



Käyttäjäkeskeisen verkkopalvelun käyttöliittymän kehittäminen Bootstrapilla

Antti Jääskeläinen

2020 Laurea



Laurea-ammattikorkeakoulu

Käyttäjäkeskeisen verkkopalvelun käyttöliittymän kehittäminen Bootstrapilla

Antti Jääskeläinen
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Huhtikuu, 2020

Antti Jääskeläinen

Käyttäjakeskeisen verkkopalvelun käyttöliittymän kehittäminen Bootstrapilla

Vuosi

2020

Sivumäärä

29

Tämän opinnäytetyön tavoitteena oli kehittää käyttäjakeskeisen verkkopalvelun käyttöliittymä hyödyntäen Bootstrap-komponenttikirjastoa. Työn aiheena oli luoda uudenlainen palvelu, joka antaa mahdollisuuden yksityishenkilöille ajaa autoalan yrityksen auto ilmaiseksi yrityksen tarjoamaan kohteeseen, saaden itse myös ilmaisen kyydin samaiselle paikkakunnalle. Toimeksiantajana toimi vanha työkaverini, jonka kanssa kehitin palvelua yhteistyössä.

Työn teoriaosuudessa käsiteltiin käytettävyyden teoriaa tietotekniikan näkökulmasta, sen vaikutusta palvelun käyttäjiin ja seurauksiin käytettävyyden saralla epäonnistumisesta. Kerroin myös teoriapohjaa Bootstrapista, sen ominaisuuksista ja toiminnoista. Opinnäytetyöni oli kehityspainotteinen, teoriaosuudessa rakensin pohjan sivuston käytettävyydelle, ja palvelun toteuttamiseen käytin hyödykseni teoriapohjaa.

Opinnäytetyöni tuloksena syntyi teoriapohjaan tukeutuen yksinkertainen verkkopalvelun käyttöliittymä, joka on käytettävyydeltään ennalta-asetettujen kriteerien mukainen. Työstä saatiin uutta tietoa Bootstrapin ja käytettävyyden yhteensopivuudesta. Bootstrap yllätti helppokäyttöisyydellään, sekä tarjosi loistavan alustan palvelun luomiselle käytettävyyden näkökulmasta ajatellen. Palvelu olisi vielä jatkokehityksen kannalta tarvinnut kunnollisen käyttäjätestauksen. Ajan puutteen vuoksi tämä jäi valitettavasti tekemättä, ja kunnollista arviota todellisen loppukäyttäjän näkökulmasta on vaikea antaa.

Asiasanat: bootstrap, verkkopalvelu, käyttäjakeskeisyys

Antti Jääskeläinen

Creating a user-centered user interface using Bootstrap

Year 2020

Pages

29

The purpose of this thesis was to develop a user interface of a web service using Bootstrap component library. Idea was to create a new service, that would give a chance for private persons to drive a car dealership car to a destination and getting a ride to the same destination for free. The service was created for a former colleague who worked with cars and their transportation.

Knowledge base consisted of theory from usability, focusing on the viewpoint of information technology, and Bootstrap, mainly about its functions and attributes. This thesis was a practice-based research. Knowledge base was created about usability theory, and then the web service was created according to the knowledge gained during the process.

End-result was a theory-based web service, that serves its function well. It became very simple to use, and it fulfilled all the tasks that we made during the planning process. I learned new information regarding usability and web services, and especially Bootstrap. Bootstrap offered a solid platform of creating user-centered web services. Web service itself could have benefitted of some usability testing. It is hard to tell if the service is usable for the end user without usability testing.

Keywords: bootstrap, web service, user-centered design

Sisällys

1	Johdanto	6
2	Työn tausta ja kuvaus	6
2.1	Tutkimuskysymykset ja tavoite.....	7
2.2	Aihealueen rajaus.....	7
2.3	Keskeiset käsitteet.....	8
3	Käytettävyys.....	9
3.1	Nielsenin käytettävyyden määritelmä.....	9
3.2	ISO standardi 9241-210.....	10
4	Käyttäjäkeskeinen suunnittelu.....	10
4.1	Käyttäjäkokemus.....	10
4.2	Käyttäjäkokemuksen kaari	11
4.3	Käyttäjän tarpeet.....	12
5	Käyttöliittymäsuunnittelu.....	13
5.1	Värit ja visuaalisuus	13
5.2	Responsiivisuus ja saatavuus.....	14
5.3	Navigaatio	14
5.4	Käytettävyydeltään onnistuneita verkkopalveluita	15
6	Bootstrap.....	16
6.1	Bootstrapin tausta	16
6.2	Bootstrapin ominaisuuksia.....	17
6.2.1	CSS-esiprosessorit	17
6.2.2	Grid	18
6.2.3	Flex	19
7	Tutkimus- ja kehittämismenetelmät.....	20
8	Siirtoajat-verkkopalvelu.....	20
8.1	Projektin eteneminen	21
8.2	Bootstrap käyttäjäkeskeisen kehittämisen tukena.....	22
9	Yhteenveto, johtopäätökset ja kehitysehdotukset	24
	Lähteet	26
	Kuviot	29

1 Johdanto

Opinnäytetyössä toteutettiin Bootstrapilla käytettävyyteen perustuva verkkopalvelu liittyen henkilöautojen siirtämiseen. Sovelluksesta kehitettiin käyttöliittymä Bootstrap-komponenttikirjastoa apuna käyttäen. Kehitystyössä oli mukana entinen työkaverini, jonka kanssa oltiin tiiviisti yhteistyössä palvelun kehittämisen ajan. Aihealue oli mieluinen, sillä olen opiskeluaikanani saatossa kiinnostunut front-end koodaamisesta, ja töiden puolesta auto- ja kuljetusala tuli hyvin tutuksi.

Verkkopalveluiden yleistyessä käytettävyyden roolista on tullut tärkeimpiä osia palvelun käyttäjäkokemuksen kannalta. Palvelu tai tuote voi olla visuaalisesti erittäin näyttävä, mutta jos perusoletuksena käytettävyys ei toteudu, niin palvelulla voi olla suuri riski menettää käyttäjäkuntaa. Teoreettisessa viitekehyksessä pohditaan käytettävyyden teoriaa tietotekniikan näkökulmasta ja kerrotaan myös, kuinka käyttäjäkokemus voi olla ratkaisevassa roolissa palvelun onnistumisessa. Käyttäjäkokemuksella on myös merkitystä taloudellisesti yrityksen digitalisaation ja käyttäjäkunnan pitämisen kannalta. Kehitetyn verkkopalvelun suunnitteluun on käytetty aikaa miettien loppukäyttäjän roolista palveluun vaadittavia toiminnallisuuksia ja käyttöliittymän ulkonäköä.

Bootstrap on nykyaikainen komponenttikirjasto, joka sisältää valmiita tyylimäärittelyitä, sekä elementtejä koodauksen helpottamiseksi. Nykyaikana verkkopalveluita käyttävät ihmiset, jotka omistavat monia erilaisia laitteita. Bootstrap valittiin avuksi kehittämiseen siksi, että siinä on järkevä pohja rakentaa palvelu, joka skaalautuu automaattisesti niin tietokoneelle kuin mobiililaitteillekin.

Tarkoituksena opinnäytetyössä oli luoda Bootstrapilla verkkopalvelu, joka on käytettävyydeltään soveltuva kaikenikäisille käyttäjille. Palvelun käyttöliittymä rakennettiin ja suunniteltiin käytettävyyden teoriapohjaan tukeutuen. Opinnäytetyön tarkoituksena on myös osoittaa se, että toimivan verkkopalvelun käyttäjäpuolen kehittäminen Bootstrapin avulla voi olla helppoa pelkillä perustiedoilla front-end kehityksestä.

2 Työn tausta ja kuvaus

Olin noin vuoden ajan töissä autonkuljettajana, ja pohdimme keväällä 2019 entisen työkaverini kanssa, jos olisi jollain tavalla mahdollista helpottaa autojen kuljetusta paikkakunnalta toiselle. Autoalan yrityksillä on omia kuljettajia, jotka hoitavat rutiiniajoja, mutta pidemmille matkoille harvemmin voidaan ottaa palkattua kuljettajaa, sillä työaikasäännökset tulevat vastaan.

Esimerkkikäyttötapaus palvelulle olisi, että yrityksellä on auto, joka halutaan kuljettaa esimerkiksi Helsingistä Ouluun. Yritys laittaa verkkopalveluun tiedon, että halutaan kuljettaja

välille Helsinki-Oulu, ja haluttuna päivämääränä. Kuljettaja, joka etsii autoa ajettavaksi samaiselle välille, ottaa yhteyttä autoliikkeeseen ja sopii tapaamisen yrityksen edustajan kanssa. Tämän jälkeen kuljettaja ajaa auton sovittuun paikkaan kohdekaupungissa, ja näin kuljettaja on saanut ilmaisen kyydin paikkakunnalle, ja autoliike on saanut siirrettyä autonsa haluttuun kohteeseen.

On olemassa jo Facebook-ryhmä, joka tarjosi käyttäjille mahdollisuuksia ajaa ilmaiseksi paikkakunnalta toiselle yritysten siirrettävillä autoilla. Ryhmään tulee yleensä ilmoituksia, jossa haetaan kuljettajaa autolle paikkakuntien välillä. Yleensä kommunikaatio tapahtui sähköpostilla tai puhelimitse. Mietimme, että saattaisi olla tarvetta verkkopalvelulle, joka hoitaa yritysten sekä kuljettajien tarpeita.

Idea lähti kehittymään, ja pohdimme yhdessä entisen työkaverini kanssa verkkopalvelun kehittämistä. Mietin samalla, että voisin tehdä tästä opinnäytetyöni. Toteutin sovelluksesta käyttöliittymän. Sovellus toteutettiin itse koodaamalla, sillä valmiista julkaisualustoista, kuten WordPress tai Drupal, ei löydy tarvittavia ominaisuuksia palvelun käytölle tai ylläpidolle. Koodaukseen käytettiin myös apuna Bootstrap-nimistä CSS- ja JavaScript-frameworkia, josta kerrotaan myös tässä opinnäytetyössä.

2.1 Tutkimuskysymykset ja tavoite

Tutkimuskysymykset opinnäytetyössäni ovat:

- Mitkä ovat käyttäjäystävällisen verkkopalvelun kriteerit ja miten sellainen tuotetaan?
- Miten Bootstrapilla voidaan toteuttaa käyttäjäystävällinen verkkopalvelu?

Tavoitteena opinnäytetyölle on luoda uudenlainen verkkopalvelu, jonka tehtävänä on palvella autoalan yrityksiä, jotka ovat halukkaita siirtämään autojaan paikkakunnalta toiselle. Toinen käyttäjäkunta on palvelua käyttävät yksityishenkilöt, jotka ovat halukkaita siirtymään paikkakunnalta toiselle yritysten tai toisten yksityishenkilöiden siirrettävillä autoilla. Opinnäytetyössä käsitellään verkkopalveluiden front-puolen kehittämistä Bootstrapilla käytettävyyden näkökulmasta, sekä kerrotaan Bootstrap-frameworkin teoriaa ja sen soveltuvuutta käytettävän verkkopalvelun rakentamiseen.

2.2 Aihealueen rajaus

Käytettävyys on itsessään todella laaja käsite, joten tässä työssä keskitytään nimenomaan verkkopalveluiden käytettävyyteen. Työstä on rajattu myös pois käytettävyydestäminen aikataulusyistä. Kaikki tuotettu materiaali pohjautuu käytettävyyden olemassa olevaan teo-

riapohjaan. Aihealue rajoittuu myös verkkokehittämisen kannalta Bootstrapin teoriaan ja käytännön osuuteen. Työssä ei käsitelty esimerkiksi syvemmillä tasolla yleisesti verkkosivujen tekemistä.

Olemassa oleva aiheeseen liittyvä Facebook-ryhmä toimii myös osittain vertailukohtana palvelun käytettävyydelle. Tässä työssä ei kuitenkaan lähdetä tarkastelemaan tarkemmin palveluiden välisiä käytettävyyseroja, sillä pohjaksi ei ole kummallekaan käyttäjätutkimusta aiheesta.

2.3 Keskeiset käsitteet

Verkkopalvelu	Internet-verkossa toimiva palvelu, voidaan rinnastaa usein myös verkkosivustoon.
Front-end	Osa sovelluksesta tai verkkosivusta, joka on silmin nähtävissä käyttäjälle. Front-endiä kirjoitetaan yleisesti HTML:llä, JavaScriptillä ja CSS:llä.
Bootstrap	Avoimeen lähdekoodiin perustuva komponenttikirjasto, joka sisältää valmiita tyylimäärittelyjä sekä elementtejä, jolla voidaan helpottaa verkkosivujen luomista.
HTML	Hypertext Markup Language, merkkaukieli, jolla verkkosivut kirjoitetaan.
CSS	Cascading Style Sheets, verkkosivujen tyylimäärittelyt kuten esimerkiksi värien, asemointien ja fonttien määrittely tapahtuu CSS:n avulla.
JavaScript	Koodikieli, jolla verkkosivuille voidaan lisätä toiminnallisuuksia. JavaScriptillä voidaan tehdä esimerkiksi merkkijonoja, toiminnallisia nappeja tai ponnahdusikkunoita.
GitHub	Kehittäjien versionhallintasovellus, jossa voidaan ylläpitää projekteja.
Framework	Ohjelmoinnin apuväline, joka nopeuttaa palveluiden kehittämistä, tarjoaa valmiiksi rakennettuja palvelun osia, joita ei tarvitse kirjoittaa aina erikseen koodiin

3 Käytettävyys

Käytettävyys terminä kuvaa sitä, miten palvelulla voidaan saavuttaa tavoitteita tehokkaasti, tuloksellisesti, ja miellyttävästi. Tehokkuuden määritelmä on ihmisen voimavarojen käyttötarpeen suhde tavoitteiden saavuttamiseen. Miellyttävyyden kokemisen edellytys on tyytyväisyys, joka tarkoittaa epämukavuuden puuttumista ja positiivista suhtautumista palvelun käyttöön. (Väyrynen, Nevala & Päivinen, 2004).

Käytettävyys liitetään usein tietotekniikkaan ja käyttäjakeskeisyys on digitalisaation myötä lisääntynyt verkko- ja tietotekniikan tuotannon saralla. Verkkopalveluiden käytettävyydessä tärkeimmässä roolissa on käyttöliittymäsuunnittelu. Tietotekniikassa käytettävyyteen liittyy usein myös palvelun tai tuotteen esteettömyys ja käyttäjäkokemus. Nykyaikana digitaalisten sisältöjen täytyy tavoittaa kaiken ikäistä väestöä, ja sen takia myöskin suunnittelijoiden on panostettava saavutettavuuteen.

Tietotekniikan puolella käytettävyyteen aloitettiin kiinnittää huomiota 1980-luvulla. Ensimmäiset julkaisut tulivat suurilta tietotekniikan yrityksiltä. Modernit käytettävyyden määritelmät kuitenkin alkoivat kehittymään vasta 1980-luvun lopulla, kun testeihin otettiin mukaan tavoitteellisuutta alkukehityksessä, prototyyppien kehittämistä ja iteratiivista arviointia (Sauro, 2013.). Jacob Nielsenä pidetään yleisesti käytettävyyden oppi-isänä tietotekniikan puolella. Hänen julkaisemansa kirja Usability Engineering oli ensimmäisiä teoksia, joka avasi käytettävyyden tärkeyttä tietotekniikan kehittyvässä maailmassa.

3.1 Nielsenin käytettävyyden määritelmä

Kirjassaan Usability Engineering, Jacob Nielsen määrittää käytettävyydelle viisi laatukomponenttia: opittavuus, tehokkuus, muistettavuus, virheettömyys sekä tyytyväisyys. (Nielsen 1994, 26.) Opittavuudella tarkoitetaan järjestelmän helppoutta oppia ja omaksua käyttöön. Käyttäjän tulisi saada omaksuttua järjestelmän käyttö mahdollisimman nopeasti. Tehokkuutta voidaan mitata siinä vaiheessa, kun käyttäjä on oppinut järjestelmän käyttämisen. Järjestelmän pitäisi olla kykeneväinen mahdollisimman korkeaan tasoon tuottavuudessa. Muistettavuuden osalta järjestelmän tulisi olla mahdollisimman helppokäyttöinen, jotta tavallinenkin käyttäjä voi palata takaisin käyttämään järjestelmää, ilman että hänen täytyy opetella kaikkea käyttöön liittyvää uudestaan. (Nielsen 1994, 26)

Virheettömyydellä pyritään siihen, että järjestelmässä on mahdollisimman pieni määrä ihmisen käyttöön liittyvää virhemarginaalia. Ja vaikka käyttäjä tekisikin virheitä ohjelman käytössä, niin hänen pitäisi olla myös kykeneväinen palaamaan takaisin normaaliin käyttöön virheen sattumisen jälkeen. Tyytyväisyyden mittarina toimii järjestelmän miellyttävyys käyttäjälle. Jos he ovat yleisesti tyytyväisiä käyttäessään järjestelmää, niin he pitävät siitä. (Nielsen 1994, 26.)

3.2 ISO standardi 9241-210

Kansainvälisen standardoimisjärjestön tuottaman ISO 9241 mukaan, käytettävyys tarkoittaa tiettyä laajuutta, jolla tuotteita, palveluita ja järjestelmiä voivat käyttää kaikki käyttäjäryhmät. Käyttäjien tarpeet, yksilöllisyys ja osaaminen on otettava huomioon. ISO standardissa käytettävyys on jaettu kolmeen eri osa-alueeseen: tuottavuus, tehokkuus ja miellyttävyys. (ISO 2019.)

Tuottavuudessa mitataan suoritettujen toimintojen määrää, virheiden määrää ihmisen toiminoissa käyttäessä järjestelmää ja suorituksen onnistuneisuutta päämääräkohtaisesti. Tehokkuudessa käsitellään tehtäviin kulunutta aikaa, ajankäytön hyödyllisyyttä, produktiivista aikaa, väsymystä ja tarpeettomien funktioiden määrää. Miellyttävyydessä arvioitavia osa-alueita ovat käyttäjien yleinen tyytyväisyys, tyytyväisyys ominaisuuksiin, ominaisuuksien hyödyntäminen, käyttäjän luottamus ja käyttäjän yleinen tyytymättömyys, sekä ominaisuuksien puute. (ISO 2019.)

4 Käyttäjakeskeinen suunnittelu

Käyttäjakeskeisen suunnittelun lähtökohtana on ollut mieltä käyttäjän tarpeita ohjelmalle. Kun käyttäjät ovat suunnittelun keskiössä, niin lopputuloksena oleva palvelu on laadultaan huippuluokkaa niin käytettävyudessa kuin käyttäjäkokemuksessa. Käyttäjakeskeisessä suunnittelussa käytetään paljon aikaa käyttäjätutkimukseen ja sen avulla pystytään todistamaan tilastollisesti tapahtumaketju palvelun käytöstä. Suoraan käyttäjiä analysoimalla voidaan myös poistaa ennakkoluuloja ja luoda uusia suuntauksia palvelun kehittämiseen. (Lowdermilk 2013, 5.)

Käyttäjiltä pyritään saamaan tarkkaa tietoa siitä, miten heidän mielestään palvelu käyttäytyy tietyn käyttötilanteen aikana. Tilannetta voidaan avata kysymällä käyttäjältä tulkintaa esimerkiksi virheilmoituksesta tai tilanteesta, jossa käyttäjä ei ole varma, kuinka palvelussa jatketaan eteenpäin. Yhdistämällä käyttäjakeskeisen suunnittelun, käytettävyyden ja käyttäjäkokemuksen, voidaan luoda palvelu, joka on kilpailukykyinen myös nykypäivän kehittyvässä markkinatilanteessa. (Lowdermilk 2013, 12.)

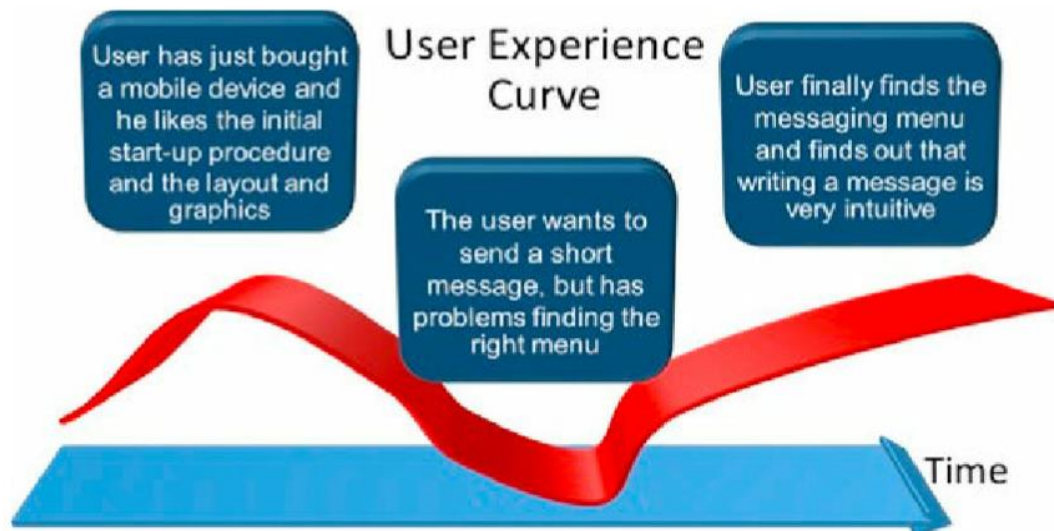
4.1 Käyttäjäkokemus

ISO standardi 9241-210 määrittelee käyttäjäkokemuksen henkilön käsityksenä ja vastauksena, joka muodostuu järjestelmän, palvelun tai tuotteen käytöstä. Käyttäjäkokemus koostuu käyttäjän tunteista, preferensseistä, uskomuksista, fyysisistä ja psyykkisistä reaktioista ja saavutuksista käytön aikana ja jälkeen. (ISO 2019.)

Käyttäjäkokeemusta suunniteltaessa ei pidä perehtyä pelkästään palvelun käytettävyyteen, vaan halutaan myös tarttua muihin osa-alueisiin kuten käytön miellyttävyyteen, hauskuuteen ja tehokkuuteen. Jokainen käyttäjäkokeemus perustuu käyttäjän yksilölliseen tarpeeseen palvelun käyttöön tai jollekin sen tietyille ominaisuudelle. (Interaction Design Foundation 2019.)

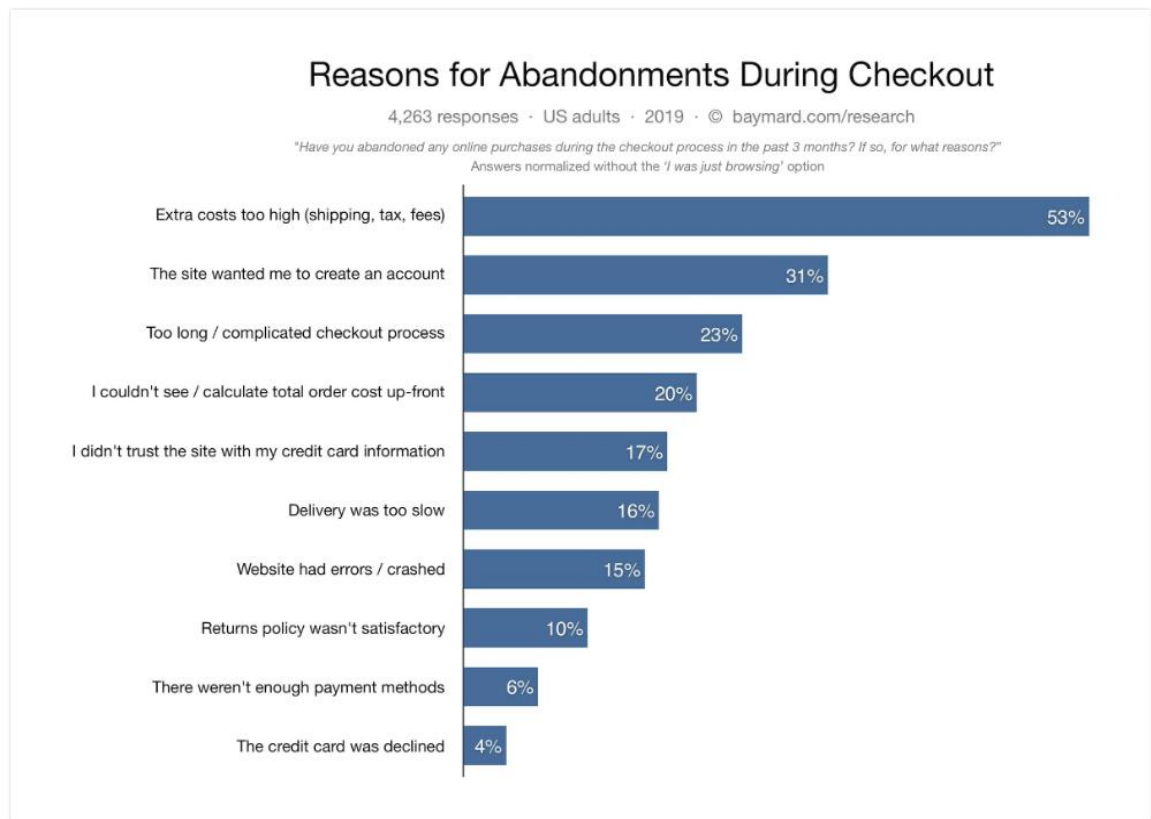
4.2 Käyttäjäkokeemuksen kaari

Kun käyttäjä käyttää palvelua, niin hänelle syntyy monenlaisia tunnetiloja erilaisissa käyttötilanteissa. Kun syntyy positiivisia käyttökokeimuksia, kaari menee ylös. Kun taas käyttökokeemukset ovat negatiivisia, niin kaari painuu alaspäin (kuvio 1). Riittävän monet vastoinkäymiset palvelun käytön yhteydessä pilaavat käyttäjäkokeemuksen, ja tämän seurauksena on mahdollista, että käyttäjä lopettaa kokonaan palvelun käyttämisen. (Kraft 2012, 2)



Kuvio 1: Käyttäjäkokeemuksen kaari. (Kraft 2012, 2.)

Kuviossa 2 listataan yleisimpiä syitä verkkosivulla olevan ostotapahtuman perumisiksi. Muutama näistä liittyy oleellisesti palvelun käyttöliittymän käyttäjäkokeemukseen. 23 % vastaajista oli sitä mieltä, että liian pitkään kestänyt tai monimutkainen vahvistusprosessi on synnyttänyt vahvistuksen perumisiksi, ja 15 % vastaajista kertoo syyksi verkkosivulla olleet viat tai koko palvelun kaatumisen.



Kuvio 2: Syitä ostopäätöksen hylkäämiseksi (Baymard 2019.)

Käyttäjien ennakoasenteet ja käsitykset palvelusta ovat myös merkittäviä palvelun kannalta. Tuntemattomammalle yritykselle pienet ennakkoluulot voivat olla merkittävä tapa erottua pienten yritysten joukosta tarjoamalla ensiluokkaisen käyttäjäkokemuksen. Toisaalta myös suuret ennakkoluulot brändiä kohtaan voivat olla hyödyllisiä, sillä se antaa mahdollisuuden veloittaa enemmän tuotteesta, sillä käyttäjät ovat valmiita maksamaan laadukkaasta palvelusta. (Kraft 2012, 3-4.)

4.3 Käyttäjän tarpeet

Jotta voi arvioida käyttäjien tarpeita palvelulle, on ensin määriteltävä käyttäjäryhmä. Opin- näytetyössä kehitetyssä palvelussa käyttäjät koostuvat käytännössä ajokortin omaavista henkilöistä, joten käyttäjien ikähaarukan voi asettaa noin 18-80 ikävuoteen. Suurempi ikähaitari tarkoittaa myöskin paljon yksilöllisiä käyttäjäkokemuksia, ja vaatii suunnittelussa ja toteutuksessa tiedostamista siitä, että käyttäjiä on paljon erilaisia.

Käyttäjien tarpeet ja käyttäjäryhmät voivat kuitenkin vaihtua kesken palvelun kehittämisen. Facebook oli alun perin tarkoitettu vain Harvardin Yliopiston opiskelijoille, sen jälkeen sitä

laajennettiin muihin kampuksiin Bostonin alueella. Vuonna 2019 Facebookilla oli 2,43 miljardia käyttäjää, ja käyttäjäkuntaan kuuluu hyvin laaja kirjo ihmisiä. Facebook toimii hyvänä esimerkkinä siitä, miten käyttäjäkunta voi vaihtua hyvinkin radikaalisti. (Kraft 2012, 3-4.)

Tarpeiden muuttuessa myös palvelua joudutaan uudistamaan, jotta se soveltuu uusille käyttäjäryhmille. Se tuo haasteita kehittämiseen, ja vaatii paljon tutkimusta, jotta saadaan mahdollisimman laaja otanta ihmisiä pysymään palvelun käyttäjinä. (Kraft 2012, 3-4.)

5 Käyttöliittymäsuunnittelu

Käyttöliittymä on palvelun normaalissa käytössä käyttäjälle näkyvä osa. Yleisesti verkkopalveluiden käyttöliittymät ovat graafisia käyttöliittymiä, joissa halutut toiminnot suoritetaan suoraan palvelusta esimerkiksi navigoiden valikoiden läpi hiirellä tai klikkaamalla painikkeita, joista pääsee eteenpäin halutulle sivulle tai toiminnolle.

Jakob Nielsenin mukaan käyttöliittymän pitäisi olla mahdollisimman yksinkertaistettu, sillä jokainen lisätty ominaisuus tai kohde näytöllä on aina käyttäjälle uusi asia opeteltavaksi. Ideaalitilanteessa käyttäjälle annetaan täsmälleen tarvittava määrä informaatiota paikkaan, jossa sitä tarvitaan. Informaation sekä operaatioiden pitäisi myös olla selkeässä järjestyksessä niin, jotta käyttäjän ajankäyttö ja toiminnallisuuksien hyöty voidaan maksimoida. (Nielsen 1993, 115-116) Lisandra Maioli kertoo tärkeäksi ymmärtää palvelua tilaavaa yritystä, sekä palvelun loppukäyttäjiä. Lopputuloksen pitäisi molempia tyydyttävä ratkaisu, joka taloudellisesti hyödyttää yritystä, sekä palvelee loppukäyttäjää itse palvelun käyttötarkoituksessa. (Maioli 2018, 19.)

5.1 Värit ja visuaalisuus

Graafisen suunnittelun pitäisi olla apuna käyttäjälle ymmärtämään käyttöliittymän rakennetta. Värit voivat auttaa käyttäjää priorisoimaan kohteita näytöllä erottamalla selkeästi taustaväristään. Ominaisuuksia voi myös korostaa esimerkiksi muuttamalla fonttia isommaksi tai vaihtamalla tekstin tai objektin kirjainkokoja isoiksi kirjaimiksi. Kuitenkaan kaikkia ominaisuuksia ei kannata liikaa korostaa, sillä käyttäjän huomio voi herpaantua helposti tärkeimmistä kohteista. (Nielsen 1993, 118-119.)

Yleismaailmalliset värikoodit pätevät myös usein verkkosuunnittelussa. Punainen väri elementissä tarkoittaa yleisesti varovaisuutta tai virhettä (esimerkiksi sulkemisikkunan ruksi punaisella taustalla tai virheilmoitukset). Vihreä on yleisesti esimerkiksi merkki eteenpäin jatkamiselle tai toiminnon hyväksymiselle. Jakob Nielsen kertoo (1993, 119.) vaaleanharmaiden tai hillittyjen pastellivärien olevan parempia kuin räikeät värit. Hyvä taustaväri sivulle on sellainen, josta teksti on helposti luettavissa. Värit auttavat käyttäjää lukemaan selkeämmin sisälön palvelun ylätunnisteesta tai runkorakenteesta. Sanomalehdistä tuttu valkoinen tausta ja musta teksti on suosittu yhdistelmä värejä. (Maioli 2018, 128.)

5.2 Responsiivisuus ja saatavuus

Responsiivisuus tarkoittaa terminä palvelun skaalautuvuutta laitteen näytön koosta huolimatta. Googlen mukaan 94 % älypuhelimien omistajista etsii tarvittavaa tietoa puhelimeltaan. Myöskin 77% kotona tai töissä tapahtuvista hauista tapahtuu puhelimella, vaikka yleisesti kyseisissä paikoissa hakuun voisi olla käytettävissä myös tietokone. (Google 2019.)

Saatavuudella tarkoitetaan palvelun käytön mahdollisuutta huolimatta siitä, minkälaisesta käyttäjästä on kyse, tai millä laitteella käyttäjä käyttää palvelua. Navigoinnin täytyy olla helppoa ja rajoituksetonta (Maioli 2018, 150.). W3C (World Wide Web Consortium) eli kansainvälinen WWW-suosituksia luova yhteisö on luonut saavutettavuudelle oman suosituksensa (kuvio 3). (W3C 2019.)

<p>Havaittavuus - Sisällön on ja komponenttien on oltava näkyvillä käyttäjille tavalla jolla he voivat sen tulkita</p>	<p>Hallittavuus - Käyttöliittymä ei voi sisältää komponentteja, jotka vaativat toimenpiteitä joita ei voi suorittaa</p>
<p>Ymmärrettävyys - Käyttöliittymän on oltava helposti tulkittava ja ominaisuus ei saa olla liian monimutkainen</p>	<p>Lujatekoisuus - Kun teknologiat kehittyvät, sisällön on oltava saavutettavissa</p>

Kuvio 3: Neljä suositusta verkkosisällön saavutettavuuteen (W3C 2019.)

5.3 Navigaatio

Sivulla navigointi voi usein johtaa käyttäjän väärälle sivustopolulle, vahvistaen käyttäjäkokemuksen negatiivista osaa. Tällaisen tilanteen sattuessa käyttäjä joutuu palaamaan sivuja tai linkkejä pitkin taaksepäin, ja tämä osaltaan pilaa käyttäjäkokemuksen. Palvelussa liikkumisen pitäisi olla mahdollisimman yksinkertaista ja järkeenkäypää. Navigaatiossa käyttäjää voidaan ohjata esimerkiksi värikoodeilla tai elementtien kokomäärittelyillä. Tarkka sivustorakenteen suunnittelu auttaa myös palvelun toteutusvaiheessa.

Toistuvat samanlaiset elementit auttavat käyttäjää navigoimaan palvelussa eteenpäin, sillä toiminnot ovat jo tuttuja aikaisesta käyttökokemuksesta. Myöskin sivun käyttämisen nopeus

kasvaa, jolloin tehokkuus sivulla on maksimoitu. Joitain kohtia palvelussa voi myös nopeuttaa automaation avulla, esimerkiksi täyttämällä valmiiksi osia lomakkeesta. (Usability.gov 2019.)

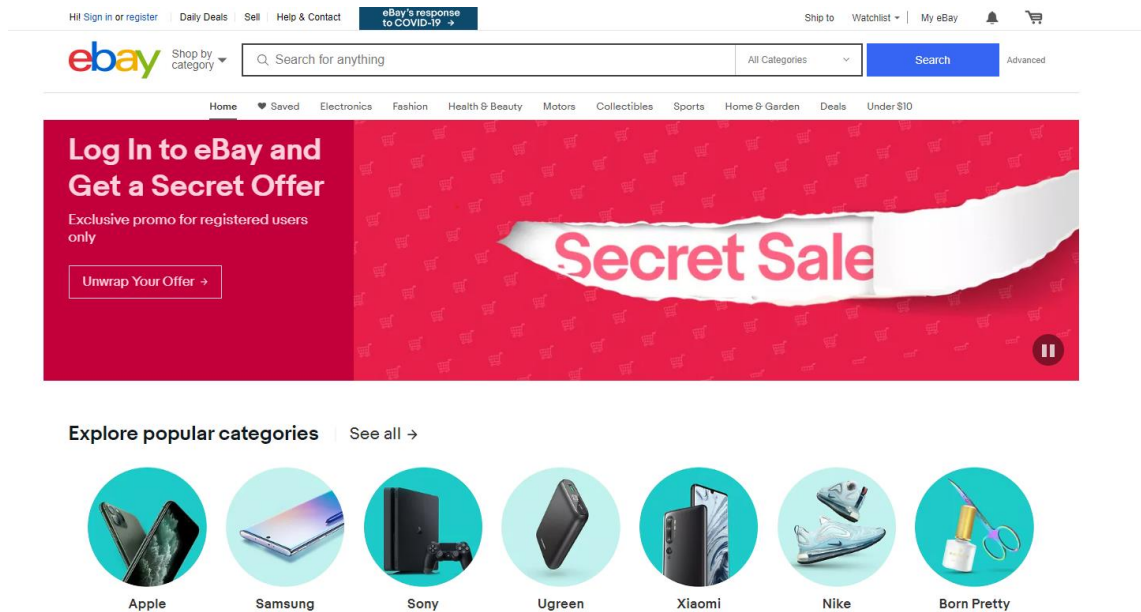
5.4 Käytettävyydeltään onnistuneita verkkopalveluita

Amazonin Alexa-palvelun tarjoama listaus The top 500 sites on the web sisältää suosituimmat 500 sivustoa koko maailmasta. (Alexa, 2019) Suosituimpien palveluiden täytyy olla käytettävyydeltään huippuluokkaa, jotta ne palvelevat käyttäjää tehokkaasti ja auttavat heitä löytämään sivustolla täytettävän tavoitteen helposti ja nopeasti. Ensimmäisenä listalla on Google.com, maailman suosituin hakukone (kuvio 4). Google on käyttöliittymältään hyvin yksinkertainen, värimaailmaltaan vaalea ja eteneminen ja tavoitteeseen pääseminen on hyvin vaivatonta. Keskiössä sivustossa on suuri hakupalkki, johon kirjoitetaan haettava asia. Tämän jälkeen sivustolta pääsee hakutuloksiin, ja valitsemaan halutun sivuston, jolle pääsee klikkaamalla linkkiä.



Kuvio 4: Googlen etusivu (Google 2020.)

Maailman suosituimpiin verkkokauppoihin kuuluva eBay on myös hyvä esimerkki käytettävyydeltään hyvästä sivustosta (kuvio 5). Sivustolla keskiössä on pyörivä banneri, jossa näkyy erinäköisiä tarjouksia. Ylhäällä on suuri hakupalkki, josta voi hakea tarvitsemaansa tuotetta verkkokaupasta. Hakupalkin alapuolella sijaitsee valmiiksi määriteltäviä hakukategorioita helpottamaan käyttäjää löytämään nopeasti omanlaisensa kategorian tuotteita. Värimaailmaan myöskin vaalea, mutta tehosteväreillä tuotu aikaan helpotusta käyttäjän navigaatioon, esimerkkinä sininen Search -painike. Ajoneuvojen osia hakiessa eBay kysyy suoraan ajoneuvon vuosimallia, jotta käyttäjän hakutuloksia voisi rajoittaa lähemmäs sitä mitä ollaan etsimässä.



Kuvio 5: eBayn etusivu (eBay 2020.)

6 Bootstrap

Bootstrap on avoimeen lähdekoodiin perustuva komponenttikirjasto. Sen tarkoituksena on toimia apuvälineenä verkkosivujen luonnissa. Bootstrap sisältää valmiita HTML ja CSS-pohjaisia tyylimalleja, joita voi hyödyntää esimerkiksi teksti- ja kirjainasettelussa, navigaation luomisessa ja nappuloiden tekemisessä Bootstrapissa on myös jQuery-liitännäisiä, joilla voi luoda erilaisia toiminnallisuuksia, kuten esimerkiksi pop-up laatikoita, kuvakaruselleja, pudotusvalikoita ja vihjetekstejä. (w3schools 2019.)

Tällä hetkellä Bootstrapista on jakelussa versio Bootstrap 4. Tähän kehitysversioon on tuotu uusia komponentteja, parempi skaalautuvuus laitekohtaisesti, sekä nopeampi tapa luoda sivustoja. Bootstrap tarjoaa maksullisena ominaisuutena myös valmiiksi tehtyjä teemoja omaa alustaa käyttäen. (Bootstrap 2019.)

6.1 Bootstrapin tausta

Bootstrapin loivat Twitteriä kehittäneet Mark Otto ja Jacob Thornton. Twitterin alkutaipaleella koodaajat käyttivät erilaisia komponenttikirjastoja heidän front-end kehitystarpeisiinsa. Monen komponenttikirjaston käyttäminen loi kuitenkin koodiin epäjohdonmukaisuutta, ja kehittäminen törmäsi ongelmiin projektinhallinnallisella tasolla. (Bootstrap 2019.)

Pieni ryhmä kehittäjiä aloitti uuden komponenttikirjaston suunnittelun johdonmukaisuuden säilyttämiseksi. Komponenttikirjasto sisälsi vanhojen kirjastojen parhaimmat ominaisuudet,

yhdistäen ne uudeksi yksittäiseksi kokonaisuudeksi. Bootstrap oli alkuperäiseltä kehitysnimeltään Twitter Blueprint. Muutaman kuukauden kehittämisen jälkeen, Twitter piti ensimmäisen Hack Week-tapahtuman. Siellä Bootstrap sai ensimmäisen stabiilin versionsa, jota Twitterin työntekijät pystyivät käyttämään verkkopalvelun kehittämiseen. Bootstrap julkistettiin avoimen lähdekoodin komponenttikirjastona elokuussa 2011. (Bootstrap 2019.)

Nykyään kirjastoa ylläpitävät edelleen Mark Otto ja Jacob Thornton, sekä pieni ryhmä kehittäjiä GitHubissa. Bootstrapista on tullut maailman suosituimpia avoimen lähdekoodin kirjastoja. Tällä hetkellä Bootstrap on GitHubissa kuudenneksi parhaiten arvosteltu projekti. (GitHub 2019.)

6.2 Bootstrapin ominaisuuksia

Bootstrap tarjoaa kehittäjälle erilaisia toimintoja ja ominaisuuksia helpottamaan ja yksinkertaistamaan koodaamista. Valmiiden CSS-tyylimallien ja toiminnallisuuksien lisäksi Bootstrap tarjoaa kehittäjälle valmiin tyylittelyn elementeille, käytännössä omaa CSS:ää ei tarvitse kirjoittaa ollenkaan, vaan HTML-koodilla pystyy tekemään valmiita toimivia sivuja.

Erikoismenettelyitä halutessaan omaa CSS:ää voi lisätä Bootstrapin oman CSS:n päälle. Bootstrapista löytyy myös valmiiden elementtien tarjonta, jolla voi helposti tehdä omanlaisen sivun ilman erillismäärittelyä.

6.2.1 CSS-esiprosessorit

Bootstrap on lisännyt CSS-tyylimäärittelyihin esiprosessorit nimeltä Less (Leaner Style Sheets), sekä Sass (Syntatically Awesome Style Sheets). Less:in ja Sass:in avulla CSS-tyylimäärittelyiden kirjoittaminen on yksinkertaisempaa ja tehokkaampaa. Valmis tyylimäärittely Lessillä ja Sassilla on johdonmukaisempaa, ja sen rakenne jäljittelee paremmin HTML-koodin rakennetta.

Less on esiprosessori, joka on Bootstrapissa ladattu CSS:n päälle erillisenä skriptinä. Less:in kehitti Alexis Sellier vuonna 2009. Se oli alun perin kirjoitettu Ruby-koodikielellä, mutta jälkeenpäin siirrettiin käytettäväksi JavaScriptillä. Less sai inspiraationsa Sass:ista, mutta piti sisällään kevyemmän joukon ominaisuuksia, ja sen kielioppisäännöt ovat lähempänä CSS:ää. Less kasvattaa jo käytössä olevia ominaisuuksia ja mahdollistaa muun muassa muuttujien määrittelyt käyttämällä @-merkkiä, luokkamäärittelyiden siirron toisiin luokkiin ja luokkamäärittelyiden moninkertaistamisen (kuvio 6). (Less 2019.) Sass on samantyyppinen esiprosessori kuin Less, mutta siinä on tuttuja elementtejä muista koodikielistä, kuten esimerkiksi jQuerystä tuttu muuttujien määrittelyyn käytetty \$-merkki. (Sass 2019.)

```
1 /* Sass ja Less esimerkki */
2 /* CSS: */
3 #ylapalkki {
4     color: #white;
5 }
6 #ylapalkki .logo {
7     width: 200px;
8     height: 200px;
9 }
10 #ylapalkki .navigation {
11     min-width: 768px;
12 }
13
14 /* Less ja Sass */
15 #ylapalkki {
16     color: #white;
17     .logo {
18         width: 200px;
19         height: 200px;
20     }
21     .navigation {
22         min-width: 768px;
23     }
24 }
```

Kuvio 6: Less ja Sass Nesting-toiminnon esimerkki.

6.2.2 Grid

Normaalisti verkkosivun tai -palvelun kehittämisessä HTML:llä joudutaan määrittämään erikseen erikokoisille näytöille tarkoitetut tyylimäärittelyt CSS:ssä. Tämä hidastaa huomattavasti kehittämisen prosessia ja palveluntarjoajat ovatkin kehittäneet verkkosivujen lisäksi myös erillisen mobiilisovelluksen. Bootstrap sisältää kuitenkin Grid-ruudukkomäärittelyn, joka jakaa verkkosivun elementin kahteentoista eri osaan. Gridin avulla pystyy vaivattomasti luoda sivun, joka skaalautuu täydellisesti mille tahansa laitteelle ilman erillisiä skaalausmäärittelyitä laitekohtaisesti. Grid-elementeille voi määrittää arvoja erillisillä luokkamäärittelyillä. (Bootstrap 2019.)

Gridin toimimiseen määritellään erillinen Container-laatikkoelementti, jonka sisään laitetaan Row-luokkamäärittely. Row määrittää vaakasuuntaisia rivejä Containerin sisään. Row-elementtien sisälle muodostetaan sarakkeita Col-luokkamäärittelyllä. Allaolevassa kuvassa on toteutettu koodinäkökulma etusivun hakutoiminnosta (kuvio 7).

```

38 <div class="jumbotron text-center">
39 <h1 class="display-3">Siirtoajot</h1>
40 <p class="lead">Mihin haluaisit tänään lähteä?</p>
41 <br>
42 <div class="container">
43 <form>
44 <div class="row justify-content-center">
45 <div class="col-4">
46 <input type="text" class="form-control" placeholder="Mistä:">
47 </div>
48 <div class="col-4">
49 <input type="text" class="form-control" placeholder="Mihin:">
50 </div>
51 </form>
52 </div>
53 </div>

```

Kuvio 7: Etusivun HTML-koodiesimerkki Grid-ruudukkomäärittelystä

Sarakkeilla pystytään määrittämään halutun sisällön leveys Containerin sisällä. Col-luokan leveys määritellään numeroin 1-12. Esimerkiksi luokka "col-12", on koko elementin levyinen sarakke. Samaan sarakkeeseen on myös mahdollista luoda useampia sarakkeita, esimerkiksi kolme "col-4"-saraketta, tai kuusi "col-2"-saraketta (kuvio 8). (Bootstrap 2019.)

col	col	col	col
col-8		col-4	

Kuvio 8: Col-elementtien määrittely. (Bootstrap 2019.)

6.2.3 Flex

Flex on elementtien yksinkertaiseen säätelyyn ja paikkamäärittelyyn tehty ominaisuus. Sillä voidaan siirrellä esimerkiksi elementtejä esimerkiksi sivun reunoille tai keskelle ilman, että luodaan työläitä erillismäärittelyjä täsmällisiin paikkoihin sivussa CSS:n avulla (kuvio 9). (Souza, 2018.) Flex täytyy osaan elementeistä käydä määrittämässä päälle Bootstrapin sisäisestä CSS:stä. Flex korvaa Bootstrap 3 versiossa olleen Float-määrittelyn kokonaan. Flex tarjoaa floatia luotettavamman tavan asetella elementtejä paikalleen. Joissain tapauksissa, kun elementti oli liian suuri sivulle, alkoi elementin toimivuus kärsimään. (Souza 2018.)

Flex item 1	Flex item 2	Flex item 3	
			Flex item 3 Flex item 2 Flex item 1

Kuvio 9: Esimerkki Flexin käyttötarkoituksesta (Bootstrap 2019.)

7 Tutkimus- ja kehittämismenetelmät

Opinnäytetyö on toiminnallinen tutkimus. Toiminnallisen tutkimuksen tarkoituksena on tuottaa uutta tietoa jo aikaisemmin tutkitusta asiasta uuden palvelun tai toteutuksen kautta. Toiminnallinen tutkimus jaetaan käytännön osaan sekä teoriaosuuteen. Teoriaosuudessa tehdään tutkimustyötä aiheesta, ja käytännön osassa luodaan uutta tietoa käytännön kokeen avulla. (Candy & Edmonds, 2018.)

Kehittämismenetelmänä työssä käytettiin kokeilevaa kehittämistä. Kokeileva kehittäminen eroaa tavallisesta kehitysprosessista siten, että normaalissa kehitysprosessissa usein palvelun suunnittelu ja kehittäminen tehdään pitkälti valmiiksi, ja tämän jälkeen vasta luovutetaan asiakkaalle tai käyttäjälle testattavaksi. Kokeilevassa kehittämisessä lähdetään hyvin nopeasti tarjoamaan palvelua eteenpäin asiakkaalle tai käyttäjälle, ja tällä pyritään selvittämään, onko tuote kehityskelpoinen. (Työ- ja elinkeinoministeriö 2018.) Kehittämiseen pyrittiin mahdollisimman nopealla aikataululla, sillä ideana palvelu oli täysin uusi, eikä kehittämisen alkaessa ollut vielä toista samanlaista palvelua kehitteillä. Ensimmäinen kehitysversio pyrittiin saamaan mahdollisimman nopeasti toimeksiantajalle valmiiksi, ja sen perusteella lähdettiin jatkokehittämään palvelua eteenpäin.

Validiteetilla tarkoitetaan mittauksen pätevyyttä, eli sen kykyä mitata sitä mitä sen tarkoitus on mitata tarpeellisella tehokkuudella ja kattavuudella. Reliabiliteetti tarkoittaa luotettavuutta ja johdonmukaisuutta, eli sitä että se mittaa kokonaisuudessaan samaa asiaa. (Kvali-MOTV 2008.)

Työssä lähdettiin kehittämään palvelua olemassa olevan teorian, sekä tutkijan ja toimeksiantajan palvelun tavoitteiden perusteella. Palvelun kehitys kuitenkin nojaa pitkälti teoriapohjaan, sillä käyttäjätutkimusta tai haastatteluja ei ajanpuutteen vuoksi saatu toteutettua. Koska aikataulullisesti kehitysversioita piti saada ulos nopeasti, kehittämismenetelmänä kokeileva kehittäminen oli paras mahdollinen vaihtoehto. Tämä mahdollisti käytännössä jatkuvan kehityksen ja yhteydenpidon toimeksiantajaan. Tämän työn kaltaiset palvelun kehittämistyöt ovat usein myös toiminnallisia tutkimuksia, sillä tietoa käytettävyyteen ja Bootstrapiin on paljon saatavilla.

8 Siirtoajot-verkkopalvelu

Tässä kohdassa kuvataan palvelun toteutumisen vaiheita, kerrotaan projektin suunnitteluvaiheesta ja avataan projektin kulkua. Kerrotaan myös, kuinka Bootstrap on vaikuttanut kehittämisen kannalta käytettävän palvelun rakentamiseen, ja lopuksi pohditaan lopputuloksia ja kerrotaan kehitysehdotuksia.

8.1 Projektin eteneminen

Projekti aloitettiin suunnitelman tekemisellä keväällä 2019. Palvelun tarpeet käytiin läpi, ja päädyttiin käyttäjäryhmän laajuuden perusteella tekemään mahdollisimman yksinkertaisesti toimivan palvelun käyttäjille.

Toteutus aloitettiin maaliskuussa, ja ensimmäinen alustavan versio tehtiin valmiiksi. Päätettiin myös tehdä yrityksille erillisen hallintasivu, josta he pääsevät katselemaan omia ajojaan, perumaan niitä tarvittaessa ja katsomaan tarvittavat tiedot kuljetuksesta. Käyttäjille tehtiin erillinen profiilisivu, josta näkee mihin on ilmoittautunut ajamaan, jos on ajon varannut itselleen.

Päätettiin, että etusivun keskiössä tulee olemaan hakutoiminto, prioriteettina yksinkertaiset hakukentät, joissa sisällä vihjetekstit Mistä ja Minne. Lisättiin myös kalenterinäkömä, josta voi lisätä hakea päivämäärän lähtö- ja paluupäivälle. Kalenteriin emme nähneet tarvetta asettaa suoranaista kellonaikaa lähdöille, vaan näimme parhaaksi työssä havaitsemiemme käyttäjäkokemusten perusteella, että on parempi asettaa pelkkä lähtöpäivä, ja sopia erikseen käyttäjän kanssa puhelimitse tai sähköpostilla tarkemmat ajankohdat (kuvio 10).

Kuvio 10: Etusivun hakutoiminto

Tulevaisuutta varten mietittiin myös mahdollisuutta Chat-palvelulle, jossa käyttäjä ja yrityksen edustaja voisivat käydä keskustelua palvelun kautta. Projektin taustalla tehtiin myös tutkimustyötä mahdollisia käyttäjiä ja yrityksiä varten. Kirjoitimme palvelulle käyttäjäehdot,

kartoitimme mahdollisia asiakasyrityksiä ja teimme selvitystä, kuinka myymme palvelua, jos mahdollisia kilpailijoita tulee.

Teimme määrytykset myös tarkempaa hakua varten, ja kartoitimme infokenttien sisällön, joita haluamme näkymään yleiselle hakutulosten sivulle. Ensimmäiseen palkkiin laitettiin ajoa tarjoavan yrityksen logo. Jos logoa ei ollut saatavilla, niin toisena määrittelynä siihen tuli yrityksen nimi. Toiseen ja kolmanteen palkkiin tuli lähtöpaikka ja kohde. Neljänteen päivämäärä milloin auton voi ajaa kyseisen välin ja viidenteen kohtaan lisätietoja, josta voi klikata itsensä eteenpäin. Lisätietoja kohdan takaa löytyy mm. tiedot autosta, yhteyshenkilö joka autoa tarjoaa (esim. automyyjä), ja polttoainekorvausmenettelyt (kuvio 11). Koska kyseessä on palvelun visuaalinen käyttöliittymä, toteutin valmiit hakumallit, jotta sivustoa voi esitellä.

Hakutulokset

Yritys	Lähtöpaikka	Kohde	Lisätietoja
Yrityksen logo	Rovaniemi	Iisalmi	Lisätietoja ajosta
Yrityksen logo	Kotka	Tampere	Lisätietoja ajosta
Yrityksen logo	Helsinki	Kuopio	Lisätietoja ajosta

Kuvio 11: Hakutulokset-sivu

8.2 Bootstrap käyttäjäkeskeisen kehittämisen tukena

Bootstrap tarjoaa Grid-ominaisuutensa vuoksi erittäin helppokäyttöisen käyttöliittymämallin. Elementtien muokkaaminen onnistuu helposti, ja kaiken saa todella näppärästi juuri halutulle paikalle. Gridin ansiosta myös mobiilikehittäminen on yksinkertaista, sillä palvelu skaalautuu normaalin näköiseksi myös mobiilissa, jos kolumnien määrittelyt on tehty oikein. Palvelusamme Bootstrap piilottaa esimerkiksi mobiiliversiossa yläpalkin niin, että se luo pudotusvalikon itsestään (kuvio 12).

Kuvio 12: Mobiilisivuston esimerkki

Hakutulokset-sivu oli erittäin yksinkertaista luoda Bootstrapilla, sillä Bootstrapin taulukkotoinnolla saa luotua helposti juuri halutunlaisen listaussivun. Bootstrapilla saa myös määritellä tiettyjen objektien poistumista, kun näyttökoot vaihtuvat. Jos olisimme halunneet taustakuvan etusivulle, niin sen olisi saanut määriteltä pois mobiililaitteille, joka vähentää huomattavasti sivun lataamisaikaa, sekä yksinkertaistaa mobiilille tarkoitettua versiota palvelusta.

Käytettävyyden teorian pohjalta päädyttiin värimaailmaltaan vaaleaan sävyyn, sivun perusväri on valkoinen, ja tausta ilman taustakuvaa harmaa. Fontin väri pidettiin mustana ja fontti on selkeästi luettava. Navigaationappuloita on väritetty, jotta käyttäjä hahmottaisi painikkeen, jolla pääsee palvelussa eteenpäin. Haku on myös itsessään yksinkertainen, käyttäjän tarvitsee syöttää vain päivämäärä, lähtö- ja tulopaikka, ja painaa Matkaan! -nappulaa, ja näin hän pääsee jo tarkastelemaan vaihtoehtoja ajettaville matkoille. Yläpalkki on myös pidetty valkoisena ja siihen on laitettu yksinkertainen navigaatio sisältäen vain yksinkertaisia linkkejä ilman pudotusvalikoita. Tällä minimoidaan sivustolla eksymisen riskit ja tehdään navigaatiosta mahdollisimman yksinkertaista.

Jos on aikaisempaa kokemusta HTML:llä tehdyistä verkkosivuista, ei Bootstrapilla verkkopalvelun luominen ole haastavaa. Bootstrapista on olemassa tietokirjallisuutta ja internet on hyvin laaja tietopankki Bootstrapista kiinnostuneille. GitHubissa toimii myös yhteisö, joka kehittää uusia ominaisuuksia käyttäjien toiveista ja vastaa kysymyksiin kehittämisestä.

9 Yhteenveto, johtopäätökset ja kehitysehdotukset

Opinnäytetyössäni toteutettiin autojen siirtoihin tarkoitetun verkkopalvelun käyttöölyttäminen. Tarkoituksena oli toteuttaa palvelun käyttäjälle näkyvä osa, joka on käytettävyydeltään soveltuva laajalle käyttäjäkunnalle ja erilaisille laitteille. Tässä työssä tuotettiin uutta tietoa luodun sovelluksen pohjalta niin itselleni, tuleville käyttäjille ja entiselle työkaverilleni, joka oli toimeksiantajana projektille. Uusi tieto pohjautuu käytettävyyden teoriapohjalle ja tuo tietoa siitä, miten käytettävyyden teoria toimii käytännössä Bootstrapilla luodun verkkosovelluksen kanssa, joka keskittyy yksityisautojen siirtoon.

Palvelulle olisi ollut hyvä toteuttaa erillistä käyttäjätutkimusta, mutta aika loppui opinnäytetyön kehittämisen kannalta kesken. Käyttäjien mielipiteet sovelluksesta ovat ratkaisevia lopputuotteen kannalta, ja olisimme myös tarvinneet kohdeyrityksen edustajia kertomaan mielipiteensä palvelun yrityksille suunnatusta puolesta. Vertailu myöskin palvelun sekä Facebook-ryhmän toimivuuden välillä olisi ollut hyvä, sillä meillä ei juurikaan potentiaalisilta käyttäjiltä ollut tietoa olisiko tämäntyyppistä palvelua oikeasti tarvittu.

Chat-palvelun tuominen jatkokehitysideana olisi mahdollinen kilpailuetu. Facebookista voi lähettää suoraan ilmoituksen tarjoajalle Facebook-viestiä, mutta yleisesti kommunikaatio tapahtuu puhelimitse. Jos verkkopalvelussa olisi Chat-palvelu (josta esimerkiksi ajon tarjoaja voisi saada ilmoituksen uudesta viestistä sähköpostitse), olisi toiminnallisuus jo huomattavasti parempi. Tämä poistaisi myöskin kynnyksen soittaa puhelimitse ajon tarjoajalle.

Palvelu olisi myöskin ehdottomasti pitänyt saada palvelimelle ja verkkoon. Tämän avulla olisimme voineet testata reaaliajassa, kuinka nopeasti Bootstrapilla luotu front-end puoli sivustosta latautuu. Ylimääräisten back-end puolen ominaisuuksien lataaminen vaatii enemmän aikaa ja vaikuttaa käyttäjäkokemukseen, mutta pelkästään front-endin latausajat minimoimalla olisimme voineet vaikuttaa ainakin sen puolen toimivuuteen ja käytettävyyteen.

Käyttäjätutkimuksen kannalta myös Bootstrapin testaaminen olisi ollut hyvä idea, sillä Bootstrapin perustajan Mark Otton kommentin mukaan heidän tiimissään on töissä ihmisiä, jolla on vuosien kokemus käyttäjäkeskeisten sivujen luonnista, mutta he eivät virallisesti testaa Bootstrapia. Bootstrapin kehittäminen käyttäjien suuntaan tapahtuu yhteisön pyytämien ominaisuuksien perusteella.

Kokonaisuutena opinnäytetyö on ollut minulle erittäin opettavainen. En ole aikaisemmin juurikaan käyttänyt Bootstrapia. Bootstrapin oppiminen ja sillä toimiminen on antanut minulle todella positiivisen kuvan kyseisestä komponenttikirjastosta. Yllätyin siitä, kuinka nopeasti Bootstrapilla on mahdollista ylipäättään tehdä järkeviä yksinkertaisia sivuja ilman erillisiä tyylimäärittelyitä. Loppujen lopuksi jouduin sivulle muokkaamaan vain muutaman värin, jolla saimme hieman eloa muuten harmaaseen sivustoon. Kaikki dokumentaatio Bootstrapista oli

helposti saatavilla heidän omilta verkkosivuiltaan sekä w3schoolsista. Kun kehittäessä tuli ongelmia, niin löysin kaiken tarvittavan etsimällä Googlesta. Useat käyttäjät olivat kyselleet samoja asioita kuin minä, joten ratkaisuja oli tarjolla useita.

Käyttäjäkunta tulisi koostumaan ajokortti-ikäisistä käyttäjistä. Bootstrapin Grid-määrittelystä automaattisesti muodostuvan responsiivisuuden avulla voidaan vaivatta tehdä mobiililaitteillekin sopiva palvelu, eikä välttämättä edes tarvita erillistä sovellusta palvelun käyttämiseen. Opinnäytetyön tavoite on saavutettu, käytettävyyden näkökulmasta projekti oli oikein onnistunut, lopullinen tuotos on yksinkertainen verkkopalvelu, joka kuvaa myöskin hyvin aikaisemmin mainittujen käytettävyyden teoriakohtien mukaisia odotteita. Palvelussa ei ole juurikaan muistettavaa palvelun käyttäjän puolella. Yritysosiossa on vielä hieman sekava käyttöliittymä, mutta sen korjaisi nopeasti, kun saisi karsittua hieman tarpeettomia tietoja pois. Palvelun jatkokehittäminen on ainakin toistaiseksi päättynyt. Back-end toteuttajaa ei ollut opinnäytetyön tekohetkellä, ja palvelun kehittämisen aikana toinen yritys on perustanut samantyyllisen verkkopalvelun.

Lähteet

Painetut

Kraft, C. User Experience Innovation: User Centered Design that Works. New York: Apress.

Lowdermilk, T. 2013. User-centered design. Boston: O'Reilly.

Maioli, L. 2018. Fixing Bad UX Designs: Master proven approaches, tools and techniques to make your user experience great again. Birmingham: Packt Publishing, Ltd.

Nielsen, J. 1993. Usability Engineering. Cambridge: Academic Press, Ltd.

Väyrynen, S., Nevala N. & Päivinen N. 2004. Ergonomia ja käytettävyys suunnittelussa. Helsinki: Teknologiateollisuus ry.

Sähköiset

Alexa, 2020. The top 500 sites on the web. Viitattu 24.04.2020

<https://www.alexacom/topsites>

Baymard Instute. 2019. 41 Cart Abandonment Rate Statistics. Viitattu 10.12.2019

<https://baymard.com/lists/cart-abandonment-rate>

Bootstrap. 2019. About. Viitattu 15.11.2019.

<https://getbootstrap.com/docs/4.3/about/overview/>

Bootstrap. 2019. Grid system. Viitattu 15.11.2019.

<https://getbootstrap.com/docs/4.0/layout/grid/>

Bootstrap. 2019. Bootstrap Themes. Viitattu 10.12.2019

<https://themes.getbootstrap.com/>

Candy, L. & Edmonds, E. 2018. Practice-Based Research in Creative Arts. Viitattu 10.12.2019

https://www.mitpressjournals.org/doi/pdf/10.1162/LEON_a_01471

Github, 2019. Usability Testing? #27979. Viitattu 15.11.2019.

<https://github.com/twbs/bootstrap/issues/27979>

GitHub, 2019. Parhaiten arvostellut projektit GitHubissa. Viitattu 15.11.2019

<https://github.com/search?q=stars:%3E1&ref=searchresults&type=Repositories&utf8=%E2%9C%93>

Google. 2019. Be mobile-friendly. Viitattu 9.12.2019

<https://developers.google.com/search/mobile-sites/>

Interaction Design Foundation. 2019. User Experience (UX) Design. Viitattu 09.12.2019

<https://www.interaction-design.org/literature/topics/ux-design>

International Organization for Standardization. 2019. ISO 9241-210:2019(en) Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. Viitattu 19.11.2019.

<https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-2:v1:en>

International Organization for Standardization. 1999. ISO 13407:1999(en) Human-centred design processes for interactive systems. Viitattu 19.11.2019.

<https://www.iso.org/obp/ui/#iso:std:iso:13407:ed-1:v1:en>

KvantiMOTV, 2008. Mittaaminen: Mittarin luotettavuus. Viitattu 23.04.2020

<https://www.fsd.tuni.fi/menetelmaopetus/mittaaminen/luotettavuus.html>

Less. 2019. History. Viitattu 15.11.2019.

<http://lesscss.org/about/>

Otto, M. 2011. Bootstrap from Twitter. Viitattu 15.11.2019.

https://blog.twitter.com/developer/en_us/a/2011/bootstrap-twitter.html

Sauro, J. 2013. A brief history of usability. Viitattu 19.11.2019.

<https://measuringu.com/usability-history/>

w3schools. 2019. Viitattu 15.11.2019.

<https://getbootstrap.com/docs/4.0/layout/grid/>

Sass. 2019. Sass Basics. Viitattu 16.11.2019

<https://sass-lang.com/guide>

Souza, D. 2018. Progressively Enhanced CSS Layouts: Floats to Flexbox & Grid. Viitattu 10.12.2019.

<https://www.sitepoint.com/css-layouts-floats-flexbox-grid/>

Työ- ja elinkeinoministeriö, 2015. Kokeileva kehittäminen. Viitattu 23.04.2020

https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/74944/TEM-rap_67_2015_web.pdf?sequence=1

Usability.gov. 2019. User Interface Design Basics. Viitattu 9.12.2019

<https://www.usability.gov/what-and-why/user-interface-design.html>

W3C. 2019. Introduction to Understanding WCAG 2.0. Viitattu 10.12.2019

<https://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html#introduction-fourprincs-head>

Kuviot

Kuvio 1: Käyttäjäkokemuksen kaari. (Kraft 2012, 2.)	11
Kuvio 2: Syitä ostopäätöksen hylkäämiseksi (Baymard 2019.)	12
Kuvio 3: Neljä suositusta verkkosisällön saavutettavuuteen (W3C 2019.)	14
Kuvio 4: Googlen etusivu (Google 2020.)	15
Kuvio 5: eBayn etusivu (eBay 2020.)	16
Kuvio 6: Less ja Sass Nesting-toiminnon esimerkki.	18
Kuvio 7: Etusivun HTML-koodiesimerkki Grid-ruudukkomäärittelystä	19
Kuvio 8: Col-elementtien määrittely. (Bootstrap 2019.).....	19
Kuvio 9: Esimerkki Flexin käyttötarkoituksesta (Bootstrap 2019.)	19
Kuvio 10: Etusivun hakutoiminto	21
Kuvio 11: Hakutulokset-sivu.....	22
Kuvio 12: Mobiilisivuston esimerkki.....	23