



Haaga-Helia
ammattikorkeakoulu Oy

Moniosaajan portfolio

Kristiina Lemmetyinen

Opinnäytetyö

Tietojenkäsittelyn koulutusoh-
jelma

2020



Tekijä(t) Kristiina Lemmetyinen	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Raportin/Opinnäytetyön nimi Moniosaajan portfolio	Sivu- ja liitesivumäärä 61 + 0
<p>Tämä portfolio-opinnäytetyö on osa Haaga-Helian Tietojenkäsittelyn koulutusohjelman tutkintoa. Opintojen valmistuessa työnhaku tuli ajankohtaiseksi ja työnhaussa käytettävä moniosaajan portfolion tarve heräsi. Olen opiskellut niin ohjelmoinnin kuin digitaalisten palveluiden profiilipintoja ja portfolio-opinnäytetyö tuo esiin monipuolisen osaamisen, joka kertyy opiskelemalla molempia linjoja yhdistellen sekä vapaa-ajalla muista lähteistä tietoa ja oppia keräten.</p> <p>Projektin lopputulos on portfolioraportti sekä työnhaussa käytettävä sähköinen portfolio työnantajia varten. Projektin lopputulos on visuaalisesti ja kirjallisesti laadukas portfolio-opinnäytetyö, joka tuo esiin osaamiseni ja tulevaisuuden kehityskohteeni sekä tavoitteeni. Portfolio-opinnäytetyön pohjalta valmistui varsinainen työnantajille välitettävä sähköinen portfolio.</p> <p>Portfolio-opinnäytetyö tuo esiin sen, millainen on hyvä portfolio. Opinnäytetyössä esittelen valikoitujen projektien tuotokset teorioineen. Tuon projektien kautta esiin osaamistani ja tulevaisuuden kehityskohteitani.</p> <p>Portfolio-opinnäytetyö on toteutettu vuoden 2020 aikana. Keväällä 2020 toteutin uusia projekteja ja jatkokehitin jo tehtyjä projektejani sekä aloitin teoriapohjan keräämisen portfolio-opinnäytetyötäni varten. Kesällä 2020 aloitin raportin kirjoittamisen ja sähköisen portfolioni suunnittelun ja toteutuksen.</p> <p>Projektin myötä opin lisää tekemistäni projekteista ja syvensin osaamistani aihealueissa. Tehtyjen uusien projektien kautta keräsin lisää työkaluja työkalupakkiini. Lopputuloksena projektista syntyi materiaali sähköisen portfolion laadintaan sekä sähköinen työnhaussa käytettävä portfolio.</p> <p>Päädyin portfolio-opinnäytetyöprojektin pohjalta luomaan käyttäjäkokemus -, käyttöliittymäsuunnittelu- ja palvelumuotoiluportfolion Behance-portfoliona ja esittelemään ohjelmointitöitä GitHub-portfoliossa. Päädyin ratkaisuun, koska tavoitteeni on saada työ käyttäjäkokemus, käyttöliittymäsuunnittelu ja palvelumuotoilun saralta. Tästä syystä haluan portfoliossa painottaa kyseessä olevien aihealueiden osaamista.</p> <p>Portfolio-opinnäytetyön tekeminen oli erittäin opettavainen ja mielenkiintoinen kokemus. Osaamiseni syveni projektin kautta kaikilla projektien osa-alueilla. Jatkokehityksenä ohjelmoin itse portfolion tämän projektin tuloksien perusteella.</p>	
Asiasanat Portfolio, moniosaaja, ohjelmointi, käyttäjäkokemus, käyttöliittymäsuunnittelu, palvelumuotoilu	

Sisällys

1	Johdanto.....	1
2	Hyvän portfolion ainekset	3
2.1	Portfolion suositeltu sisältö.....	4
2.2	Vaihtoehtoisia portfolion toteutustapoja	7
3	Oman osaamiseni esiintuonti projektien kautta	9
3.1	Käyttäjäkokemustutkimus.....	9
3.2	Palveluiden tuotteistaminen ja palvelumuotoilu.....	14
3.3	Käyttäjäkokeemus ja käyttöliittymäsuunnittelu	34
3.4	Käyttöliittymäohjelmointi.....	42
3.5	Palvelinohjelmointi.....	47
3.6	Rajapinta- ja konttitekniikat	52
4	Lopputulokset, portfolion esittely ja jatkokehitys.....	58
	Lähteet	62

1 Johdanto

Ajatus portfolio-opinnäytetyön tekemisestä heräsi opintojeni loppusuoralla työhaun tullessa ajankohtaiseksi ja kun huomasin portfolion tarpeellisuuden työhaussa. Oman osaamisen tunnistaminen ja esiintuonti pyörivät mielessä ja pohdin, miten saisin niin visuaalisesti kuin kirjallisesti tuotua esiin monialaisen osaamiseni työhaussa.

Opiskelen Haaga-Heliassa Tietojenkäsittelyn koulutusohjelmassa ohjelmistotuotannon sekä digitaalisten palveluiden profiilopintoja. Lisäksi olen vapaa-ajallani panostanut niin ohjelmointiin, palvelumuotoiluun kuin käyttäjäkokemus ja käyttöliittymäsuunnittelun opintoihin. Päätin tehdä opinnäytetyönäni portfolio-opinnäytetyön, joka tulee toimimaan työhaussa käytettävän sähköisen portfolion pohjana.

Tavoitteena on portfolio-opinnäytetyön avulla kuvata ja kehittää monialaista osaamistani. Tarkoitus on selvittää, millainen on hyvä portfolio. Eri profiilopintojen yhdisteleminen laajemmin oli omalla kohdallani toimiva vaihtoehto toteuttaa opintoni Haaga-Heliassa ja toivon portfolio-opinnäytetyöni toimivan inspiraationa uusille opiskelijoille, jotka miettivät omaa opintopolkuaan ammattilaiseksi. Yksi projektin tavoitteista on kuvata osaamiseni monialainenkaari projektitöiden avulla. Tavoitteena on tuottaa visuaalisesti ja kirjallisesti laadukas portfolio-opinnäytetyö, joka esittelee osaamiseni ja tulevaisuuden kehityskohteeni sekä tavoitteeni. Lopputuloksen pohjalta luon työnantajille välitettävän sähköisen portfolion työnhakua varten.

Selvitän ensin, millainen on hyvä portfolio. Tämän jälkeen esittelen valikoitujen projektieni tuotokset teorioineen. Valitsin portfolio-opinnäytetyössä esiteltävät projektit tarkan harkinnan kautta. Kriteereinä valinnoissa olivat työn tuoma lisäarvo työnhakuun (mitä osaamisalueita valittu työ tuo esiin), osaamisen esittelyn monimuotoisuus (valitsin projekteja niin ohjelmoinnin, palvelumuotoiluun kuin käyttäjäkokemuksen puolelta) ja viimeisenä, mutta ei vähäisimpänä, valitsin parhaimmat tuotokset yllä mainituilta osa-alueilta. Valitut projektit muodostavat kokonaisuuden, joka tuo esiin monipuolisen osaamiseni. Eri projekteissa opimani asiat nivoutuvat yhteen ja projektit tuovat esiin sen, miten hallitsen kehittämisen polun käyttäjäkokemuksen tutkimisen kautta aina lopulliseen tuotteeseen ja testaamiseen asti.

Ensimmäisenä esittelen kouluprojektin Käyttäjäkokelemustutkimus Käyttäjäkokelemus opintojaksolta. Projektissa opin laaja-alaisesti käyttäjäkokemuksen tutkimiseen ja kehittämiseen liittyviä metodeja. Projektityössä edetään ideasta käyttäjäkokemustutkimuksen kautta uuden palvelun ensimmäiseen luonnokseen asti.

Toisena esittelen kouluprojektin Palveluiden tuotteistaminen ja palvelumuotoilu opintojaksoilta. Projektissa opin palveluiden tuotteistamisen ja palvelumuotoilun paljon käytetyt teorit ja tekniikat sekä sovelsin oppimaani projektin asiakasyritykseen.

Kolmanneksi esittelen vapaa-ajan projektin käyttäjäkokemuksen ja käyttöliittymäsuunnittelun puolelta. Projektin aikana opin paljon käyttäjälähtöisestä käyttöliittymäsuunnittelusta sekä suunnitteluohjelma Adobe XD:n käytöstä.

Kolmen ensimmäisen projektin aikana esittelen kehittämisen elinkaaren alkupäätä. Sitä, miten ensin on ymmärrettävä se, mitä lähdetään tekemään ja kenelle lopputulos on tarkoitettu. Viimeiset kolme projektia keskittyvät siihen, mitä tehdään suunnitteluvaiheen jälkeen.

Neljännessä projektissa esittelen kouluprojektin React-sovelluksen. Sovelluksella tarkoitetaan tietokoneella suoritettavaa ohjelmaa, jonka tarkoituksena on suorittaa tietty tai tietyt tehtävät (itewiki). Projektissa opin ohjelmoimaan modernia web-käyttöliittymäpuolen sovelluskehystä ja tekemään sen avulla sovelluksen.

Viidennessä projektissani esittelen kouluprojektin Spring Boot-sovelluksen. Tämän projektin aikana opin Spring Boot-sovelluskehysten käyttöä web-sovelluksessa, yksikkötestausta sekä syvensin Java-ohjelmointikielen osaamista.

Viimeisenä esittelen kouluprojektin GraphQL ja Docker. Projektin aikana toteutin sovelluksen, jossa opin hyödyntämään modernia rajapintateknologiaa GraphQL:ää sovelluslogiikan yksinkertaistamiseen. Projektissa opin myös Docker-konttitekniikan, jonka avulla voidaan helpottaa ohjelmointiprojektien yhteiskäyttöä sekä vähentää fyysisen koneen kuormittamista ohjelmointityössä.

Kokonaisuutena valitut projektit esittelevät kattavasti osaamistani suunnittelun ja kehittämisen saralla aina ideasta lopulliseen testattavaan tuotteeseen asti. Valitut työt tuovat portfolioon esiin monialaisen osaamisen, jonka saavutin opiskelemalla tutkintoni aikana niin digitaalisten palveluiden kuin ohjelmoinnin profiiliin opintojaksoja.

2 Hyvän portfolion ainekset

Portfoliolla tarkoitetaan joko fyysistä tai digitaalista tuotetta, jonka avulla tuodaan esiin osaamista, tapaa työskennellä sekä persoonallisuutta projektien esittelyn kautta. (Krause 2019).

Kerätessäni tietoa siitä, miten rakentaa hyvä portfolio, törmäsin aiheeseen LinkedInissä. Useimmat rekrytoivat henkilöt käsittelivät aihetta keskustelussa ja pyysin kahdelta heistä, saisinko lainata heidän mielipiteitään portfolio-opinnäytetyössäni.

Panu Korhonen, Design Lead VR Groupilta kuvaili hyvää portfolioa ja sen roolia työhaussa seuraavasti:

Portfolion rooli omissa haastatteluissa on keskustelun herättäminen. Erityisen mielellään käyn joitain kandidaatin mielestä edustavia caseja läpi vähän yksityiskohtaisemmin, että mitä prosessin aikana tapahtui, mitkä menetelmät toimi hyvin, mitä opittiin jne. Tästä aika hyvin käy mun mielestä ilmi, että ymmärtääkö oikeasti mitä on tekemässä. (Korhonen 2020.)

Matias Pietilä, Head of Design, partner Qvikiltä kuvaili ajatuksiaan portfolioista työhaussa seuraavasti:

Olen itse ennen kaikkea kiinnostunut siitä, kuinka tekijä ajattelee. Kaikkea ajattelua on vaikea toki kuvata portfolioissa, siksi mieluusti kuulen projektista tarkemmin haastattelussa. Varsinkin vähemmän visuaalisia tuotoksia tekevien suunnittelijoiden kohdalla korostuu kyllä, että näyttävän portfolion tekeminen voi olla vaikeaa. Yksi suositeltava keino on tuoda työnsä tuloksia esiin mittareiden kautta. Mitä oletettiin, mitä testattiin, mitä tuloksia saatiin? (Pietilä 2020.)

Molempien henkilöiden ajatuksissa tulee vahvasti esiin, että portfolio toimii keskustelun avaajana rekrytoinneissa sekä oman osaamisen esiintuojana. Portfolion avulla saadaan tuotua esiin työtehtävissä tarvittavia osaamisia ja työkaluja. Portfolio toimii käytikorttina työelämässä.

Nielsen Norman Groupin artikkelissa "5 Steps to Creating a UX-Design Portfolio" Rachel Krause (Krause 2019) kertoo tutkimuksesta, jossa he kysyivät rekrytoivilta käyttäjäkokemuksen ammattilaisilta, mitä he etsivät hakijoiden portfolioista. Vastaajat kertoivat: "Näytä, kuinka aloitit mahdollisuudesta ja tuotit todellista arvoa käyttäjälle ja organisaatiolle.", "Minua kiinnostaa tietää, mitä ei ole suunnittelussa mukana ja miksi, aivan kuten haluaisin tietää, miksi tietyt elementit ovat mukana", "Älä vain näytä minulle lopputuotetta. Haluan nähdä sotkuisen prosessin ja kaiken työn ja tutkimuksen, joka tehtiin ennen suunnittelua.

Kerro minulle ongelma, jonka yritit ratkaista, roolisi, mahdolliset rajoitukset, projektin aikataulu ja miten tutkimus antoi tiedon suunnittelua varten”. (Krause 2019.)

2.1 Portfolion suositeltu sisältö

Portfolion kohderyhmä ovat rekrytoivat henkilöt, joten portfolion on vedottava juuri heihin. Portfoliota suunniteltaessa on tärkeää ajatella sitä, mitä ominaisuuksia ja taitoja halutaan esitellä ja miten kohderyhmä ymmärtää ja kokee esiteltyt asiat. (Krause 2019.)

Ennen kuin asiassa edetään sen pidemmälle, on tärkeää priorisoida se, mitä halutaan kommunikoida portfolion kohderyhmälle. Mitkä ovat portfolion tekijästä ja hänen töistään kolme tärkeintä asiaa, jotka portfolion lukijan tulisi huomata ja muistaa portfolion lukemisen jälkeen. Kun portfolion tekemisen alussa nämä asiat kirkastetaan esiin, niin portfolion valmistumisen jälkeen on helppo testata portfolion toimivuus testaamalla välittyvätkö valmiista tuotteesta halutut asiat lukijalle. (Krause 2019.)

Molempien osaamisalueiden - suunnittelun ja kehittämisen osaamisen esiintuonnissa portfolion avulla pätevät useat samat säännöt. Portfolion alkuun on hyvä kirjoittaa lyhyt ja selkeä kuvaus tekijästä. Näin rekrytoijat tietävät heti, kuka hakija on, mitä hän haluaa esitellä ja mitä häneltä voi odottaa. On tärkeää muistaa, että portfolio toimii käyntikorttina. Se on usein ensimmäinen asia, jonka rekrytoija hakijasta näkee ja sen pohjalta tehdään päätöksiä siitä, haluaako rekrytoija keskustella hakijan kanssa lisää. Siksi on erittäin tärkeää miettiä, miten herättää katsojan mielenkiinto ja minkälaisen mielikuvan hakija haluaa itseltään antaa portfolion avulla. (Siang and the Interaction Design Foundation 2020.)

Kuvauksessa tulee käydä ilmi hakijan nimi, yhteystiedot, nykyinen työtehtävä tai työtehtävä, jossa hakija toivoo seuraavaksi työskentelevänsä. Myös sijainti tulee ilmoittaa, jos sillä on merkitystä työnannon kannalta. Esittely on kuitenkin hyvä pitää lyhyenä, muutaman lauseen pituisena. Kuvaus on hyvä paikka tuoda esiin omaa persoonallisuuttaan ja näin erottua muista hakijoista. (Siang and the Interaction Design Foundation 2020.)

Käyttäjäkokemus-, käyttöliittymäsuunnittelu- ja palvelumuotoilunportfoliota tehdessä suositellaan valitsemaan ja esittelemään kahdesta kolmeen parhaista projekteista. On tärkeää valita nimenomaan visuaalisesti ja informatiivisesti parhaimmat työt ja esitellä nämä kattavasti ja tiiviisti. Töitä valitessa on tärkeää miettiä, että työt tuovat esiin haettavan roolin kannalta oleellisia taitoja. Töitä valitessa on hyvä miettiä, mitä juuri tämä työ tuo esiin tekijästä, hänen osaamisestaan sekä mitä työt viestivät kohderyhmälle. (Siang and the Interaction Design Foundation 2020.)

Valittujen projektien valmiiden visuaalisten tuotosten lisäksi on tärkeää tuoda esiin koko suunnitteluprosessi, ongelma tai tavoite, joka ratkaistiin, portfolion omistajan rooli projektissa, ketkä projektiin osallistuivat ja vastuiden jakautuminen projektissa. On tärkeää tuoda esiin se mitä tehtiin, miksi tehtiin mitä tehtiin sekä mihin tietoon valitut asiat perustuvat. Jos projektista on saatu mitattavia tuloksia, kuten esimerkiksi myynnin kasvu, on tämä tuotava esiin. On myös hyvä tuoda esiin ratkaisut, jotka päätettiin prosessin aikana hylätä sekä mahdolliset haasteet projektin aikana. Viimeisenä, mutta ei suinkaan vähäisimpänä, on tärkeää tuoda esiin mitä projektista opittiin ja mitä kenties tehtäisiin nykyisen tiedon perusteella toisin. (Krause 2019.)

Valitsipa portfolion toteutustavaksi minkä vain, niin on tärkeää muistaa portfolion visuaalisuus ja käytettävyys. Käytettävyys on avain onnistuneeseen portfolioon. Rekrytoijat arvioivat portfolion kuten yksittäisetkin projektit ja käytettävyysongelmat portfolioissa ovat yhtä huono juttu kuin käytettävyysongelmat itse projekteissa. Tästä päästäänkin taas siihen, että hakijan on syytä testata portfolionsa. Näin portfolion laatija saa ajoissa selville mahdolliset ongelmat portfolion käytettävyydessä. (Siang and the Interaction Design Foundation 2020.)

Portfolion tulisi sisältää projekteista asiankuuluvat valokuvat, kuvakaappaukset ja muu materiaali, jotka visuaalisesti kertovat projektin tarinan. Esimerkiksi varhaisen vaiheen luonnokset, tutkimusmateriaalit ja lopulliset tuotokset on kuvattava portfolioissa. On tärkeää muistaa, että projektin valmiit tuotokset kertovat vain osan tarinasta. Rekrytoivat henkilöt haluavat ymmärtää laajemmin hakijan tapaa työskennellä ja tämän tavoitteen saavuttaa parhaiten, kun kuvaa projektin koko prosessin alusta loppuun ja näin tuo esiin laajempaa kuvaa tekijän ammattitaidosta, luonteesta sekä tavasta työskennellä. (Krause 2019.)

Portfolioon voi halutessaan luoda laajemman esittelyn itsestään esimerkiksi tietoja -sivun avulla. Sivulla voidaan kertoa tekijästä lisää tietoa, kuten esimerkiksi harrastuksista ja kiinnostuksen kohteista, työhistoriasta ja koulutustaustasta sekä siellä voi jakaa sosiaalisen median linkit ammattiprofiileihin. (Siang and the Interaction Design Foundation 2020.)

On tärkeää valita portfolioon kahdesta kolmeen parhainta työtä, mutta portfolioissa voi esitellä myös enemmän projekteja omana osionaan. Nämä muut työt voidaan portfolioissa sijoittaa esimerkiksi omalle välilehdelle ja näin saadaan yleisilme pysymään siistinä ja yksinkertaisena. Vaikka portfolioon suositellaan vain muutamaa tarkkaan valittua työtä esi-

teltäväksi ensin, niin on tärkeää sisällyttää muutkin taidonnäytteet portfolioon. Tämä auttaa esittelemään lisää tekijän osaamista, intohimoa, oppimisen nälkää sekä antavat rekrytoijille paremman käsityksen hakijan persoonallisuudesta ja siitä, miten hakijan ajatuksien juoksu suunnittelijana kulkee. (Siang and the Interaction Design Foundation 2020.)

On myös tärkeää muistaa, että hyvä on tarpeeksi hyvä. Hakijan tulee pyrkiä luomaan ja levittämään tarpeeksi hyvää versiota portfolioista sen sijaan, että viilaisi sitä kuukausia pöytälaatikossa. Kun portfolio on julkaistu alkaa jatkuva portfolioon jatkokehitystyö. Portfolio ei ole koskaan valmis, vaan se kehittyy alati, kuten hakijakin ammattilaisena. (Siang and the Interaction Design Foundation 2020.)

Ohjelmoinninportfolio ei ole pakollinen rekrytoinneissa, mutta varsinkin juniortason ohjelmoijat hyötyvät siitä usein. Hyvin tehty portfolio kompensoi mahdollisia muiden osa-alueiden puutteita, kuten esimerkiksi työkokemuksen puutetta. Ohjelmoinninportfolio voi olla monella tavalla toteutettu, mutta tunnettu työkalu ohjelmoinninportfoliossa on versionhallintatyökalu GitHub. (Fecak 2020.) Esittelen GitHub-palvelun myöhemmin raportissa kohdassa 2.2.

Ohjelmoinninportfolio on sisällön suunnittelultaan hyvin samanlainen kuin käyttäjäkokemus- ja käyttöliittymäsuunnitteluportfolio. Kun rekrytoija käy läpi ohjelmoinnin projekteja, häntä kiinnostavat työt, joita hakija pitää parhaimpina töinään, aivan kuten käyttäjäkokemus- ja käyttöliittymäsuunnitteluportfoliossakin. Esimerkiksi GitHub-portfoliossa voidaan käyttää toimintoa, jonka avulla voidaan kiinnittää parhaat työt profiiliin etusivulle. (Laakkonen 2020.)

Ohjelmoinninportfoliossa on tärkeää kiinnittää huomiota siihen, että esiteltyjen projektien teknologiat ovat haettavan työn kannalta oleellisia. Hyvä vinkki on lisätä CV:een lisätyn taidon kohdalle linkki ohjelmointiprojektiin, jossa tämä taito näkyy käytännössä. (Laakkonen 2020.)

Rekrytoija ei tutki jokaista hakijan ohjelmointiprojektia läpi yksityiskohtaisesti, vaan keskittyy yleiskuvaan. Tämän vuoksi ensivaikutelma on tärkeä osa myös ohjelmoinninportfoliossa. Selkeä ja standardinmukainen projektirakenne sekä selkeä tiivistelmä projektista ovat hyvä keino antaa hyvä vaikutelma ja herättää katselijan kiinnostusta. (Laakkonen 2020.)

Juniortason portfoliossa on hyvä muistaa, että työnantajat eivät odota juniorohjelmoijan käsittelevän, suunnittelevan ja toteuttavan itsenäisesti esimerkiksi seuraavaa Facebookia.

Jos ohjelmointiprojektin tavoitteen asettaa liian korkealle, voivat projektit jäädä kesken ja näin portfolioista ei tule selkeää kokonaisuutta. On tärkeää rajata työt ja valita itselleen projekteja, joiden tavoite on saavutettavissa. (Fecak 2020.)

Ohjelmoinninportfolioa tehdessä on muutamia vinkkejä, joita seuraamalla kokonaisuudesta tulee selkeämpi ja katselmoijalle kiinnostavampi:

1. Monipuolisuus. Jos portfolioissa on peli toteutettuna Java-ohjelmointikielellä, niin seuraavaksi voisi toteuttaa saman esimerkiksi Python-ohjelmointikielellä.
2. Projektit tulee saattaa loppuun. Rekrytoijat haluavat nähdä mieluummin muutaman hyvän, loppuun asti viedyn projektin kuin useamman keskeneräisen.
3. Lähdekoodin toimivuus. On tärkeää varmistaa, että projektin lähdekoodi toimii kuten tiedoissa sanottu sen toimivan.
4. Lähdekoodin luettavuus. Lähdekoodi tulee olla kirjoitettu niin, että se on helposti ymmärrettävää.
5. On tärkeää myös dokumentoida projekti ja kertoa, mikä projektin tarkoitus on, mitä teknologioita käytettiin sekä antaa käyttäjälle käyttöohjeet projektiin. (Fecak 2020.)

2.2 Vaihtoehtoisia portfolion toteutustapoja

Portfolion voi toteuttaa monella tapaa, kuten esimerkiksi ohjelmoimalla sen itse tai käyttämällä valmista palvelua sen toteuttamiseen. Seuraavaksi käyn läpi eri palveluita, joiden avulla voi tehdä portfolion ilman, että tarvitsee osata ohjelmoida. Nämä palvelut ovat tunnettuja ja paljon käytettyjä palveluita, mutta myös muita vastaavia tuotteita löytyy paljon.

Ensimmäiseksi esittelen Behance-palvelun, joka on suosittu visuaalisen suunnittelun portfoliopalvelu. Palvelu löytyy osoitteesta <https://www.behance.net/>. Behance-palvelussa on yli kymmenen miljoonaa jäsentä ja palvelu on erittäin tunnettu. Käyttäjä voi liittyä palveluun ilmaiseksi ja julkaista palvelussa portfolionsa. Behance on online-portfolio sivusto, jonka avulla voi esitellä projekteja sekä verkostoitua muiden suunnittelijoiden kanssa ja saada inspiraatiota. Behance-projektieditori on helppokäyttöinen. Behance-palvelussa portfolio on muiden käyttäjien portfolioiden rinnalla. (Siang and the Interaction Design Foundation 2020.)

Jos taas halutaan portfolioiksi oma verkkosivu, mutta ilman ohjelmointia, niin WordPress-palvelu on tällöin hyvä vaihtoehto. Palvelu löytyy osoitteesta <https://wordpress.com/>. WordPress-palvelusta löytyy sekä ilmainen että maksullinen versio. Ilmainen versio antaa

käyttäjälle mukautetun ".wordpress.com" osoitteen, mutta käyttäjä voi milloin vain päivittää maksulliseen versioon ja tällöin on mahdollista päivittää verkkosivunsa URL-osoitetta edelleen esimerkiksi "sinunverkkosivustosi.com". Näin saadaan yksilöllisempi osoite ja muokattavampi portfoliorakenne. (Siang and the Interaction Design Foundation 2020.)

WordPress tarjoaa ilmaisia sekä maksullisia teemoja, joiden avulla voi muokata portfolioa juuri sellaiseksi, kun tekijä itse haluaa ohjelmoimatta laisinkaan. Helppokäyttöisen WordPress-editorin avulla käyttäjä voi luoda portfolion helposti ja nopeasti.

Jos portfolion tekijä hallitsee ohjelmoinnin tai hänellä on tietty visio siitä, millainen portfolion tulee olla ja sen toteutukseen tarvitsee ohjelmointia, niin hyvä vaihtoehto on ohjelmoida portfolio itse ja tehdä samalla uusi työnäyte osaamisesta. Ensimmäisen portfolion voi ohjelmoida esimerkiksi opettelemalla HTML- ja CSS-teknologioita. Jos portfolion tekijä on jo kokeneempi ohjelmoinnin parissa niin, varteenotettavia vaihtoehtoja voivat olla esimerkiksi React- tai React Native-teknologiat. Niin käyttäjäkokemus- ja käyttöliittymäsuunnittelun- kuin ohjelmoinninportfolion voi toteuttaa itse ohjelmoimalla ja näin saavuttaa juuri halutunlaisen portfolion.

GitHub-palvelu on yleinen paikka ohjelmoinnin projektien versionhallinnaksi ja portfolioiksi. GitHub löytyy osoitteesta [GitHub.com](https://github.com) ja oppimateriaalia sen käyttämiseen löytyy laajasti. (Magana 2018.)

GitHub on kehitysalusta ohjelmoinnin projekteille. Palvelusta löytyy niin avoimenlähdekoodin palvelu kuin yrityksille suunnattu palvelu. Palvelussa voi säilyttää ja ylläpitää projekteja yhdessä kolmenkymmenen miljoonan muun kehittäjän kanssa. Yksityishenkilöille palvelu on ilmainen. (Magana 2018.)

GitHub-palvelun tarkoitus on tukea projekteissa tehtävien muutosten seuranta. Versionhallinnassa on aina projektin toimivin lähdekoodi ja versionhallinta tukee tiedostoon tehtyjen muutosten peruuttamista ja koodikantaan tehtyjen muutosten merkintää. GitHub-palvelussa voi myös ylläpitää ja kehittää projektin dokumentaatiota. GitHub-palvelua voi käyttää graafisen käyttöliittymän tai komentorivin kautta. (Magana 2018.)

3 Oman osaamisen esiintuonti projektien kautta

Portfolio-opinnäytetyöhön valitsemani projektit tuovat teorian ja käytännön kautta esiin monipuolisen osaamisen tuomalla esiin sen, miten hallitsen käyttäjäkokemuksen kehittämisen, suunnittelun ja testaamisen sekä lopullisen tuotteen rakentamisen. Valitsin esiteltäväksi projekteja, joissa koen oppineeni paljon nykyaikaisista teknologioista sekä projekteja, joista olen nauttinut suuresti. Valitessani projekteja yksi valintaperusteista oli myös se, että näistä projekteista haluan oppia lisää paneutumalla aihepiirien teoriaan ja projektien läpikäyntiin uudelleen. Jokainen projekti kuvaa hyvin laajaa osaamisen kirjoa, jonka olen opinnoissani saavuttanut ja jokainen työ esittelee taitoja, joita unelmieni työtehtävissä tarvitaan.

3.1 Käyttäjäkokeutumustutkimus

Käyttäjäkokeutus projektityö on Käyttäjäkokeutus opintojakson kurssityönä tehty projekti. Kurssityö on tehty ryhmätyönä ja työn arvosanaksi tuli 5 asteikolla 1-5. Ryhmässä teimme tiiviinä ryhmänä töitä ja kokonaisuus on yhdessä tehty. Vastasin projektissa vuorollani projektin hallinnasta, valittu aihe oli ideoimani, otin vetovastuun tutkimuksen perusteella ideoiden esiin tuonnista analysoinnin kautta sekä kirjoitin kysymykset kyselyä sekä haastattelua varten.

Valitsin projektin mukaan portfolio-opinnäytetyöhön, sillä projekti oli kokonaisuudessaan erittäin onnistunut ja projektissa tulee hyvin esiin laaja skaala käyttäjäkokeutumisen kehittämisessä hyödynnettäviä työkaluja sovellettuna käytäntöön. Erityisesti koin affiniteetti-analyysin oppimisen hienona ja hyödyllisenä kokemuksena.

Affiniteettianalyysi tarkoittaa tutkimuksessa esiin tulleiden tulosten järjestämistä kategorioittain. Ensin kirjoitetaan esiin nousseet asiat lapuille ja sen jälkeen laput järjestetään kategorioittain. Tämän jälkeen analysoidaan esiin tulleita asioita tai ideoita ja priorisoidaan laput. Lopuksi päätetään jokaisen kategorian ja esiin tulleen asia kohdalla jatkotoimenpiteet: jatketaanko kenties tutkimista, vai löytyykö jo tarpeeksi informaatiota jatkosuunnittelua varten. (Kara, 2018.)

Opintojakson tavoitteena oli, että käytyään opintojakson opiskelija osaa kehittää palveluja tuomalla käyttäjän äänen mukaan kehitykseen. Tavoitteena oli, että opiskelija ymmärtää opintojakson suoritettuaan käyttäjäkokeutumisen koostuvan käyttäjän kannalta merkityksellisestä arvontuotannon prosessista, johon liittyy erilaisia palvelutuokioita ja kontaktipisteitä

palveluntuottajan, erilaisten käyttöliittymien ja muiden palvelun käyttäjien kanssa. Opintojakson käytyään opiskelija osaa tarkastella ja kehittää sekä kokonaiskokemusta että sen osia asiakaslähtöisesti. Tavoitteena oli myös, että opiskelija osaa opintojakson käytyään käyttää erilaisia menetelmiä käyttäjäkokemuksen kartoittamiseen ja suunnitteluun sekä soveltaa niitä kuhunkin tilanteeseen sopivalla tavalla. (Moodle 2019.)

Projektissa toteutettiin käyttäjäkokemustutkimus Haaga-Helian lounasruokalan maksuprosessista. Tutkimus oli kattava ongelman rajaamisesta, kyselyn ja analysoinnin kautta palvelun ensimmäiseen luonnokseen (prototyyppiin) asti. Opintojakson aikataulun ja sisällön vuoksi prototyypistä on yksi kuva pääsivusta. Jatkokehityksenä prototyyppiä tulisi tehdä pidemmälle ja ottaa mukaan jatkuva käyttäjälähtöinen testaus. Tämä ei kuitenkaan opintojakson puitteissa ollut mahdollista.

Projektin tavoitteena oli tarkastella opiskelijoiden kokemuksia Pääraiteen (SLK:n ravintola) lounaan maksamisprosessista ja opiskelijastatuksen todistamisesta ruokalassa maksamisen yhteydessä. Projektissa toteutettiin sekä kyselylomakkeella kysely, että yhdeksän kappaletta yksilöhaastatteluja. Kyselylomakkeeseen vastauksia tuli yhteensä sata. Haastattelut purettiin auki, analysoitiin ja tulosten perusteella löydettiin vastaajien kommentteista uuden konseptin toiminnalle Pääraiteessa.

Uusi konsepti muodostui tutkimuksen perusteella kaksiosaiseksi:

1. **Automatisointi.** Tuloksen pohjalta ehdotettiin maksuautomaatteja ruokalan aulaan.
2. **Mobiilisovellus.** Tuloksen pohjalta ehdotettiin mobiilisovellusta, jolla opiskelija voi maksaa ruokailun, ostaa ruokakertoja etukäteen sekä seurata ruokalan reaaliaikaista jonotilannetta sekä tutkia jonohistoriaa.

Aluksi kerättiin datasta nousseet asiat muistilapuille ja järjesteltiin esiin tulleet asiat kategoriaittain. Kuvassa 1 näkyy esimerkki projektityössä hyödynnetystä affiniteettianalyysistä.

Tutkimus toi esiin, että opiskelijat arvostavat helppoa, nopeaa sekä nykyaikaista tapaa tunnistautua ja maksaa ravintola Pääraiteessa. Enemmistö toivoi, että tunnistamisen ja maksamisen prosessia kehitettäisiin. Tuloksissa nousi esiin useamman kerran vastaajien ideoissa sanat automaatio, maksaminen etukäteen, tunnistaminen ja maksaminen samassa sekä jonojen poistaminen. Uuden tavan toimia lounasruokala Pääraiteessa pitää olla siis nykyaikainen, nopea ja helppo.

Nykyisessä toimintatavassa nousi esiin negatiivisena puolena muun muassa asiointin hitaus eli se, että opiskelijat kaivavat maksukortit, puhelimen ja Kela-ateriatukilapun esiin vasta kassalla, sekä kassahenkilökunnan hitaus. Vaikka toimintaan oltiin tyytyväisiä pääosin, niin suurin osa vastanneista toivoi silti, että toimintaa kehitettäisiin joustavammaksi.

Aineistoa analysoidessa löydettiin neljä suurta kehityskohtaa, joita lähdettiin tutkimaan tarkemmin: **hitaus, sujumattomuus, monimutkaisuus** sekä **jonot**. Uudella konseptilla tarjotaan ratkaisut näihin ongelmiin.

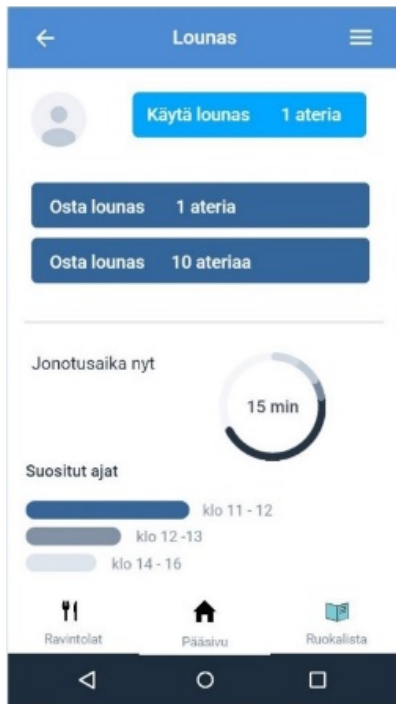
1. Automatisointi: ruokalan aulaan asennetaan itsepalvelumaksuautomaatteja tunnistautumista ja maksamista varten. Automaatista valitaan lounas etukäteen. Maksamisen yhteydessä näytetään Kela-korttia laitteelle, jolloin laite tunnistaa onko henkilö oikeutettu opiskelijahintaiseen lounaaseen vai ei. Varsinainen maksaminen suoritetaan lopuksi automaattiin maksukortilla. Automaatti antaa käyttäjälle poletin, jolla käyttäjä todistaa maksaneensa ruuan.

2. Mobiilisovellus: tunnistautuminen ja maksaminen voidaan hoitaa myös sovelluksella. Sovellus tarjoaa mahdollisuuden ostaa yhden ruokailukerran tai käyttäjä voi ostaa sovelluksella esimerkiksi 20 ruokakertaa. Linjaston alussa on laite, johon sovelluksesta näytetään maksetun ruoan koodi. Laite varmistaa samalla, että henkilö on oikeutettu opiskelijahintaiseen lounaaseen tarkistamalla ruokailija opiskelijastatuksen.

3. Jonot pois: Sovelluksessa löytyy myös toiminto, jolla käyttäjä voi tarkastella jonotilannetta. Sovellus näyttää jonotushistoriaa pitkältä ajalta. Sovelluksesta voi tarkastella tilastoja Pääraiteen jonoista eri päivinä ja kellonaikoina. Sovellus näyttää myös senhetkisen tilanteen, eli sovelluksesta voi tarkistaa joutuuko ruokalassa jonottamaan vai ei. Ruuhkatileen voi tarkistaa jo luokassa ja tarvittaessa siirtää ruokailua hieman myöhemmäksi tai aikaisemmaksi.

4. Käteisellä maksaminen: Jos käyttäjä haluaa jatkossa maksaa käteisellä, tulee hänen mennä päätalon kahvilaan tunnistautumaan ja maksamaan lounas. Kahvilasta hän saa potetin, joka annetaan kassahenkilölle ruokalassa.

Kuvassa 2 on esimerkki siitä, minkälainen uusi sovellus ruuan maksamista ja jonojen seuraamista varten voisi olla.



Kuva 2. Ensimmäisen vaiheen luonnos mobiilisovelluksesta

Kurssityön tekeminen oli hieno kokemus ja opin paljon ongelman ratkaisusta asiakaslähteisesti. Työkaluna affiniteettianalyysin opettelin koin mielisana ja tärkeänä. Erityisesti innostuin tutkimuksesta ja tiedon analysoinnista affiniteettianalyysin avulla. Oli hienoa nähdä, miten työstämällä kerättyä tietoa saimme esiin idean uudesta konseptista.

3.2 Palveluiden tuotteistaminen ja palvelumuotoilu

Tämä projektityö on Palvelujen tuotteistaminen ja palvelumuotoilu opintojakson kurssityönä tehty projektityö. Työ on tehty yksilötyönä ja arvosanaksi tuli 5 asteikolla 1-5.

Opintojakson tavoitteena oli, että suoritettuaan opintojakson opiskelija ymmärtää palvelun tuotteistamisen sekä palvelumuotoilun periaatteet ja prosessit. Opintojakson teemoina olivat palvelumuotoilun teoriat, menetelmät ja työkalut sekä niiden soveltaminen käytäntöön kurssityön kautta. Opintojakson projektityönä opiskelijat tarkastelivat olemassa olevaa palvelua asiakaslähtöisesti hyödyntäen palvelumuotoilun prosesseja sekä menetelmiä. Projektityössä oli tavoitteena esitellä palvelumuotoilun prosessit ja menetelmät kirjallisesti ja sen jälkeen soveltaa opittua valittuun palvelutuotteeseen. (Moodle 2019b.)

Valitsin projektin portfolio-opinnäytetyöhön, sillä projekti oli laaja ja kattava palvelumuotoilun ja tuotteistamisen yksilötyönä toteutettu projektityö. Projektista tulee esiin laaja skaala palvelumuotoilussa käytettyjä metodeja ensin teorian muodossa ja lopuksi sovellettuna käytäntöön. Projektityö tuo kattavasti esiin laajan perusosaamiseni palvelumuotoilun saralla ja se on erinomainen työnäyte portfolioon.

Kutsun projektin kohdeyritystä nimellä yritys X. Projektia ei ole tehty yhteistyössä yrityksen kanssa. Opintojaksolla valikoitiin mielenkiintoinen palvelutuote ja tutkimus tehtiin julkisesti saatavilla olevista materiaaleista. Tästä syystä en käytä projektissa yrityksestä yrityksen oikeaa nimeä.

Valitsin tutkittavaksi palvelutuotteeksi yritys X:n kauppakassipalvelun. Projektissa käsitteelin palvelumuotoilun teorioita ja esittelin sekä analysoin valitsemaani palvelutuotetta yritys X:n kauppakassipalvelua oppimani avulla.

Palvelumuotoilu on käyttäjälähtöistä suunnittelua, jossa pyritään tekemään kokonaisvaltaisesti palvelusta asiakkaalle sopivampi ja parempi. Tarkoitus on tuottaa mahdollisimman hyvä asiakaskokemus. (Jaakkola, Orava & Varjonen 2009, 7.)

Palvelun tuotteistaminen on keino tehdä palveluiden kehittämisestä systemaattista niin, että yritys menestyy ja on kilpailukykyinen. Kilpailukykyille keskeisiä ominaisuuksia ovat hyvin suunniteltu ja johdettu tuotekehitysohjelma, jatkuva innovointi, kustannustehokkuus sekä asiakaslähtöisyys. (Jaakkola, Orava & Varjonen 2009, 7.)

Tuotteistamisella voidaan tarkoitaa uusien ja olemassa olevien palvelujen määrittelyä, systematisoimista ja vakiointia, joka voi kohdistua sekä yrityksen sisäisiin että ulkoisiin prosesseihin. (Jaakkola, Orava & Varjonen 2009, 6.)

Tuotteistaminen ja palvelumuotoilu on järkevää, sillä tänä päivänä asiakkailta on varaa mistä valita, joten vain parhaimmat yritykset ja palvelut pärjäävät. Lisäksi tuotteiden ollessa hyvinkin samakaltaisia, ylivertaisen asiakaskokemuksen merkitys kasvaa suureksi asiakkaan valitessa palveluntarjoajaa. Tarjotun palvelun on oltava kustannustehokas, mutta silti juuri sitä, mitä asiakkaat haluavat. Tuotteistamisen hyötyjä ovat esimerkiksi palvelun tasainen laatu, toistettavuus, sisäinen kommunikointi ja markkinoinnin sekä myynnin tehostuminen.

Palvelumuotoilussa lähdetään miettimään asioita asiakas edellä. Palveluiden tuotteistamisen tavoite on tehdä palveluiden kehittämisestä systemaattista. Palvelumuotoilu on jatkuva prosessi, joka muotoilee palvelua käyttäjäystävällisemmäksi.

Kuvassa 3 on palvelumuotoiluprosessin neljä vaihetta.



Kuva 3. Palvelumuotoiluprosessi (Innanen 2018)

Palvelumuotoilunprosessissa on neljä vaihetta:

1. Määrittele palvelun nykytilanne. Tavoitteena on selkeyttää organisaation sisäisiä ja liiketoiminnallisia tavoitteita sekä kerätä ymmärrystä sekä tietoa palvelutuotteen mahdollisista ongelmakohtista. Nykytilannetta tutkimalla tunnistetaan ne asiat, joita halutaan tutkia paremmin. (Innanen 2018.)

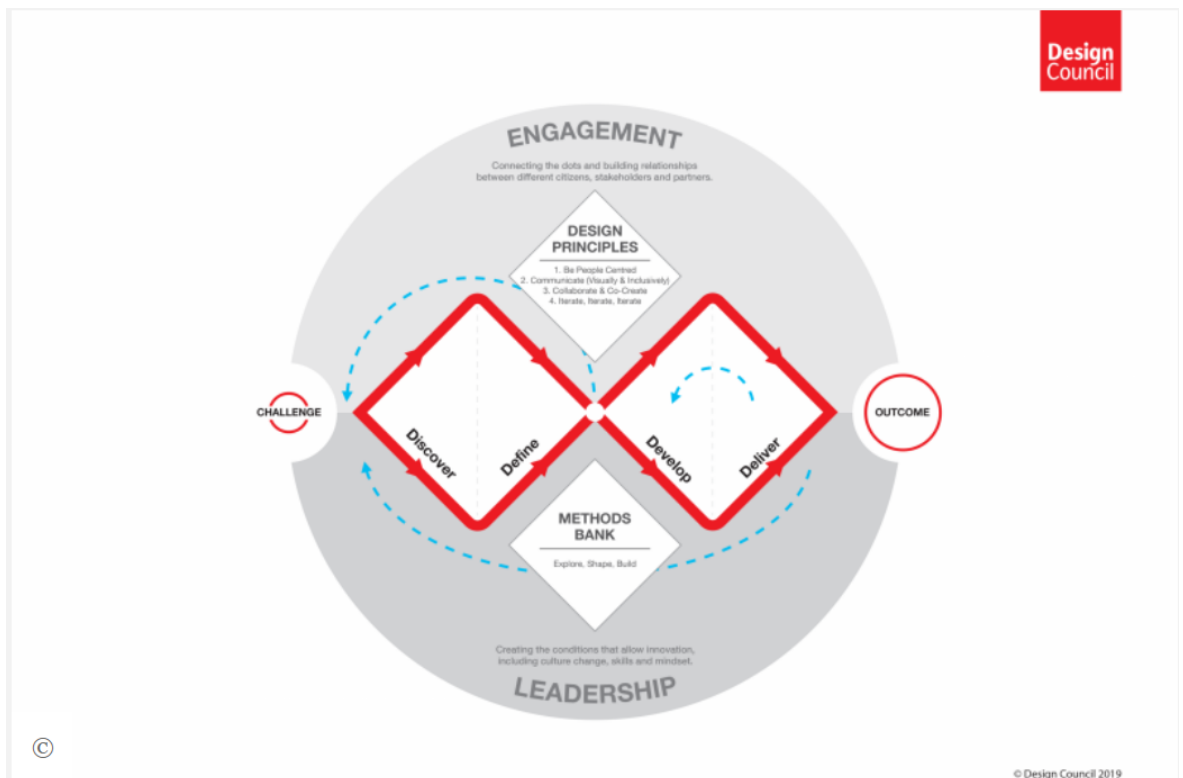
2. Tutki ja kiteytä. Tässä prosessin vaiheessa kasvatetaan ymmärrystä palvelutuotteen kohderyhmän tarpeista. Tavoitteena on laajempi asiakasymmärrys ja löytää mahdolliset

piilevät asiakastarpeet tai haasteet, jotka vaikuttavat havaitun haasteen ratkaisemiseen. (Innanen 2018.)

3. Ideoi, kuvaa ja kokeile. Ideoidaan mahdollisimman paljon vaihtoehtoisia ratkaisuja asiakaslähtöisesti organisaation tavoitteet huomioiden. Ideat jalostuvat kuvauksien ja konkreettisuuden avulla toteutuskelpoisiksi konsepteiksi. Tässä vaiheessa hyödynnetään esimerkiksi prototyyppien (luonnosten) tekoa. Ideoista saadaan näin haettua toteutumiskelpoiset ja jatkokehityksen arvoiset ideat. (Innanen 2018.)

4. Testaa ja toteuta. Uusia ratkaisuja viilataan loppuun ja lopuksi viedään asiakkaiden saataville. Tähän vaiheeseen kuuluvat oleellisesti testaus, pilotointi, mittaaminen sekä sen varmistaminen, että palvelutuote on kilpailukykyinen ja että projektissa on saavutettu sille määritellyt tavoitteet. (Innanen 2018.)

Yksi tunnetuimmista muotoilun prosessimalleista on British Councilin kuvassa 4 kuvattu tuplatimanttimalli (The Double Diamond). Malli on palvelumuotoilun asiakaslähtöinen kehitysmalli.



Kuva 4. Tuplatimanttikehitysmalli (Design Council 2015)

Kuvassa nähtävissä olevassa ensimmäisessä timantissa pyritään ymmärtämään ratkaisu-
tava ongelma. Toisessa timantissa päästään ratkaisun kehittämiseen. Mallin timantit ku-
vastavat prosessia, jossa asiaa tutkitaan laajemmin ja syvällisemmin ja sen jälkeen tartu-
taan rajatun ongelman kimppuun. Tuplatimanttimallissa prosessi jaetaan neljään osaan:
selvitä, määrittele, kehitä ja toimita. Moni palvelumuotoiluprosessien teorioista on ottanut
mallia tuplatimanttikuvauksesta. Palvelumuotoilussa keskeistä on, että asiakas on mu-
kana kehityksessä alusta asti. Palvelumuotoilu on asiakaslähtöistä kehittämistä. (Design
Council 2015.)

Valitsin palvelutuotteeksi projektiin yritys X:n ruokakassipalvelun, sillä palvelu on innos-
tava ja moderni palvelutuote. Yritys X:n ruokakassipalvelu on helppo tapa kokeilla uusia
reseptejä ja saada uusia ideoita. Laadukkaat raaka-aineet toimitetaan kotiin kuljetettuina
asiakkaille (halutessaan tuotteen voi myös noutaa kaupalta) ja selkeiden reseptien avulla
asiakas saa loihdittua herkullisia aterioita kassin sisällöstä. Asiakas voi valita minkä, mil-
loin ja kuinka monelle hengelle ruokakassin haluaa. Palvelun idea on tuoda helpotusta ar-
keen ja toimittaa asiakkaalle valmiiksi raaka-aineet ja reseptit. Vain ruuanvalmistus jää
asiakkaalle itselleen.

Asiakas voi valita kauppakassipalvelusta neljästä eri kassivaihtoehdosta: arki-, vege-,
kala-, kasvis- tai muksukassi. Välillä tarjonnasta löytyy kasseja teeman mukaan, kuten
esimerkiksi äitienpäiväkassi. Ruokakassin koon voi valita joko kahdelle tai neljälle hen-
gelle. Tilauksen keston kertatilauksena tai kestotilauksena. Tuotteessa on vakioituja osia,
mutta asiakkaalla on myös varaa mistä valita. Palvelu ei ole täysin tuotteistettu standardi-
paketti kaikille samanlaisena, mutta perusrunko on kaikille sama. Palvelu on pitkälle tuot-
teistettu.

Yritys ei voi realistisesti tarjota jokaiselle sopivia tuotteita tai palveluja, joten yrityksen on
rajattava toimintaansa. Segmentoinnissa kohdemarkkina jaetaan pienempiin ryhmiin eli
segmentteihin. Segmentointia voidaan tehdä esimerkiksi maantieteellisen sijainnin, demo-
grafisten tekijöiden, asenteiden tai asiakasuskollisuudentason perusteella. Käytännössä
segmentoinnissa yhdistellään edellä mainittuja ja myös muita segmentoinnin tapoja riip-
puen markkinoitavasta tuotteesta ja markkinasta. Onnistunut segmentointi on yksi yrityk-
sen menestystekijöistä. (Pitkänen 2019.)

Yritys X:n ruokakassipalvelun kohderyhmään kuuluvat kokeilunhaluiset kotitaloudet yrityk-
sen toimitusalueella Uudellamaalla. Kohderyhmän henkilöt kuuluvat työssäkäyvään ryh-
mään sekä ryhmään, jotka kokevat intoa kokeilla uusia palveluita sekä toivovat helpotusta
arkeen. Kohderyhmään kuuluvat henkilöt arvostavat laatua ja tuoreutta.

Toimitusalueen ulkopuolella olevat henkilöt rajautuvat pois kohderyhmästä kuljetuksen osilta, mutta halutessaan he voivat tilata ja noutaa kassin kaupalta. Erikoisruokavalion omaavat (esimerkiksi vegaanit tai diabeetikot) rajautuvat myös pois käyttäjäryhmästä, sillä tarjonnasta ei tällä hetkellä löydy heille sopivaa vaihtoehtoa. Kassin minimikoko on kahdelle henkilölle, joten myös pienet kotitaloudet rajautuvat pois kohderyhmästä. Palvelun kohderyhmään kuuluvat haluavat etunenässä ottaa käyttöön uusia tuotteita ja palveluita. Lisäksi he ovat kokeilunhaluisia ja toivovat kiireisen arjen keskelle helpotusta. Kohderyhmä on loppukäyttäjät, palveluntilaaja maksaa ja myös saa palvelun.

Kuvassa 5 on kuvattuna palvelun kohderyhmän edustajat Matti 35 ja Maija 34, pari Helsingistä.



Kuva 5. Yritys X:n kohderyhmän edustajat

Matti on ammatiltaan ohjelmoija ja Maija suunnittelija. Pari asuu Ullanlinnassa Helsingissä. Heillä ei ole lapsia. Perheeseen kuuluu kissa nimeltä Miuku. Matti ja Maija ovat trendikkäitä kaupunkilaisia. He haluavat syödä terveellisesti ja ekologisesti. He arvostavat paikallista ruokatuotantoa ja luonnonmukaisia tuotteita. Kiireisestä elämästään huolimatta he haluavat investoida laatuun ravinnossaan. He arvostavat laatua, orgaanista ja ekologista ruokaa sekä helppoa että nopeaa palvelua.

Palvelua hinnoitellessa on tärkeää lähteä miettimään hintaa asiakkaan kannalta. Asiakas ei mieti mitkä kustannukset palvelusta yritykselle ovat, sillä asiakas vertaa hinnan lisäksi hinta-laatusuhdetta toisiin vastaavien palveluihin. Hinnoittelua mietittäessä on tärkeää miettiä, minkälaisia asiakkaita hinnoitellulla tavoitellaan, kuinka ainutlaatuinen palvelu on, mitä asiakas olisi valmis palvelusta maksamaan, millaista mielikuvaa hinnalla halutaan

palvelusta luoda sekä mitkä ovat keskeiset hinnoittelumenetelmät ja poikkeavatko hinnat toisistaan palvelun eri elämänkaaren vaiheissa. (Jaakkola, Orava & Varjonen 2009, 33.)

Hinnoittelumenetelmät voidaan jakaa neljään ryhmään:

- Tuotosperusteinen hinnoittelu. Asiakas maksaa palvelun tuotoksesta kiinteän hinnan.
- Resurssipohjainen hinnoittelu. Hinnat perustuvat palveluun käytettyyn aikaan tai veloitukseen varatusta henkilö-, tila- tai laitekapasiteetista.
- Hyöty- ja arvoperusteinen hinnoittelu. Hinta määräytyy asiakkaan palvelusta saaman hyödyn perusteella.
- Käyttöoikeusperusteinen hinnoittelu. Asiakkaalle myydään jokin käyttöoikeus.

(Jaakkola, Orava & Varjonen 2009, 33.)

Yritys X:n ruokakassipalvelussa on seitsemän erilaista ja eri hintaista pakettiyhdistelmää. Kaikille asiakkaille on samat kiinteät hinnat valitun valmispaketin ja paketin koon mukaan. Hintaan vaikuttaa kuinka monelle hengelle ruokakassi on ja mitä kassi sisältää. Hinnoittelumalli viittaa tuottoperusteisen suuntaan. Asiakas maksaa tuotteesta kiinteän hinnan valintojensa perusteella. Toisaalta hinnoittelussa on hieman hyöty- ja arvoperusteista hinnoittelua, eri kassit ovat saadun hyödyn mukaan eri hintaisia.

Esittelen teorian ja käytännön kautta palvelupolun, empatiakartan sekä palveluketjuanalyysin (service blueprint) toimintaa sekä esittelen visuaalisesti yritys X:n kauppakassipalvelua näiden työkalujen avulla. Valitsin kohderyhmästä Matin tarkempaan tarkasteluun palvelupolkua, empatiakarttaa sekä palvelupolkukuvausta varten.

Palvelupolku on visuaalinen kuvaus kohderyhmän toiminnan eri vaiheista. Palvelupolku on helppo työkalu, jonka avulla voidaan kuvata asiakkaan matkaa palvelussa helposti ja nopeasti. Palvelupolkukuvauksessa kuvataan asiakkaan polkua kolmessa eri vaiheessa: ennen palvelua, palvelun aikana sekä palvelun jälkeen. Tarkoitus on ymmärtää asiakkaan matkaa palvelussa, jotta häntä voidaan palvella juuri oikealla tavalla oikeaan aikaan. (Väätäinen 2019.)

Palvelupolkukuvauksen avulla asiakkaan palvelukokemus tuodaan visuaaliseen muotoon ja sen keskiössä on asiakkaan kokemus eri vaiheista. Palvelupolussa haetaan kosketuspisteitä. Kosketuspisteitä voivat olla esimerkiksi fyysiset tilat, käyttöliittymät tai fyysinen palvelu. Kosketuspisteet kertovat miten yritys on vuorovaikutuksessa asiakkaan kanssa.

Palvelupolkua voidaan hyödyntää uuden palvelukokemuksen luontiin tai jo olemassa olevan palvelukokemuksen tutkimiseen ja parantamiseen. Palvelupolku on hyvä työkalu asiakaslähtöiseen kehittämiseen ja sitä voidaan myös käyttää pohjana palveluketjuanalyysin tekoon. (Aalto University 2019.)

Kuvaan yritys x:n kauppakassipalvelun palvelupolkua Solitan (Väättäinen 2019.) kirjoituksen avulla. Ensin mietin, mitä tapahtuu ennen palvelun ostoa. Tämä herätti ajattelemaan sitä, miksi asiakas tarvitsee palvelua ja mistä tarve kumpuaa. Yrityksen tulee vastata juuri asiakkaan tarpeeseen ja täten yrityksen tulee tuntea asiakkaansa ja heidän tarpeensa. Kohderyhmän henkilöllä Matilla on kiireinen arki ja hän toivoo arkeen helpotusta. Ruokapalvelun tilaaminen kotiin vapauttaa aikaa arjesta.

Tämän jälkeen mietin, mitä tapahtuu palvelun oston jälkeen. Esimerkiksi kohderyhmän edustaja Matti rientää illan viettoon vaimonsa luo tehtyään ruokatilauksen. Jos Matti ei tekisi tilausta hän menisi kauppaan ostamaan tarpeet ja suunnittelemaan menuun. Ostamalla palvelun, jossa raaka-aineet ja reseptit tuodaan kotiin, Matti osti samalla itselleen myös vapaa-aikaa ja näin yrityksen kannattaakin palvelua markkinoida.

Kolmanneksi mietin, miten asiakkaan matka palvelussa kulkee. Asiakas saa ensimmäisen kosketuksensa palveluun törmätessään yrityksen internetsivuihin tietoa etsiessään. Toisaalta ensimmäinen kosketus yritykseen voi olla jo kohdassa, jossa asiakas kuulee yrityksestä. Esimerkiksi jos asiakkaan ystävä on kokeillut palvelua ja kertoo kokemuksistaan. Toinen kosketuspinta syntyy, kun asiakas päättää mennä yrityksen internetsivuille ja aloittaa palveluun tutustumisen. Kolmas kosketuspinta syntyy, kun asiakas tekee tilauksen ja saa tilausvahvistuksen. Neljäs kosketuspinta syntyy, kun tilaus toimitetaan asiakkaan toimitusosoitteeseen ja kuljettaja ojentaa kassin asiakkaalle.

Kuvassa 6 on kuvattuna visuaalisesti yritys X:n ruokakassipalvelun palvelupolku. Palvelupolkuun tutustuminen ja palvelupolun visualisoiminen auttavat ymmärtämään palvelun käyttöä asiakkaan kannalta.

1.Ennen

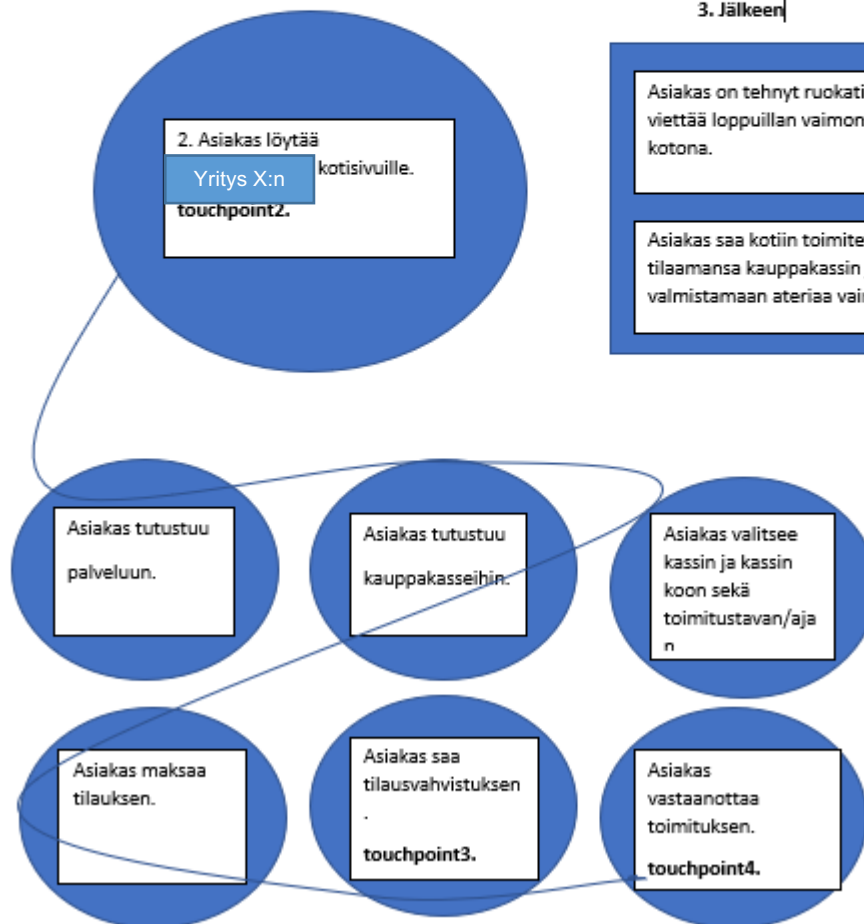
Asiakas lähtee töistä ja muistaa, että hän on luvannut hoitaa ensi viikon illalliset kotona. Kauppaan ei huvita mennä ja reseptien keksiminen on hankalaa. Asiakas juttelee asiasta töissä ja saa vinkkejä palveluista.

Asiakas muistaa kuulleensa ruuantilauspalveluista. Asiakas googlettaa palveluita. **Touchpoint1**

3. Jälkeen

Asiakas on tehnyt ruokatilauksen ja viettää loppuillan vaimonsa kanssa kotona.

Asiakas saa kotiin toimitettuna tilaamansa kauppakassin ja alkaa valmistamaan ateriaa vaimolleen.



Kuva 6. Yritys X:n palvelupolku sekä kosketuspinnat

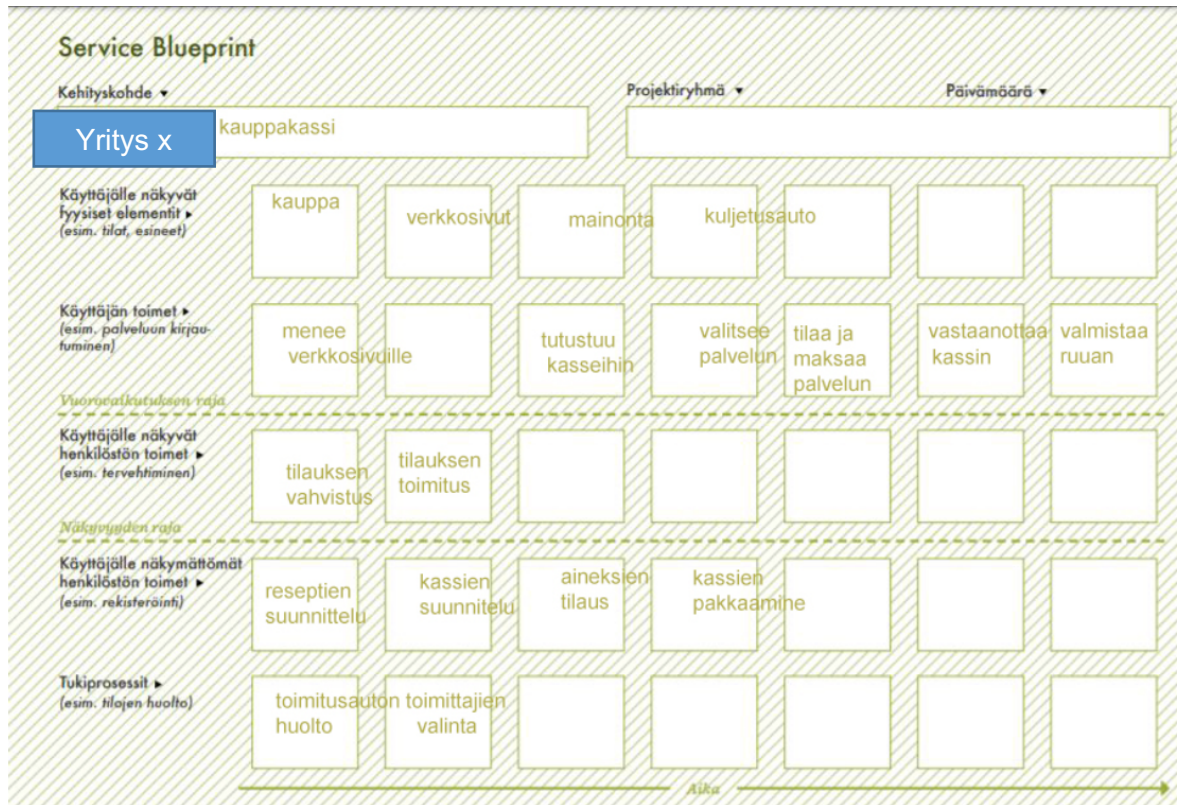
Palveluketjuanalyysi (service blueprint) tarkoittaa palvelusuunnitelmaa eli yksinkertaista kuvausta palvelun toiminnasta. Kuvaus näyttää palvelun lineaarisena toimintana. Kuvaus auttaa suunnittelemaan palvelun tuottamista huomioiden palvelutapahtumaan liittyvät toimet, kuten palvelukanavat, palvelusysteemit sekä palveluvälineet. Kuvaus jakaa tapahtumat näkyviin ja näkymättömiin tapahtumiin. Osa tapahtumista on asiakkaalle näkyvää ja

osa tapahtuu niin sanotusti konepellin alla. Kuvauksen avulla voidaan kuvata palvelun nykyistä tilaa tai tulevaisuuden visioita. (Aalto University 2019.)

Palveluketjuanalyysi kuvaa palvelun vaiheet tarkasti ja objektiivisesti. Kuvauksen tarkoituksena on kiinnittää huomiota kehittämistä ja analysointia tarvitseviin palvelun vaiheisiin sekä palvelun laadun kannalta kriittisiin kohtiin. Kuvauksen avulla voidaan hahmottaa, missä kohdin asiakkaan ja palveluntuottajan prosessit kohtaavat. (Jaakkola, Orava & Varjonen 2009, 16.)

Hyödynsin valmista palveluketjuanalyysi-pohjaa analysoinnissa kuvassa 7 (Aalto University 2019) havaitakseni niin näkyviä kuin piilossa olevia tapahtumia asiakkaalle sekä palvelun tukitoimenpiteitä yritys X:n kauppakassipalvelussa. Havaitsin, että vaikka moni asia on asiakkaalle piilossa, niin jollain tapaa asiat tulevat asiakkaalle esiin kuitenkin. Esimerkiksi reseptien suunnittelu ei näy asiakkaalle, mutta valmis resepti kyllä. Samoin kauppakassin pakkaaminen ei näy asiakkaalle, mutta kun asiakas saa kassin, niin pakkaamisen laatu näkyy asiakkaalle. Kassien kuljetusauton huolto ei näy asiakkaalle, mutta jos huolto jää tekemättä, niin seuraukset näkyvät asiakkaalle viivästyksinä toimituksissa.

Yrityksen verkkosivut on oltava niin innostavat, että asiakas siirtyy seuraavaan vaiheeseen, eli tutkimaan kassivaihtoehtoja sekä hintoja. Kasseista on löydettävä asiakasta kiinnostava vaihtoehto, jotta palvelun tilaaminen etenee. Toisaalta on mietittävä, että saako asiakas kaiken tarvitsemansa. Onko palvelun arvo asiakkaalle tarpeeksi suuri, jos hän joutuu palvelun tilaamisen lisäksi käymään kaupassa ostamassa jotakin puuttuvaa kaappiin. Tilausvahvistuksen oikeellisuus sekä kassin toimituksen onnistuminen luovat pohjaa uuden tilauksen tekemiselle.

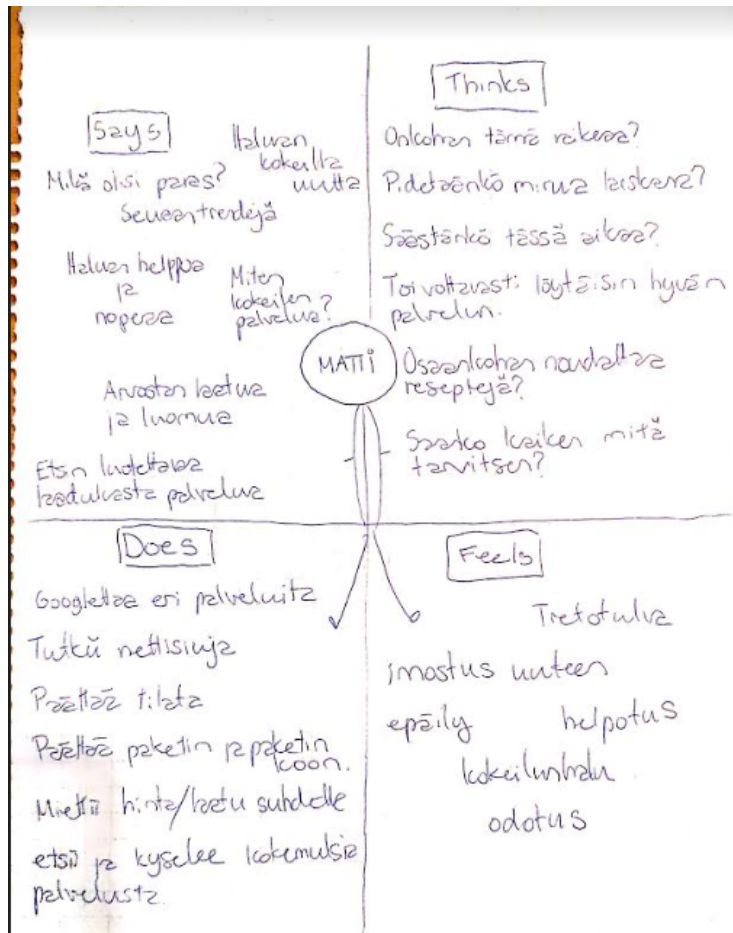


Kuva 7. Yritys X ruokakassipalvelun palveluketjuanalyysi (Aalto University 2019)

Prosessissa mietin asiakkaan mieltä empatiakartan avulla. Empatiakartan tavoitteena on ymmärtää asiakasta ja se on tärkeä osa asiakaslähtöistä suunnittelua, jotta yritys ymmärtää, mitä asiakas tarvitsee, haluaa ja odottaa. Empatiakartta on nopea, tehokas ja edullinen tapa kasvattaa asiakasymmärrystä. Työkalun avulla kasvatetaan asiakasymmärrystä tiimissä ja tiivistetään tietoa visuaaliseen muotoon. Empatiakarttaa on myös helppo muovata sen mukaan, mitä informaatioita kohderyhmästä on saatavilla. Ideana on hakea asiakkaan ajatuksia, puheita, tekoja ja tunteita saatavilla olevasta informaatiosta ja muodostaa niistä visuaalinen kuvaus asiakasymmärryksen kasvattamiseksi (Brown 2018.)

Kuvasin kohderyhmän edustajan Matin ajatuksia, puheita, tunteita ja tekoja empatiakartan avulla kuvaan 8. Visuaalisuuden avulla saadaan tarkempi kuva siitä, mitä käyttäjä kokee palvelupolun aikana ja miten tilannetta voisi kehittää.

Palvelupolkukuvauksen sekä empatiakartan avulla kuvasin visuaalisesti kohderyhmän palvelun käyttämistä ja käyttäjäkokemusta.



Kuva 8. Empatiakartta

Seuraavaksi avaan arvolutupauksen ja asiakasarvokartan teorian sekä sovellan teorioita yritys X:n kauppakassipalveluun. Tavoitteena on saada tarkemmin esiin kohderyhmän edustajan Mattin tarpeita ja miten yritys X:n kauppakassipalvelulla yritys pystyy näihin tarpeisiin vastaamaan.

Asiakkaan kokema arvo on keskeisessä osassa tuotekehitystyötä. Yritykset ovat alkaneet sisäistämään, että asiakkaan vallinnat eri tuotteiden välillä muodostuvat kasvavassa määrin tuotteen tai palvelun funktionaalisista hyödyistä ja valinnat painottuvat yhä enemmän asiakkaan arvioon yrityksestä ja ihmisistä sen takana. (Payne & Pennie 2013, 97.)

Arvolutupaus tarkoittaa suhdetta sen välillä mitä yritys tarjoaa ja mitä asiakas haluaa ostaa, määrittelemällä miten yritys vastaa asiakkaan tarpeeseen tuotteen tai palvelun käytön elinkaaren eri vaiheissa. Kaiken liiketoiminnan tavoitteena on luoda asiakkaille enemmän arvoa, mutta samalla kannattavammin kuin mitä kilpailijat luovat omilla tuotteillaan ja palveluillaan. (Payne & Pennie 2013, 98.)

Arvosuunnittelu auttaa yrityksiä suunnittelemaan, testaamaan ja toimittamaan asiakkaille juuri sitä, mitä asiakkaat haluavat ja tarvitsevat. Asiakasarvokartta on työkalu arvosuunnittelussa, joka auttaa yrityksiä luomaan arvoa asiakkailleen. Asiakasarvokartassa on kaksi eri puolta. Asiakaspuolen kanssa kirkastetaan asiakasymmärrystä ja asiakkaan tarpeita, haluja ja esteitä tarpeen täyttämiseen. Arvopuolella kuvataan, miten arvoa tuotetaan asiakkaalle (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 3.)

Arvokartta kuvaa yksityiskohtaisesti yrityksen arvolupauksen. Se purkaa arvolupauksen tuotteeksi ja palveluksi sekä kivunlievittäjiksi ja hyödynluojiksi. Hyödynluojat kuvaavat, miten yrityksen palvelu tai tuote luo asiakkaalle hyötyä. Kivunlievittäjät kuvaavat, miten tuote tai palvelu helpottaa asiakkaan ongelmia. Asiakasprofiili kuvaa tietyn asiakasprofiilin yrityksessä jäsenellyllä ja yksityiskohtaisella tavalla. Se avaa asiakkaan ajatukset ja jakaa ne asiakkaan töiksi, kivuiksi ja saavutuksiksi. Asiakkaan tavoitteet kuvaavat sitä, mitä asiakas pyrkii saavuttamaan. Kivut kuvaavat huonoja seurauksia, huolia, riskejä ja murheita, joita asiakkaalla liittyy tavoitteen suorittamiseen. Hyödyillä kuvataan mitä asiakas pyrkii saavuttamaan. Yritys onnistuu luomaan yhteensopivuutta asiakkaan kanssa, jos yrityksen arvokartta kohtaa asiakkaan profiilin tarpeiden kanssa. Toisin sanoen, kun tuote tai palvelu vastaa ja ratkaisee yhden tai useamman asiakkaan tavoitteen, kipukohdan tai saavutuksen. (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 7-10.)

Arvokartta auttaa yrityksiä luomaan arvoa asiakkailleen. Kuvassa 9 on yritys X:n kauppakassipalvelu kuvattuna arvokartan avulla. Tarkasteluun on valittu kohderyhmän edustaja Matti.



Kuva 9. Yritys X:n kauppakassipalvelun arvokartta (Helsinki 2019)

Asiakkaan tavoitteilla, jotka halutaan saada suoritettua, tarkoitetaan asioita, joita yrityksen asiakas pyrkii saamaan suoritetuksi työssään tai elämässään ylipäättänsä. Asiakkaan tavoite voi olla esimerkiksi tehtävä, joka hänen tulee suorittaa, ongelma, mikä hänen tulee ratkaista tai tarve, jonka hän haluaa tyydyttää. Asiakkaan tavoitteet tulevat tutkia yrityksen hallussa olevasta kohderyhmän asiakasdatasta. Asiakkaan tavoitteet voidaan jakaa funktionaalisiin (esimerkiksi asiakkaan pitää ajaa autolla Lappiin), sosiaalisiin tavoitteisiin (halu näyttää tietynlaiselta ulospäin, saavuttaa statusta tai valtaa) ja persoonallisiin tai emotionaalisiin tavoitteisiin (asiakas pyrkii tiettyyn tunnetilaan esimerkiksi vapaudentunteeseen tai mielenrauhaan). (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 12-13.)

Kohderyhmän edustaja Matti haluaa terveellistä ruokaa vähällä vaivalla kotiin toimitettuna. Hänellä on siis toive ja tarve ratkaistavaksi. Matti haluaa vähentää aikaa, jonka hän käyttää aterioiden suunnitteluun sekä kaupassa käyntiin ja suunnata tarmonsensa sen sijaan vaimon kanssa vietettyyn aikaan. Matilla on kiireinen arki ja hän haluaa lisätä laatu-aikaa arjessa. Matti haluaa vähentää kiireen tuntua arjessa.

Asiakkaan saamat hyödyt ja arvo kuvaavat lopputulosta ja hyötyjä, joita asiakas haluaa saavuttaa. Jotkut näistä ovat ehdottomia ja vaadittuja, mutta osa voi olla sellaisia, joista asiakas yllättyisi iloisesti. Hyödyt ja arvot sisältävät funktionaaliset hyödyt, sosiaaliset hyödyt, positiiviset tunnetilat sekä kulusäästöt. (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 16.)

Matti haluaa saavuttaa lisää vapaa-aikaa ja vähentää kaupassa käyntiä. Tätä kautta syntyy myös säästöä, kun heräteostokset vähenevät. Matti haluaa helpotusta arjen ruokahuoltoon. Monipuolinen ruokavalio ja ruuan laatu ovat Matille perheineen tärkeitä ja se on yksi isoista toiveista, joita Matti kaipaa palvelulta.

Asiakkaan huolet ja murheet kuvaavat kaikkea sitä, mikä ärsyttää asiakasta ennen, jälkeen ja tavoitteen suorittamisen aikana tai yksinkertaisesti asiaa, joka estää asiakasta suoriutumasta tavoitteestaan. Asiakkaan huolet ja murheet kuvaavat myös sitä riskiä, mikä on olemassa, jos asiakas ei saa tavoitetta suoritettua. Huolet ja murheet voidaan jakaa epätoivottuun lopputulokseen, ongelmiin, esteisiin sekä riskeihin. (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 14.)

Mattia mietityttää ruuantilauspalvelussa tuleeko toimitus ajallaan ja kohtaako hinta-laatusuhde. Matti miettii myös löytääkö hän aina mieluisan vaihtoehdon ja mitä jos hän haluaa ostaa jotain ylimääräistä tietyllä viikolla. Hän miettii osaako hän tehdä aineksista reseptien mukaisen aterian vai ovatko reseptit liian monimutkaisia.

Tuotteet ja palvelut tarkoittavat yksikertaisesti sitä, mitä yritys tarjoaa. Esimerkiksi tämä voi tarkoittaa kaikkia niitä tavaroita, joita asiakas näkee yrityksen näyteikkunassa. Tuote voi olla esimerkiksi kotiin tilattava kauppakassipalvelu. On tärkeää muistaa, että tuote tai palvelu ei yksistään luo arvoa. Arvonmuodostus tapahtuu, kun tuote tai palvelu yhdistyy tietyn asiakassegmentin ja heidän tavoitteidensa sekä kipupisteidensä kanssa. (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 30.)

Yritys X:n kauppakassipalvelu tarjoaa reseptit ja raaka-aineet valmiina kassissa kotiin toimitettuna tai kaupalta noudettuna. Asiakas voi valita useista ruokakassi vaihtoehtoista itselleen sopivan vaihtoehdon. Palvelu sisältää myös sesonkivaihtoehtoja. Yritys X:n ruokakassi on helppo tapa kokeilla uusia reseptejä sekä saada uusia ideoita arjen ruokahuoltoon. Laadukkaat raaka-aineet tulevat kotiin kuljetettuina ja selkeiden reseptien avulla asiakas saa loihdittua herkullisia aterioita. Asiakas voi valita sopivan ruokakassin kerta- tai kestopalveluksella. Asiakas voi päättää tilausrytmistä ja valita vaikka joka viikko eri kassin.

Yrityksen arvontuottajat kuvaavat, miten yrityksen tuotteet ja palvelut luovat asiakkaalle arvoa. Nämä arvontuottajat linjaavat selkeästi, miten yritys aikoo tuottaa asiakkaalle sen mitä asiakas toivoo, tarvitsee ja haluaa. Arvontuottajia pohiessa on hyvä muistaa, ettei yritä tyydyttää kaikkia asiakkaan tarpeita vaan keskittyy siihen, mikä on kohderyhmälle tärkeä ja ratkaiseva asia palvelua tai tuotetta valitessa. (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 33-34.)

Yritys X:n kauppakassipalvelun avulla kaupassa vietetty aika vähenee, ruokavalio monipuolistuu ja yhdessä ruokapöydässä vietetty aika lisääntyy. Palvelu tarjoaa kotiin toimitettuna tuoret ja laadukkaat raaka-aineet ja tarkasti laaditut reseptit. Kasseissa on mahdollisimman paljon lähi- ja luomuruokaa sekä yksinkertaisia ja helppoja reseptejä. Kassi sisältää aina tuoretta kalaa, lihaa, leipää ja kasviksia, sillä ruokakassi pakataan toimituspäivänä. Kassin sisältöön nähden tuote on edullinen.

Yrityksen tarjoamat murheenpoistajat kuvaavat sitä, miten yrityksen tuotteet ja palvelut vastaavat asiakkaan tiettyyn murheeseen. Murheenpoistajat kuvaavat, miten yritys aikoo eliminoida tai vähentää asiakkaan kokemia esteitä. Murheenpoistajissa on hyvä keskittyä poistamaan tai lievittämään esteitä asiakkaalle tärkeissä asioissa. Kaikkia esteitä ei tarvitse saada korjatuksi. (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 31.)

Yritys X:n kauppakassi toimitetaan tiistaisin asiakkaan toivomaan osoitteeseen ja asiakas halutessaan voi myös noutaa tilaamansa kassin yrityksen kaupoilta. Toimitus tapahtuu

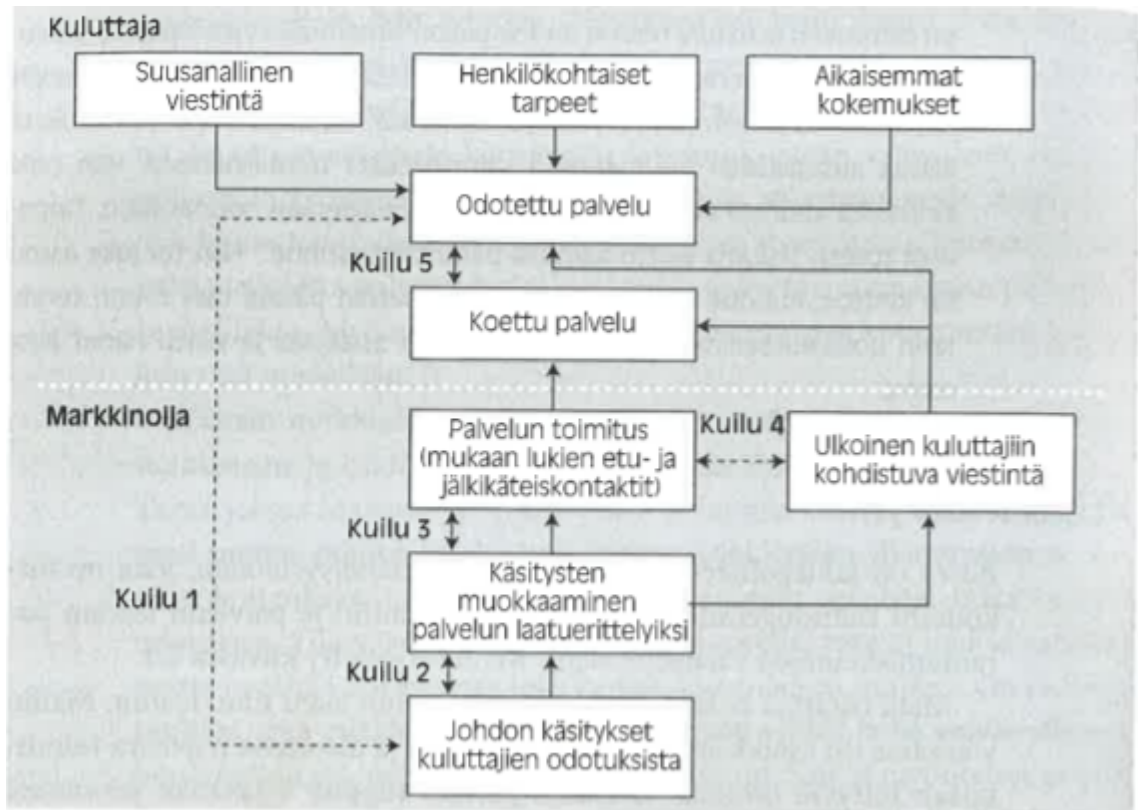
asiakkaan valitsemana toimitusaikana. Palvelu on asiakkaiden suosittelu ja referenssejä löytyy yrityksen nettisivuilla. Palvelu tarjoaa viikoittain vaihtuvat kassit. Tarjonnassa on myös sesonkikasseja. Yritys X:n kauppakassi sisältää tuoretta, laadukasta lähiruokaa, helposti kotiin toimitettuna. Tilaaminen on tehty helpoksi internetsivujen kautta ja asiakas saa aina vahvistuksen tilauksesta. Yrityksen kotisivuilla on myös ohje tilauksen tekoon.

Seuraavaksi projektissa paneudun palvelun laatuun. ”Tärkeää on laatu sellaisena kuin asiakas sen kokee” (Grönroos 2009, 100). Laatu luo yritykselle uskollisia asiakkaita. Erinomainen palvelu maksaa itsensä takaisin moninkertaisena, sillä erinomainen laatu luo uskollisia asiakkaita. Uskolliset asiakkaat käyttävät yrityksen palveluita uudelleen ja uskolliset asiakkaat kehuvat palvelua edelleen. (Zeithaml, Parasuraman & Berry 1990, 9.)

Asiakkaan kokema laatu koostuu kolmesta eri osasta: yrityksen tai sen osan imagosta, lopputuloksen laadusta (mitä) ja prosessien toiminnallisesta laadusta (miten) (Grönroos 2009, 103). Laatu on yksi yritysten avaintekijöistä. Yrityksen kilpailuedun sanotaan olevan riippuvaisia sen tarjoamien tuotteiden ja palveluiden laadusta ja arvosta asiakkaalle. Yrityksen menestykseen ei riitä erinomaiset tekniset laadut, vaan erottautumiseen markkinoilla tarvitaan erinomaisesti johdetut ja hoidetut vuorovaikutustilanteet, eli kosketuspisteet asiakkaan kanssa. (Grönroos 2009, 104.)

Asiakkaan kokema kokonaislaatu muodostuu odotetun laadun ja koetun laadun kokonaisuudesta. Laatu on hyvä, jos asiakkaan kokema laatu vastaa hänen odotuksiaan palvelun tai tuotteen laadusta. Jos asiakkaan odotukset ovat epärealistisia, niin asiakkaan kokema kokonaislaatu on alhainen, vaikka muilla mittareilla laatu voi olla koettu hyväksi. (Grönroos 2009, 105.)

Laatukuilumalli on tarkoitettu laatuongelmien lähteiden analysointiin ja laadun parantamisen keinojen hakemiseen (Grönroos 2009, 143). Kuvassa 10 on palvelun laadun kuiluanalyysinmalli.



Kuva 10. Palvelun laatuhiilumalli (Grönroos 2009, 144)

Laatuhiilumallin kautta saadaan esiin, miten palvelun laatu muodostuu. Mallin yläosassa näkyy asiakkaaseen liittyviä ilmiöitä ja alaosassa palvelun toimintaan liittyviä ilmiöitä. Mallissa näkyvä odotettu palvelu riippuu asiakkaan aikaisemmista kokemuksista sekä henkilökohtaisista tarpeista. Siihen vaikuttavat myös yrityksen markkinointitoimenpiteet. Koettu palvelu syntyy yrityksen sisäisten päätösten sekä niihin liittyvien toimenpiteiden seurauksena. Asiakas kokee palvelua käyttäessään laadun kaksi tekijää: palvelun toimitus- ja tuotantoprosessin (toiminnallinen laatu) ja prosessin seurauksena olevan ratkaisun lopputuloksen laadun. Laatuhiilumallin avulla saadaan selville mitä toimenpiteitä on harkittava palvelun laadun parantamiseksi. (Grönroos 2009, 143-144.)

Mallissa on viisi perusrakenteen osien välistä laatuhiilua. Laatuhiilut ovat seurausta laadunjohtamisprosessin epäjohtonmukaisuuksista. Lopullinen hiilu, eli odotetun ja koetun palvelun välinen hiilu, on riippuvainen muista prosessin hiiluista. (Grönroos 2009, 144.)

Hiilu yksi on johdon näkemyksen hiilu. Tämä hiilu tarkoittaa sitä, että johto näkee laatuodotukset puutteellisesti. Syitä tähän voivat olla puutteelliset tiedot tutkimuksista, virheellisesti tulkitut tiedot odotuksista tai esimerkiksi, että johto saa puutteellista tietoa organisaatiolta. (Grönroos 2009, 144-145.)

Kuilu kaksi tarkoittaa sitä, että palvelun laatuvaatimukset eivät ole yhdenmukaisia johdon laatuvaatimusten kanssa. Syitä tähän saattavat olla esimerkiksi suunnitteluvirheet, organisaation puutteelliset tavoiteasettelut tai puutteellinen ylimmän johdon tuki. (Grönroos 2009, 145-146.)

Kuilu kolme tarkoittaa sitä, että palvelun tuotanto- ja toimitusprosessissa ei noudateta laatuvaatimuksia. Syitä tähän voivat olla liian jäykät vaatimukset tai se, että yrityksen laatuvaatimukset eivät kohtaa yrityksen työntekijöiden vaatimusten kanssa. Myös sisäinen markkinointi voi olla puutteellista tai käytössä oleva tekniikka ja järjestelmät eivät tue vaatimusten mukaista toimintaa. (Grönroos 2009, 146 – 148.)

Kuilu neljä tarkoittaa sitä, että markkinointiviestinnässä annetut lupaukset eivät kohtaa asiakkaan saamaan palvelun kanssa. Syitä tähän voivat olla, että markkinointia ja tuotantoa ei ole koordinoitu riittävästi, organisaatiossa ei noudateta sitä, mitä asiakkaalle on markkinointiviestinnässä luvattu tai on luvattu liikoja ja liioiteltu asiakkaan saamaa arvoa ja laatua. (Grönroos 2009, 148.)

Kuilu viisi tarkoittaa, että koettu palvelu ei ole yhdenmukainen odotetun palvelun kanssa. Tästä seuraa se, että asiakas kokee laadun huonona ja luultavasti myös jättää suosittelematta palvelua. Seurauksena voi olla kielteinen vaikutus yrityksen ja palvelun imagoon ja pahimmillaan jopa liiketoiminnan menetys. Viides kuilu voi olla myös positiivinen, jolloin tämä johtaa hyvään tai jopa liian hyvään laatuun. Koetun palvelun kuilu voi johtua yhdestä tai useammasta edellä mainituista kuiluista. Kuiluanalyysi on tapa löytää epä johdonmukaisuuksia toimittajan ja asiakkaan näkemysten välillä. (Grönroos 2009, 149.)

Lähdin sovittamaan kohderyhmän edustajasta Matista projektin aikana keräämääni tietoa palvelunlaatu kuilumalliin ja vertasin yrityksen markkinoimaa ja toimittavaa palvelua kohderyhmän edustajan käsityksiin palvelusta, odotuksiin palvelusta sekä hänen tarpeisiinsa ja haluihinsa. Kuvassa 11 on laatu kuilumalliin sovellettuna kohderyhmän edustajan Matin kokemuksia. Sovittamalla kohderyhmän edustajasta kerättyä tietoa laatu kuilumalliin sain tuotua esiin haasteita, joita mielestäni olisi hyvä lähteä ratkaisemaan, jotta koettu palvelunlaatu kasvaisi ja oltaisiin lähempänä ylivertaista asiakaskokemusta.



Kuva 11. Laatukuilumalli sovellettuina kerättyyn tietoon kohderyhmän edustajasta (Grönroos 2009, 149)

Asiakas odottaa yrityksen markkinoinnin perusteella nykyaikaista, modernia ja laadukasta palvelua, joka vapauttaa häneltä aikaa arjesta. Pohtiessani kuiluja neljä ja viisi huomasin, että vaikka palvelun laatu on hyvää ja se kohtaa asiakkaan odottaman laadun kanssa niin puutoksia laadussa on silti. Modernia palvelua käyttäessä kiireinen asiakas haluaa käyttää mobiilisovellusta palvelun tilaamiseen. Mobiili on asiakkaalla käytössä jatkuvasti, mutta tietokone ei. Markkinoinnin perusteella on syytä olettaa, että nykyaikainen väline palvelun tilaamiseen löytyy mutta tutkimuksien perusteella asiakas voi tilata palvelun vain selaimen kautta. Tämä puutos mielikuvan ja todellisuuden välillä voi muodostua isoksi ongelmaksi kohderyhmässä.

Toiseksi haasteeksi muodostuu se, että vaikka asiakas saa tilattua kassin kotiin ja sitä kautta hän saa ateriat ja reseptit, niin hän ei voi tilata tilauksen mukana esimerkiksi maitoa tai suklaata. Tämä aiheuttaa sen, että asiakas joutuu uhraamaan vapaa-aikaansa kaupassa käyntiin kuitenkin tai hän joutuu olemaan ilman lisätuotteita. Myöskään kassivaihtoehtoja ei ole kuin muutamat. Esimerkiksi vegaanikassi tai kassi erikoisruokavaliolle kuten laihduttajalle tai diabeetikolle puuttuu valikoimasta, vaikka markkinoinnin perusteella asiakkaalle välittyy mielikuva monipuolisesta tarjonnasta. Kolmanneksi kun asiakas tilaa pal-

velun hän saa tilauksen tekemiseen hyvän ohjeistuksen ja tilausvahvistuksen, mutta jälkihuolto puuttuu. Palvelun toimituksen jälkeen asiakkaalle olisi hyvä lähteä nopea ja helppo palautekysely.

Kohderyhmän edustajan tarpeiden mukaan havaitut haasteet on helppo asettaa tärkeysjärjestykseen. Ensimmäisenä on tärkeä korjata kanava, jonka avulla tilaus saadaan tehtyä. On hyvä, että tilauksen voi tehdä selaimessa, mutta kohderyhmän edustajan tarpeet ja odotukset nykyaikaiselta palvelulta vaativat ehdottomasti mobiilisovelluksen. Pidemmän päälle palvelun käyttö takkuu ilman modernia palvelun tilauksenteon välinettä.

Toiseksi priorisoinnissa nousee palvelun sisällön kehittäminen. Asiakkaan aikaa säästää se, jos hän saa kaiken tietyn tarpeen tyydyttämiseen yhdestä paikasta.

Kolmanneksi tulisi pohtia palautteenannon kanavaa. Yrityksellä on hyvin esillä yhteystietoja palautteenannolle, mutta kohderyhmän edustaja ei lähde niitä erikseen etsimään. Palaute tulisi voida antaa palvelun käytön jälkeen automaationa helposti.

Yritys X:n kauppakassipalvelussa löysin haasteita kuiluissa neljä ja viisi. Asiakkaan markkinoinnissa saama mielikuva ei täysin vastaa saatua palvelutuotetta.

Koin opintojakson erittäin mielenkiintoiseksi ja opin paljon lisää palvelumuotoilusta. Jokainen opintojaksolla käytetty työkalu oli minulle projektia tehdessä uusi. Koin mielekkään ensin opetella teoriaa aiheesta ja sen jälkeen soveltaa oppimaani valitsemaani palvelutuotteeseen. Näin teorian sai heti nivottua yhteen käytännön kanssa.

Raportin ensimmäisessä kappaleessa esittelin tuplatimanttimallin. Esittelen lopuksi vielä kiteytettynä, että mitä palvelumuotoilun työkaluja ja menetelmiä käytin projektissani tutkiessani yritys X:n kauppakassipalvelua.

Nyt olen toteuttanut projektin ensimmäisen vaiheen. Projektin ensimmäisessä vaiheessa hankin tietoa käyttäjistä ja tiedon perusteella muodostin käyttäjäpersoonat. Tässä vaiheessa hyödynsin jo olemassa olevaa tietoa käyttäjistä, havainnointia sekä suoritin pienen kyselyn lähipiirissäni oleville kohderyhmän edustajille. Tämän jälkeen ensimmäisessä vaiheessa aloitin ongelman määrittelyn jatkamalla asian tutkimista käyttäjälähtöisesti.

Opintojakson puitteissa tein projektin vaiheen yksi. Seuraava olisi vuorossa vaihe kaksi. Ensin vaiheessa kaksi ideoidaan esiin nousseiden haasteiden perusteella asiaa korjaavia

ideoita. Tässä hyödyntäisin ideoiden tiimissä kehittämistä ja tekisin vielä tarkempaa kyselyä asiakkaille. Seuraavaksi visualisoisin, luonnostelisin ja jalostaisin ideoita, joita on noussut esiin tarkemmissa tutkimuksissa. Testaisin ja validoisin ideoita ja loppujain lopuksi käyttöönottaisın parhaan, jalostetun idean ja parantaisin palvelun laatua sekä asiakaskokemusta.

Mielestäni onnistuin haastavassa opintojaksossa erittäin hyvin. Opintojakso oli erittäin laaja yksilötyönä suoritettavaksi. Sovelsin hyvin ja innovatiivisesti palvelumuotoilun prosessia sekä työvälineitä valitsemaani palvelutuotteeseen ja sen liiketoimintaympäristöön. Olisi ollut erittäin mielenkiintoista päästä prototyypivaiheeseen havaitsemieni haasteiden kanssa opintojakson puitteissa. Jatkokehityslistalle jääkin vaiheen kaksi tekeminen projektille.

Sain hyvin projektissa haettua asiakasryhmäkohtaista tietoa ja asiakasymmärrystä esiin ja sitä kautta sain haettua haasteita ja konkreettisia kehitysehdotuksia palvelukokemuksen parantamiseksi. Sain myös hyvin perusteltua, miten ja miksi valitsemani palvelutuotteen laatua tulisi asiakaslähtöisesti kehittää. Suoritin opintojakson yksilötyönä ja aikataulussa. Kehitysehdotuksena parantaisin projektini visuaalisuutta ja tekemällä projektista visuaalisen tiivistelmän. Työ on laajuuden, kattavuuden ja onnistuneen sisällön vuoksi erittäin hyvä projekti portfolioon valittavaksi.

Opettajan antama palaute projektityöstäni oli seuraava:

” tehtävä on suoritettu erinomaisella tarkkuudella ja arviointikriteerit huomioiden. Työssäsi korostuu lähteiden ja tehtävien välillä käyty syvä pohdinta. Lähteitä on käytetty erinomaisella tavalla ja olet löytänyt hyviä lähteitä. Oivallukset ja opit nousevat esiin hyvin ja sinulla vahva kyky ilmaista uusia ajatuksia. Kirjallisten lähteiden käyttö on kattavaa. Käsittelet uusien asioiden kohdalla teoriataustaa ja sovellat tehtäviä teoriaan peilaten. Kirjallinen ulosanti on erinomaista ja Haaga-Helien raportointipohjaa noudattavaa. Visualisoi myös hyvin vastauksesi. Asiakaspolku, Empatiakartta, Value Proposition Canvas on kuvattu johdonmukaisesti ja olet osannut yhdistää tehtävän personan näkökulmaan. Laatuksiluanalyysi on kattava, teoriaa ja personan näkökulmaa yhdistävä. Lisäksi kehitysehdotukset on kiteytetty loistavasti ja oma pohdinta rikasta. Loistavaa työtä!” (Moodle 2019f.)

3.3 Käyttäjäkokeemus ja käyttöliittymäsuunnittelu

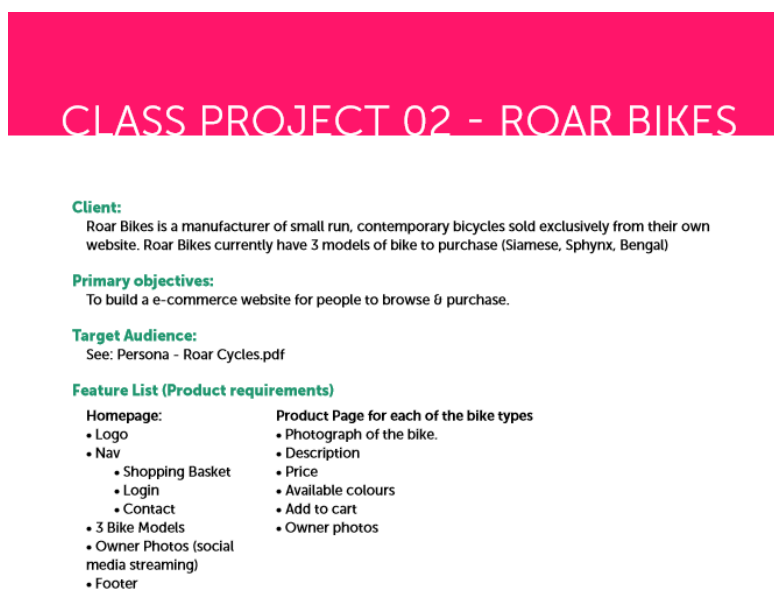
Roar Bikes Bikestore-projekti on Adobe XD-suunnitteluohjelmalla toteutettu käyttäjäkokeemus- (UX) ja käyttöliittymäsuunnitteluprojekti (UI). Projekti on tehty kurssityönä verkko-opintojaksolla User Experience Design Essentials - Adobe XD UI UX Design Udemy.comissa.

Valitsin projektin mukaan portfolio-opinnäytetyöhön, sillä projektissa tulee käyttäjäkokeuksen lisäksi esiin suunnittelutyössä tarvittavan Adobe XD-ohjelmiston hallinta sekä kykyä niin käyttäjäkokeuksen kuin visuaalisen ilmeen suunnittelijana.

Tavoitteena oli oppia lisää käyttäjälähtöisestä käyttöliittymäsuunnittelusta sekä Adobe XD-suunnitteluohjelman käytöstä. Hyödynsin projektissa niin opintojakson materiaalia kuin aikaisemmin keräämiäni oppeja käyttäjäkokeuksen, käyttöliittymäsuunnittelun ja palvelumuotoilun keinoista.

Projektin tavoite oli tuottaa suunnitelma verkkosivusta yritykselle Roar Bikes Bikestore. Yritys on pieni, uniikkeja pyöriä valmistava ja myyvä yritys. Yritys myy tuotteitaan vain verkkosivujensa kautta. Yrityksellä on tällä hetkellä myynnissä kolme eri pyörä mallia: Siamese, Sphynx ja Bengal.

Kuvassa 12 ovat projektin tavoitteet ja konseptin haluttu sisältö. Projektin tavoitteena oli tuottaa Roar Bikes Bikestorelle suunnitelma verkkosivusta, jossa asiakkaat voivat tutustua yritykseen ja ostaa yrityksen tuotteita. Projektin tavoitteena oli luoda suunnitelma yrityksen verkkosivujen pääsivusta sekä tuotesivusta.




Kuva 12. Roar Bikes Bikestoren projektikuvaus

Yrityksen kohderyhmänä ovat nuoret, kolmenkymmenen vuoden korvilla olevat, ympäristöä ja uniikkeja tuotteita arvostavat miehet. Ensimmäiseksi projektissa tutustin sekä kohderyhmään että heidän tarpeisiinsa ja toteutin pienen tutkimuksen kilpailijoiden vastaaviin tuotteisiin. Näin sain kerättyä laajasti tietopohjaa tuotteen suunnittelua varten.

Kohderyhmän edustaja visualisoidaan käyttäjäpersoonan avulla. Käyttäjäpersoonana on esimerkki kuvitteellisesta palvelun käyttäjästä. Persooniin liitetään tietoja, joita on kerätty käyttäjistä eli kohderyhmästä. Käyttäjäpersoonat heräävät eloon kehittäjien mielessä ja ne on hyvä kuvata niin visuaalisesti kuin sanallisesti. (Anne 2011.) Käyttäjäpersoonan visuaalinen kuvaus auttaa suunnittelijaa pitämään mielessä sen kenelle tuotetta suunnitellaan.

Kuvassa 13 on Roar Bikes Bikestoren käyttäjäpersoonana Jake. Jake on 28-vuotias graafinen suunnittelija Kaliforniasta. Hän arvostaa uniikkeja tuotteita, joita ei kaupungilla tule koko ajan vastaan. Hän arvostaa pieniä yrittäjiä suurien korporaatioiden sijaan ja on ympäristöystävällinen arjessaan. Jake ei omista autoa vaan hän pyöräilee joka paikkaan ja siksi hän tarvitsee käytännöllisen, mutta tyyliinsä sopivan uniikin pyörän.



Jake 28 / GRAPHIC DESIGNER / CALIFORNIA, USA

DESCRIPTION

Jake is a graphic designer for a large design agency in Los Angeles. Jake prides himself on having strange & interesting versions of everyone else's everyday objects. His shoes are hand made & his backpack is pink.

He likes to make his purchase (big or small) from local responsible crafts people. He doesn't own a car. He always brings a reusable cup to the coffee shop.

Jake cycles to work and wants his bicycle to be practical but unique.

"I like things that are well made & unexpected."

Kuva 13. Jake, Roar Bikes Bikestoren kohderyhmän edustaja

Tutustuin kohderyhmän tarpeisiin käyttäjäpersoonan avulla. Lähdin hakemaan tunnelmaa, joka houkuttelisi asiakkaan tutustumaan Roar Bikes Bikestoren verkkosivuun tarkemmin sekä tekemään ostoksia verkkosivulla. Kilpailijoiden sivuilta tutkin, mitkä asiat heidän sivuillaan toimivat tai eivät toimineet asiakaskokemuksen kannalta.

Hyödynsin verkkosivun suunnittelussa arvokarttaa (value proposition canvas). Arvosuunnittelu (value proposition design) auttaa yrityksiä suunnittelemaan, testaamaan ja toimittamaan asiakkaille juuri sitä mitä asiakkaat haluavat ja tarvitsevat. Arvokartta on työkalu arvosuunnittelussa ja se auttaa yrityksiä luomaan arvoa asiakkailleen. Arvokartassa löytyy kaksi eri puolta. Asiakaspuolen kanssa kirkastetaan asiakasymmärrystä ja asiakkaan tarpeita, haluja sekä esteitä tarpeen täyttämiseen. Arvo puolella kuvataan, miten arvoa tuotetaan asiakkaalle. (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 3.)

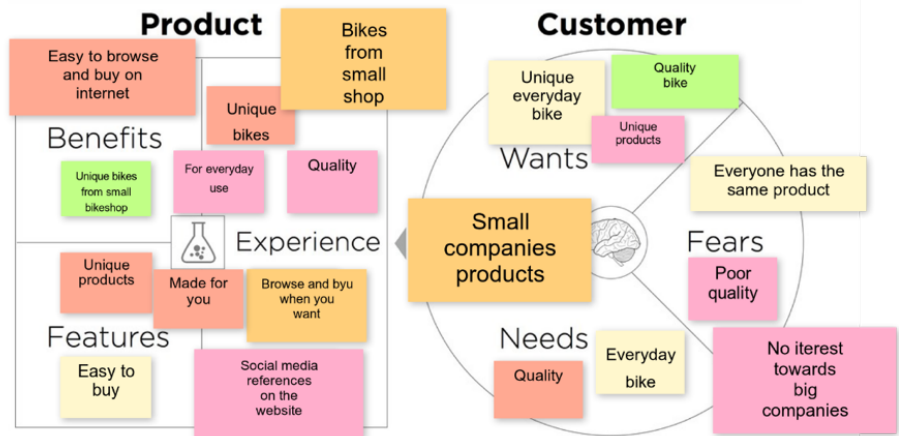
Arvokartta kuvaa yksityiskohtaisesti yrityksen arvolupauksen. Se purkaa arvolupauksen tuotteeksi ja palveluksi, kivun lievittäjiksi ja hyödyn luojiksi. Hyödyn luojat kuvaavat, miten yrityksen palvelu tai tuote luo asiakkaalle hyötyä. Kivunlievittäjät kuvaavat, miten tuote tai palvelu helpottaa asiakkaan ongelmia. (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 3.)

Asiakasprofiili kuvaa tietyn asiakasprofiilin yrityksessä jäsennellyllä ja yksityiskohtaisella tavalla. Se avaa asiakkaan ajatukset ja jakaa ne asiakkaan töiksi, kivuiksi ja saavutuksiksi. Asiakkaan tavoitteet kuvaavat sitä, mitä asiakas pyrkii saavuttamaan. Kivut kuvaavat huonoja seurauksia, huolia, riskejä ja murheita, joita asiakkaalla liittyy tavoitteen suorittamiseen. Hyödyillä kuvataan mitä asiakas pyrkii saavuttamaan. (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 7-10.)

Yritys onnistuu luomaan yhteensopivuutta asiakkaan kanssa, jos yrityksen arvokartta kohtaa asiakkaan profiilin tarpeiden kanssa. Kun tuote tai palvelu vastaa ja ratkaisee yhden tai useamman asiakkaan työn tavoitteen, kipukohdan tai saavutukseen. (Osterwalder, Pigneur, Bernarda & Smith Hoboken 2014, 7-10.)

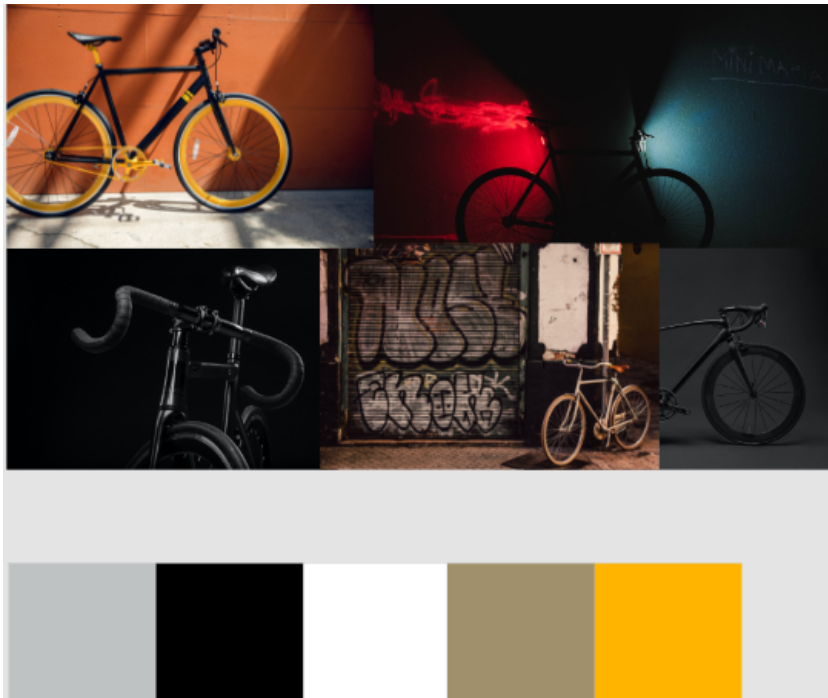
Kuvassa 14 on visuaalisessa muodossa Roar Bikes Bikestore-projektin arvokartta. Asiakkaan halut, tarpeet ja pelot on kuvattu käyttäjäpersoonan Jaken kannalta katsottuna.

Value Proposition Canvas



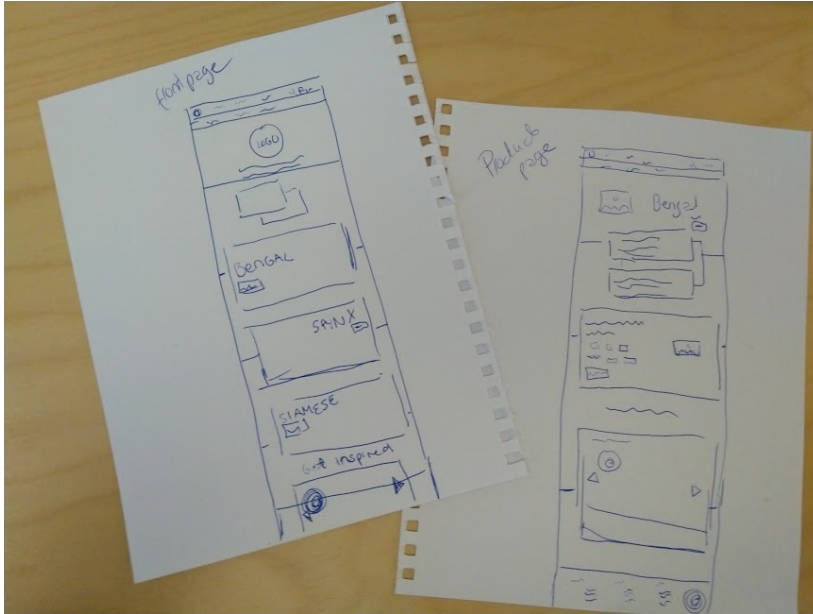
Kuva 14. Roar Bikes-projektin arvokartta (awwapp)

Ennen palvelun luonnostelua lähdin väri- ja kuvamaailman avulla suunnittelemaan palvelun tunnelmaa. Käyttäjäpersoonan Jaken perusteella päädyin hakemaan Berliinimäistä tunnelmaa, persoonallista ja uniikkia. Moodboardin (tunnekuvaus) (kuva 15) avulla visualisoin tuotteen tunnelmaa ja värimaailmaa. Moodboardin avulla kuvataan tuotteen visuaalista tunnelmaa keräämällä yhteen visuaalisia elementtejä, jotka kuvaavat projektin tavoiteltua tunnelmaa.



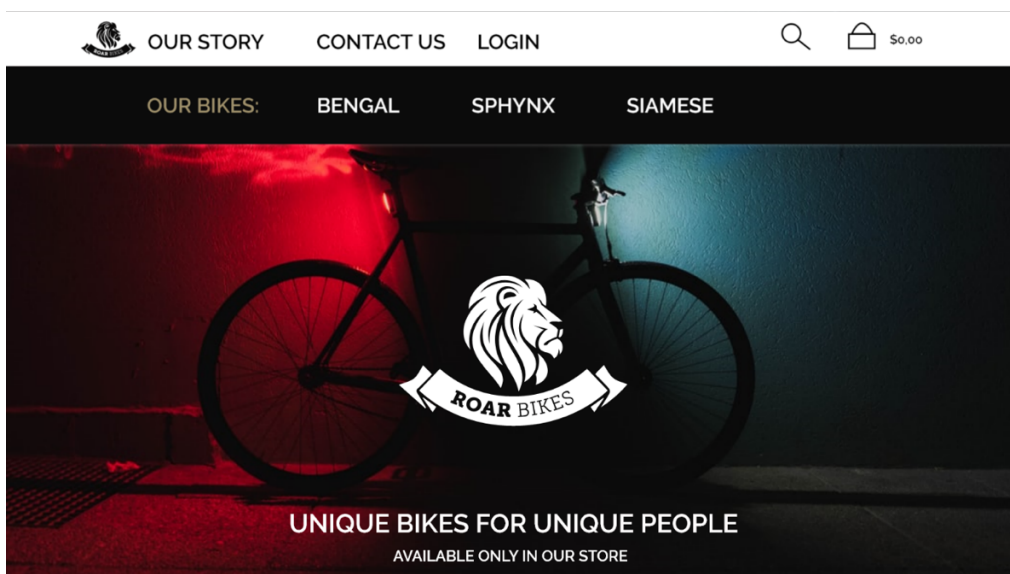
Kuva 15. Roar Bikes Bikestore-projektin moodboard tunnekuvaus

Taustatyön tehtyäni lähdin tekemään ensimmäisen vaiheen prototyyppiä (luonnosta) vapaamuotoisesti kynällä ja paperilla. Lähdin matalalla kynnyksellä piirtämään sitä, minkälainen uusi palvelu voisi aikaisempien tutkimuksien pohjalta olla. Kuvassa 16 näkyvät luonnokset Roar Bikes Bikestoren-verkkosivun etusivusta ja tuotesivusta.



Kuva 16. Roar Bikes Bikestore-projektin ensimmäisen vaiheen luonnos tuotteesta

Luonnostelun jälkeen lähdin varsinaisen lopullisen mallin kimppuun. Kuvassa 17 on projektin suunnitellun etusivun ensimmäisen osa (Main Hero). Main Hero on ensimmäinen asia, jonka asiakas näkee verkkosivuille tullessaan. Ensimmäisessä tuoteversiossa laitoin ostoskorin alle tekstin shoppingbag (ostoskori), mutta testausvaiheessa palautteen saatua otin tekstin pois. Näin etusivusta tuli selkeämpi ja navigaatiopalkista vähemmän ruuhkainen.



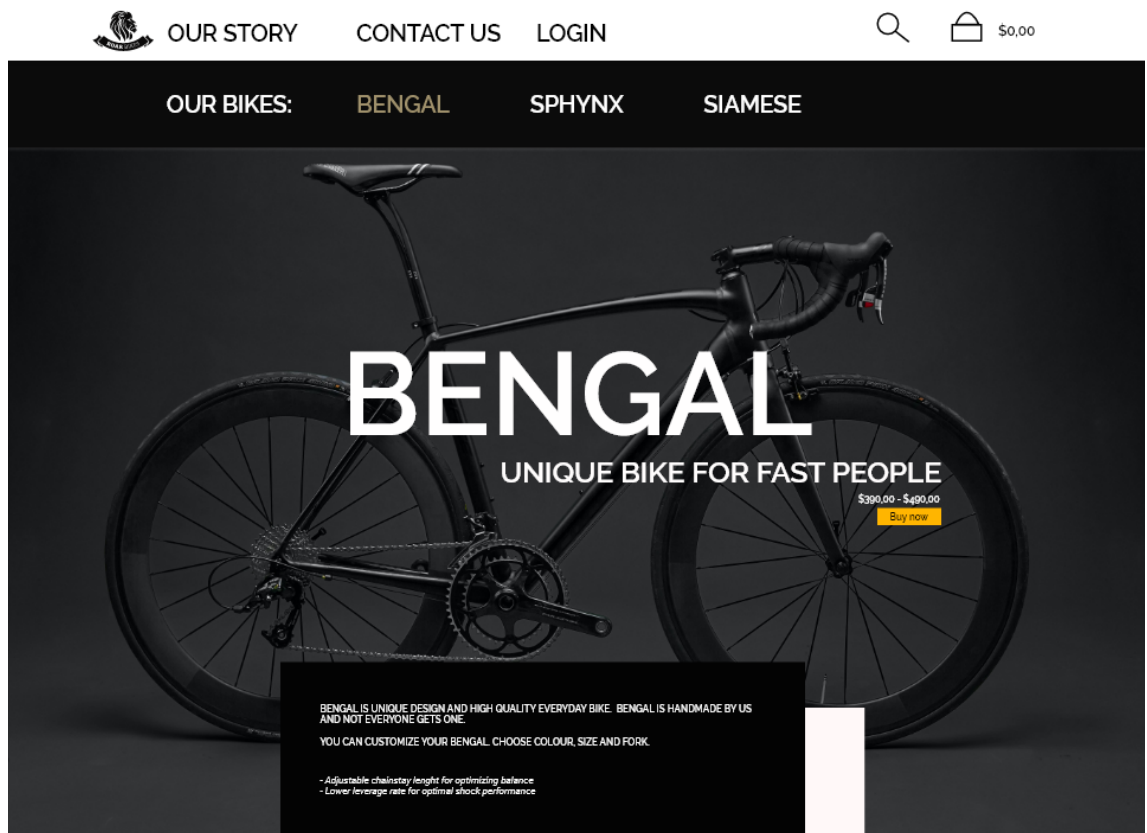
Kuva 17. Main Hero Roar Bikes Bikestoren etusivulla

Kuvassa 18 on Roar Bikes Bikestoren verkkosivun ensimmäinen sivu kokonaisuudessaan. Asiakkaalle tuodaan sivulla esiin selkeästi yrityksen nimi ja se, että yritys tarjoaa uniikkeja tuotteita vain omilla sivuillaan myytynä. Heti etusivulla näytetään asiakkaalle yrityksen tarjoamia tuotteita selkeiden kuvien avulla. Asiakas voi jatkaa tutustumista tuotteisiin klikkaamalla tuotekuvaa tai painamalla osta painiketta, jolloin ohjaudutaan suoraan ostos kohtaan tuotesivulla. Tuotteeseen tutustuminen ja tuotteen hankinta on haluttu tehdä asiakkaalle yksinkertaiseksi.



Kuva 18. Roar Bikes Bikestoren verkkosivun etusivu kokonaisuudessaan

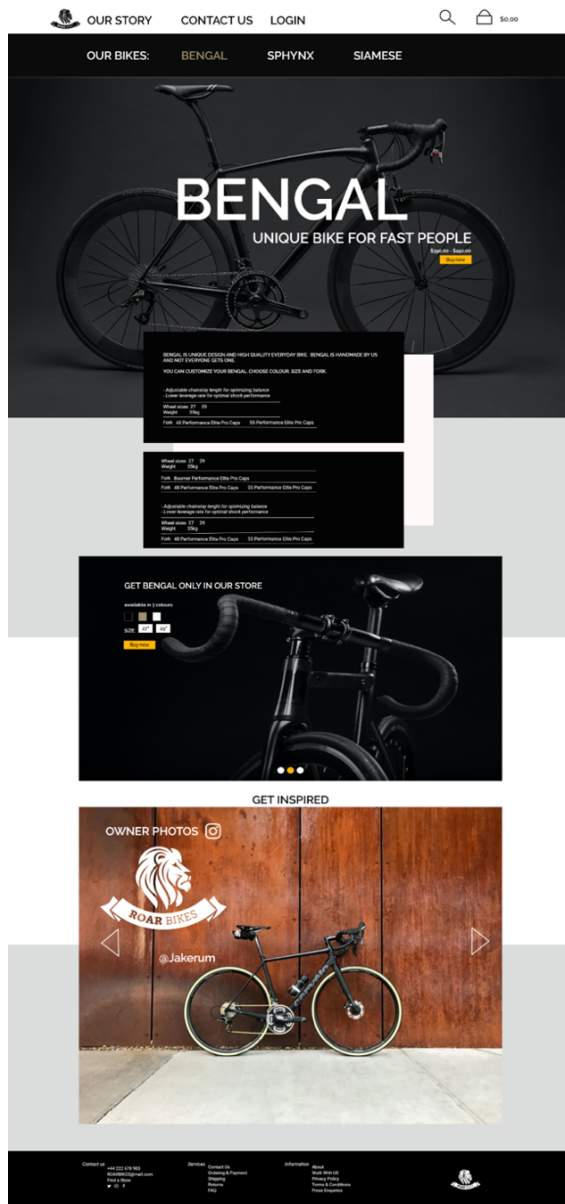
Yksittäisen tuotteen tuotesivu on pidetty myös teemalle uskollisen yksinkertaisena ja kar-
sittuna. Asiakas voi ostaa pyörän heti tai tutustua ensin pyörän tarkempiin tuotetietoihin.
Kuvassa 19 on Bengal tuotteen tuotesivun ensimmäinen näkymä.



Kuva 19. Bengal tuotesivun ensimmäinen näkymä

Bengal tuotesivulla pääkuvassa on haluttu tehdä ostaminen asiakkaalle helpoksi. Asiakas voi joko jatkaa sivuun tutustumista tai jos ostopäätös on jo tehty asiakas voi ostaa tuotteen Buy now -painikkeen kautta heti sivulle saapuessaan.

Kuvassa 20 on koko Bengal tuotteen tuotesivu, johon asiakas pääsee tutustumaan siirtymällä sivua alaspäin.



Kuva 20. Tuotesivu Bengal kokonaisuudessaan

Opin projektin aikana todella paljon suunnitteluprosessista. Erityisen tyytyväinen olen projektin kirjasintyyppin (fonts) valintaan. Tämän projektin aikana pyrin panostamaan kirjasintyyppin valinnan harjoitteluun. Mielestäni sain myös erinomaisesti vangittua kohderyhmän edustajalle sopivan tunnelman kautta koko verkkosivun. Taitoni myös suunnitteluohjelma Adobe XD:n käyttäjänä vahvistuivat hyvälle tasolle. Kokonaisuudessaan olen erittäin tyytyväinen projektin lopputulokseen ja projektin aikana karttuneisiin uusiin taitoihini.

Parannusehdotuksena tulen kehittämään projektissa vielä tuotesivun tuotetietokaavioita. Tuotetietokaavio on tällä hetkellä projektissa keskeneräisen oloinen ja vaatii vielä viilausta ja yksinkertaistamista. Myös näin jälkikäteen huomaan, että painikkeiden koko vaatii vielä viilaamista. Jatkokehityksenä projektiin toteutetaan tuotesivut jokaiselle tuotteelle sekä mobiiliversio Roar Bikes Bikestoresta.

3.4 Käyttöliittymäohjelmointi

Ensimmäiseksi ohjelmoinnin projekteista esittelen koulun opintojakson Front-end-ohjelmointi projektityön. Sain projektityöstä arvosanan 4 asteikolla 1-5.

Valitsin projektityön portfoliooni, sillä se tuo esiin taitojani käyttöliittymäohjelmoinnin puolelta modernista, paljon käytetystä teknologiasta ja lisää osaamiseni esittelyn monipuolisuutta portfolioissani.

Front-end-ohjelmointi opintojakson tavoitteena oli, että opiskelija oppii tekemään yksinkertaisen sivuston käyttäen modernia Front-end-sovelluskehystä Reactia. Opintojakson aikana käytettiin teknologiana React JS-ohjelmointikieltä. React-sovelluskehysten avulla opeteltiin Front-end-sovelluskehysellä sivuston tekeminen, käyttöliittymän tekeminen käyttäen käyttöliittymäkirjastoa, navigaation toteuttaminen, JSON-muotoisen tiedon hakeminen ja käsittely React-sovelluskehysellä sekä ulkoisten palveluiden käyttäminen REST-rajapinnalla. (Moodle 2019c.)

Projektityössä (kuva 21) toteutin sivuston nimeltä Book Store. Sivustolla käyttäjä voi lisätä, listata, poistaa sekä muokata kirjoja. Käyttäjä saa sivustolla myös kirjavinkkejä ja hän voi selata Book Storen henkilökunnan tietoja.



Kuva 21. Kuva Book Store sivuston käyttöliittymästä

React on JavaScript-kirjasto käyttöliittymien toteuttamiseen. React on komponenttipohjainen sovelluskehys. Komponentit ovat itsenäisiä ja uudelleenkäytettäviä sovelluksen osia. Komponenttiajattelu on React-sovelluskehysten ydinajatus. (Hinkula 2019.)

Projektistani löytyvät esimerkiksi komponentit kirjojen lisäämiseen sovellukseen sekä lisättyjen kirjojen listaamiseen. Molemmat komponentit ovat toisistaan riippumattomia, itsenäisiä sovelluksen osia, joita voidaan ottaa käyttöön sovelluksen eri osissa tarpeen mukaan.

Reactjs.org sivustolla on hyvä esimerkki yksinkertaisesta komponentista (kuva 22). Kuvassa näkyy, miten kuvatus komponentin on tarkoitus tuottaa käyttöliittymään viesti "Hello" ja sovellukseen annettu käyttäjän nimi. Render metodi ottaa vastaan annetun tiedon, tässä tapauksessa nimen Taylor, ja palauttaa sen, mitä käyttöliittymään on tarkoitus tuottaa. Tässä tapauksessa palautetaan sana Hello ja annettu nimi Taylor. (Reactjs.) Esimerkin komponentin tarkoitus on tuottaa yksinkertainen viesti selaimen.

LIVE JSX EDITOR	<input checked="" type="checkbox"/> JSX?	RESULT
<pre>class HelloMessage extends React.Component { render() { return (<div> Hello {this.props.name} </div>); } } ReactDOM.render(<HelloMessage name="Taylor" />, document.getElementById('hello-example'));</pre>		Hello Taylor

Kuva 22. Esimerkki komponentin toiminnasta (Reactjs)

Kuvassa 23 on ylläolevaa komponentti esimerkkiä vastaava komponentti projektityöstäni. Kuvassa on projektin komponentti kirjojen listaamiseen käyttöliittymään. Komponentti palauttaa otsikon "kirjat listassa" sekä kirjalistan jokaisen osan omana kuvakkeenaan.

```

1  import React from 'react';
2  import Grid from '@material-ui/core/Grid'
3  import Card from '@material-ui/core/Card';
4  import CardHeader from '@material-ui/core/CardHeader';
5  import CardContent from '@material-ui/core/CardContent';
6  import Typography from '@material-ui/core/Typography';
7
8  function KirjaLista (props) {
9    return (
10     <Grid spacing={4} container>
11       <CardHeader>Kirjat listassa:</CardHeader>
12
13       {
14         props.kirjat.map(kirja => {
15           return(
16
17             <Grid item key={kirja.id}>
18               <Card style={{maxWidth: 350, minWidth: 350}}>
19                 <CardHeader title='Lukemani kirja'/>
20                 <CardContent>
21                   <Typography variant='h6'> Kirjan nimi: {kirja.nimi}</Typography>
22                   <Typography variant='body2'> Kirjaailija: {kirja.kirjaailija} </Typography>
23                 </CardContent>
24
25               </Card>
26             </Grid>
27           )
28         }
29       )
30     )
31   </Grid>
32 )
33 }
34
35 export default KirjaLista;

```

Kuva 23. Projektin komponentti kirjojen listaamista varten

Kuvassa 24 on kirjalistakomponentin tuottama tulos käyttöliittymässä kuvattuna.



Kuva 24. Kirjojen listauskomponentin palauttama tulos käyttöliittymässä

Projektin sivuston navigaatio (kuva 25) on toteutettu Material UI kokonaisuuden tarjoaman Drawer-kirjaston avulla. Material UI tarjoaa ohjelmoijalle valmiita työkaluja, jotka nopeuttavat ja helpottavat komponenttien ohjelmointia.

```

const toggleDrawer = (side, open) => event => {
  if (event.type === 'keydown' && (event.key === 'Tab' || event.key === 'Shift')) {
    return;
  }

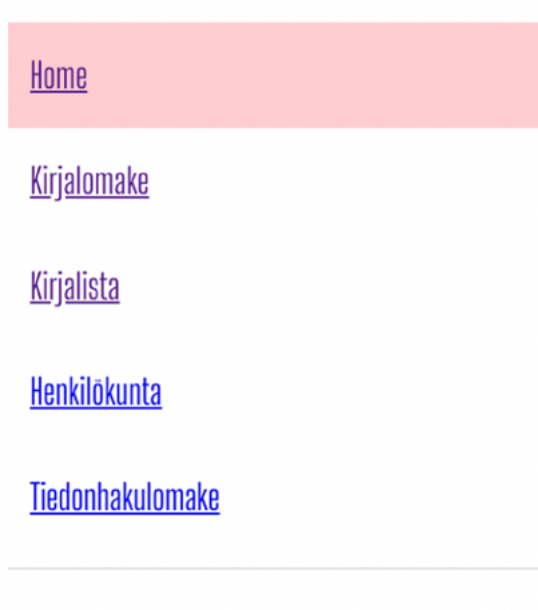
  setState({ ...state, [side]: open });
};

const sideList = side => (
  <div
    className={classes.list}
    role="presentation"
    onClick={toggleDrawer(side, false)}
    onKeyDown={toggleDrawer(side, false)}
  >
    {/**<Link to="/Home">Home</Link>,**/}
    <List>
      {[<Link to="/">Home</Link>, <Link to="/Kirjalomake">Kirjalomake</Link>, <Link to="/Kirjalista">Kirjalista</Link>,
        <ListItem button key={text}>
          <ListItemText primary={text} />
        </ListItem>
      ]}
    </List>
    <Divider />
  </div>
);

```

Kuva 25. Projektin Drawer-navigaation lähdekoodi

Navigaatiokomponentti tuottaa käyttöliittymään sivunavigaation (kuva 26), joka aukeaa mobiilissa käyttäjän pyyhkäisystä tai selaimessa käyttäjän napin painalluksesta.



Kuva 26. Projektin Drawer-navigaatio

Työssäni hyödynsin Material UI:n sekä Reactin tarjoamia kirjastoja ja komponentteja omissa komponenteissani.

Opintojakso oli haastava ja sisältö monipuolinen. Pidin opintojaksosta erittäin paljon ja siksi valitsin projektityön esiteltäväksi portfolioissani. Koin opintojakson haastavaksi ja mielenkiintoiseksi. Opintojakson puitteissa työni lopputulos on hyvä, mutta jatkokehityksenä haluan suunnitella projektin käyttöliittymän uudelleen ja saada lopputuloksesta visuaalisemman. Projektia tehdessäni panostin toiminnallisuuden harjoitteluun ja visuaalinen ilme jäi työstä puuttumaan.

Opintojakson aikana opin monipuolisesti hyödyntämään komponenttijaottelua ohjelmoinnissa sekä opin hyödyntämään valmiina olevia ratkaisuja osana itse ohjelmoituja komponentteja. Haastavaksi projektissa osoittautui ulkoisen rajapinnan käyttö ja tähän tulen paneutumaan seuraavassa React-projektissani tarkemmin.

3.5 Palvelinohjelmointi

Palvelinohjelmoinnin projektityö on Palvelinohjelmointi opintojakson kurssityö. Sain työstä arvosanaksi 5 asteikolla 1-5. Projektin esittelyssä keskityn projektin pääteknologioihin Spring Boottiin ja Javaan. Projektissa hyödynsin myös Git-versionhallintaa. Valitsin projektityön portfolioon, sillä projektityö tuo esiin taitojani monipuolisesti ohjelmoinnin puolelta suosituista teknologioista ja näin lisää osaamiseni esittelyn monipuolisuutta portfolioissa. Projektityö oli vaativa, kattava ja taitoni ohjelmoijana kasvoivat isoin harppauksin opintojakson aikana. Opin projektissa ensimmäistä kertaa laajemmin ja syvemmin, miten meitä arjessa ympäröivät sovellukset rakentuvat.

Opintojakson tavoitteena oli, että opintojakson suoritettuaan opiskelija ymmärtää ja osaa kuvata palvelinohjelmoinnin kokonaisuuden, osaa toimia laadukasta jälkeä tuottavana palvelinohjelmoijana, pystyy analysoimaan ongelmia, etsimään tietoa, soveltamaan tietoa ja pystyy varmistamaan luomansa ratkaisun toimivuuden testauksen avulla. (Moodle 2019d.)

Opintojakson projektityön tavoitteena oli tehdä sovellus, jolla voi ylläpitää jotain opiskelijalle mielenkiintoista tietoa kuten elokuvat, musiikki tai urheilu. Hallintaliittymään pääsee ainoastaan sisäänkirjautunut käyttäjä, mutta tiedon katselunäkymään pääsee tarkastelemaan kuka tahansa sisäänkirjautumaton vierailija. Tavoitteena oli, että opiskelija suunnittelee ja rakentaa sovellusta varten tietokannan. Tietokannassa tuli olla ainakin kaksi toisiinsa liittyvää tietokantataulua. (Moodle 2019d.)

Projektin tavoite oli tuottaa Spring Boot-sovellus, joka on toteutettu käyttämällä tekniikkoina Spring Boot, Java, Thymeleaf templates, JPA (H2), Spring Boot Security, Bean Validation, Spring Boot REST (JSON) ja ainakin kahta muuta seuraavista tekniikoista: AJAX JS/JQuery/React, Testaus (JUnit), I18n (monikielisyys) tai muu vapaavalintainen Spring Boot -toiminnallisuus. (Moodle 2019d.)

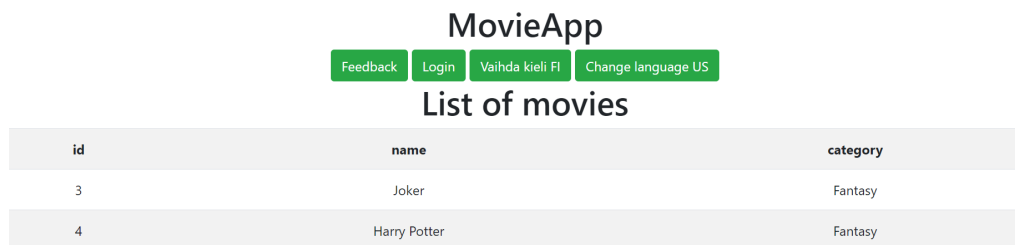
Iso osa web-sovellusten rakentamisesta perustuu valmiiden sovelluskehysten hyödyntämiseen. Sovelluskehukset sisältävät valmiiksi ohjelmituista osia. Näin ohjelmoijan ei tarvitse halutessaan ohjelmoida kaikkea itse alusta alkaen, vaan työskentelyssä voidaan hyödyntää Spring Bootin kaltaista sovelluskehystä. (Helsingin yliopisto 2019.)

Hyvä esimerkki Spring Boot-sovelluskehysten ideasta on, että tehdessään web-sovellusta ohjelmoija toteuttaa sovelluskehysten puitteissa luokkia, mutta ei luo niistä olioita, vaan olioiden luominen on Spring-sovelluskehysten vastuulla. Ohjelmoija kertoo, minkälainen

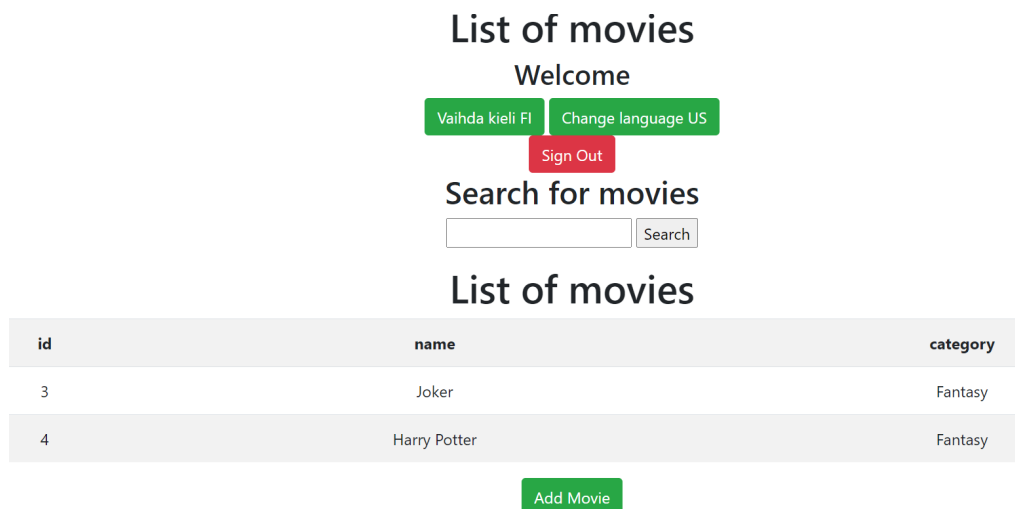
olio tehdään ja Spring-sovelluskehys toteuttaa olion ohjeiden mukaan. (Helsingin yliopisto 2019.)

Spring Boot-sovelluskehys tekee Java-ohjelmoinnin nopeammaksi, helpommaksi ja turvallisemmaksi. Sovelluskehys fokusoi nopeuteen, yksinkertaisuuteen sekä tuottavuuteen ja siksi se on maailman suosituin Java-sovelluskehys. (Spring.io 2020.)

Projektin aiheeksi valikoitui katsomani elokuvat-sovellus. Sovellukseen voi lisätä sisäänkirjautuneena katsomiaan elokuvia ja myös muokata tai poistaa niitä. Kuka vaan voi nähdä elokuvalistan, mutta lisäyksiä ja muokkauksia voi tehdä vain sisäänkirjautunut käyttäjä. Sovelluksen kieltä voi vaihtaa joko suomeksi tai englanniksi. Kuvassa 27 on projektin elokuva sovelluksen etusivu ja kuvassa 28 näkymä, kun käyttäjä on kirjautunut sisään sovellukseen.



Kuva 27. Spring Boot-projektin etusivu



Kuva 28. Spring Boot-projektin etusivu sisäänkirjautumisen jälkeen

Teknologioina sovelluksessa ovat Spring Boot-sovelluskehiksen ja muiden pakollisten vaatimuksien lisäksi JUnit-testausta, security validation (sisäänkirjautujan tunnistus) ja Internationalization (sovelluksen selaimen kielen vaihtaminen). Toteutin myös rekisteröitymisen sovellukseen.

Projektin ohjelmointikieli on Java. Java-ohjelmointikieli on tänä päivänä yksi laajimmin käytetyistä ohjelmointikielistä. Alun perin Java on julkaistu vuonna 1996. (Cosmina 2018.) Java on hyvin yhteensopiva kieli taaksepäin vanhojen versioiden kanssa (Gee 2020). Tämä tekee kielestä suosittu valinnan organisaatioissa, joissa halutaan varmistua sovelluksen toimivuudesta vuosikymmeniksi eteenpäin.

Harjoittelin projektityössä testausta opettelemalla JUnit-kirjaston avulla yksikkötestausta. Toin sovellukseeni mukaan JUnit-riippuvuuden ja näin pääsin hyödyntämään sen tarjoamia lisäpalveluita projektissa. Kuvassa 29 on esimerkki projektiin kirjoittamistani testeistä JUnit-riippuvuuden avulla. Testit varmistavat kirjoitetun lähdekoodin toimivuuden.

```
import org.junit.Test;

//annotation tells JUnit to run using Spring's testing support
@RunWith(SpringRunner.class)
@DataJpaTest
//annotation will configure in-memory database, JPA and Spring
//Data. It also turns on SQL logging
public class MovieRepositoryTest {

    //This annotation allows Spring to resolve
    //and inject collaborating beans into your bean.
    @Autowired
    private MovieRepository mrepository;

    @Test
    public void findByName() {
        Optional<Movie> movies = mrepository.findByName("Joker");
        assertThat(movies.get().getName()).isEqualTo("Joker");
    }

    //annotation defines a method to be test method
    @Test
    public void createNewStudent() {

        Movie movie = new Movie(null, "Joker", new Category("Drama"));
        mrepository.save(movie);
        assertThat(movie.getCategory()).isNotNull();
        assertThat(movie.getName()).startsWith("J").endsWith("r");
    }
}
```

Kuva 29. Esimerkki projektin JUnit-testistä kirjoitettuna ohjelmointikielellä Java

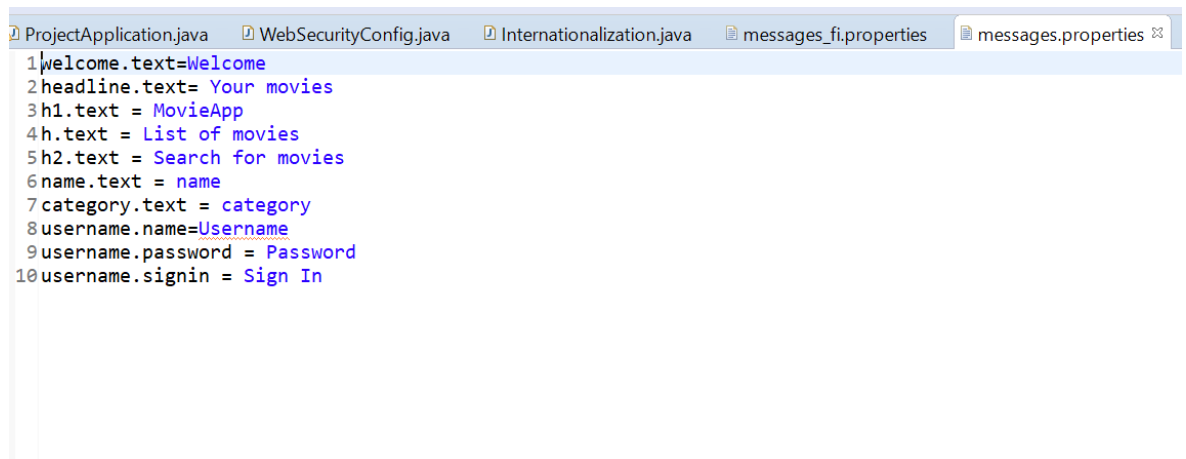
Kuvassa 29 olevassa testissä varmistan, että sovellukseni toimii kuten olen tarkoittanut. Testissä tarkistan, että sovellus luo pyynnöstä uuden elokuvan ja tallentaa sen tietokantaan. Varsinainen testi on se, että juuri luotu elokuva löytyy tietokannasta.

Yksikkötestauksen idea on kirjoittaa ohjelman yksittäisille osille, kuten yksittäiselle metodille (toiminnallisuudelle) automaattiset testit. Testien avulla tarkastetaan, että ohjelmoijan kirjoittama metodi toimii halutulla tavalla. (Helsingin Yliopisto 2019b.)

Lisäosana projektissa halusin tuoda myös mukaan sovelluksen käyttöliittymän kielenvaihtamisen. Sovelluksen etusivulla on painikkeet, joiden avulla selaimessa näytettävän kielen voi vaihtaa englannista suomeksi. Toimintoa varten harjoittelin Internationalization riippuvuuden käyttöä sovelluksessa.

Internationalization Spring Bootin avulla tarkoittaa prosessia, joka tekee sovelluksesta mukautettavan eri kielille ja alueille ilman, että lähdekoodiin tehdään teknisiä muutoksia aina kun käyttäjä haluaa eri kielellä toimivan version sovelluksesta. Sovellukseen on riippuvuuden avulla jo luotu mahdollisuus kielen vaihtamiseen ja riittää kun käyttäjä painaa esimerkiksi nappia kielen vaihtamista varten. (Tutorialspoint.)

Kielen vaihtamista varten määrittelin sovellukseen millä kielillä sovellusta on mahdollista käyttää ja tein jokaista kieltä varten kielitiedoston, jonka perusteella sovellus osaa kunkin kielen mukaan tuoda oikein käännöksen käyttäjän nähtäväksi selaimen (Kuva 30). Käyttäjän painaessa nappia vaihtaakseen sovelluksen kielen, sovellukseen ohjelmoitu localeChangeInterceptor tulkitsee, että mikä kieli on kyseessä ja lopuksi palautetaan käyttäjälle käyttöliittymän pyydetyn kielen mukainen käyttöliittymä.



```
ProjectApplication.java WebSecurityConfig.java Internationalization.java messages_fi.properties messages.properties
1welcome.text=Welcome
2headline.text= Your movies
3h1.text = MovieApp
4h.text = List of movies
5h2.text = Search for movies
6name.text = name
7category.text = category
8username.name=Username
9username.password = Password
10username.signin = Sign In
```

Kuva 30. Kielitiedosto

Kuvassa 31 on kielenvaihtamisen mahdollistava ohjelmointiesimerkki.

```

4 @Configuration
5 public class Internationalization implements WebMvcConfigurer {
6     //default kielen asetus tässä
7 @Bean
8 public SessionLocaleResolver localeResolver( ) {
9     SessionLocaleResolver sessionLocaleResolver = new SessionLocaleResolver();
10    sessionLocaleResolver.setDefaultLocale(Locale.US);
11    return sessionLocaleResolver;
12 }
13
14 //tulkitsee
15 @Bean
16 public LocaleChangeInterceptor localeChangeInterceptor() {
17     LocaleChangeInterceptor localeChangeInterceptor = new LocaleChangeInterceptor();
18     localeChangeInterceptor.setParamName("language");
19     return localeChangeInterceptor;
20 }
21
22 @Override
23 public void addInterceptors(InterceptorRegistry registry) {
24     registry.addInterceptor(localeChangeInterceptor());
25 }
26
27 }
28

```

Kuva 31. Ohjelmointiesimerkki sovelluksen kielen vaihtamisesta

Opintojakso oli kattava katselmus Spring Boot–sovelluskehiksen ja Java-ohjelmointikielen maailmaan. Oli mielenkiintoista kokeilla sovelluskehystä ja huomata, miten paljon löytyy valmiiksi ohjelmoituja kirjastoja, joita ohjelmoijat voivat hyödyntää omassa työssään nopeuttaakseen ja optimoidakseen ohjelmointiaan. Tämän projektityön aikana myös Java-ohjelmointikielen perusteet vahvistuivat sekä taitoni hyödyntää Git-versionhallintatyökalua kasvoivat.

Jatkokehityksenä tulen tekemään tuottamalleni palvelulle uuden käyttöliittymän ja liittämään sen projektiin. Tulevaisuudessa haluan liittää sovellukseen myös varsinaisen tietokannan käyttämäni h2-testitietokannan sijasta.

3.6 Rajapinta- ja konttitekniologiat

Tämä projekti on Ohjelmistokehityksen teknologioita opintojakson projektityö. Projektityöstä sain arvosanaksi 5 asteikolla 1-5.

Opintojakson tavoitteena oli, että opiskelija osaa itsenäisesti kasvattaa osaamistaan opiskelijalle aiemmin tuntemattomista teknologioista ja oppii arvioimaan sekä valitsemaan projekteihin soveltuvia teknologioita ja menetelmiä. (Moodle 2020e.)

Valitsin projektin mukaan portfolio-opinnäytetyöhön projektin monipuolisuuden vuoksi sekä koska koen, että projektissa oppimani asiat ovat erittäin monipuolisia ja nykyaikaisia ja näin ollen projekti esittelee erinomaisesti monipuolista ja ajankohtaista osaamistani ohjelmoinnin saralla.

Lähdin projektissa tutustumaan teknologioihin GraphQL ja Docker. Molemmista teknologioista olin kuullut aiemmin, mutta minulla ei ollut aikaisempaa käytännön kokemusta kummastakaan. Päätin projektityönä perehtyä mainittuihin teknologioihin ja tehdä käytännön harjoituksen niiden avulla. Hyödynsin oppimateriaalina projektissa internetiä sekä Sebastian Greben kirjaa ”Hands on full-stack web development with GraphQL and React”.

Muita tekniikoita projektissa ovat React, Apollo, Node.js ja Express. React toimii projektin sovelluksessa käyttöliittymän sovelluskehiksenä. Apollo on teknologia, joka mahdollistaa GraphQL-tekniologian käytön yhdessä Reactin kanssa. Express toimii projektissa web-palvelimena, joka pyörii Node.js:n avulla. Näihin edellä mainittuihin teknologioihin en paneudu tässä esittelyssä, sillä projektin tavoitteena oli opetella ja keskittyä GraphQL- ja Docker-tekniologioihin.

GraphQL on alun perin Facebookin kehittämä teknologia, jota voidaan hyödyntää sovelluksen ja palvelimien väliseen kommunikointiin. GraphQL:n toimintaperiaate on, että selaimen lähdekoodi muodostaa kyselyn, joka sisältää kuvaelman halutusta datasta. Tämän jälkeen selain lähettää sen eteenpäin rajapinnalle HTTP-POST pyynnöllä. GraphQL:ssä kaikki pyynnot ovat POST-tyyppisiä pyyntöjä ja rajapintoja on vain yksi, eli pyynnot kohdistuvat aina samaan osoitteeseen. Palvelin vastaa pyyntöön JSON-oliolla, joka sisältää pyydetyn datan. GraphQL:n avulla sovelluslogiikka pysyy yksinkertaisena ja haluttu tieto saadaan palvelimelta yksittäisellä kyselyllä. (Helsingin yliopisto 2020.)

GraphQL-sovelluksissa on aina skeema, joka määrittelee minkä muotoista dataa sovelluksessa välitetään sovelluksen ja palvelimien välillä. Ensin skeemassa määritellään tyyppi,

jossa määritellään haettavan tiedon sisältö ja kenttien muoto. Toiseksi määritellään tyyppi Query, jossa määritellään, että mitä kyselyjä rajapintaan voidaan tehdä. Kuvassa 32 on esimerkki skeeman sisällöstä. (Helsingin yliopisto 2020.)

```
type Person {
  name: String!
  phone: String
  street: String!
  city: String!
  id: ID!
}

type Query {
  personCount: Int!
  allPersons: [Person!]!
  findPerson(name: String!): Person
}
```

Kuva 32. Esimerkki skeemasta (Helsingin yliopisto 2020)

Skeema määrittelee kaksi eri tyyppiä. Näistä tyypeistä ylempi "Person" määrittelee, että henkilöillä on viisi kenttää. Kentistä neljä on merkkijono muotoista tietoa. Näistä String-arvoisista kentistä muilla paitsi puhelinnumerolla on oltava arvo. Tämä on merkitty skeemaan huutomerkillä. Huutomerkki tarkoittaa pakollista kenttää. (Helsingin yliopisto 2020.)

Toinen määritellyistä tyypeistä on Query. Query kertoo, minkälaisia kyselyjä rajapintaan on mahdollista tehdä. Kuvan skeemassa on määritelty kolme erilaista kyselyä. PersonCount palauttaa kokonaisluvun, allPersons palauttaa listan Person-olioita, findPerson saa parametrina arvon ja palauttaa tietyn Personin. (Helsingin yliopisto 2020.)

Skeema on tärkeä osa GraphQL-sovellusta, sillä se määrittelee mitä kyselyjä käyttöliittymä pystyy palvelimelta tekemään, minkälaisia parametreja kyselyillä voi olla sekä sen, minkä muotoista kyselyjen palauttama tieto (data) on.

Skeema kuvaa ainoastaan palvelimen ja sitä käyttävien käyttöliittymien välillä liikkuvan tiedon muodon. Tieto voi olla kuitenkin tallennettuna palvelimella missä muodossa vain. (Helsingin yliopisto 2020.)

GraphQL mahdollistaa sovelluksessa tiedon välittämisen käyttöliittymän (clientin) ja palvelimen välillä. Kuvassa 33 on esimerkki siitä, miten projektissa määrittelin skeeman. Skeemassa on kerrottu tarkasti, minkä muotoista sovelluksen tieto on.

```

const typeDefinitions = gql`
  type Post {
    id: Int!
    text: String!
    user: User!
  }

  type User {
    id: Int!
    avatar: String!
    username: String!
  }

  type Message {
    id: Int!
    text: String!
    chat: Chat!
    user: User!
  }

  type Chat {
    id: Int!
    messages: [Message!]
    users: [User!]
    lastMessage: Message!
  }
`

```

Kuva 33. Esimerkkikuva projektin skeemasta

GraphQL:ssä skeeman lisäksi mutaatiot ovat tärkeässä roolissa. Kaikki muutoksen aiheuttavat operaatiot tehdään mutaatioiden avulla. (Helsingin yliopisto 2020.) Kuvassa 34 on käyttäjän lisäävä mutaatio.

```

type Mutation {
  addPerson(
    name: String!
    phone: String
    street: String!
    city: String!
  ): Person!
}

```

Kuva 34. Mutaatioesimerkki (Helsingin yliopisto 2020)

Kuvan mutaatio saa parametreina käyttäjän tiedot. Parametreista phone on ainoa ei pakollinen kenttä. Mutaatioilla on parametrien lisäksi paluuarvo. Paluuarvo on tässä esimerkissä tyyppiä Person. Mutaation ideana on, että se palauttaa operaation onnistuessa lisätyn henkilön tiedot. (Helsingin yliopisto 2020.)

Kuvassa 35 on esimerkki projektin mutaatiosta, joka palauttaa käyttöliittymään käyttäjän luoman uuden uutisen.

```

RootMutation: {
  addPost(root, { post }, context) {
    logger.log({
      level: 'info',
      message: 'Post was created',
    });

    return User.findAll().then((users) => {
      const usersRow = users[0];

      return Post.create({
        ...post,
      }).then((newPost) => {
        return Promise.all([
          newPost.setUser(usersRow.id),
        ]).then(() => {
          return newPost;
        });
      });
    });
  },
},

```

Kuva 35. Esimerkki projektin mutaatiosta

Toimiakseen mutaatiot tarvitsevat resolverit. GraphQL toimii niin, että jokaisella tyypillä on mutaatio ja resolveri, joka toteuttaa mutaation. Jokaista kenttää kohden on oma resolveri. Kun kenttää kutsutaan, niin sitä vastaava resolveri kutsutaan toteuttamaan seuraava arvo. (GraphQL.)

Kuvassa 36 on esimerkki projektin resolveista, jotka toteuttavat projektissa skeeman mukaisesti määritellyt mutaatiot.

```

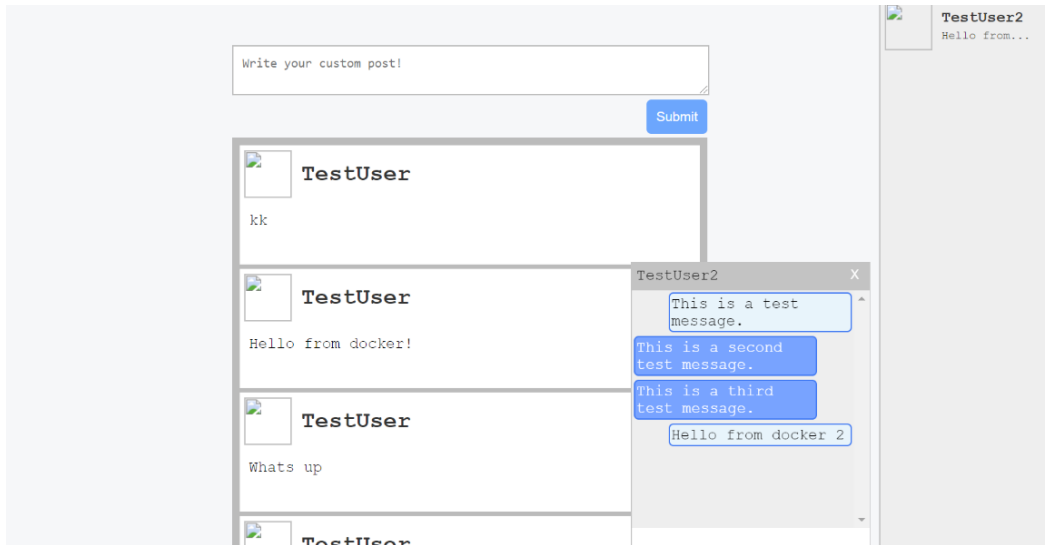
const resolvers = {
  Post: {
    user(post, args, context) {
      return post.getUser();
    },
  },
  Message: {
    user(message, args, context) {
      return message.getUser();
    },
    chat(message, args, context) {
      return message.getChat();
    },
  },
  chat: {
    messages(chat, args, context) {
      return chat.getMessages({ order: [['id', 'ASC']] });
    },
    users(chat, args, context) {
      return chat.getUsers();
    },
  },
  lastMessage(chat, args, context) {
    return chat.getMessages({ limit: 1, order: [['id', 'DESC']] }).then((message) => {
      return message[0];
    });
  },
},

```

Kuva 36. Esimerkki projektin resolveista

Projektityössä toteutin pienen React-sovelluksen, jossa voidaan lisätä, muokata ja poistaa uutisia. Sovelluksessa on myös chat-toiminto. Kuvassa 37 on sovelluksen selainäkymä.

Sovelluksessa ei ole paneuduttu visuaalisuuteen, vaan tarkoituksena oli harjoitella GraphQL- ja Docker-tekniikoita teorian ja käytännön kautta.



Kuva 37. Projektin selainnäkö

Projektissa opettelin myös Docker-konttien käyttöä, jotta voisin ohjelmoida asentamatta omalle tietokoneelleni kaikkia projektissa tarvitsemiä työvälineitä. Docker-kontit ovat keino pakata sovelluksia ja niiden riippuvuuksia standardisoidulla tavalla. Dockeria voisi kuvata niin, että Dockerin avulla voi helposti pakata ohjelmoinnin projektin ja välittää sen esimerkiksi työkaverille tai jakaa projektin esimerkiksi GitHubissa. Koska projekti on Docker-kontissa, voi toinen henkilö käynnistää projektin standardoidulla tavalla ilman, että hänen tarvitsee tietää projektistasi sen enempää tai ilman, että hänen täytyy ladata koneelleen projektin vaatimia riippuvuuksia. Docker-kontit hyödyntävät pilviresursseja ja näin ollen projektien resurssit eivät kuormita fyysistä konetta niin paljon. (Schenker 2020.)

Projektissa opettelin Dockerin perusteita. Asensin Dockerin tietokoneelleni ja sen jälkeen rakensin projektiin Docker-kontteja, oman projektin back-endille, front-endille ja oman tietokannalle. Nimenomaan tietokannan osalta Dockerin rooli projektissa oli merkittävä. Työssä hyödynsin MySQL-tietokantaa, mutta en halunnut asentaa sitä omalle tietokoneelleni. Sen sijaan tein Docker-kontin, jonka tehtävä on toimia projektini tietokantana. Projektin käynnistyessä Docker-kontin rakennusohjeet ajetaan ja MySQL-kontin rakennusohjeet tuottavat projektille MySQL-tietokannan ilman, että käyttäjän tarvitsee mitään itse asentaa tietokoneelleen.

Alla olevassa kuvassa 38 näkyvät rakennusohjeet, joiden avulla Docker käynnistyessään osaa rakentaa juuri projektiin tarvittavan kontin tietokantaa varten. Docker-kontti tarvitsee

aina rakennusohjeet, jonka perusteella se tietää mitä tehdä, kun sovellus käynnistyy. Ohjeet perustuvat peruskuvaan (base image). Kuvassa 38 on MySQL-tietokannan rakennusohjeet, joten peruskuva on viimeisin MySQL-kuva. Sovelluksen käynnistyessä haetaan ensin peruskuva. Sen jälkeen kopioidaan luodut sql-lausekkeet Docker-konttiin ja avataan ohjeissa annettu portti, jossa tietokanta sitten pyörii. Lopuksi ajetaan käynnistyskomento. Kun Docker käynnistetään, niin määritellyt Dockerfilessa olevat rakennusohjeet ajetaan heti ja halutut kontit sisältöineen käynnistyvät.

```
Dockerfile X
docker_mysql > Dockerfile > ...
1 #RAKENNUSOHJEET TIETOKANNALLE
2
3 #haetaan base image mihin ohjeet pohjautuu
4 FROM mysql:latest
5
6 ENV MYSQL_ROOT_PASSWORD root
7
8 #kopiodaan omat sql- käyttäjän luonti lausekkeet konttiin
9 COPY v01_initial_script.sql /docker-entrypoint-initdb.d/
10
11 #varataan portti 3306
12 EXPOSE 3306
13
14 #käynnistys komento, jolla hommaa starttaa
15 CMD ["mysqld"]
16
17
18
```

Kuva 38. Projektin MySQL-tietokannan Docker-rakennusohjeet

Projektityö oli laaja ja haastava, mutta mielenkiintoinen. Koin valitsemani teknologiat erittäin mielenkiintoisiksi ja aioin jatkossa perehtyä molempiin teknologioihin lisää. Kehitysehdotuksena näin jälkikäteen olisin rajannut aiheita hieman pienemmäksi. Vaikkakin pidin teknologioista ja opin paljon, niin opintojakson puitteissa samaan lopputulokseen olisi riittänyt vain toisen teknologian harjoittelu. Näin olisin voinut viedä sovellusta myös muilta osin pidemmälle ja perehtyä yksityiskohtaisemmin valitsemaani teknologiaan. Toisaalta olen erittäin tyytyväinen, että projektini aikana sain kokonaiskuvan niin GraphQL- kuin Docker-teknologioista ja nyt minulla on perustaa minkä päälle kasvattaa osaamistani aiheista.

4 Lopputulokset, portfolioon esittely ja jatkokehitys

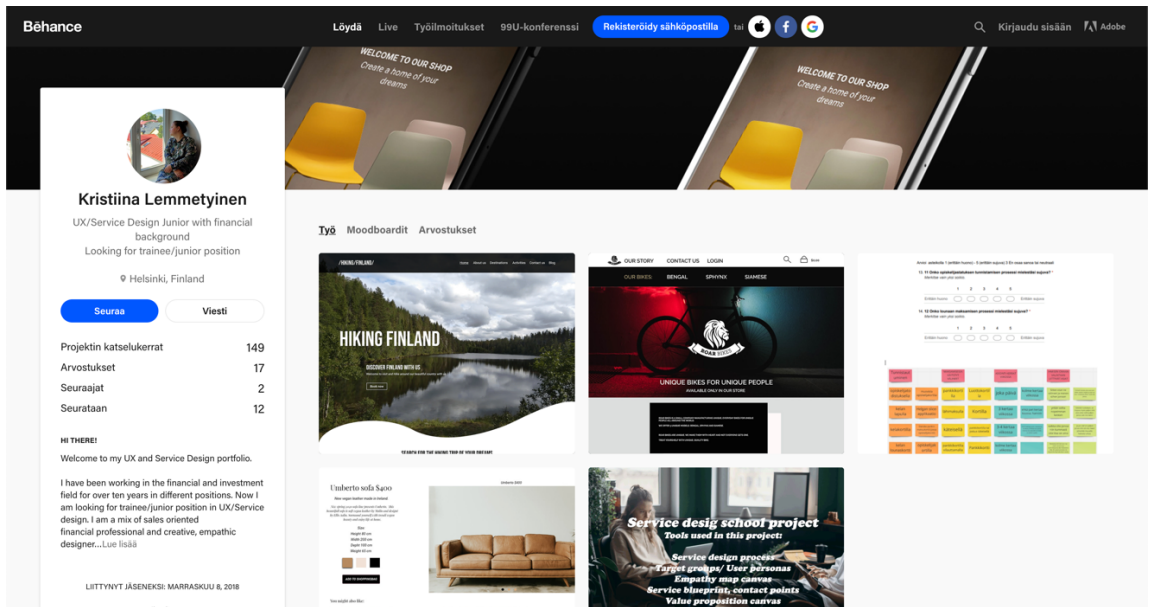
Päästyäni portfolio-opinnäytetyöni loppuvaiheeseen oli minulla kasassa tarvitsemaani tietoa siitä, miten voin luoda portfolioon, minkälainen on hyvä portfolio sekä tarkoin valitut projektit analysoituna portfolioa varten. Projektin aikaisen oppimisprosessin perusteella päädyin ratkaisuun, että minulle parhain tapa rakentaa portfolio on jakaa sen kahteen osaan: käyttäjäkokemus-, käyttöliittymäsuunnittelu- ja palvelumuotoiluportfolioon sekä ohjelmoinninportfolioon. Valitsin käyttäjäkokemus-, käyttöliittymäsuunnittelun ja palvelumuotoilun portfolioon toteutustavaksi Behance-portfolioon ja ohjelmoinninportfolioon toteutustavaksi GitHub-portfolioon.

Päädyin ratkaisuun, koska tällä hetkellä haen käyttäjäkokemus-, käyttöliittymäsuunnittelu- ja palvelumuotoilun tehtäviä. Haluan hakemuksen liitteenä välittää juuri näihin tehtäviin tarvittavan osaamisen portfolioon. Hakemukseen liitän myös ohjelmoinninportfolioon, jotta monipuolinen osaamiseni tulee esiin, mutta koen Portfolio-opinnäytetyön aikana tekemäni tutkimukseni perusteella, että on selkeämpää, että portfolio on siisti, yksinkertainen ja rekrytoivan henkilön nopeasti selattavissa juuri kyseisen haettavan tehtävän osaamisalueiden saralta.

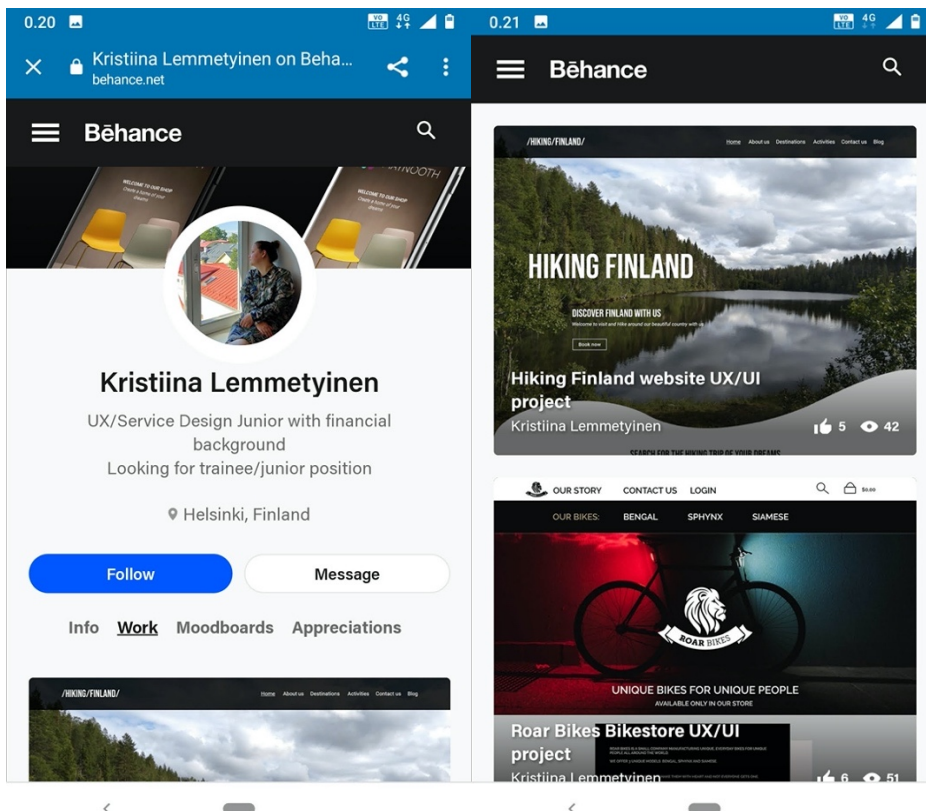
Jos hakemukseni on kiinnostava, niin tällöin ohjelmoinninportfolio on plussaa ja tästä syystä liitän hakemukseen myös GitHub-portfolioon. Prosessi toimii myös toisinpäin, jos päätän hakea ohjelmoinnin tehtäviä, voin hakea nimenomaan ohjelmoinnin projekteilla hyödyntäen GitHub-portfolioani, mutta samalla välitän myös Behance-portfolioon, jotta monipuolinen osaamiseni koko kehittämisen elinkaarelta tulee esiin hakemuksessa.

Päädyin toteuttamaan käyttäjäkokemus-, käyttöliittymäsuunnittelu- ja palvelumuotoilun portfolioon Behance-palvelussa, sillä palvelu on yleisesti tunnettu, helppokäyttöinen ja portfolio on nopea rakentaa palvelun avulla. Portfolio toimii niin selaimessa kuin mobiilissa. Ensimmäiseksi portfolioon näkyy lyhyt, kahden rivin esittely minusta. Esittelyn vieressä on valittujen projektien etusivut. Viemällä osoittimen projektin etusivun päälle, ilmestyy esiin projektin otsikko. Projektin etusivua klikkaamalla pääsee tutustumaan yksittäiseen projektiin. Lyhyen esittelyn alapuolelta löytyy lisää tietoja minusta ja osaamisestani. Portfolio on toteutettu englanniksi, sillä en halua sulkea pois mahdollisuuksia kansainvälisissä yrityksissä. Portfolioon pääsee tutustumaan tarkemmin osoitteessa <https://www.behance.net/Hagridaa>.

Portfoliossa esittelen portfolio-opinnäytetyötäni varten valitsemani ja analysoimani projektit. Lisään myös uusia projekteja portfolioon sitä mukaan, kun niitä valmistuu ja osaamiseni kasvaa. Olen järjestänyt työt portfolioon niin, että mielestäni parhain työ on ensimmäisenä. Kuvassa 39 on Behance-portfolioni etusivu selainversiossa ja kuvassa 40 esimerkki Behance-portfolioni näkymästä mobiiliversiossa.



Kuva 39. Behance-portfolioni etusivu selainversiossa



Kuva 40. Behance-portfolioni etusivu mobiilikäyttöliittymässä

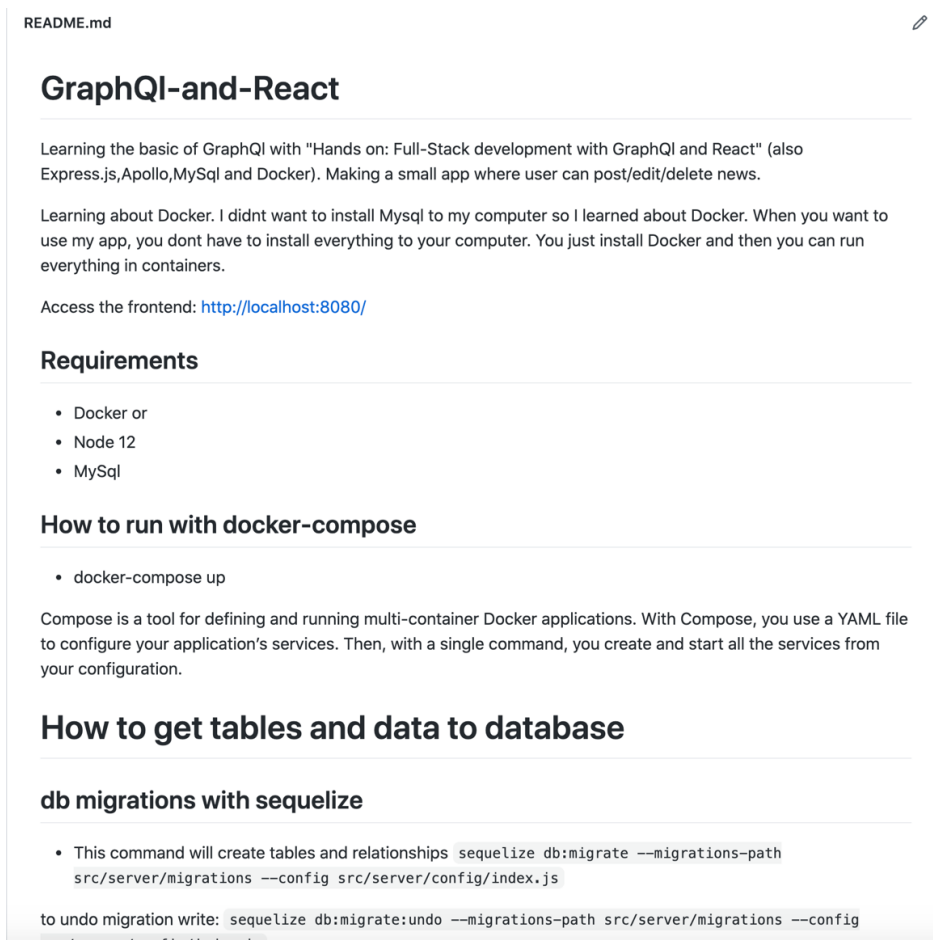
Ohjelmoinninportfolio on toteutettu GitHub-portfoliona. Kuvassa 41 on GitHub-portfolion etusivu. Etusivulla näkyy lyhyt kuvaus minusta sekä etusivulle valitsemani työt. Jokaisen työn kohdalla on lyhyt kuvaus projektin sisällöstä. Etusivulla alareunassa näkyy myös kaavio, jossa aktiivisuuteni GitHub-palvelussa on esitelty vihrein kuvioin. Mitä enemmän ja mitä tummempaa vihreää, niin sitä aktiivisempi olen ollut ohjelmoinnin saralla.

GitHub-portfoliossani tutustuja näkee yhdellä silmäyksellä paljon tietoa osaamisestani ja aktiivisuudestani. Portfolio toimii niin selaimessa kuin mobiilissa. Portfolio on toteutettu englanniksi, sillä en halua sulkea pois mahdollisuuksia kansainvälisissä yrityksissä.

The screenshot displays a GitHub profile for Kristiina Hagridaa. The profile includes a bio: "Hi, I am a third year ICT-student. I study Programming and Digital Services. I am a designer and a user-oriented problem solver." and a "Pinned" section with six repositories: "MovieApp-Springboot", "React-Book-app", "Bookstore", "questionnaire", "RobotFramework-testing", and "GraphQL-and-React". Below the repositories is a contribution calendar for the last year, showing activity with green squares. The calendar indicates 283 contributions in the last year.

Kuva 41. GitHub-portfolioni etusivu

GitHub-portfolion etusivulta projektia klikkaamalla pääsee tutustumaan jokaiseen projektiin tarkemmin. Tarkemmassa projektin esittelyssä on tarkka kuvaus projektin sisällöstä sekä ohjeet sovelluksen käynnistämiseen. Kuvassa 42 on esimerkki projektin esittelysivusta.



Kuva 42. Esimerkki GitHub-portfolioni projektin esittelystä

Ohjelmoinninportfoliooni pääsee tutustumaan tarkemmin osoitteessa <https://github.com/Hagridaa>.

Tavoitteenani oli kolmella sanalla kuvaten antaa itsestäni tietynlainen kuva portfolion avulla: luova, innokas, monipuolinen. Mielestäni onnistuin tavoitteessani erinomaisesti. Tavoitteenani oli toteuttaa myös portfolioiden käyttäjättestaus, mutta valitettavasti en saanut kattavaa testiryhmää kasaan rekrytoivien henkilöiden osalta ja näin ollen testaus jää toiseen kertaan. Jatkokehityksenä ohjelmoin nyt toteutettujen portfolioiden perusteella itselleni portfolioit. Aion toteuttaa portfolion ohjelmoinnin React-sovelluskehityksellä. Näin saan portfolioistakin osaamisen näytteen työnhakua varten.

Portfolio-opinnäytetyö oli kattava kokonaisuus, jonka aikana opin suuresti lisää projekteistani. Osaamiseni lisääntyi kaikilla tasoilla merkittävästi. Myös toinen tavoitteeni täyttyi ja toteutin tämän portfolio-opinnäytetyön pohjalta työhaussa käytettävät portfolioit. Suositelen portfolio-opinnäytetyön tekemistä muillekin opiskelijoille. Projekti oli mahtava sukellus omaan osaamiseen ja sen kehittämiseen.

Lähteet

Aalto University 2019. Sivistystoimen työkalupakki palvelumuotoiluun. Luettavissa: https://designresearch.aalto.fi/groups/encore/wp-content/uploads/2013/11/Sivistystoimen_tyokalupakki_palvelumuotoiluun2.pdf . Luettu 24.10.2019

Anne 2011. Käyttäjäpersoonat herättävät käyttäjät eloon. Aalto University Innovaatiopalvelu blogs. Luettavissa: <https://blogs.aalto.fi/itainnovaatiopalvelu/2011/10/20/kayttajapersoonat-herattavat-kayttajat-eloon/>. Luettu 2.6.2020.

Awwapp. Value propotion canvas template. Luettavissa: <https://awwapp.com/info/business-templates/value-proposition-canvas/>. Luettu 1.1.2020.

Brown, J. 2018. Empathy Mapping: A Guide to Getting Inside a User's Head. Luettavissa: <https://www.uxbooth.com/articles/empathy-mapping-a-guide-to-getting-inside-a-users-head/>. Luettu 2.11.2019.

Cosmina, J. 2018. Java for Absolute Beginners. Learn to Program the Fundamentals the Java 9+ Way. Luettavissa <https://learning.oreilly.com/library/view/java-for-absolute/9781484237786/html>. Luettu 1.7.2020.

Design Council 2015. What is the framework for innovation? Design Council's evolved Double Diamond. Luettavissa: <https://www.designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond> . Luettu 15.10.2019.

Fecak, D. What do job seeking developers need their GitHub. Blogi techbeacon. Luettavissa: <https://techbeacon.com/app-dev-testing/what-do-job-seeking-developers-need-their-github> Luettu 29.5.2020.

Gee, T. 2020. 25 Things We Love About Java. Luettavissa: <https://blog.jetbrains.com/idea/2020/05/25-things-we-love-about-java/>. Luettu 9.8.2020.

GraphQL. Learn execution. Luettavissa: <https://graphql.org/learn/execution/>. Luettu 22.6.2020.

Grönroos, C. 2009. Palveluiden johtaminen ja markkinointi. 3. uudistettu painos. Christian Grönroos, Juva: WS Bookweil Oy, 2009.

Helsingin yliopisto 2019. Web palvelinohjelmointi syksy 2019. Spring-sovelluskehys. Luettavissa: <https://web-palvelinohjelmointi-s19.mooc.fi/osa-1/3-spring-sovelluskehys>. Luettu 6.8.2020.

Helsingin Yliopisto 2019b. Ohjelmointi-19 Mooc. Luettavissa: <https://ohjelmointi-19.mooc.fi/osa-5/6-testauksen-alkeet-jatkuvat>. Luettu 2.7.2020.

Helsingin yliopisto 2020. Fullstackopen. Luettavissa: https://fullstackopen.com/osa8/graph_ql_palvelin. Luettu 15.7.2020.

Hinkula, J. 2019. Hands-On Full Stack Development with Spring Boot 2 and React - Second Edition. Luettavissa: <https://learning.oreilly.com/library/view/hands-on-full-stack/9781838822361/f558131a-c489-43ea-bafe-bc1cd3237da1.html>, Luettu 10.8.2020.

Itewiki. Sovelluskehitys. Luettavissa: <https://www.itewiki.fi/opas/sovelluskehitys/>, Luettu 22.9.2020.

Jaakkola, E., Orava, M. & Varjonen. 2009. Palveluiden tuotteistamisesta kilpailuetua opas yrityksille. Tekes. Luettavissa:

https://www.businessfinland.fi/globalassets/julkaisut/palvelujen_tuotteistamisesta_kilpailuetua.pdf?t_id=1B2M2Y8AsgTpgAmY7PhCfg%3d%3d&t_q=palvelujen+tuotteistamisesta+kilpailuetua&t_tags=language%3afi%2csiteid%3a53b34a16-7ce7-4ab0-8c7e-f06c83547e28&t_ip=77.86.196.74&t_hit.id=Finpro_Web_Features_MediaData_GenericMediaData/_1c92e6d0-e728-4066-ae1d-5e15e7b784b0&t_hit.pos=1 . Luettu 22.10.2019.

Kara, P. 2018. Affinity Diagramming for Collaboratively Sorting UX Findings and sign Ideas. Luettavissa: <https://www.nngroup.com/articles/affinity-diagram/>. Luettu 27.6.2020.

Krause, R. 4.8.2019. 5 Steps To Creating a UX-Design Portfolio. NN Groups articles. Luettavissa: <https://www.nngroup.com/articles/ux-design-portfolios/>. Luettu 1.6.2020.

Korhonen, P. 2020. LinkedIn keskustelu. Luettavissa: https://www.linkedin.com/posts/activity-6645949317410545664-2GD8?lipi=urn%3Ali%3Apage%3Ad_flagship3_profile_view_base_recent_activity_details_shares%3B6P5Tsl%2BPRtu3YDqmgQVgTA%3D%3D . Luettu 15.5.2020.

Laakkonen, C. GitHub-cv. Sofokus Blogi.

Luettavissa: <https://www.sofokus.com/fi/blogi/2020/01/29/github-cv/> Luettu 26.5.2020.

Magana, A. 2018. Version Control with Git and GitHub.

Luettavissa: https://learning.oreilly.com/library/view/Version+Control+with+Git+and+GitHub/9781789808971/C11892_1_comm_Final.html#_idTextAnchor015 Version Control with Git and GitHub. E-kirja. Magana Alex. Packt Publishign 2018. Luettu 31.5.2020.

Moodle 2019. Haaga-Helia sisäinen työtila. Käyttäjäkokemus DIG1TN002-3004 kurssin tavoitteet. Luettu 20.5.2020.

Moodle 2019b. Haaga-Helia sisäinen työtila. Palvelujen tuotteistaminen ja palvelumuotoilu CUS4LH101A-3007 kurssin tavoitteet. Luettu 20.5.2020.

Moodle 2019c. Haaga-Helia sisäinen työtila. Front-End Ohjelmointi kurssin tavoitteet. Luettu 20.5.2020.

Moodle 2019d. Haaga-Helia sisäinen työtila. Palvelinohjelmointi SWD4TN020-3008 kurssin tavoitteet. Luettu 20.5.2020.

Moodle 2020e. Haaga-Helia sisäinen työtila. Ohjelmistokehityksen teknologioita SWD4TN023-3006 kurssin tavoitteet. Luettu 20.5.2020.

Moodle 2020f. Haaga-Helia sisäinen työtila. Palvelujen tuotteistaminen ja palvelumuotoilu CUS4LH101A-3007 kurssipalaute. Luettu 20.5.2020.

Osterwalder, A., Pigneur, Y., Bernarda, G. & Smith Hoboken, A. 2014. Value proposition design: how to create products and services customers want. Alexander Osterwalder, Yves Pigneur, Greg Bernarda, Alan Smith Hoboken, New Jersey: John Wiley & Sons, Inc 2014.

Payne, A. & Pennie, F. 2013.

Payne, Adrian, and Pennie Frow. Strategic Customer Management: Integrating Relationship Marketing and CRM, Cambridge University Press, 2013. ProQuest Ebook Central, <http://ebookcentral.proquest.com/lib/haaga/detail.action?docID=1139582>.

Pietilä, M. 2020. LinkedIn keskustelu. Luettavissa: https://www.linkedin.com/posts/activity-6645949317410545664-2GD8?lipi=urn%3Ali%3Apage%3Ad_flagship3_profile_view_base_recent_activity_details_shares%3B6P5Tsl%2BPRtu3YDqmgQVgTA%3D%3D . Luettu 15.5.2020.

Reactjs. Luettavissa: <https://reactjs.org/>, Luettu: 20.8.2020.

Siang, Y, T and the Interaction Design Foundation 2020. How to create an online ux design portfolio? Luettavissa: https://www.interaction-design.org/literature/article/how-to-create-an-online-ux-design-portfolio?utm_source=linkedin&utm_medium=sm Luettu 26.5.2020.

Schenker, G. 2020. Learn Docker – Fundamentals of Docker 19.x – Second Edition. Luettavissa: <https://learning.oreilly.com/library/view/learn-docker-/9781838827472/?ar>. Luettu 13.6.2020.

Spring.io 2020. Why Spring. Luettavissa: <https://spring.io/why-spring>. Luettu 1.8.2020.

Pitkänen, T. 2019. Miten tunnistaa kohderyhmä. Schmidt Communications kirjoitus. Luettavissa: <https://s-com.fi/tuomas/miten-tunnistaa-kohderyhma/> .Luettu 20.10.2019.

Jaakkola, E. 2019. Turun kauppakorkeakoulu. Youtube-video. Luettavissa https://www.youtube.com/watch?v=gh_TS_oyQvM . Luettu 20.8.2019.

Tutorialspoint. Spring Boot internationalization. Luettavissa: https://www.tutorialspoint.com/spring_boot/spring_boot_internationalization.htm. Luettu 4.8.2020.

Väätäinen, M. 2019. Miksi jokaisen johtajan tulisi ymmärtää palvelupolku. Luettavissa: <https://www.solita.fi/blogit/miksi-jokaisen-johtajan-tulisi-ymmartaa-palvelupolku/>. Luettu 1.11.2019.

Zeithaml, V., Parasuraman, A. & Berry, L. 1990.
Delivering Quality Service Balancing customer perceptions and expectations.
The Free Press A division of Macmillan, Inc.