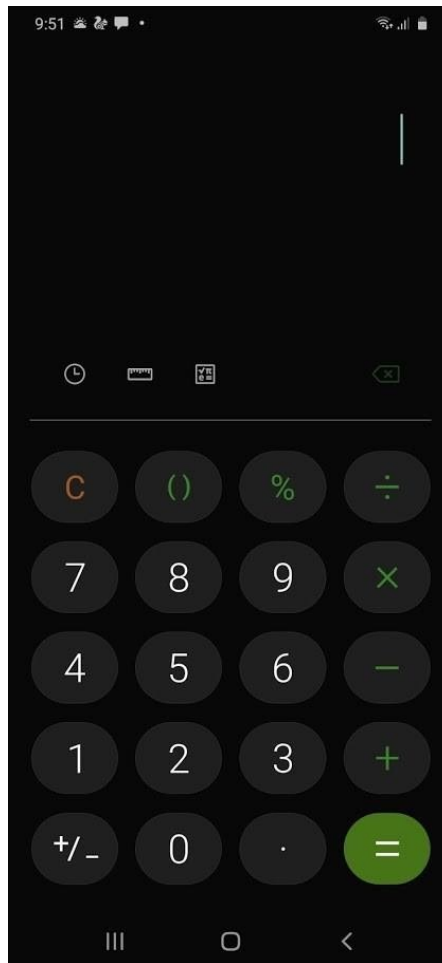
Työkalu-sovellukset, jotka ovat yleisempiä mobiililaitteissa ja kannettavissa tietokoneissa. Melkein poikkeuksetta sisältävät erittäin yksinkertaisen käyttöliittymän, joka

on tehty mahdollisimman selväksi ilman ylimääräisiä elementtejä. Esimerkkejä työkalu sovelluksista ovat laskimet, muistiot, taskulamput, kamerat, sää ja kalenterit. Yleishyödylliset sovellukset, joiden tarkoituksena on yksinkertaisesti helpottaa ja nopeuttaa jonkin asian tekemistä, kuten vaikka kirjanpitoa, dokumentointia tai vaikka pikaista laskujen maksamista tai rahan siirtämistä. Usein tämän tyyppisissä sovelluksissa on visuaalisesti yksinkertainen käyttöliittymä, joka kertoo mahdollisimman selvästi ja tarkasti, mikä toiminnon tarkoitus on. Tämä ei silti välttämättä aina takaa helppoa tai miellyttävää käyttökokemusta. Esimerkkejä yleishyödyllisistä sovelluksista ovat Google Docs, Google Sheets, Microsoft Excel ja Mobile Pay.

Esimerkkejä sovellusten käyttöliittymistä:



Kuva 3. Esimerkki lompakko sovelluksen käyttöliittymästä. (Mobile App – Crypto Wallet/uixninja)



Kuva 4. Kuva Samsungin mobiililaitteista löytyvästä laskimesta.

3.3.3 Verkkosivujen käyttöliittymän suunnittelu

Verkkosivujen käyttöliittymät harvoin eroavat toisistaan ja silloin kun eroja löytyy, erot eivät ole yhtä radikaaleja, mitä sovelluksien käyttöliittymien välillä voi olla. Tämä johtuu lähinnä siitä, että suurin osa verkkosivujen käyttöliittymäistä on luotu samantyyppisellä tai saman tyyppisellä teknologialla.

3.3.4 Erityyppisistä verkkosivuista

Verkkosivuja löytyy moneen eri tarkoitukseen, mutta suurin osa niistä seuraa verkkosivuille tyypillistä kaavaa käyttöliittymän suunnittelun puolesta. Suurimmalle osalle

aikaisemmin käsitellyistä sovelluksista löytyy niitä vastaava verkkosivu ja usein sovellukset on tehty tarpeesta siirtää verkkosivun tarjoamat palvelut mobiililaitteille suuremman käyttäjäkunnan saavuttamiseksi.

Toisin kuin sovelluksissa, verkkosivujen käyttöliittymät ovat usein samanlaisia toistensa kanssa. Tämä on johtanut siihen, että käyttäjän on helpompi navigoida uusien verkkosivujen käyttöliittymää koska ne ovat jo ennestään käyttäjälle tuttuja saman tyyppisten käyttöliittymien rakenteen vuoksi.

Sosiaalisen median verkkosivut ovat olennainen osa internetiä ja yksi suosituimmista ja tunnetuimmista verkkosivujen tyypeistä. Sosiaalisen median verkkosivujen tuottavuus kulkee käsi kädessä niiden koukuttavuuden kanssa. Kyseinen koukuttavuus saavutetaan suunnittelemalla mahdollisimman paljon mielenkiintoista sisältöä jatkuvasti syöttävä käyttöliittymä, jolloin käyttäjälle tulee koko ajan uutta sisältöä katseltavaksi.

Sosiaalisen median verkkosivut usein sisältävät enemmän toimintoja yhdessä näkyvässä kuin niistä tehdyt sovellukset. Tämän lisäksi verkkosivun käyttöliittymässä on tilaa näyttää enemmän sisältöä, joka vaatisi sovelluksessa oman näkymän tilan puutteen vuoksi.

Lisäksi on mainittava hitaasti katoava sosiaalisen median verkkosivun tyyppi, eli keskustelupalstat. Keskustelupalstat eivät ole kokonaan verkosta poistumassa, mutta ovat usein vähemmän suosittuja kuin muut sosiaalisen median verkkosivut, koska keskustelupalstoilta sisältöä on usein hitaampaa ja vaikeampaa löytää, eikä kyseinen sisältö esiinny keskustelupalstan käyttöliittymässä yhtä tiheään tahtiin kuin muissa sosiaalisen median verkkosivuissa. Esimerkkejä sosiaalisen median verkkosivuista ovat Facebook, Twitter ja Twitch.

Blogien verkkosivuja voidaan pitää eräänlaisena sosiaalisen median verkkosivujen tyyppinä, mutta niiden käyttöliittymä eroaa varsinaisista sosiaalisen median verkkosivuista huomattavasti. Yksi syy tähän huomattavaan eroon on usein blogin omistajan tekemät visuaaliset muutokset blogin verkkosivuun. Tämä taas johtaa siihen, että käyttöliittymään ja yleiseen ulkoasuun tehdyt muutokset ovat täysin mielivaltaisia ja perustuvat lähinnä siihen, mitä blogia pitävä henkilö pitää visuaalisesti miellyttävänä, eikä esimerkiksi käyttäjäkokemusta tai luettavuutta parantavana.

Vaikka blogit yleensä keskittyvät blogin pitäjän päivityksiin, sisältää useat blogit mahdollisuuden kommentoinnille ja tykkäyksille. Esimerkkejä blogien palveluntarjoajista ovat Wix, Squarespace ja WordPress.

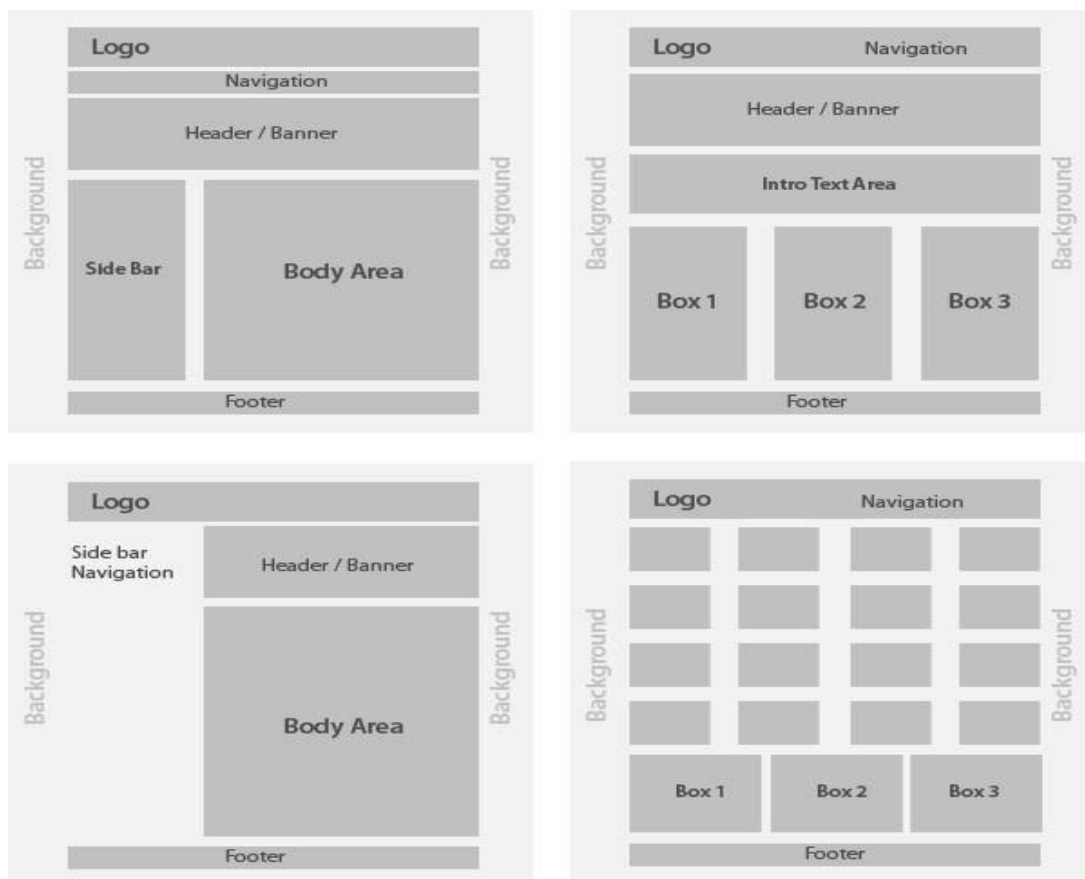
Kotisivut ovat digitalisaation aikakaudella jokaiselle yritykselle, yhdistykselle, kerholle tai muulle vastaavalle ehdottoman tärkeä löytyä, sillä nykyään tietoa edellä mainituista etsittäessä, etsivät potentiaaliset käyttäjät sitä verkosta. Usein tämän tyyppiset sivut sisältävät vähintään perustietoa yrityksestä, muutamia kuvia liittyen sen toimintaan ja yhteystietoja sekä linkkejä sosiaaliseen mediaan.

Joskus kotisivut saattavat sisältää hinnaston yrityksen tarjoamista tuotteista, joissakin tapauksissa saattaa löytyä jopa pieni verkkokauppa tai kauppapaikka, joka on osana kotisivua.

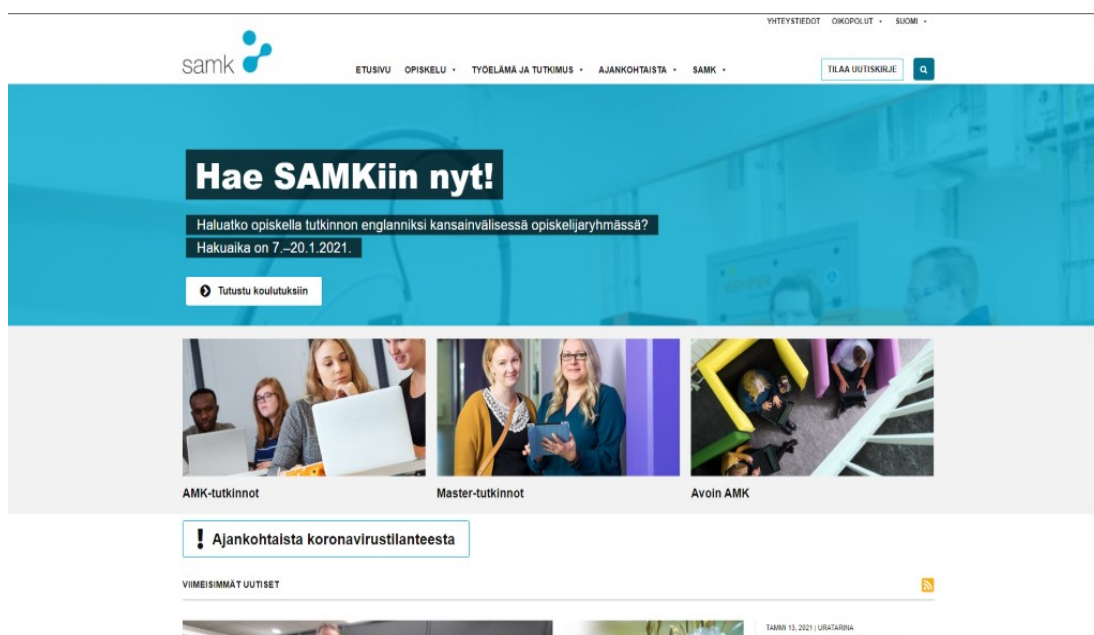
Usein kotisivut tehdään joko itse tai tilataan verkkosivuja tuottavilta yrityksiltä tai freelancereilta, riippuen kuinka suurista kotisivuista on kyse.

Verkkokauppojen verkkosivut ja eCommerce. Melkein jokaiselta yritykseltä, joka haluaa saavuttaa laajan, mahdollisesti kansainvälisen asiakaskunnan, löytyy verkkokauppa. Verkkokauppojen verkkosivujen käyttöliittymät seuraavat yleensä samaa kaavaa kuin muut verkkokaupat ja usein ainoat erot verkkokauppojen käyttöliittymien välillä ovat yrityksen brändiin ja joissakin tapauksissa myytäviin palveluihin tai tuotteisiin liittyvät erot. Esimerkkejä verkkokaupoista ovat Verkkokauppa.com, Gigantti ja Wolt.

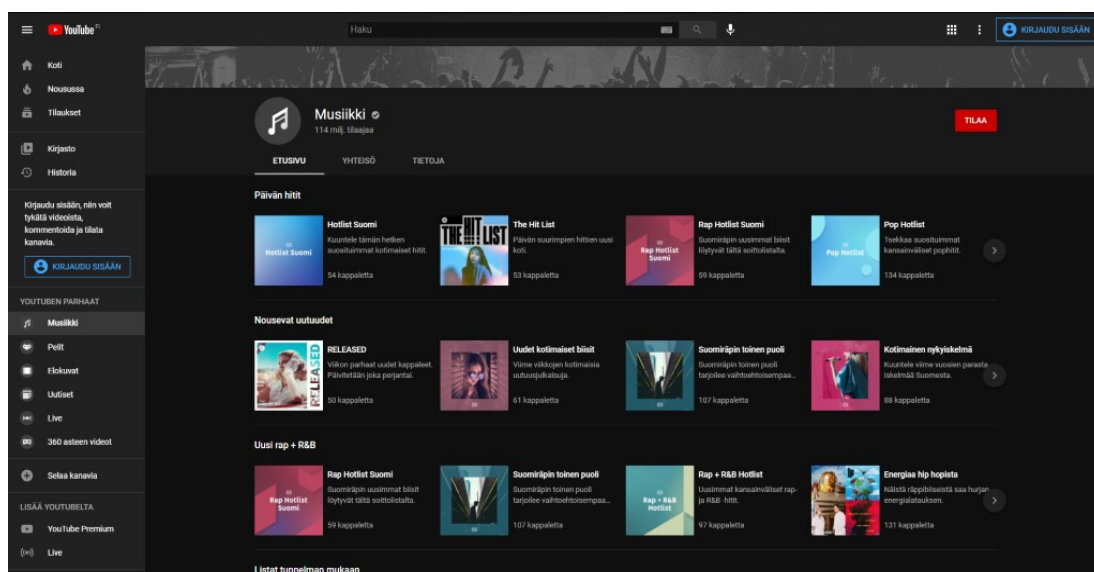
3.3.5 Esimerkkejä verkkosivujen käyttöliittymistä



Kuva 5. Esimerkkejä tyypillisestä käyttöliittymän elementtien asettelusta.



Kuva 6. Kuvankaappaus SAMKin sivun käyttöliittymästä.



Kuva 7. Kuva Youtuben Musiikki-sivun käyttöliittymästä.

3.4 Suunnittelun työkaluja

3.4.1 Figma

Käyttöliittymän suunnitteluun ja prototyypittämiseen tarkoitettu vektorigrafiikka editori [Figma](#) on yksi suosituista ohjelmista, joita käyttöliittymän visuaalisessa suunnittelussa käytetään. Figma on ilmainen, joten se soveltuu hyvin esimerkiksi aloitteleville suunnittelijoille tai vaikka ohjelmistokehittäjälle, joka tarvitsee kevyttä vektorigrafiikka työkalua työssään.

Figman käyttöön liittyvät perusteet on helppo oppia muutamassa tunnissa ja ohjelmalla onkin helppo toteuttaa ja kokeilla yksinkertaisia käyttöliittymän leiskoja. Figmasta löytyvien työkalujen avulla pystyy luomaan esimerkiksi mobiililaitteella käytettävän sovelluksen käyttöliittymän valitsemalla leiskan alustaksi haluamansa mobiililaitteen. Myös käyttöliittymästä löytyvien nappien toiminnallisuutta pystytään esittelemään yksinkertaistettuna Figmasta löytyvien toimintojen avulla, eli työkalun avulla on helppo esittää, minkälainen näkymä avautuu näytöllä napin painamisen jälkeen. (Figma.)

Figman kanssa saman tyyppisiä ohjelmia löytyy useita, maininnan arvoisena Sketch ja Adoben tuotteet

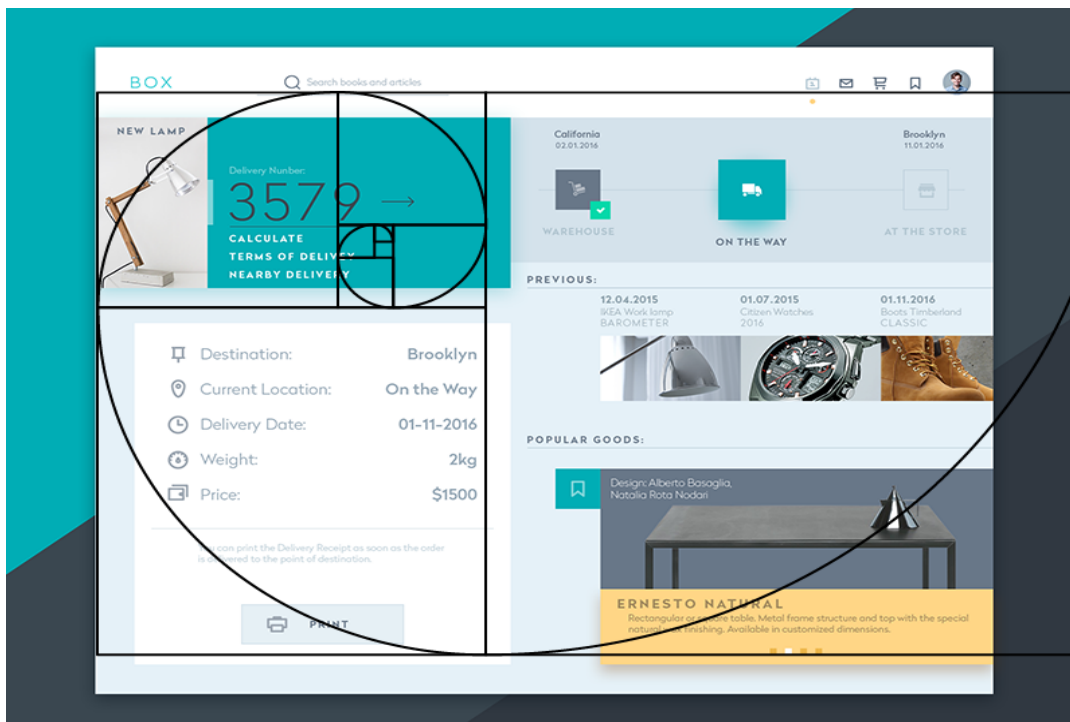
3.4.2 HTML ja CSS

Verkkosivun tai sovelluksen käyttöliittymän suunnittelijalle hyödyllisiä taitoja graafisten editorien käyttämisen lisäksi ovat HTML:n ja CSS:n perusteiden osaaminen. Näiden avulla suunnittelija pystyy kokeilemaan käytännössä, miltä hänen suunnittelemansa käyttöliittymä näyttää esimerkiksi verkkoselaimessa. Erityisesti HTML ja CSS osaaminen tulee suunnittelijalle hyödylliseksi, sillä HTML ja CSS kertovat selaimelle mitä näytetään ja miten. HTML: n avulla luodaan verkkosivulle pohja, jonka päälle lisätään halutut visuaaliset lisäykset CSS:n avulla. (Toptal 2020.)

3.5 Käyttöliittymän suunnittelun tekniikoita ja metodeja

3.5.1 Kultainen leikkaus

Käytettävyydeltään miellyttävä ja tehokas käyttöliittymän malli saavutetaan tekemällä siitä mahdollisimman helppo ymmärtää ja käyttää. Tämän saavuttamiseksi on tärkeä pyrkiä mahdollisimman tasapainoiseen elementtien käyttöön hyvän käyttäjäkokemuksen takaamiseksi. Kulusta leikkausta hyödyntämällä on mahdollista suunnitella visuaalisesti miellyttävä ja tasapainoinen käyttöliittymän malli. Kultainen leikkaus on Fibonaccin lukujonoon perustuva elementtien välistä matemaattista suhdetta toisiinsa mittaava suunnittelussa hyödynnettävä tekniikka, mutta sitä esiintyy usein myös luonnontieteissä. Kulusta leikkausta oikeaoppisesti hyödyntämällä uskotaan saavutettavan ihmisille estetiikaltaan mahdollisimman miellyttävä lopputulos. (Tubik Studio 2018)



Kuva 8. Esimerkki kultaisen leikkauksen soveltamisesta. (UX Planet – Tracking Widget)

3.5.2 Parallel design

Parallel Design tekniikkaa sovellettaessa useampi suunnittelija suunnittelee alustavan version verkkosivun tai sovelluksen käyttöliittymästä. Jokainen suunnittelija työskentelee itsenäisesti oman alustavan versionsa kanssa ja valmistuttuaan jakaa suunnittelemansa käyttöliittymän muiden suunnitteluryhmän jäsenten kanssa. Tämän jälkeen jokainen suunnitteluryhmän jäsen käy läpi jokaisen alustavan version ja valitsee niistä parhaat ideat ja soveltaa niitä omaan käyttöliittymäänsä. (Usability.gov.)

Tätä tekniikkaa sovellettaessa saavutetaan useita erilaisia ideoita ja mahdollisuuksia nopeaan tahtiin ja kustannustehokkaasti, mikä taas vaikuttaa kehityksen aikatauluun. Lopputuloksena on käyttöliittymän leiska, joka hyödyntää jokaisen suunnitellun alustavan version parhaita ideoita.

Tätä tekniikkaa sovellettaessa tärkeää on kommunikoida ryhmän kanssa tavoitteista ja

siitä, minkälaisia ideoita tarkalleen etsitään. Lisäksi on tärkeä saavuttaa yhteisymmärrys siitä, mitä ideoita pidetään hyvinä. Tätä helpottaa ryhmän jäsenten taitojen samantasoisuus.

Parallel Design tekniikan vaiheet lyhyesti:

- Jokainen ryhmän jäsen aloittaa itsenäisesti suunnittelemaan.
- Esitellään alustavat käyttöliittymien leiskat ryhmälle.
- Arvioidaan kaikkien ryhmän jäsenten suunnittelemat käyttöliittymät.
- Valitaan parhaat ja toimivat asiat ja toistetaan prosessi hyödyntäen uusia ideoita.

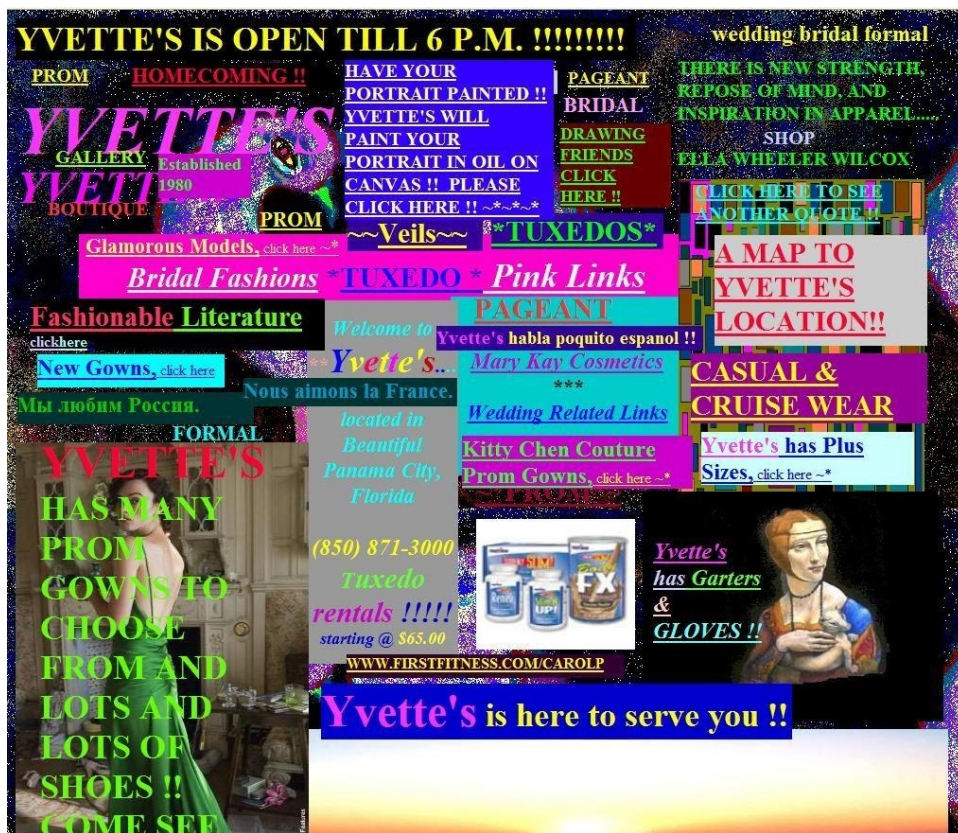
3.6 Käyttöliittymän suunnittelun virheitä

Käyttöliittymän suunnittelussa on helppo tehdä virheitä, varsinkin jos suunnittelun toteuttaa vähemmän kokenut tai täysin kokematon henkilö. Tämä usein ilmenee ylimääräisinä toimintoina, visuaalisina efekteinä tai liian monimutkaisina ja normaalista poikkeavina toteutuksina. Tästä johtuen on tärkeä seurata yleisesti hyväksytyjä ja toimivia tapoja miellyttävän käyttäjäkokemuksen takaamiseksi.

Joissakin tapauksissa virhe ei välttämättä johdu suunnittelijasta, vaan verkkosivun tai sovelluksen tilannut asiakas on vaatinut, että siihen lisätään elementtejä, komponentteja ja ominaisuuksia, jotka eivät välttämättä seuraa hyväksi todettuja käyttöliittymän visuaalisen suunnittelun tapoja.

- Liian sekavalla ja ristiriitaisella väriteemalla suunniteltu verkkosivun tai sovelluksen käyttöliittymä on yleinen virhe käyttöliittymän visuaalisessa suunnittelussa, varsinkin aloittelevilla suunnittelijoilla. Kaoottinen väriteema johtaa helposti epämiellyttävään käyttäjäkokemuksen eikä näytä silmää miellyttävältä, joka saattaa ajaa mahdolliset käyttäjät ja asiakkaat pois sivustolta tai sovelluksesta.

- Tiheään pakattu verkkosivu tai sovellus on usein vaikea käyttää johtuen siitä, että sisältöä selatessa käyttäjälle saattaa tulla epävarma olo siitä, missä relevantti sisältö jatkuu, liittykö kaikki näytöllä näkyvä sisältö käyttäjän lukemaan sisältöön ja tiheään pakatussa näkymässä on helppo jättää huomioimatta käyttäjälle mahdollisesti tärkeitä kohtia.
- Luettavuudeltaan huono ja epäselvä verkkosivu tai sovellus muodostuu helposti lisäämällä liian monta erialista elementtiä, komponenttia tai visuaalista efektiä sivustolle. Myös sisällön piilottaminen liian monen nappulan tai muun vastaavan toiminnon taakse johtaa huonoon käyttäjäkokemukseen.
- Liian monen erilaisen fontin käyttäminen yhdessä verkkosivussa tai sovelluksessa on usein virhe, jonka aloitteleva suunnittelija tekee. Liian monen erilaisen fontin käyttö vaikuttaa sivun sisällön luettavuuteen negatiivisesti ja hankaloittaa sisällön lukemista käyttäjälle, varsinkin jos muutoksia fontissa tapahtuu tiheään tahtiin. Yleisesti hyväksyttävä määrä eri fontteja on kahdesta kolmeen.



Kuva 9. Esimerkki, mistä löytyy useita virheitä. (Kuvankaappaus Yvette's -yrityksen kotisivusta.)

4 TUTKIMUSONGELMA

Tässä osassa opinnäytetyötä esitellään tutkittavaksi valittu ongelma Visma Tampuurin tuotteessa, sekä sen korjaamiseen liittyviä ratkaisuja.

4.1 Opinnäytetyössä käsiteltävän alueen määrittely ja rajaaminen

Opinnäytetyötä varten tarkoituksena oli löytää työpaikan kautta jokin aihe tai projekti, josta hyötyisi niin opinnäytetyön tekijä kuin työnantaja. Tavoitteena oli löytää työnantajan kanssa aihe, joka on tällä hetkellä ajankohtainen.

Opinnäytetyössä Visma Tampuurille tehtävä konkreettinen osio aloitetaan määrittelemällä vaatimukset niin opinnäytetyön kuin yrityksenkin puolesta. Tämän jälkeen rajataan yrityksen tuotteesta tietty alue, mihin opinnäytetyöhön liittyvä konkreettinen kehitystyö tullaan tekemään.

Tätä opinnäytetyötä varten valikoitui Tampuurin avaintenhallinnan allekirjoitusalue. Avaintenhallinnan allekirjoitusalueella on käytössä QR-skanneri, jota käytetään mobiililaitteella tehtävien allekirjoitusten QR-koodien skannaamista varten. Kyseisessä toiminnossa on käytettävyyden kannalta ikävä ongelma, mikä on tarkoitus tässä opinnäytetyössä saada korjattua. Kyseisen käytettävyyteen liittyvän ongelman korjaamisen lisäksi tarkoituksena on päivittää kyseinen toiminto uudella teknologialla. Tätä uutta teknologiaa ollaan ottamassa käyttöön Visma Tampuurilla laajemmin tulevaisuudessa, joten tämä opinnäytetyö toimii eräänlaisena pilotointina uudelle teknologialle.

4.2 Allekirjoitusalue

Mikä on allekirjoitusalue? Allekirjoitusalue kehitettiin paperikuorman vähennystä varten. Sitä käytetään Tampuurin -tuotteen avaintenhallinnan moduulissa. Tampuurin

allekirjoituslupaus –ratkaisulla on haettu korvaajaa nykyisille tätä tarkoitusta varten tehdyille laitteistopohjaisille ratkaisuille, joissa rajapinnat ovat suljettuja ja vain tiettyyn laitteistomalliin sopivia. Uudella ratkaisulla voidaan valjastaa mikä tahansa Android –pohjainen mobiililaitte toimimaan allekirjoituksia vastaanottavana laitteena.

Kuvaus toiminnasta:

1. Asiakas saapuu vastaanottamaan avaimet ja toimittamaan allekirjoituksen.
2. Asiakaspalvelija avaa Tampuurin avainhallinnan, tekee lainauksen merkaton asiakkaan tiedot sekä “Allekirjoita” -laatikkoon ruksin ja valitsee “Lainaa valitut”.

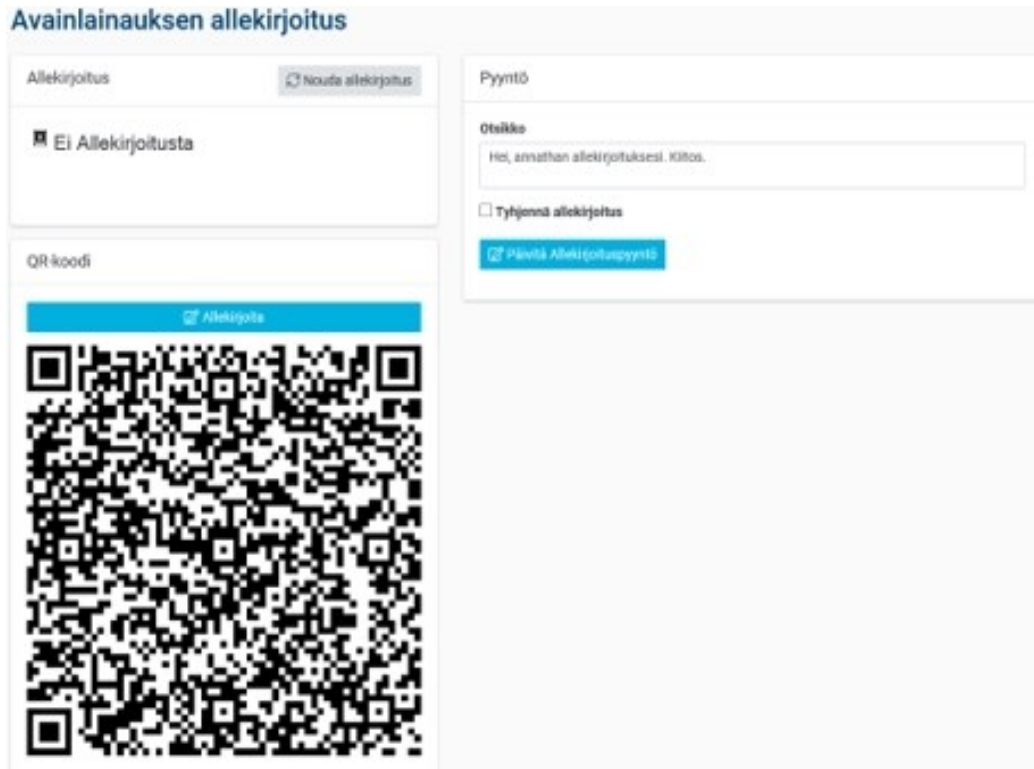
<input type="checkbox"/>	Avainleima	Tyyppi: (Kaikki)	Kohde
<input checked="" type="checkbox"/>	ABC/1	Askartelu + mankeli	001 As Oy Matinraitti 7/7
<input type="checkbox"/>	ABC/2	Askartelu + mankeli	001 As Oy Matinraitti 7/7
<input type="checkbox"/>	ABC/3	Askartelu + mankeli	001 As Oy Matinraitti 7/7

Kuva 10.

3. Allekirjoitus –kohdan ollessa valittuna avautuu allekirjoituspyynnön aktiivointinäköymä.

Kuva 11.

4. Tämän jälkeen asiakaspalvelija valitsee “Luo allekirjoituspyyntö”, jolloin avaintenhallinta luo QR-koodin mobiililaitetta varten, jolla otetaan asiakkaan allekirjoitus.



Kuva 12.

5. Asiakaspalvelija valitsee mobiililaitteen allekirjoitusta varten ja käynnistää sen. Mobiililaite on oletuksena odottaa-tilassa akun säästämistä varten. QR – koodin skannaus aktivoidaan painamalla “Käynnistä kamera” painiketta.

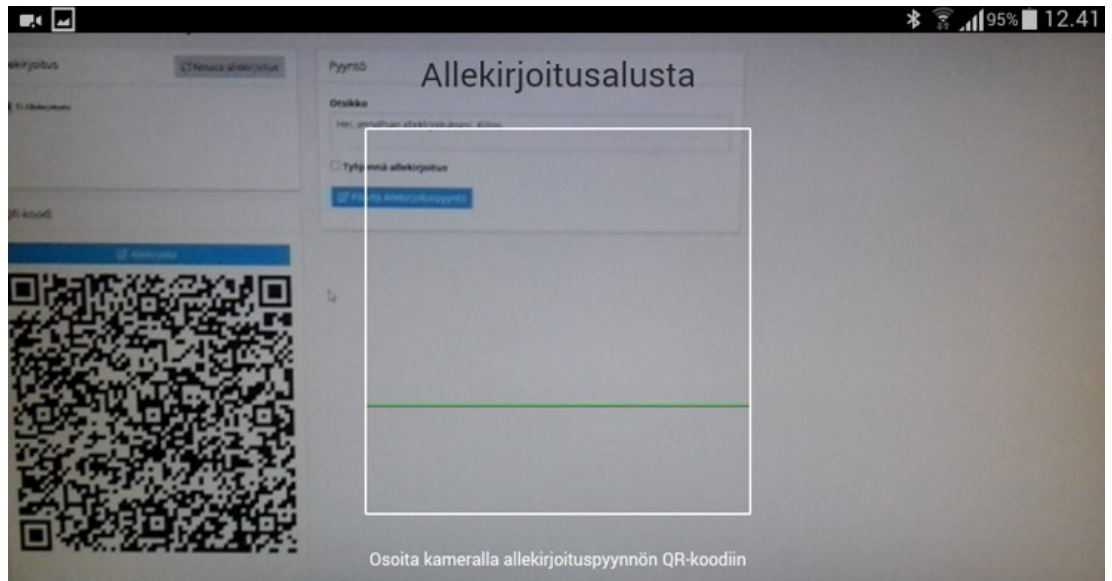


Allekirjoitusalue

Käynnistä kamera

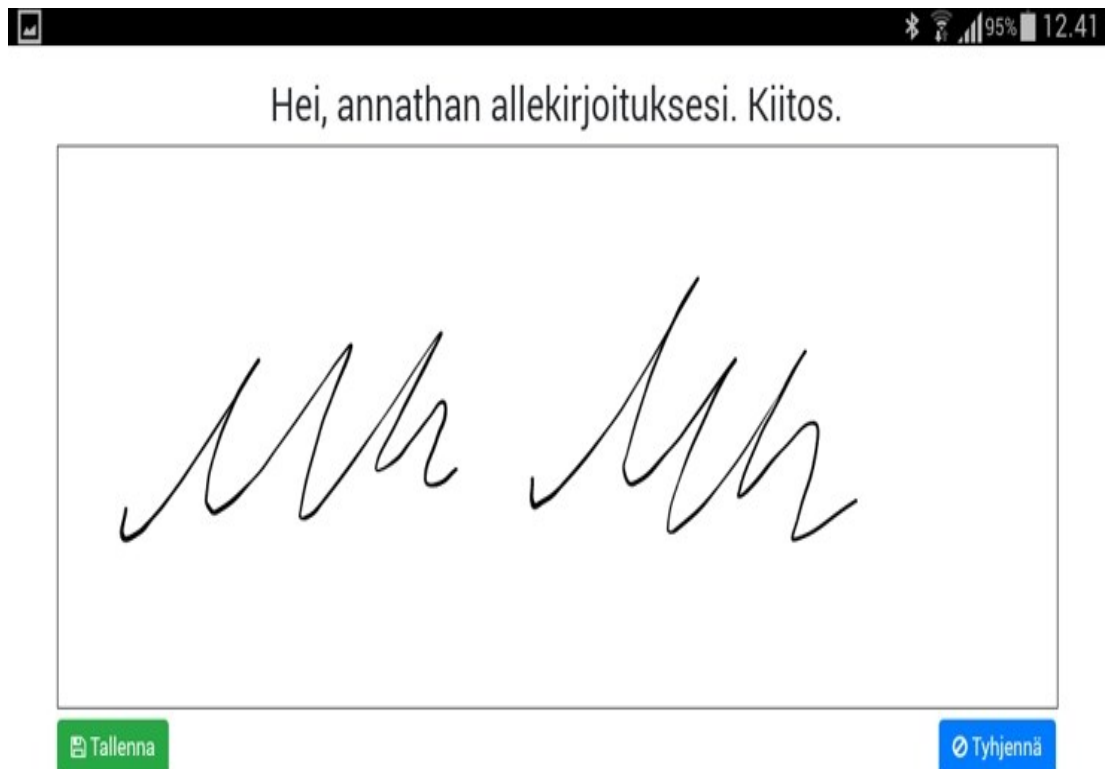
Kuva 13.

6. Asiakaspalvelija osoittaa mobiililaitteen kameralla rajatun neliöalueen kohdalle QR –koodin.



Kuva 14.

7. Kun Koodi on skannattu, mobiililaitte kytkeytyy automaattisesti allekirjoitusta odottavaan tilaan. Tässä vaiheessa asiakaspalvelija ohjeistaa asiakasta painamaan vihreän väristä "Tallenna" painiketta ja ojentamaan laitteen takaisin kiitosviestin jälkeen.



Kuva 15.

8. Kun allekirjoitus on vastaanotettu, ilmestyy ruudulle kiitosviesti. Kyseinen viesti pysyy ruudulla n. 10 sekuntia, minkä jälkeen mobiililaitte palaa takaisin uutta allekirjoitustapahtumaa odottamaan.

Kiitos


Allekirjoituksesi on vastaanotettu.

Kuva 16.


9. Asiakaspalvelija varmistaa allekirjoituksen painamalla oman laitteensa ruudulta “Nouda allekirjoitus” -painiketta.

Avainlainauksen allekirjoitus

Allekirjoitus Nouda allekirjoitus



QR-koodi Allekirjoita



Pyyntö

Otsikko

Hei, annathan allekirjoituksesi. Kiitos.

Tyhjennä allekirjoitus

Päivitä Allekirjoituspyyntö

Kuva 17.

10. Tämän jälkeen allekirjoitus on vastaanotettu ja se löytyy avainlainauksen tiedoista sekä tulosteista.

Avainlistaan Lainaa valitut

Avainten massakuittaus

Lainausaika: 11.06.2020 -



Kuva 18.

11. Allekirjoitus on nähtävissä Avainlainauksen tiedot -näkyvässä ja tulosteissa panttikuitti sekä Lainauskuitti.

Avainleima	Status	Lainaus tilanne	Alkaa	Päätyy	Le
  123/1	Käytössä	Lainassa	11.06.2020		Pi
  123/2	Käytössä	Lainattavissa			

Avainlainauksen tiedot - Google Chrome








isannointi.tampuuri.fi/Agenteq.Tampuuri.Avaintenhallinta.Web/ah_lainaus.aspx?avainid=8868&lain

Avainlainauksen tiedot

Kohde [9999] As Oy TESTI - Keskuskatu 3 , 00100 TESTILÄ

Avaimen nimi

Avaimen numero

		Avaimen luovuttaja Avaimen lainaaja Avaimen lainaajan yritys Alkaa - Määräaika	Kuittaus Liite
		 Lainaukset  Panttikuitti  Perintäkiri  Lainauskuitti	Piirainen Tomi Piirainen Tomi Testi 11.06.2020 -  Näytä liitteet (0 kpl)

Kuva 19.

Avainlistaan Lainaa valitut Noutoautomaattiin Valittuja avaimia 1 kpl

Avainten massakuittaus

Lainausaika: 09.04.2018 - []

Luovuttaja: Alhainen Mika

Lainaja: Alhainen Mika

Lainajan yritys: Agenteq Tiimi

Allekirjoita:

Lainauksen lisätiedot: Testilainaus

Valitse viestipohja: [v]

Noutoautomaattiviesti: []

Tulosta lainakuitti:

<input type="checkbox"/>	Avainleima	Tyyppi: (Kaikki) [v]	Kohde
<input checked="" type="checkbox"/>	ABC/1	Askarlelu + mankeli	001 As Oy Mafinraitti 7/7
<input type="checkbox"/>	ABC/2	Askarlelu + mankeli	001 As Oy Mafinraitti 7/7
<input type="checkbox"/>	ABC/3	Askarlelu + mankeli	001 As Oy Mafinraitti 7/7

Kuva 20.

4.3 Tutkimusongelma

Visma Tampuurin asiakkaat ovat raportoineet, että Avaintenhallinnan allekirjoituksessa käytettävä QR-skanneri avaa mobiililaitteella kameran laitteen väärältä puolelta. Tästä syystä on siis tutkittava, onko mahdollista kameran kääntöominaisuuden lisääminen tai kameran toiminnan korjaaminen. Kamera ei siis tällä hetkellä sisällä kääntöominaisuutta, joten kuvan ottaminen koituu luonnottomaksi toimenpiteeksi kameran avautuessa satunnaisesti väärään suuntaan, johtaen ikävään ja epäkäytännölliseen käyttäjäkokemukseen.

Lisäksi on tutkittava, onko käyttöliittymän uudistaminen web-komponenttipohjaisella teknologialla mahdollista, käyttäen joko olemassa olevia web-komponentteja tai luoden uuden web-komponentin tarpeen mukaan.

Lisäksi on selvitettävä, onko kyseiselle komponentille mahdollista kehittää Android-tuki kaikille uusille käyttöjärjestelmäversioille. Asiakaspalvelun mukaan joillakin Android -käyttöjärjestelmän omaavilla mobiililaitteilla esiintyy ongelmia alustan käytön yhteydessä, joten Android tuen lisääminen on tarpeellinen.

4.4 Tavoitetila

Visma Tampuurin tuotteeseen tehtävän kehityksen tavoitteena on käytettävyydeltään parempi käyttöliittymä. Tavoitteena on saada avautumaan kamera mobiililaitteen oikealta puolelta. Toisin sanoen mobiililaitte avaa oletuksena takakameran, jolloin on helpompi kohdistaa QR-skanneri oikeaan kohtaan QR-koodin skannaamista varten, koska skannattavan alueen näkeminen mobiililaitteen näytöltä on tehty mahdolliseksi.

Lisäksi tavoitteena tuottaa pilotointi uudelle kirjastolle sekä sen kanssa käytettävälle teknologialle, tarkoituksena tuoda esiin mahdolliset hyödyt ja haasteet. Visma Tampuurilla on menossa sisäinen kehityshanke käyttöliittymän uudistukselle ja tämä opinnäytetyö toimii pilotointina eteen tuleville haasteille.

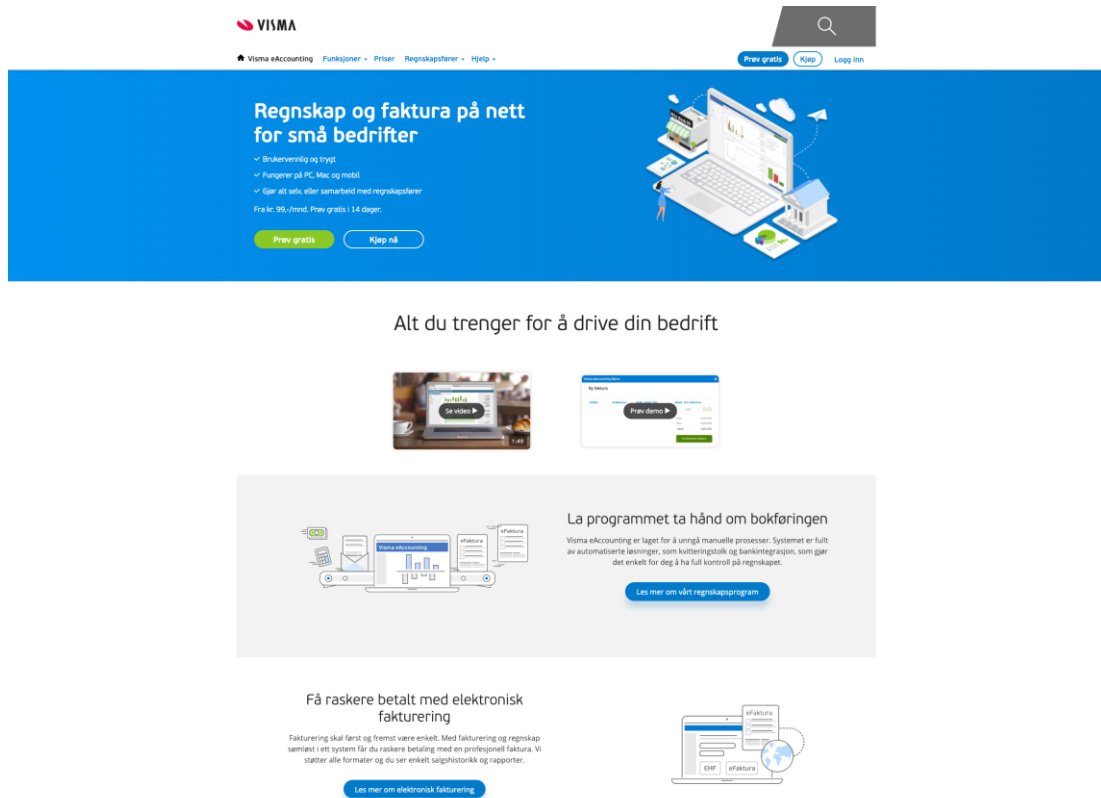
4.5 Nordic Cool 4

Tässä opinnäytetyössä on tarkoitus selvittää mikä on Nordic Cool 4, kuinka se toimii ja miten sitä pystytään hyödyntämään Visma Tampuurin tuotteissa. Tarkoituksena on käydä läpi Nordic Cool 4:n tuomia visuaalisia muutoksia käyttöliittymään sekä sen vaatimia teknisiä muutoksia itse tuotteen ohjelmistoon.

Käyttöliittymän uudistamista tehdään Visma Tampuurille, joten opinnäytetyön tavoitteena on tutkimusongelman ratkaisemisen lisäksi työskennellä yrityksen standardien ja vaatimusten mukaisesti.

Nordic Cool 4 on Visma konsernin kehittämä käyttöliittymän ulkoasu, jota ollaan ottamassa käyttöön useissa Visman tuotteissa ja palveluissa. Nordic Cool 4:n on tarkoitus korvata aikaisemmin käytössä olleet käyttöliittymän ulkoasut.

Nordic Cool 4 on suunniteltu korjaamaan sitä edeltäneen Nordic Cool 3:n ongelmakohdat. Nordic Cool 4:n uusi design muuttaa sivun yläosassa olevat valikoiden taustat sinisiksi, mikä on katsottu tärkeäksi yrityksen brändin puolesta, sillä useissa Visman tuotteissa käytetään käyttöliittymässä sinisen eri sävyjä. Lisäksi tarkoituksena on hyödyntää koko sivun leveyttä ja valkoista tilaa, mikä tekee käyttöliittymästä taspainoisemman ja ilmavamman.



The image shows the Visma eAccounting website homepage. At the top, there is a navigation bar with the Visma logo, menu items like 'Visma eAccounting', 'Funksjoner', 'Priser', 'Regnskapsfæser', and 'Hjelp', and user options 'Pruv gratis', 'Kjøp', and 'Logg inn'. The main content area features a large blue banner with the heading 'Regnskap og faktura på nett for små bedrifter'. Below the heading, there are three bullet points: '✓ Brukervennlig og trygt', '✓ Fungere på PC, Mac, og mobil', and '✓ Gjør alt selv, eller samarbeid med regnskapsfæser'. A price range 'Fra kr. 99,-/mnd, Pruv gratis i 14 dager.' is shown, along with 'Pruv gratis' and 'Kjøp nå' buttons. To the right of the text is an illustration of a laptop, a smartphone, and a tablet displaying various data charts and graphs. Below the banner, the text 'Alt du trenger for å drive din bedrift' is centered. Underneath, there are two small images: one showing a laptop with a 'Se video' button and another showing a 'Pruv demo' button. A larger section titled 'La programmet ta hånd om bokføringen' includes a diagram of the accounting process and a 'Les mer om vårt regnskapsprogram' button. At the bottom, there is a section for 'Få raskere betalt med elektronisk fakturering' with a 'Les mer om elektronisk fakturering' button and an illustration of a laptop and a globe.

Kuva 21. Esimerkki Nordic Cool 3 käyttöliittymästä.

Gjør det du er god på

Hold fokus med vårt automatiserte regnskapsprogram. Visma eAccounting er så enkelt at du kan ha 100% fokus på å lykkes.

[Prøv gratis](#)

Alt du trenger for å drive din bedrift

Vi i Visma eAccounting vet at regnskapsprogram er en ren nødvendighet, ikke et ønske. Derfor har vi gjort Visma eAccounting så enkel og automatisert at du kan fokusere på akkurat det du liker.

[Se hvordan](#)

Inv. No.	Invokt. No.	Cu. No.	Customer name	Type	Category	Invoice date	Due date	Unpaid amount
C2	88	26	Army Of Denmark	Debit Invoice	Normal	2016-09-31	2016-06-30	1427.00
C1	87	2	Arles Society	Debit Invoice	Normal	2016-09-31	2016-06-30	1160.00

Account	Debit	Credit
1510 - Kundfordringer	1160.00	
3342 - Forvalgt lønn 6% mv		1084.00
2021 - Utgående moms på forsalping inom Sverige, 6%		65.64
3740 - Dros och kronavgiftning		0.36

Kuva 22. Esimerkki Nordic Cool 4 käyttöliittymästä.

Varsinaisten tuotteisiin kohdistuneiden muutosten lisäksi Nordic Cool 4 otetaan käyttöön laajemmin myös muualla, kuten esimerkiksi Visman sosiaalisen median sivuilla ja uutiskirjeissä. Tarkoituksena on, että Nordic Cool 4:n design on mukana kaikessa mikä näkyy asiakkaalle, luoden yhdenmukaisen käyttäjäkokemuksen käyttäjälle eli asiakkaalle.

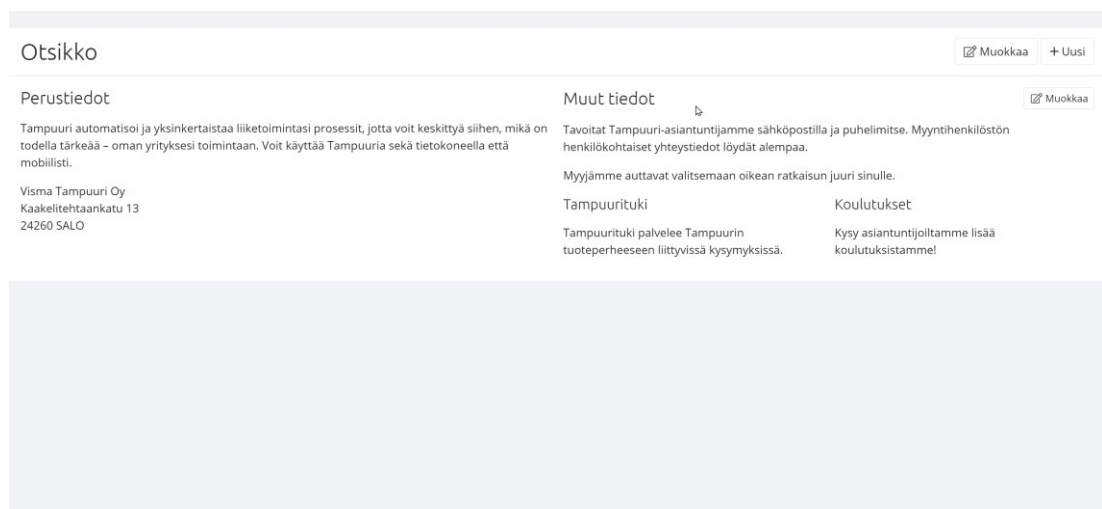
4.6 Nordic Cool 4 Kirjaston esittely

Visma Tampuurin Nordic Cool 4 komponenttikirjasto sisältää useita erilaisia web-komponentteja, joiden avulla on mahdollista luoda moderneja verkkosivuja. Kyseessä

oleva web-komponenttikirjasto on vielä keskeneräinen ja siihen tehdään jatkuvasti kehitystyötä ja uusia lisäyksiä. Kehitystyön nopeuttamiseksi ja helpottamiseksi Visma Tampuurin Nordic Cool 4 web-komponenttikirjastosta löytyvien komponenttien esimerkkejä on mahdollista selata Storybookin avulla. Tätä UI kehityksen työkalua hyödynnettäessä on mahdollista selata yksittäisiä web-komponentteja sekä tarkastella niiden toimintoja kätevästi käyttämättä aikaa ympäristön asentamista ja ylläpitoa varten.

4.6.1 Sivun päänäkymä

Tam-layout-main on Nordic Cool 4 komponenttikirjaston verkkosivun perusnäkö. Kyseistä web-komponenttia siis käytetään verkkosivuissa aina silloin, kun sivun tarpeisiin ei löydy tarkemmin räätälöityä web-komponenttia. Tam-layout-main sisältää kolme paikkaa(slot) komponenteille: Header, content ja footer. Header on sivun otsikko, mikä on yleensä navigaatiopalkin ja linkkipolun alapuolella. Content on sivun sisältö, mikä voidaan jakaa yhteen, kahteen tai kolmeen osaan. Footer toimii sivun ”alapalkkina”, eikä sitä välttämättä aina oteta käyttöön.



Kuva 23. Esimerkki tam-layout-main web-komponentista.

4.6.2 Hakusivun päänäkymä

Tam-layout-search on web-komponenttikirjaston perusnäkyminen sivuille, joissa on tarkoitus suorittaa tietyn tyyppinen haku, kuten esimerkiksi jonkin yrityksen omistaman kohteen asukkaat listaava haku. Alla oleva esimerkki sisältää web-komponenttiin liittyvät kaksi komponenttia, jotka ovat hakuehdot ja hakutulokset. Hakuehdot ovat kuvan vasemmalla reunalla ja niihin voi täydentää pudotusvalikoita sekä päivämäärän valitsimia. Hakutulokset vievät sivusta suuremman osan, listaten perustiedot, muut tiedot sekä itse hakutuloksen.

Otsikko Muokkaa + Uusi

Sukunimi

Etunimi

[Hae](#)

Perustiedot Muokkaa

Tampuuri automatisoi ja yksinkertaistaa liiketoimintasi prosessit, jotta voit keskittyä siihen, mikä on todella tärkeää - oman yrityksesi toimintaan. Voit käyttää Tampuuria sekä tietokoneella että mobiilisti.

Visma Tampuuri Oy
Kaakelitehtaankatu 13
24260 SALO

Työntekijät

#	Sukunimi	Etunimi	Palkka	Syntymäaika
4	Möttönen	Matti	2 200€	05.10.1952
6	Makkonen	Maija	3 750€	01.02.1982
8	Seppänen	Rauha	4 220€	21.11.1962
10	Rauhala	Timo	150€	19.12.2009
17	Ankka	Roope	1 245 222,5€	27.01.1951

Kuva 24. Esimerkki tam-layout-search web-komponentista.

4.6.3 Taulukot

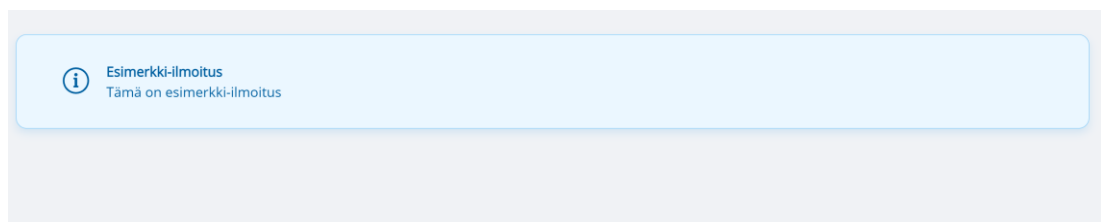
Komponenttikirjastosta löytyy kahdenlaisia taulukoita, jotka ovat ulkoasultaan samanlaisia, mutta eroavat koodin puolella toisistaan. Taulukoista voi valita joko html-pohjaisen taulukon tai array-pohjaisen taulukon. Taulukosta on mahdollista tehdä interaktiivinen, kuten alla olevassa esimerkissä näkyy linkeiksi muunnetuista id numeroista.

#	Sukunimi	Etunimi	Palkka	Syntymäaika
4	Möttönen	Matti	2 200€	05.10.1952
6	Makkonen	Maija	3 750€	01.02.1982
8	Seppänen	Rauha	4 220€	21.11.1962
10	Rauhala	Timo	150€	19.12.2009
17	Ankka	Roope	1 245 222,5€	27.01.1951

Kuva 25. Esimerkki taulukko komponentista.

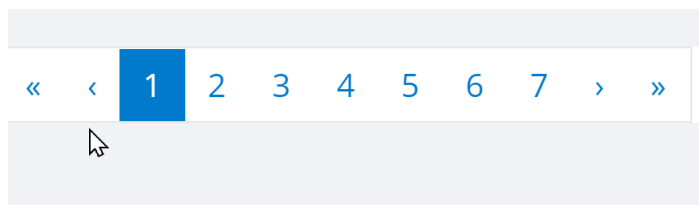
4.6.4 Ilmoitusviesti & Sivutus

Tam-alert on Nordic Cool 4 tyylillä tehty ilmoitusviesti, jonka on tarkoitus joko kadota tietyn ajan jälkeen tai pysyä näytöllä, kunnes käyttäjä kuittaa sen luetuksi.



Kuva 26. Esimerkki ilmoituksesta.

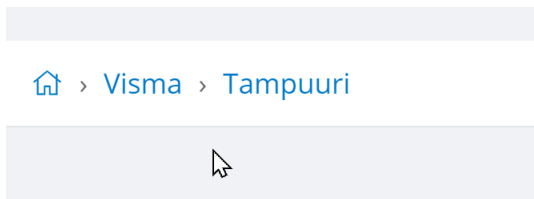
Tam-pagination on sivutuksen komponentti. Sivutuksen avulla voidaan jakaa sivulla näkyvän tiedon määrää, esimerkkinä hakutulosten sivutus, mikäli hakutuloksien määrä ylittää tietyn koodissa asetetun arvon.



Kuva 27. Esimerkki sivutus komponentista.

4.6.5 Linkkipolku

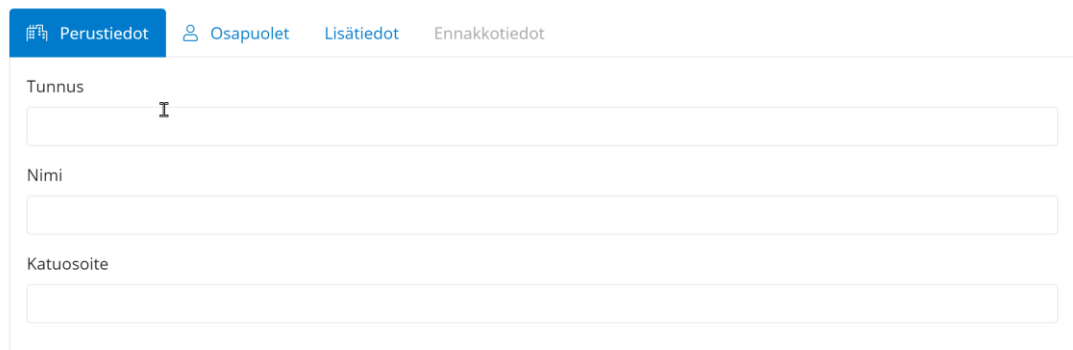
Tam-breadcrumb on Nordic Cool 4 tyylillä tehty linkkipolku. Kyseinen linkkipolku löytyy usein verkkosivun tai sovelluksen navigaatiopalkin alapuolelta ja varsinaisen sivun sisällön yläpuolelta tai osana varsinaista sisältöä. Linkkipolku kasvaa pidemmäksi sitä mukaa kun käyttäjä siirtyy jossakin tuotteessa syvemmälle tasolle. Linkkipolku auttaa käyttäjää tuotteen navigoinnissa sekä kuvaa tarkasti mitä kautta käyttäjä on päätenyt sillä hetkellä näytöllä näkyvälle sivulle.



Kuva 28. Esimerkki linkkipolusta.

4.6.6 Välilehdet

Tam-tabs on web-komponentti, millä on mahdollista lisätä käyttöliittymään tarvittaessa välilehtiä, joihin on mahdollista lisätä muita komponentteja tarpeen mukaan. Komponenttiin sisältyy myös toiminto, millä on mahdollista osoittaa käyttäjälle, mikä välilehdistä on kiireinen, esimerkiksi jonkin keskeneräisen prosessin takia. Lisäksi on mahdollista poistaa käytöstä jokin välilehti, mikäli se ei ole jossakin tiettyssä tilanteessa tarpeellinen tai mikäli kyseisen välilehden avautuminen vaatii tiettyjen ehtojen täyttymisen.

The image shows a web form with a tabbed interface. The tabs are 'Perustiedot' (selected), 'Osapuolet', 'Lisätiedot', and 'Ennakkotiedot'. Below the tabs are three input fields labeled 'Tunnus', 'Nimi', and 'Katuosoite'. The 'Tunnus' field has a small 'I' icon inside it.

Kuva 29. Esimerkki välilehti komponentista.

Tuotteen nimi

Tuotteen yksilöivä nimi

Kuva 30. Esimerkki viestilaatikon komponentista.

4.6.7 Ikonit

Nordic Cool 4 tyylillä kehitettyjä web-komponentteja varten on luotu myös ikonikirjasto, mikä sisältää useita tuotteen kannalta tarpeellisia ikoneita. Useissa web-komponenteissa on erikseen koodin puolella määritelty kohta ikoneita varten, mikä mahdollistaa niiden käytön tarpeen mukaan.

Laajan ikoni valikoiman lisäksi ikoneita löytyy myös erikokoisina. Ikoneille on kehitetty valmiiksi tuetut ikonien koot, joita käytettäessä käyttöliittymässä esiintyvät eri ikonit ovat kooltaan yhdenmukaisia. Ikonille määriteltävä koko riippuu siitä, onko se osana varsinaista sisältöä vai ei.

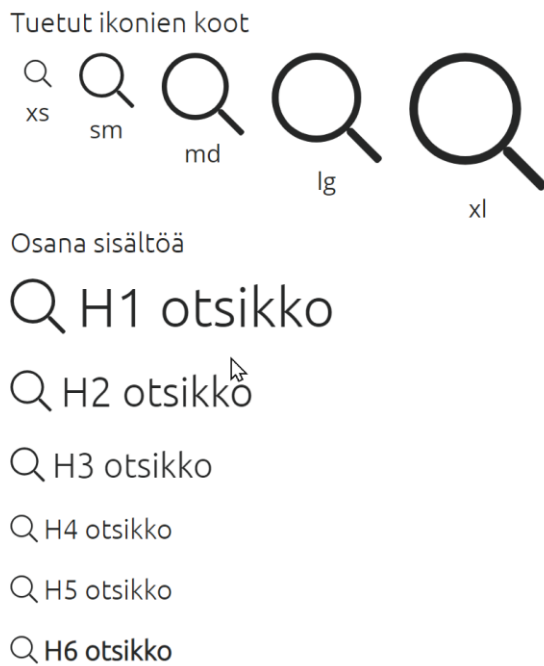
Ikonikirjasto



Kuva 31. Esimerkki 1 NC4 tyylitellyistä ikoneista.



Kuva 32. Esimerkki 2 NC4 tyylitellyistä ikoneista.



Kuva 33. Esimerkki eri kokoisista ikoneista.

5 KONKRETIA

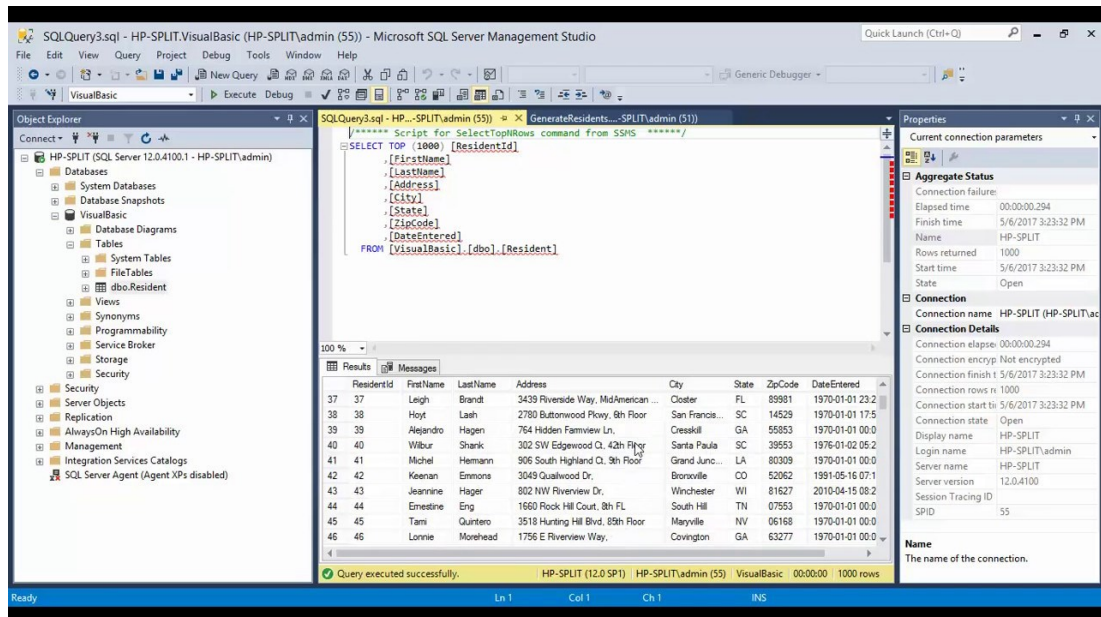
Tässä osassa opinnäytetyötä käsitellään työssä tehtyä varsinaista konkreettista kehitystä Visma Tampuurin tuotteeseen, sekä kehitystyössä käytettyjä työkaluja ja teknikoita.

5.1 Opinnäytetyössä käytettävät työkalut

Opinnäytetyötä varten valitun projektin kehitystyötä sekä testausta varten käytettiin useita eri työkaluja, perustuen Visma Tampuurin tuotteen kehitysympäristön vaatimuksiin sekä sen testaukseen liittyviin tarpeisiin. Suurin osa alla mainituista työkaluista ja teknologioista on ohjelmistokehityksen parissa opiskelevalle tai työskentelevälle entuudestaan tuttuja. Osa kyseessä olevista työkaluista ja teknologiasta ovat käyttäjälleen maksullisia, tosin monessa tapauksessa on mahdollista löytää tuotteesta ilmaiseksi käytettävää versiota. Osa näistä työkaluista ja teknologioista perustuu avoimeen lähdekoodiin, joten ne ovat käyttäjilleen ilmaisia. Työkalujen käyttöä ei tarkemmin esitellä opinnäytetyössä, sillä moni tässä opinnäytetyössä lyhyesti esitellyistä työkaluista voisi olla kokonaan oma aihe opinnäytetyötä varten.

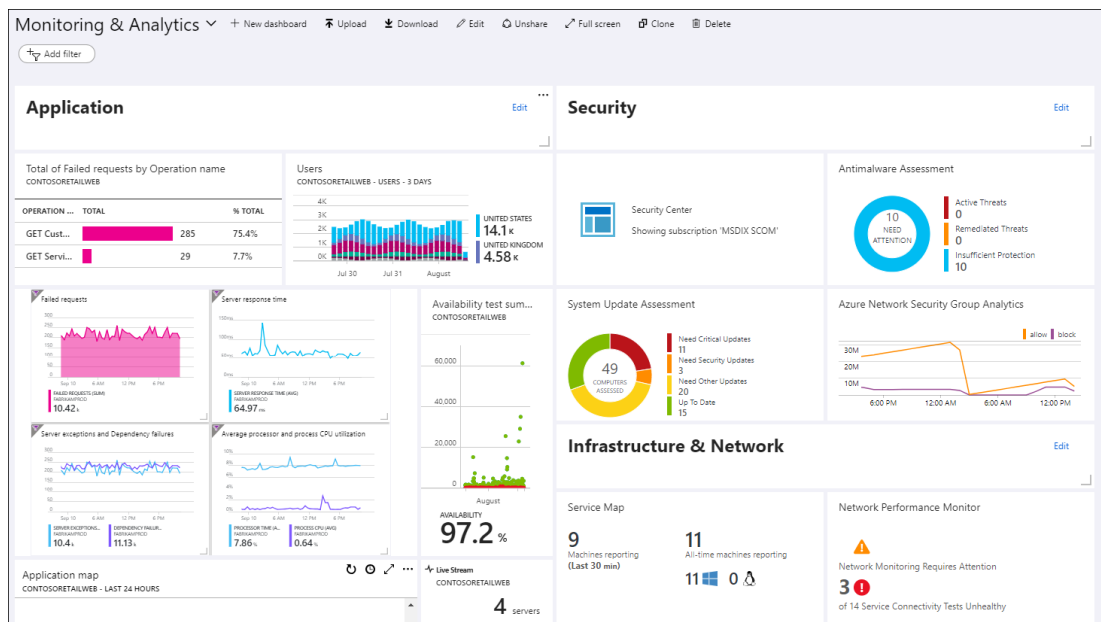
Microsoft Visual Studio 2019 toimii projektin varsinaisen koodiin vaikuttavan kehitystyön pääsijaisena alustana. Kyseisen **IDE:n** käyttöön liittyy yhteensopivuuteen liittyviä tekijöitä, sekä tarve käyttää **.NET** kehitysympäristöä.

Microsoft SQL Server Management Studion käyttäminen oli opinnäytetyön puolesta tarpeellista johtuen yksinkertaisesti siitä, että yritys, jolle opinnäytetyö tehdään, käyttää **SQL-tietokantaa** esimerkiksi käyttäjien ja käyttöäoikeuksien sekä asetusten hallitsemiseen.



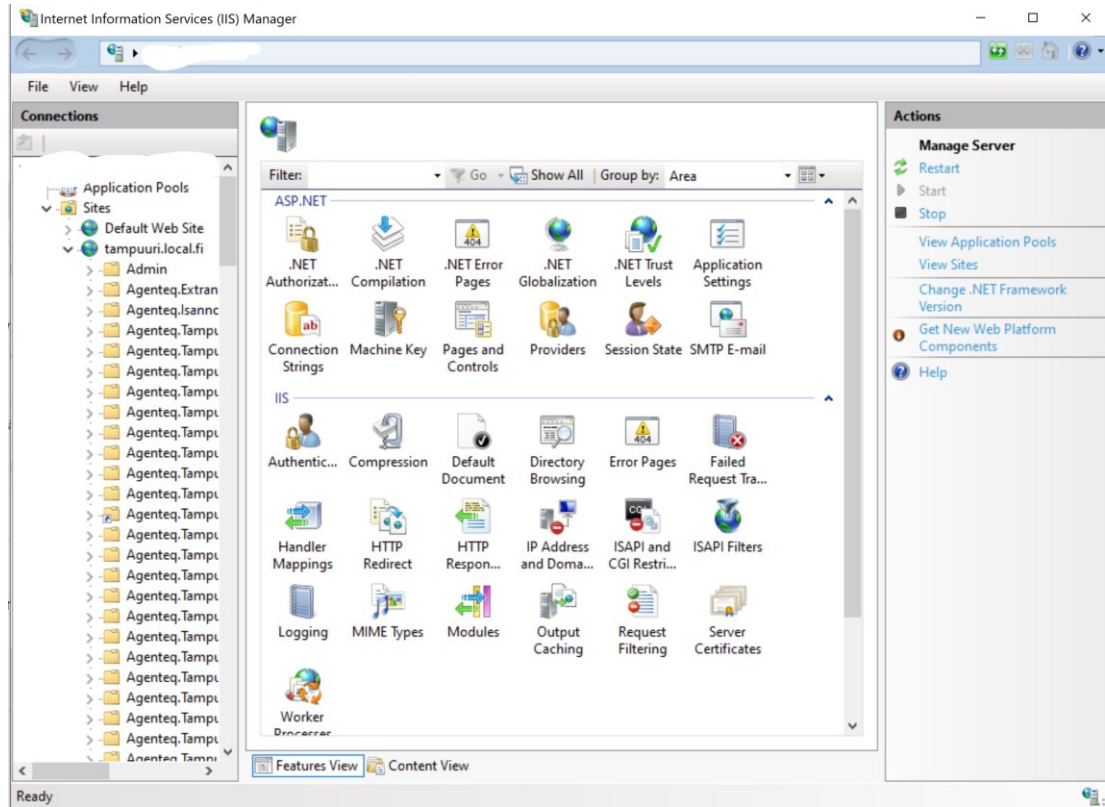
Kuva 34. SQL Server Management Studio

Opinnäytetyössä tehtävän projektin testausta varten otettiin käyttöön pilvi -teknologiaan perustuva **Microsoft Azure**. Kyseisen palvelun käyttö tuli tarpeelliseksi, koska projektissa kehitettävää ohjelmistoa oli pystyttävä testaamaan mobiililaitteilla, jotka eivät ole yhteydessä yrityksen sisäiseen verkkoon. Tämän lisäksi Microsoft Azuren avulla on mahdollista seurata pilvipalvelimella olevan sovelluksen tai verkkosivun tapahtumia ja dataa.



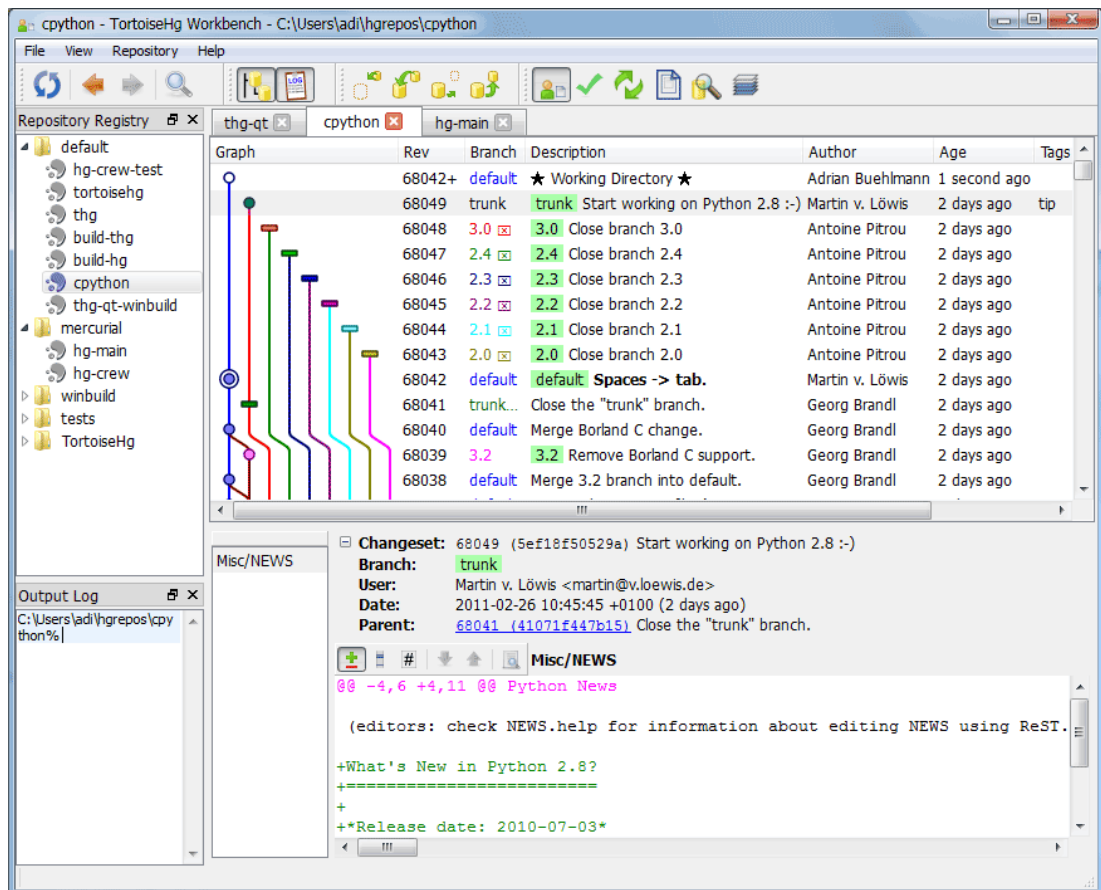
Kuva 35. Microsoft Azure

Internet Information Services (IIS) käytetään yleensä ASP.NET -pohjaisten web-sovelluksien ja staattisten verkkosivujen ylläpitämiseen. Tässä tapauksessa opinnäytetyön projektin kehitysympäristön paikallinen versio Tampuurista toimii IIS:in avulla.



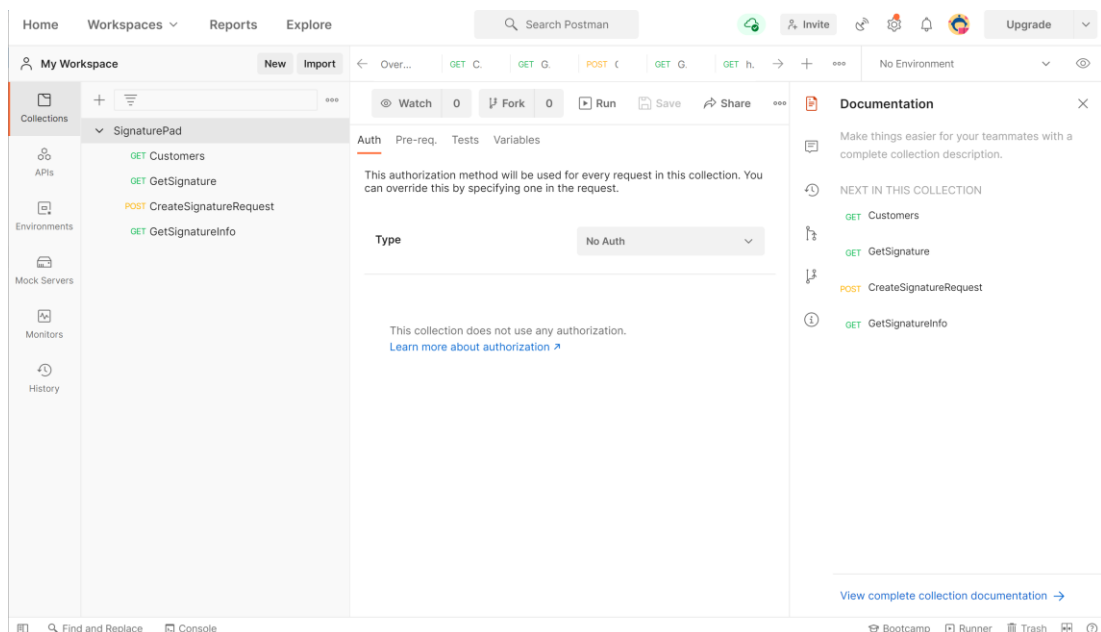
Kuva 36. IIS

TortoiseHg toimii opinnäytetyön projektin versionhallinnan työkaluna, tarjoten kätevän versionhallinnan työkalun graafisella käyttöliittymällä. Varsinainen versionhallinta on Mercurial-pohjainen.



Kuva 37. TortoiseHg graafinen käyttöliittymä.

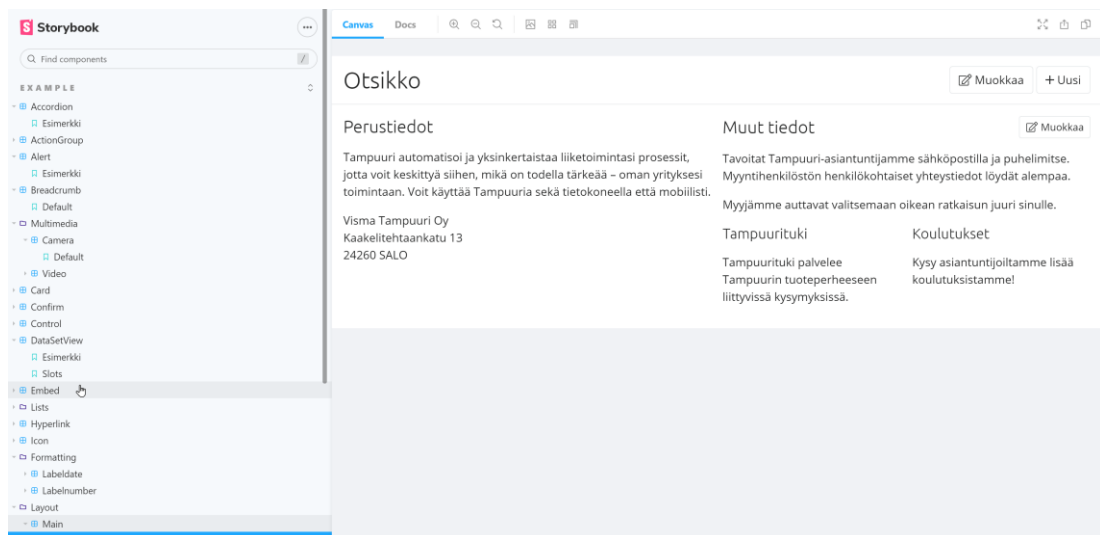
Postmanin avulla voidaan lähettää sovellukseen erilaisia pyyntöjä, jonka avulla sovelluksen toimintaa voidaan testata kätevästi ilman suurempaa säätämistä.



Kuva 38. Postman käyttöliittymä.

Node.js on ohjelmistokehityksessä käytettävä alusta, jonka avulla on mahdollista rakentaa helposti nopeita ja skaalautuvia verkkosovelluksia. Node.js on kevyt ja tehokas, tehden siitä täydellisen alustan dataintensiivisille sovelluksille. Tämän lisäksi projektissa käytetään **npm**:ää, mikä on Node.js:n vakio paketti manageri. Npm tulee tarpeelliseksi projektissa sen vaatimien riippuvuuksien takia. Riippuvuuksia on mahdollista asentaa ja hallita kätevästi npm:n avulla.

Storybook on kätevä UI:n kehityksen työkalu, minkä avulla voidaan paremmin kehittää yksittäisiä UI-komponentteja. Storybookia hyödyntämällä aikaa kuluu vähemmän UI-kehitysympäristön pystyttämiseen ja ylläpitoon, sekä mahdollistaa kätevämmän tavan selvittää UI-komponenttien toimintaa ilman sovelluksen navigointiin kuluva aikaa.



Kuva 39. Storybook

LitElement on yksinkertainen luokkapohja, minkä avulla on mahdollista luoda kevyitä web-komponentteja, mitkä toimivat millä vain verkkosivulla riippumatta Frameworkista. Lit-element tulee tässä opinnäytetyössä tarpeelliseksi, sillä se on yksi Visma Tampuurin kehittämien web-komponenttien riippuvuus.

5.2 Projektin aloittaminen

Projekti aloitetaan selvittämällä projektin ensimmäinen vaihe ja alustavat vaatimukset. Projektin aloittamista varten kävimme arkkitehdin kanssa läpi projektin kannalta tärkeitä päivityksiä ja muutoksia. Näistä ensimmäisenä esille nousi Target Frameworkin korotus uudempaan versioon. Tämän jälkeen tarpeelliseksi tulisi selvittää kuinka QR-skanneri toimii. Selvitetään myös, että onko projektiin mahdollista soveltaa web-komponenttikirjastosta löytyvää tam-camera komponenttia vai luodaanko uusi komponentti, joka on räätälöity nimenomaan QR-skannerin toimintaa varten. Lisäksi projektin kannalta on tärkeää selvittää miten TypeScript toimii ja mitä kaikkea on huomioitava sitä käyttöönottaessa.

5.2.1 Target .NET Framework 2.0 nosto 3.1 versioon

Talokeskus.Microservice.SignaturePad -projektin Target .NET Framework nostettiin versioon 3.1. Tämä tehtiin perustuen NC4 ja web -komponentti pohjaisten kirjastojen vaatimuksiin. Lisäksi projektista poistettiin nugetti Microsoft.AspNetCore.All(2.0.5). Target .NET Frameworkin muuttamisen jälkeen projektissa nousi esille ongelmia middlewaren kanssa. Ongelmat liittyivät lähinnä mvc:hen ja actionContextiin. Projektissa otettiin käyttöön endpointit, sekä muutettiin actionContext httpContextiksi, joka korjasi Target .NET Frameworkin noston yhteydessä nousseet ongelmat. Frameworkin nosto oli tarpeellinen, koska se mahdollisti web-komponenttien käytön projektissa. Noston yhteydessä ilmaantui haasteita, koska version noston yhteydessä middlewareen käsiksi pääsyä oli rajoitettu. Rajoitusten takia jouduimme muuttamaan muun muassa IActionContextAccessor rajapinnan tilalle IHttpContextAccessorin, jonka avulla pääsee käsiksi middlewaren tietoihin (Kuva 40). Näistä Microsoftin version välisten muutosten takia oli muutettava käyttöön endpointit, jotka mahdollistivat projektin kääntymisen jälleen (Kuva 41).

```

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc.Infrastructure;
using Microsoft.AspNetCore.Routing;
using System;
using System.Collections.Generic;
using System.Linq;
// -11,19 -12,19 //
{
    public sealed class SignatureRequestFactory
    {
        - public SignatureRequestFactory( IActionContextAccessor actionContextAccessor, ISignatureRequestReader signatureRequestReader, IHttpContextAccessor httpContextAccessor, ISignatureRequestReader signatureRequestReader )
        +
        {
            this.ActionContextAccessor = actionContextAccessor;
            this.httpContextAccessor = httpContextAccessor;
            this.SignatureRequestReader = signatureRequestReader;
        }

        public SignatureRequest Create()
        {
            - var result = SignatureRequestReader.Read( this.ActionContextAccessor.ActionContext.RouteData.Values );
            + var result = SignatureRequestReader.Read( this.httpContextAccessor.HttpContext.GetRouteData().Values );
            return result;
        }

        - private readonly IActionContextAccessor ActionContextAccessor;
        + private readonly IHttpContextAccessor httpContextAccessor;
        private readonly ISignatureRequestReader SignatureRequestReader;
    }
}

```

Kuva 40. Tässä nähdään muutokset middlewarea varten.

```

using Microsoft.AspNetCore.Hosting;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Talokeskus.Microservice.SignaturePad.Repositories.Models;
namespace Talokeskus.Microservice.SignaturePad
// -25,7 -30,7 //
{
    // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
    - public void Configure(IApplicationBuilder app, IHostingEnvironment env)
    + public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
    {
        if (env.IsDevelopment())
        {
            // -42,12 -43,12 //
        }

        app.UseStaticFiles();
        - app.UseMvc(routes =>
        + app.UseRouting();
        + app.UseEndpoints( endpoints =>
        {
            routes.MapRoute(
            endpoints.MapControllerRoute(
            name: "default",
            template: "{controller=Sign}/{action=Index}/{id?}");
            - pattern: "{controller=Sign}/{action=Index}/{id?}");
        });
    }
}

```

Kuva 41. Kuvasta näkee käyttöön otetut endpointit.

5.2.2 QR-Skannerin päivitys uusimpaan versioon

Allekirjoituslupastassa käytössä olevalle QR-skannerille oli saatavilla jonkin verran päivityksiä, sillä viimeksi kyseiseen projektiin oli tehty kehitystä vuonna 2018. Kyseinen toiminto pohjautuu GitHubista löytyvään lähdekoodiin ja sitä kautta tuotiin tarpeelliset päivitykset projektiin. Toimimme ainoastaan ominaisuuksia, jotka koimme tarpeellisiksi.

Uusia päivityksiä, jotka koettiin tarpeelliseksi, oli saatavilla QR-skannerin JavaScript tiedostoon sekä QR-skannerin dekooderi JavaScript tiedostoon. Tämän lisäksi päivityksiä tuli kohtiin, joissa käytössä oleva laite on jokin muu kuin Android-käyttöjärjestelmän omaava. Tämä päivitys parantaa vikasietoisuutta ja lisää variaatiota käyttäen QR-koodilukijaa esimerkiksi iOS-laitteilla (Kuva 42).

```

+       window.isMediaStreamAPISupported = navigator && navigator.mediaDevices && 'enumerateDevices' in navigator.mediaDevices;
+       window.noCameraPermission = false;

       var scanningEle = document.querySelector('.custom-scanner');
       var videoElement = document.querySelector('video');
       window.appOverlay = document.querySelector('.app__overlay');

       $("#scan-btn").on("click", function () {
-         QRReader.init();
-         if (!window.iOS) {
-           $("#scan-button-container").hide();
-           $("#scanner-container").show();
-           scan();
-         }
+         setTimeout(() => {
+           QRReader.init();
+           if (window.isMediaStreamAPISupported) {
+             $("#scan-button-container").hide();
+             $("#scanner-container").show();
+             scan();
+           }
        }
    }

```

Kuva 42. Uutena ominaisuutena isMediaStreamAPISupported.

5.2.3 TypeScriptin käyttöönotto

Visma Tampuurilla ollaan ottamassa käyttöön Microsoftin kehittämää TypeScriptia. TypeScript on moderni supersetti JavaScriptille. Sen suurin hyöty saadaan kielen tuomista tyypeistä, joilla voidaan käännoaikaisesti ennaltaehkäistä virhetilanteita, joita tavallisesti JavaScript ei tunnistaisi ajonaikaisesti. Otimme TypeScriptin tähän projektiin mukaan pilotoivana esimerkkinä ja todisteena sen vahvuuksista. Kirjoitimme qrscan JavaScript-tiedoston uudestaan TypeScript-kielellä. Lopuksi poistimme alkuperäisen JavaScript tiedoston, koska sille ei ollut enää käyttöä.

TypeScriptin käyttöönottaminen vaatii projektilta tiettyjä toimenpiteitä. Ensimmäiseksi aloitettiin asentamalla Visual Studioon kautta Microsoftin tarjoama TypeScript NuGet-package (Kuva 43). Tämän mukana asentui automaattisesti TypeScriptin vaatima compiler, joka kääntää TypeScript koodia JavaScript muotoon.

```

+ <ItemGroup>
+ <PackageReference Include="Microsoft.TypeScript.MSBuild" Version="4.2.2">
+ <PrivateAssets>all</PrivateAssets>
+ <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>
+ </PackageReference>
+ <PackageReference Include="Microsoft.VisualStudio.Web.BrowserLink" Version="2.2.0" />
+ <PackageReference Include="Newtonsoft.Json" Version="11.0.2" />
+ <PackageReference Include="WindowsAzure.Storage" Version="9.1.0" />
+ @ -23,4 +31,8 @
+ <None Include="wwwroot\css\site.css" />
+ </ItemGroup>
+ <ItemGroup>
+ <TypeScriptCompile Include="wwwroot\js\qrscan2.ts" />
+ </ItemGroup>
+

```

Kuva 43. Projektitiedostosta otettu kuvankaappaus, jossa näkyy uusi PackageReference eli NuGet-package.

Projektin valmistelemisen jälkeen aloitettiin JavaScript-tiedoston kääntäminen TypeScript-kielelle. Kävimme koko tiedoston läpi, jotta pystyimme hyödyntämään mahdollisimman paljon TypeScriptin tuomia etuja. Kuvassa näkyy eroavaisuus alkuperäiseen JavaScript (Kuva 44) tiedostoon ja TypeScriptin (Kuva 45) tuomat tyypitykset.

```
3 var QRReader = {};  
4  
5 QRReader.active = false;  
6 QRReader.webcam = null;  
7 QRReader.canvas = null;  
8 QRReader.ctx = null;  
9 QRReader.decoder = null;  
10
```

Kuva 44. Vanha versio JavaScript kielen muodossa.

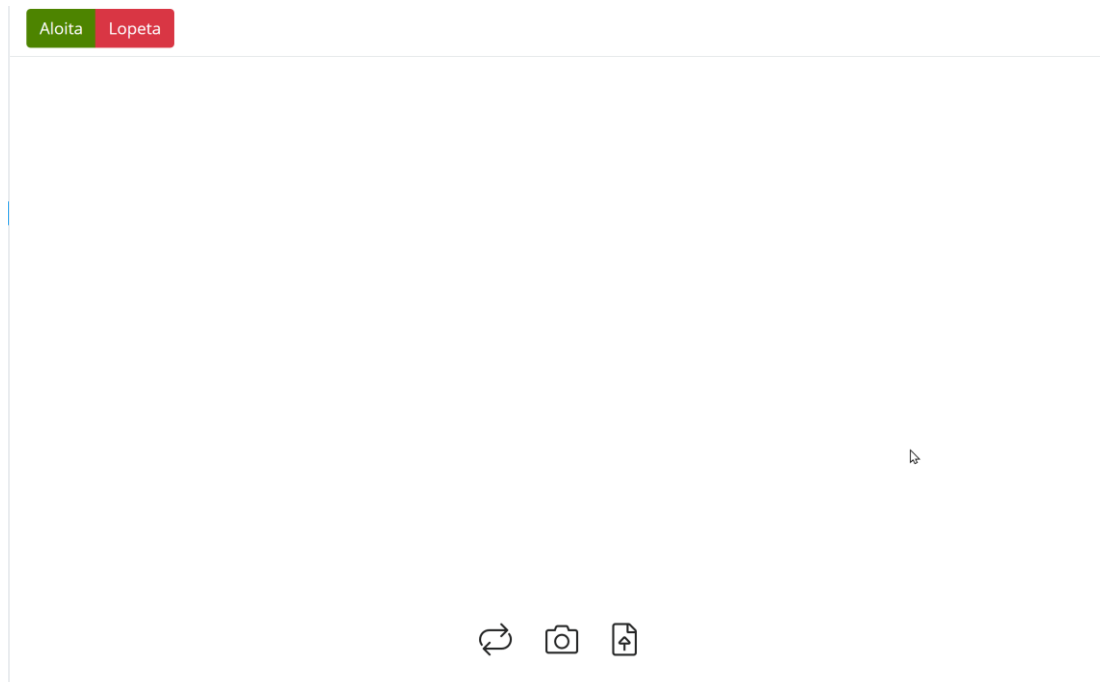
```
1 declare global {  
2   interface Window { isMediaStreamAPISupported: boolean; iOS: boolean }  
3 };  
4  
5 export const QRReader: {  
6   active: boolean;  
7   webcam: any;  
8   canvas: HTMLCanvasElement;  
9   ctx: CanvasRenderingContext2D;  
10  decoder: Worker;  
11  setCanvas: any;  
12  init: any;  
13  scan: any;  
14 } = {  
15   active: false,  
16   webcam: null,  
17   canvas: null,  
18   ctx: null,  
19   decoder: null,  
20   setCanvas: function () {},  
21   init: function () {},  
22   scan: function () {},  
23 }  
24
```

Kuva 45. Uudempi ja tyyppitetty QRReader-olio TypeScript-kielellä.

5.3 Selvitys NC4 komponenttien käyttöönotosta

TypeScriptin onnistuneen käyttöönoton jälkeen tarkoituksena oli selvittää, olisiko mahdollista korjata avaintenhallinnan allekirjoitusalueen kameraan liittyvä ongelma ottamalla käyttöön NC4 komponenttikirjastosta löytyvä kamerakomponentti. Kyseisellä kamerakomponentilla on toiminto, minkä avulla pystyy tarvittaessa vaihtamaan mobiililaitteen käyttämää kameraa etukamerasta takakameraksi yhtä nappia painamalla.

Allekirjoitushallinnan kameran korjaamiseksi selvitimme, olisiko mahdollista soveltaa NC4 komponenttikirjastosta löytyvää kamerakomponenttia nimeltä **tam-camera**.



Kuva 46. Kamerakomponentin näkymä.

Kamerakomponentin vasemmassa yläkulmassa sijaitsevat ”Aloita” ja ”Lopeta” napit. Tässä projektissa niitä ei välttämättä tarvita, sillä sivulla, mistä QR-koodin skannaava kamera avautuu, on jo nappi kameran avaamista varten. Varsinainen hyöty kamerakomponentin käytöstä tulisi keskellä alhaalla olevista napeista, tarkemmin vasemmanpuoleisesta napista, missä kaksi nuolta muodostaa ovaalin. Tämän napin avulla käyttäjän olisi mahdollista itse päättää, kumpaa kameraa käyttää QR-koodin skannausta varten.

Jotta kyseessä oleva uusi komponentti saataisiin toimimaan, oli sille tuotava tarvittavat moduulit ja riippuvuudet. Jotta nämä kaikki moduulit toimivat keskenään web-sovelluksessa, on niitä varten ladattava bundler. Bundler ”pakkaa” moduulit yhdeksi ”paketiksi”. Nämä riippuvuudet ovat seuraavat:

- Tam-htmlElement
- Lit-html
- Tam-camerapictures
- Tam-video
- Parsing (työkalu)
- Tam-actionstack

Yllä mainitut tuotiin projektiin importin avulla. Näitä varten oli myös tuotava niiden omia importteja. **Tam-htmlElement** vaatii seuraavat:

- Throttle (työkalu/tools)
- Element (työkalu/tools)
- Nanoid

Tam-camerapictures vaatimukset:

- Tam-htmlElement
- Lit-html
- Tam-actionselect
- Tam-actionstack

Tam-video vaatimukset:

- Tam-htmlElement
- Lit-html
- Videoadapter (työkalu)
- Parsing (työkalu)
- Tam-icon

Tässä projektissa arkkitehdin suositukseen perustuen jätimme NC4 komponenttikirjaston moduulit käyttämättä, johtuen yhteensopivuuteen liittyvistä ongelmista QR-skannerin kanssa.

5.4 Web-komponenttien luominen ja käyttöönotto

Koska web-komponenttikirjastosta löytyvää tam-camera -komponenttia ei voitu hyödyntää käytettävyysongelman ratkaisemiseksi QR-skannerin yhteensopivuuteen liittyvistä ongelmista johtuen, oli ongelman ratkaisemiseksi luotava uusi tarpeen mukaan räätälöity komponentti. Jotta tätä ongelman ratkaisua varten räätälöity komponentti olisi teknisesti samanlainen web-komponenttikirjastossa olevien komponenttien kanssa on sitä varten projektissa otettava käyttöön rollup-pakkaus työkalu ja LitElement, joka toimii alustavana luokkana (Base class) web-komponenteille. Lisäksi qrscan-komponentin toiminnallisuudessa ilmeni joitakin ongelmia eri selainten kanssa, joten komponentti oli korjattava.

Web-komponenttien luominen ja käyttöönotto on oikeiden työkalujen kanssa yksinkertaista. Periaatteessa mikä vain JavaScript tai TypeScript tiedosto voidaan muuttaa web-komponentiksi ja tässä projektissa qrscan-nimisessä tiedostossa otettiin käyttöön LitElement.

5.4.1 Kamerakomponentin korjaaminen

Ensimmäisessä kuvassa näkyy (Kuva 7) alkuperäinen versio, jossa ei ole otettu kantaa takakameran käyttöönottamiseen. Toisessa kuvassa (Kuva 8) näkyvät muutokset, jotka vaikuttivat kameran käyttäytymiseen. Kuvista voidaan päätellä koodin pituuden osalta, että siitä on karsittu runsaasti turhaa pois. Tärkein huomio keskittyy video-ominaisuuden ympärille, jonka sisällä tapahtuvaa logiikan käyttäytymistä muutettiin radikaalilla tavalla. Alkuperäisessä versiossa käytetään mandatory-nimellä määriteltyä ominaisuutta, joka pakottaa mahdollisesti löytyvän kameran deviceId:n käytettäväksi ensisijaisena vaihtoehtona. Uudessa toteutuksessa etsitään laitteesta mahdollisesti löytyvä takakamera ja käytetään sitä ensisijaisesti. Takakameran puuttuessa käytetään alkuperäisen mandatory-ominaisuuden logiikkaa hyväksi ja näin varmistetaan toiminta muillakin kuin mobiililaitteilla. Tehtyjen muutosten avulla takakamera toimii nyt myös iOS-laitteilla.

```

69 if (window.isMediaStreamAPISupported) {
70   navigator.mediaDevices
71     .enumerateDevices()
72     .then(function(devices) {
73       var device = devices.filter(function(device) {
74         var deviceLabel = device.label.split(',')[1];
75         if (device.kind === 'videoinput') {
76           return device;
77         }
78       });
79     });
80   var constraints;
81   if (device.length > 1) {
82     constraints = {
83       video: {
84         mandatory: {
85           sourceId: device[device.length - 1].deviceId ? device[device.length - 1].deviceId : null
86         }
87       },
88       audio: false
89     };
90   }
91   if (window.iOS) {
92     constraints.video.facingMode = 'environment';
93   }
94   startCapture(constraints);
95 }
96
97 else if (device.length) {
98   constraints = {
99     video: {
100      mandatory: {
101        sourceId: device[0].deviceId ? device[0].deviceId : null
102      }
103    },
104    audio: false
105  };
106  if (window.iOS) {
107    constraints.video.facingMode = 'environment';
108  }
109  if (!constraints.video.mandatory.sourceId && !window.iOS) {
110    startCapture({ video: true });
111  }
112  else {
113    startCapture(constraints);
114  }
115 }
116
117 else {
118   startCapture({ video: true });
119 }
120
121
122
123

```

Kuva 47. GitHubista otettu vanha JavaScript versio qrscan-tiedostosta.

```

85
86 if (window.isMediaStreamAPISupported) {
87   navigator.mediaDevices
88     .enumerateDevices()
89     .then(function (devices: MediaDeviceInfo[]): void {
90       const device: MediaDeviceInfo[] = devices.filter(function (device: MediaDeviceInfo) {
91         if (device.kind === 'videoinput') {
92           return device;
93         }
94       });
95     });
96   let constraints;
97   if (device.length > 1) {
98     constraints = {
99       video: {
100        facingMode: "environment"
101      },
102      audio: false
103    };
104   }
105   else if (device.length) {
106     constraints = {
107       video: {
108        facingMode: "environment"
109      },
110      audio: false
111    };
112   }
113   if (!constraints.video.facingMode) {
114     startCapture({ video: device[0].deviceId ? device[0].deviceId : null, audio: false });
115   }
116   else {
117     startCapture(constraints);
118   }
119 }
120
121
122
123
124

```

Kuva 48. Viimeisin versio ja siistimpi toteutus TypeScriptillä samasta tiedostosta.

Kuvassa sinisellä maalatut alueet painottavat tärkeitä kohtia koodissa.

5.4.2 LitElement

LitElement toimii web-komponenttien alustavana luokkana, mahdollistaen niiden helpon luomisen, käyttöönoton sekä jakamisen. LitElement toimii JavaScriptin ja TypeScriptin kanssa, sekä eri Frameworkien kuten Reactin tai Vuen kanssa.

```
import { LitElement, html, property, customElement } from 'lit-element';

@customElement('simple-greeting')
export class SimpleGreeting extends LitElement {
  @property() name = 'World';

  render() {
    return html`<p>Hello, ${this.name}!</p>`;
  }
}
```

```
<simple-greeting name="Everyone"></simple-greeting>
```

Kuva 49. Esimerkki LitElementin käytöstä.

Projektissa luodun LitElementin luokan sisältä löytyy suurin osa sivun Front-Endin koodista. Mikäli tiedoston ulkopuolelta tuodaan moduuleja tai komponentteja, on ne tuotava tiedostoon importin avulla. Tässä projektissa main-tiedostoon tuotiin importin avulla qrscan-niminen komponentti.

```
1 import { html, LitElement, TemplateResult } from 'lit-element';
2 import { QRScan } from './components/component-qrscan';
3 import './components/component-qrscan';
```

Kuva 50. Esimerkki LitElementin moduulien ja komponenttien tuomisesta.

```

export class MainPage extends LitElement {
  static get properties() {
    return {
      visibleButtons: {
        type: Boolean
      }
    };
  }
  constructor() {
    super();
    if ('serviceWorker' in navigator) {
      navigator.serviceWorker.register("./sw.js");
    }
  }
  protected createRenderRoot(): Element {
    return this;
  }
  protected render(): TemplateResult {
    return html` ${this.scanner}
    <div id="scan-button-container" class="${this.visibleButtons ? 'd-block' : 'd-none'}">
      <button class="btn btn-success" id="scan-btn" @click="${this.start.bind(this)}>Käynnistä kamera</button>
    </div> `;
  }
}

```

Kuva 51. Esimerkki LitElementiä sisältävästä tiedostosta.

```

private start(): void {
  this.scanner.start();
  this.visibleButtons = false;
}

```

Kuva 52. LitElementin luokan sisällä oleva qr-skannerin käynnistys.

5.4.3 Rollup käyttöönotto

Rollup otetaan projektissa käyttöön käyttämällä npm install-komentoa komentorivillä projektin juuressa. Lisäksi komentoon on lisättävä ”-D”, jotta se asennetaan oikeiden riippuvuuksien eli ”devDependencies” alle projektin package.json tiedostossa.

```

C:\Reput\Talokeskus.Microservice.SignaturePad\Talokeskus.Microservice.SignaturePad>npm install rollup -D

```

Kuva 53. rollup asennuksen komento

Lisäksi projektiin asennetaan rollupille kaksi pluginia npm install-komennolla, muistaen lisätä ”-D” jotta ne menevät oikean riippuvuuden alle package.json tiedostossa. Lisäksi package.json tiedostoon lisätään ”build”: ”rollup - -config”.

```

1  {
2    "name": "myproject",
3    "version": "1.0.0",
4    "scripts": {
5      "test": "echo \"Error: no test specified\" && exit 1",
6      "build": "rollup --config"
7    },
8    "devDependencies": {
9      "@rollup/plugin-commonjs": "^17.1.0",
10     "@rollup/plugin-node-resolve": "^11.2.0",
11     "rollup": "^2.41.0"
12   },
13   "dependencies": {
14     "bootstrap": "4.0.0-beta.3",
15     "font-awesome": "^4.7.0",
16     "lit-element": "^2.4.0",
17     "signature_pad": "^2.3.2",
18     "workbox-sw": "^3.0.1"
19   }
20 }
21

```

Kuva 54. package.json tiedosto.

Rollup.config.js tiedosto on myös täytettävä oikein, jotta rollup toimii. Huomaa kaksi kappaletta plugin-moduuleja, jotka on tuotu importin avulla.

```

1  import nodeResolve from '@rollup/plugin-node-resolve';
2  import commonjs from '@rollup/plugin-commonjs';
3
4  export default {
5    input: 'src/lib/main.js',
6    output: {
7      file: 'wwwroot/js/app.js'
8    },
9    plugins: [nodeResolve(), commonjs()]
10 };

```

Kuva 55. rollup.config.js tiedosto.

Rollupin käyttöönoton jälkeen projektin toimivuutta testattaessa IDE:n avulla on muistettava koodin kääntämisen jälkeen kirjoittaa komentoriville projektin juureen komento ”npm run build”.

```
C:\Reput\Talokeskus.Microservice.SignaturePad\Talokeskus.Microservice.SignaturePad>npm run build
> myproject@1.0.0 build C:\Reput\Talokeskus.Microservice.SignaturePad\Talokeskus.Microservice.SignaturePad
> rollup --config

src/lib/main.js → wwwroot/js/app.js...
created wwwroot/js/app.js in 423ms

C:\Reput\Talokeskus.Microservice.SignaturePad\Talokeskus.Microservice.SignaturePad>
```

Kuva 56. Onnistuneesti läpi mennyt ”npm run build”.

6 PROJEKTIN LOPPUTULOS JA POHDINTA

Web-komponentteja hyödyntäen projektin lopputulokseksi saatiin käytettävyydeltään parempi kameraominaisuus QR-koodin skannausta varten Visma Tampuurin tuotteeseen. Tuotteen kehityksen alustavan testaamisen jälkeen projekti siirtyy kehityksestä seuraavaan prosessiin, tässä tapauksessa ohjelmistoarkkitehdin tarkastettavaksi. Tämän jälkeen siirretään kehitykset tarkemman testauksen pariin.

Alustavien testien ja pilotoinnin perusteella tuotteeseen tehdyt muutokset on todettu hyväiksi ja täten hyväksytyt. Hyvien ominaisuuksien ansiosta kehityksemme sisältyvät tulevaisuudessa Visma Tampuurin tuotteeseen tehtävään julkaisuun.

Tämän opinnäytetyön jälkeen Visma Tampuurilla on siirrytty Nordic Cool 4 tyylin mukaan suunnitellun komponenttikirjaston käyttöönottoon. Tuleva kehitys tulee olemaan pääosin puhtaasti uutta kehitystä web-komponenttien kanssa, käyttäen myös uutta Nordic Cool 4 tyyliä kyseisissä komponenteissa. Toisin kuin opinnäytetyössä, jossa uudistimme olemassa olevaa koodia toimivaksi komponentiksi.

6.1.1 Matiaksen pohdinta

Uuden teknologian käyttöönotto ja käytettävyyden parantaminen Visma Tampuurin tuotteeseen oli mielestäni haastavaa ja mielenkiintoista. Erityisesti uuden teknologian opiskelu ja sen soveltaminen avaintenhallinnan allekirjoitusalueesta löytyvään toimintoon oli haastavaa, mutta kokemuksena arvokasta. Opinnäytetyön vaatiman työn vähentyessä olenkin siirtynyt työskentelemään tehtävien parissa, joissa opinnäytetyön tekemisen aikana opittuja asioita on päässyt laajemmin soveltamaan.

Miellyttävä käyttöliittymä ja käytettävyys on viime vuosien aikana noussut jatkuvasti tärkeämmäksi osaksi ohjelmistojen, sovellusten ja verkkosivujen suunnittelua ja tässä opinnäytetyössä tehty kehitys Visma Tampuurin tuotteen käytettävyyteen oli erinomainen esimerkki käytettävyyden merkityksestä. Lisäksi tuotteen tekninen päivittäminen modernilla teknologialla toimii ratkaisuna esimerkiksi vähentämällä teknistä velkaa, mikä on usein laajoja kokonaisuuksia sisältävien teknologia-alan tuotteiden ongelma.

6.1.2 Kasperin pohdinta

Asiakkaiden tarpeisiin vastaaminen pilotoimalla on ohjelmistoalalla varteenotettava näkökulma esimerkiksi sen yhteydessä esiintyvien haasteiden selvittämisen sekä käyttöön perustuvien kokemusten kautta. Tämä opinnäytetyö toimii hienona esimerkkinä uusien teknologioiden käyttöönottamisesta. Pilotoinnin tarkoituksena oli saada mahdollisimman realistinen kokemus web-komponenteilla rakennettavasta ohjelmistoratkaisusta. Kokonaan uudestaan rakentamamme web-komponentti on käyttöönotettavissa myös yrityksen sisäisen hankkeen edistämiseksi, jossa keskitytään muun muassa käyttökokemuksen parantamiseen.

Kameraan kohdistuvat ohjelmointiasiat tulivat täysin uutena kokonaisuutena ja opettivat valtavasti tulevaisuudessa tehtäviä ohjelmistokehityksiä varten. Tästä tekemästä työstä hyötyy varmasti yritys niin taloudellisesti kuin teoreettisestikin. Yrityksessä tullaan taatusti jatkossakin tekemään samankaltaisia ponnistuksia käyttöliittymän parantamiseksi.

LÄHTEET

Collapickin www-sivut 2020. Viitattu 1.12.2020.

<https://www.collapick.com>

Creative Commonnin www-sivut 2021. Viitattu 19.01.2020.

<https://creativecommons.org/>

Crothers, B. 2011 Storyboarding & UX - part 1: an introduction. Viitattu 22.12.2020.

<http://johnnyholland.org/>

Curtis Designs Website Development. 16.1.2021. <https://www.ecurtisdesigns.com/web-layout-design/>

Dam, R. & Siang, T. 2020. Personas - A Simple Introduction.

Viitattu 27.12.2020.

<https://www.interaction-design.org/>

Deanna deBara 2019. 'Design 101: The 8 graphic design basics you need to know'.

99designs. 2019. Viitattu 11.1.2021. <https://99designs.com/blog/tips/graphic-design-basics/>

Editorial Team UX Booth 2018. Complete Beginner's Guide to UX Research.

Viitattu 4.12.2020.

<https://www.uxbooth.com/>

Figma. Viitattu 14.1.2021. <https://www.figma.com/>

Kosola, M. n.d. CTA suomeksi ja viisi käytännön vinkkiä. Viitattu 16.01.2021.

<https://www.wtfdesign.fi/blogi/cta-suomeksi-ja-viisi-kaytannon-vinkkia>

Morovián, L. UX Storyboard Creation: A Complete Guide For Beginners.

Viitattu 19.01.2021.

<https://uxstudioteam.com/ux-blog/ux-storyboard/>

Olander, I. 2015. Käyttökokemus ratkaisee verkossa. Viitattu 17.01.2021.

<https://sometek.fi/kayttokokemus-ratkaisee-verkossa/>

Parveen, Y. 2019. Wireframes in UX Design — What, Why, When and How? Viitattu 20.12.2020.

<https://blog.prototypr.io/>

Sali, S. 2020. The What, Why, & When of Storyboarding in UX Design.

Viitattu 17.01.2021.

<https://medium.com/thinking-design/the-what-why-when-of-storyboarding-in-ux-design-38db2f955e23>

Tapio, K. n.d. Parempi käyttökokemus-onnistunut tuote. Viitattu 17.01.2021.

<https://www.hoyry.net/blogi/parempi-kayttokokemus-onnistunut-tuote/>

Tracking Widget. UX Planet. Viitattu 16.1.2021. <https://uxplanet.org/golden-ratio-bring-balance-in-ui-design-765c954f0ff9>

Tromm, T. 2020. User Testing: Definition, How To, Tools and Methods. Viitattu 4.12.2020.

<https://userpeek.com/blog/user-experience-testing/>

Tubik Studio. 2018. Golden Ratio. Bring balance in UI design. UX Planet. 9.1.2019. Viitattu 16.1.2021. <https://uxplanet.org/golden-ratio-bring-balance-in-ui-design-765c954f0ff9>

Valagroupin www-sivut 2018. Miten ymmärtää käytettävyydestestauksen tärkeys, ja saada itse testaus tehokkaaksi? Viitattu 16.01.2021.

<https://www.valagroup.com/fi/2018/04/miten-ymmartaa-kaytettavyystestauksen-tarkeys-ja-saada-itse-testaus-tehokkaaksi/>

Usability.gov. Viitattu. 11.1.2021. <https://www.usability.gov/what-and-why/visual-design.html>

Usability.gov. Viitattu. 15.1.2021. <https://www.usability.gov/how-to-and-tools/methods/parallel-design.html>

Usability.gov. Viitattu. 16.1.2021. <https://www.usability.gov/how-to-and-tools/methods/user-interface-elements.html>

Uxpin www-sivut 2020. Viitattu 1.12.2020.

<https://www.uxpin.com/>

Ville Tervo 2016. 'Visual design basics'. Futurice. 7.3.2016. Viitattu 11.1.2021.

<https://www.futurice.com/blog/visual-design-basics/>

Virtanen, J. 2020. UX-design ja UI-design: Mitä eroa niillä on? Viitattu 1.12.2020.

<https://contrast.fi/ux-design-ja-ui-design-mita-eroa-niilla-on/>

White, C. 2021. What Does A UX Designer Actually Do? Viitattu 17.01.2021.

<https://careerfoundry.com/en/blog/ux-design/what-does-a-ux-designer-actually-do/#2-what-does-a-ux-designer-do>

White, J. 2017. 'Visual Design 101'. Medium 10.11.2017. Viitattu 11.1.2021.

<https://www.medium.com/>

Kuva 3. Uixninja. Mobile App – Crypto Wallet. Viitattu 16.1.2021.

<https://dribbble.com/uixninja>

Kuva 8. Tracking Widget. UX Planet. Viitattu 16.1.2021. <https://uxplanet.org/golden-ratio-bring-balance-in-ui-design-765c954f0ff9>

Kuva 9. Yvette's. Viitattu 16.1.2021. <https://yvettesbridalformal.p1r8.net/>

Kuva 21. Esimerkki Nordic Cool 3 käyttöliittymästä. Viitattu 24.4.2021

<https://www.visma.com/brandbook/nordic-cool-4/>

Kuva 22. Esimerkki Nordic Cool 4 käyttöliittymästä. Viitattu 24.4.2021

<https://www.visma.com/brandbook/nordic-cool-4/>

Kuva 37. Tortoise Hg käyttöliittymä. Viitattu 24.4.2021 <https://tortoisehg.bitbucket.io/>