

Junior web-kehittäjänä Suomen Vahinkovakuutus Oy:ssä

Joa Foudila

Tekijä Joa Foudila	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Junior web-kehittäjänä Suomen Vahinkovakuutus Oy:ssä	Sivu- ja liitesivumäärä 86 + 2
Opinnäytetyön otsikko englanniksi Working as a Junior Web-Developer at Finnish P&C Insurance	
<p>Tämä Haaga-Helia ammattikorkeakoulun portfoliomainen päiväkirjaopinnäytetyö käsittelee opiskelijan työskentelyä ja ammatillista kehitystä Junior Web Developerin eli nuoremman ohjelmistokehittäjän työtehtävissä Suomen Vahinkovakuutus Oy:ssä. Opinnäytetyössä kuvataan ja analysoidaan opiskelijan työntekoa ja työtehtäviä 14.1. - 22.3.2019 välisenä aikana. Päiväkirjamuotoisen opinnäytetyön tavoitteena on opiskelijan työssä vaadittavien taitojen parantaminen ja ammatillinen kasvu.</p> <p>Opinnäytetyö koostuu johdannosta, lähtötilanteen kuvauksesta, kymmenestä seurantaviikosta, sekä pohdinnasta ja päätelmästä. Johdannossa kuvataan opiskelijan ohjelmistokehittäjän työtehtäviä, sekä käytettyjä työmenetelmiä. Johdantoa seuraa lähtötilanteen kuvaus, joka sisältää opiskelijan nykyisen työn analysointia. Päiväkirjaraportointia on kirjoitettu kymmenen seurantaviikon aikana, jolloin opiskelija on kirjannut ja kuvaillut viikon aikana työstämäänsä työtehtäviä. Jokaisesta seurantaviikosta on viikkoanalyysi, jossa tutkitaan viikon aikana kohdattuja ongelmia, niiden ratkaisumalleja, ja analysoidaan, mitä uutta opiskelija on oppinut.</p> <p>Opiskelijan työtehtäviin kuuluvat ohjelmistokehitystehtävät, mihin kuuluvat vakuutusyhtiön verkkopalveluiden kehittämiseen ja ylläpitoon liittyviä työtehtäviä, kuten uusien ominaisuuksien luominen, sisällön muokkaaminen, ja virheiden korjaaminen.</p> <p>Opinnäytetyön lopussa ovat pohdinta ja päätelmät, jossa opiskelijan kehitystä verrataan lähtötilanteen kuvaukseen. Päätelmissä käydään läpi viikkoanalyysien sisältöä peilaten havaintoja opiskelijan nykytilanteeseen kymmenen seurantaviikon jälkeen. Huomattavin kehitys opiskelijan taidoissa on tapahtunut JavaScript-ohjelmoinnissa, ja ammatillisen kasvun myötä opiskelijan itsevarmuus ja luottamus omiin taitoihin kasvoivat.</p>	
Asiasanat Ohjelmistokehitys, front end, JavaScript, HTML, CSS	

Sisällys

1	Johdanto	1
1.1	Keskeiset käsitteet.....	4
2	Lähtötilanteen kuvaus	5
2.1	Oman nykyisen työn analyysi	5
2.2	Sidosryhmät työpaikalla.....	7
2.3	Vuorovaikutustaidot työpaikalla	8
3	Päiväkirjaraportointi	10
3.1	Seurantaviikko 1	10
3.2	Seurantaviikko 2	16
3.3	Seurantaviikko 3	20
3.4	Seurantaviikko 4	24
3.5	Seurantaviikko 5	29
3.6	Seurantaviikko 6	37
3.7	Seurantaviikko 7	43
3.8	Seurantaviikko 8	50
3.9	Seurantaviikko 9	56
3.10	Seurantaviikko 10	68
4	Pohdinta ja päätelmät	77
	Lähteet.....	81
	Liitteet	87
	Liite 1. Lyhenteet.....	87
	Liite 2. Muut käsitteet sekä käytetyt työkalut ja ohjelmistot.....	88

1 Johdanto

Opinnäytetyö on Haaga-Helian päiväkirjamuotoinen opinnäytetyö, jossa kirjoitan työskentelystäni junior web-kehittäjänä Suomen Vahinkovakuutus Oy:ssä. Päiväkirjamuotoinen opinnäytetyö koostuu oman lähtötilanteen kuvauksesta, kymmenestä seurantaviikosta ja niiden viikkoanalyyseistä, sekä päätelmästä, jossa verrataan opittua lähtötilanteeseen. Opinnäytetyön päiväkirjamerkinnät sijoittuvat 14.1. - 22.3.2019 väliselle ajalle. Työn tarkoituksena on seurata ja kehittää työssä vaadittavia ammatillisia taitoja.

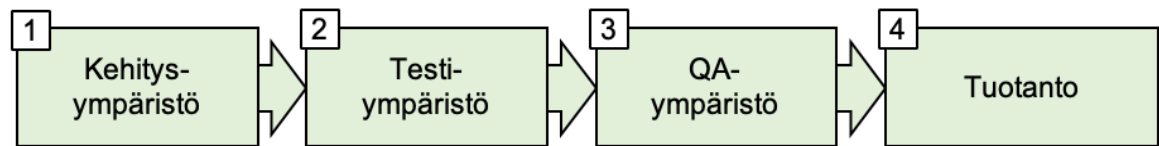
Työskentelen Suomen Vahinkovakuutus Oy:ssä ohjelmistokehittäjänä. Suomen Vahinkovakuutus Oy tarjoaa POP Pankki -ryhmän sekä Säästöpankkiryhmän vakuutuksia, keskittyen henkilöasiakkaiden vahinkovakuutuksiin ja niiden digitaaliseen jakeluun ja markkinointiin (Suomen Vahinkovakuutus 2019). Toimin yrityksen ICT-osastolla yhtenä yrityksemme kolmesta ohjelmistokehittäjästä. Yrityksessämme työskentelykieli on suomi. POP Vakuutuksen palveluita tarjotaan vain suomenkielellä, ja Säästöpankin vakuutusten palveluita tarjotaan suomen lisäksi myös ruotsiksi.

Ohjelmistokehittäjänä työtehtäväni ovat vakuutusyhtiön verkkopalveluiden kehittäminen ja ylläpitotehtävät. Työtehtävien luontiin, ohjaukseen ja seurantaan käytämme apuna Jira- ja HelpDesk-tikettijärjestelmiä. Jiraan luodut tiketit ovat POP-Vakuutuksen tai Säästöpankin vakuutusten verkkosivujen tai vakuutusten myyntiin käytettävän myyntisovelluksen liittyviä kehitystehtäviä. HelpDesk on tarkoitettu pääosin yrityksemme työntekijöiden kohtaamien teknisten ongelmien kirjaamiseen ja ratkaisemiseen. Työssäni vaaditaan ohjelmoinnin ja ohjelmistokehityksen taitoja, loogista ajattelua ja ongelmanratkaisun taitoja, sekä hyviä viestinnäntaitoja. Työtehtäväni ovat valtaosin Jira-tikettijärjestelmään luotujen kehitys-, ylläpito-, ja korjaustehtävien ratkaisemista.

ICT-osastolla työskentelemme Scrum viitekehyksessä käytettävien sprint-jaksojen kaltaisissa kahden viikon kehitysjaksoissa. Emme kuitenkaan noudata puhdasta Scrum-kehitysmallia, vaan olemme ottaneet käyttöön parhaaksi näkemämme osat. Työskentelymme tapahtuu kahden viikon jaksoissa eli sprinteissä, mutta meillä on käynnissä samanaikaisesti kaksi eri sprinttiä, mikä ei ole Scrumin mukaista. Sprintit ovat nimetty sprintin aloitusviikon viikkonumeron mukaan.

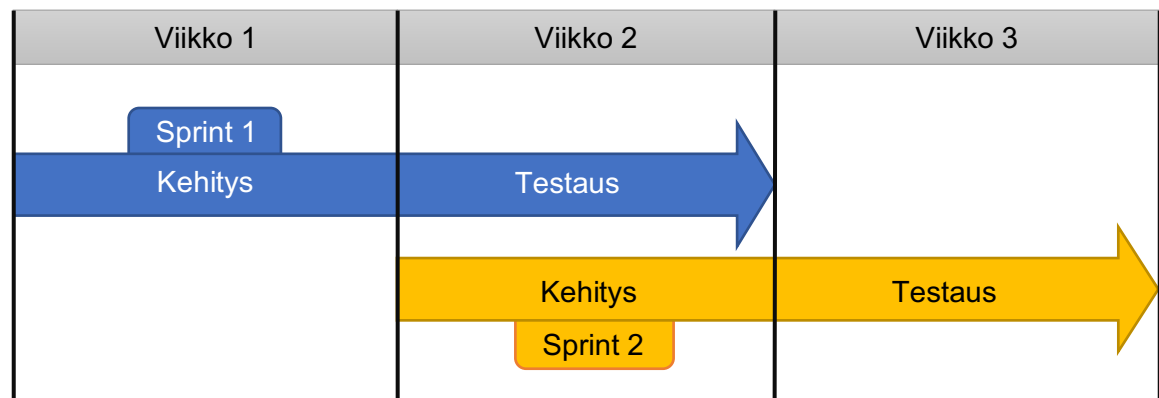
Kehitystöiden eteneminen on nelivaiheinen (kuvio 1). Ensimmäisessä vaiheessa tiketit työstetään omassa kehitysympäristössä omalla koneella eli lokaalisti. Valmiit muutokset viedään niiden valmistuttua Git-versionhallinnan testiympäristöön eli develop-haaraan. Toisessa vaiheessa muutokset testataan develop-haarassa. Kolmannessa vaiheessa

muutokset viedään testihaarasta QA-, eli Quality Assurance -haaraan, jossa tarkistetaan, että kaikki testihaarasta QA:lle tuodut muutokset varmasti toimivat kuten niiden pitäisi. Neljännessä vaiheessa muutokset siirretään production-haaraan, eli tuotantoympäristöön verkkosivuillemme sekä myyntisovellukseen.



Kuvio 1. Muutosten eteneminen vaiheittain kehitys- ja testausympäristöistä tuotantoon

Kehitystyö tapahtuu kaksi viikkoa kestävänsä sprint-jakson aikana, jotka ovat epätyypillisesti osittain päällekkäin. Keskiviikkoisin pidämme sprint-palaverin, jossa valitsemme seuraavan sprintin aikana työnettävät tiketit. Uusi sprint käynnistetään välittömästi sprint-palaverin jälkeen. Ensimmäisen sprintin ensimmäinen viikko on kehitysviikko, joka kestää seuraavaan viikon tiistaihin. Kehitysviikon aikana sprinttiin otetut tiketit tulee saada ratkaisuksi ja vietyä testiympäristöön eli develop-haaraan. Keskiviikkoaamuna develop-haaran muutokset viedään QA:lle, ja ensimmäisen sprintin testausviikko alkaa (kuvio 2).



Kuvio 2. Kahden päällekkäisen sprintin malli

Keskiviikkona pidetään taas uusi sprint-palaveri, jossa käymme läpi edellisen sprintin aikaansaannokset, minkä jälkeen käynnistämme toisen sprintin ensimmäisen rinnalle. Ohjelmistokehittäjät siirtyvät työskentelemään uuden sprintin kehitystehtäviä, samalla kun ensimmäisen sprintin muutokset ovat testattavana.

Ohjelmistokehittäjän työssä tarvitaan tietoutta verkkosivujen ja verkkopalveluiden ohjelmonnoista, ja työni keskeinen osa on verkkosivujen ulkoasujen ja toiminnollisuuksien ohjelmointityöt. Internetiä käytetään yhä enemmän älypuhelimilla, tableteilla ja muilla älylaitteilla. Mobiililaitteiden ruutukoot ovat perinteistä työpöytäkoneen monitorin ruutua

pienempiä, minkä vuoksi ohjelmistokehittäjän työssä on olennaisen tärkeää ymmärtää mobiilikäyttäjien tarpeita. Työni tueksi valitsin Matthew Carverin vuonna 2015 julkaistun kirjan *The Responsive Web*, joka käsittelee verkkosivujen responsiivista kehitystä, eli sivuston sisällön ja ulkoasun luomista skaalautuvaksi jokaiselle näytön koolle.

Carver (2015) kertoo kirjan tarkoituksena olevan opettaa sekä suunnittelijoita että ohjelmistokehittäjiä paremmin ymmärtämään responsiivista kehitystyötä, sekä tarjota työvälineet ja ohjeet responsiivisen kehitystyön sisäistämiseksi. Kirjassa käsitellään responsiivisen verkkokehityksen keskeisiä konsepteja ja teknologioita sekä hyviä käytänteitä, ja opetetaan, miten sivustot suunnitellaan, sekä luodaan alusta alkaen responsiivisiksi. Teoksessa opastetaan esimerkein ja kuvin, kuinka sivuille kannattaa suunnitella, ja on mahdollista luoda, esittää, piilottaa, muokata, skaalautuvia elementtejä.

Työssä vaaditaan tietoa ja taitoa JavaScript-ohjelmointikielestä, sillä suurin osa koodista on kirjoitettu JavaScriptillä tai siihen liittyvällä React.js -laajennuskirjastolla. Tietoperustaa kartuttaakseni valitsin Douglas Crockfordin kirjan *JavaScript: The Good Parts* vuodelta 2008. Kirja käsittelee JavaScript-ohjelmointia, ja se on tarkoitettu aloittelevalle ohjelmointijalle. Kirjassaan Crockford (2008) tuo esiin JavaScriptin hyvät puolet, ja kuinka käyttää JavaScriptiä oikein. Kirjassa käydään läpi, kuinka käyttää oikeaoppisesti JavaScriptin ominaisuuksia, objekteja, periytyvyyttä, sekä oikeaa kieliasua. Kirjassa kerrotaan myös JavaScriptin huonoista puolista ja ominaisuuksista, ja miten virheiltä olisi helpompi välttyä. JavaScriptin osaaminen on tärkeää, sillä se on yksi maailman käytetyimmistä ohjelmointikielistä (Crockford 2008). Uskon kirjan auttavan JavaScriptin paremmassa hallitsemisessa ja tietoperustani laajentamisessa.

1.1 Keskeiset käsitteet

Contentful on sisällönhallintaan käytettävä alusta, jolla on mahdollista luoda, hallita ja jakaa sisältöä millä tahansa alustalla (Contentful 2019). Monet verkkosivustamme rakentuvat Contentfuliin luodusta sisällöstä.

Git-versionhallintaa käytämme kehitystöiden muutosten hallinnointiin, siirtämiseen ja seurantaan.

HelpDesk on virheiden raportointiin käytettävä tikettijärjestelmä.

JavaScript on ohjelmointikieli, joka mahdollistaa muun muassa verkkosivujen ulkoasun muuttamisen, tiedon siirtämisen ja päivittämisen sivuston käytön aikana (MDN web docs 2019a).

Jenkins on automaatiopalvelin koodien rakentamiseen.

Jira on muun muassa projektinhallinnassa käytettävä työkalu, jolla voidaan helposti hallita ja seurata tehtävien etenemistä (Guru99 2019). Yrityksessämme hyödynnämme Jiraa kehitystehtävien tikettien luomiseen ja seurantaan.

Kommitointi (engl. commit) on Git-versionhallinnassa muutosten tallentamiseen käytettävä komento.

Murupolkunavigaatio (engl. breadcrumb navigation) on toissijainen navigaatio, jossa esitetään minkä sivujen alla kulloinenkin selattava sivu on, mikä auttaa havainnollistamaan sivuston rakennetta käyttäjälle (Gube 2009).

Sprint tarkoittaa kahden viikon kehitysjaksoa, jonka aikana työstettäväksi otettujen tiketien tehtävät tehdään ja testataan lopulta tuotantoon vietäväksi.

Prove on ulkoinen testipalvelu, jolle on mahdollista ohjata Jira-tikettejä testattavaksi.

Tiketti on Jira- tai HelpDesk-järjestelmiin luotu kuvaus ongelmasta tai kehitystarpeesta.

2 Lähtötilanteen kuvaus

2.1 Oman nykyisen työn analyysi

Työtehtäväni Suomenvahinkovakuutus Oy:ssä ovat POP Vakuutus ja Säästöpankin vakuutukset -brändien verkkosivujen sekä myyntisovellusten ohjelmistokehitystehtäviä. Työtehtäväni ovat

- uusien ominaisuuksien kehittäminen ja luominen
- olemassa olevien toimintojen ja ominaisuuksien ylläpito
- esiin nousseiden ongelmien ja virheiden korjaaminen.

Työtehtäväni ovat pääasiallisesti front-end, eli käyttöliittymän kehitystehtäviä. Kehitystehtävät ovat uusien sivujen ja ominaisuuksien luomista, esimerkiksi uuden elementin ja sen toiminnollisuuksien luomista. Ylläpittotehtäviä ovat esimerkiksi tekstien muokkaukset, kuvien muuttamiset tai poistamiset, sekä värien muutokset verkkosivuilla. Virhetilanteiden ratkaisu on esimerkiksi uusissa ominaisuuksissa ilmenneiden virheiden korjausta, tai uusien ominaisuuksien välillisesti aiheutuneiden virheiden tai ongelmien ratkaisua.

Työpaikallamme on käytössä tikettijärjestelmät Jira ja HelpDesk. Työni on näiden järjestelmien kautta tulleiden ylläpito-, korjaus, ja kehittämistehtävien toteuttamista. Tikettijärjestelmään tulleita tikettejä ohjataan eri kehittäjille tai testaajille. Tiketit voivat olla parannuksia nykyisiin ominaisuuksiin, kuten muutoksia verkkosivujen ulkonäköön. Tiketti voivat liittyä virheiden eli bugien korjauksiin, tai ne voivat olla uusien ominaisuuksien luomista. Muutokset toteutetaan joko POP Vakuutuksen tai Säästöpankin vakuutusten asiakkaalle näkyville verkkosivuille, tai vakuutuksien myyntiin ja hallinnointiin käytettävään myyntisovellukseen.

Työtehtävissä tarvitaan taitoja eri ohjelmointikielissä. Keskeisimpiä vaadittavia taitoja ovat HTML- ja CSS-merkitäkielien, sekä JavaScript-ohjelmointikielen hyvä hallinta sekä perusosaaminen olio-ohjelmoinnista. Sivustomme eri sivut rakentuvat hieman eri tavoin; osa sivuista on tehty PHP:llä, osa JavaScriptillä, osa on tehty käyttäen JavaScriptin React.js-laajennuskirjastoa. Ohjelmistokehityksessä ja ohjelmoinnissa olennaista on omata hyvät ongelmanratkaisutaidot, ja kykyä etsiä tietoa. Työssäni on myös tärkeää omata hyvät kommunikointitaidot, sillä työskentelemme avokonttorissa, ja valtaosa yrityksemme työntekijöistä työskentelee samassa toimistossa, minkä vuoksi sosiaalisia kanssakäymisiä on päivän aikana paljon.

Olen vapaa-ajallani opiskellut ja työtä tehdessä oppinut PHP-kielellä ohjelmointia, josta minulla ei ollut aiempaa kokemusta. Tämän lisäksi olen kehittynyt työtä tehdessäni

JavaScript-koodin luomisessa, ja oppinut ohjelmoimaan paremmin React.js:llä, sekä luomaan sivuja Handlebars.js -laajennuskirjastoa käyttäen. Edellä mainittujen ohjelmointikielten hallinta on olennainen osa, ja edellytys selviytyäkseni työn tekemisessä.

Kulloisenkin työtehtäväni vaatimukset ovat selitettynä Jira-tiketissä. Tiketeissä kerrotaan, mitä tulee lisätä, poistaa, muokata tai korjata. Tiketin luoja ei välttämättä ole aina antanut tarpeeksi informaatiota tehtävän ratkaisemiseksi, missä tapauksessa on kysyttävä lisätietoja tiketin luojalta, jotta tehtävä tulee tehdyksi oikein ja lopputulos on halutun mukainen. Oikeiden kysymysten esittäminen ja hyvät kommunikointitaidot ovat olennainen osa ohjelmistokehitystä.

Lähtötilanteessa arvioin osaamiseni tason aloittelevaksi toimijaksi. Onnistun itsenäisesti luomaan toimivaa koodia, ja ratkaisemaan ongelmia, mutta joudun turvautumaan ohjelmoijakollegoihini varmistaakseni koodin toimivuuden ja laadun. Kohtaan myös usein ongelmia, joita en osaa itse ratkaista. Aloitin työni yrityksessä työharjoittelulla Junior Web Developerina ilman merkittävää aikaisempaa työelämän kokemusta ohjelmistokehityksestä ja ohjelmoinnista. Minuun kohdistuvat odotukset Junior Developerina oli suhteutettu hyvin osaamiseni tasoon. Minulle ei ole asetettu liian korkeita tavoitteita, vaan merkittävimpänä tavoitteenani on oppia ja kehittyä työtä tekemällä. Arvioisin suoriutuvani tehtävistäni kohtalaisen hyvin.

Ammatillinen kehittyminen on vielä aloittelijan tasolla. Aloitin työni Suomen Vahinkovakuutusyhtiössä toukokuussa 2018, ja aikaisempi työkokemukseni ei ole ollut ICT-alalla tai tietotekniikan parissa. Päiväkirjamuotoisen opinnäytetyön aloittaessani olin mielestäni saanut vasta alustavan, perustason ymmärryksen yrityksemme monimutkaisten järjestelmien toiminnasta, ja minulla oli vielä paljon opittavaa. Verkkosivujemme ja myyntisovelluksemme toiminnollisuudet rakentuvat sadoista eri lähdekoodin tiedostoista, joten oikean tiedoston löytäminen on minulle joskus vaikeaa.

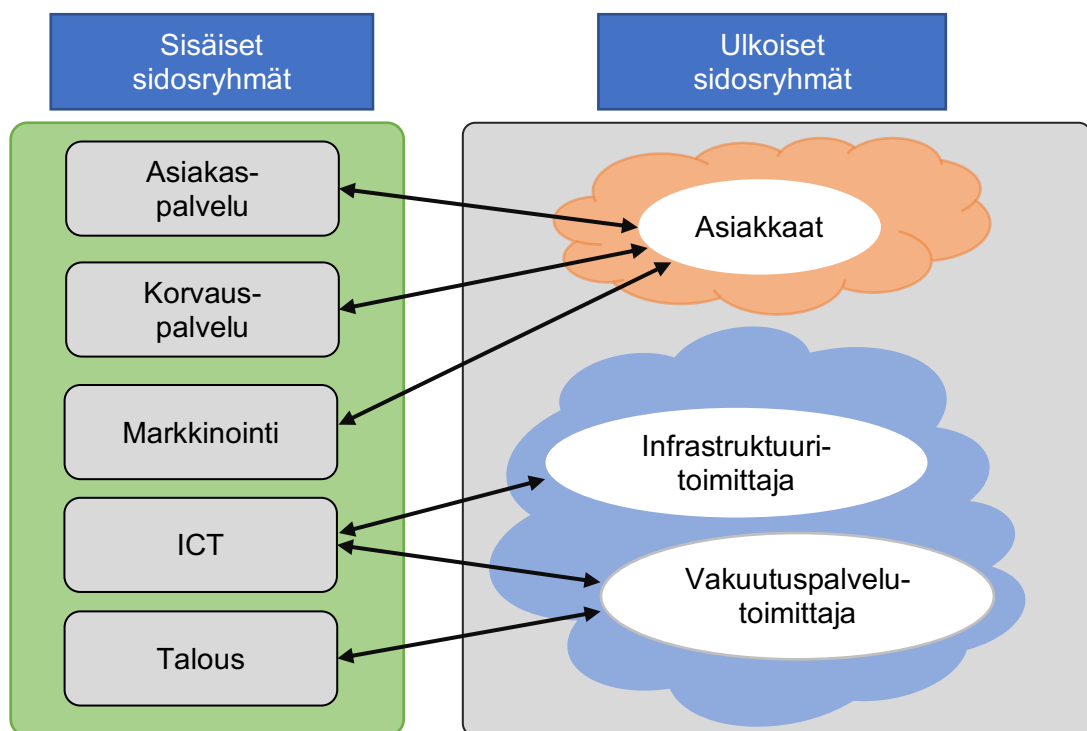
Hallitsen hyvin CSS-merkintäkielen ja olen työssäni oppinut PHP:n perusteita, mutta minulla on vielä paljon kehittymisen varaa. Työharjoitteluni aikana heinäkuussa korjasin POP Vakuutuksen tarjouspohjien ulkoasut, jotka olivat rakennettu PHP-kielillä, josta minulla ei ollut laisinkaan aiempaa kokemusta. Koodiamme on kirjoitettu myös paljon JavaScriptillä ja sen React.js-laajennuskirjastolla. Minulla oli niistä hieman aikaisempaa kokemusta, mutta osaamiseni ei ole vielä sillä tasolla, että osaisin kirjoittaa itsenäisesti toimivaa React.js-koodia, ja merkittävimmät kehittämisen kohteet ovat oppia lukemaan ja kirjoittamaan JavaScript ja React.js -koodia paremmin.

2.2 Sidosryhmät työpaikalla

Suomen Vahinkovakuutus Oy:ssä on yli sata työntekijää, joista lähes kaikki työskentelevät yhtiön Espoon toimistolla. Yrityksemme organisaatio on matala; esimiehet ovat helposti lähestyttäviä. Työyhteisömme yhteishenki on hyvä, ja työpaikallamme kaikki tervehtivät toisiaan. Matala organisaatio tulee esille muun muassa siinä, että kaikki esimiehet ruokailvat aina yhdessä muiden työntekijöiden kanssa, mikäli heidän aikataulunsa sallivat.

Työskentelen ICT-osastolla, johon kuuluu yksitoista henkilöä. ICT-tiimissämme on lisäksi kaksi muuta ohjelmistokehittäjää, kaksi integraatio-, eli rajapintakehittäjää, kaksi testaajaa, kaksi järjestelmäasiantuntijaa ja yksi käyttöliittymäsuunnittelijasta, sekä ICT-tiimimme esimies. Sisäisiin sidosryhmiin kuuluvat ICT-ryhmän lisäksi yrityksen asiakaspalvelu-, korvauspalvelu-, markkinointi- ja talousosastot. Työssäni olen eniten tekemisissä ohjelmistokehittäjäkollegoideni, sekä käyttöliittymäsuunnittelijan kanssa. Olen myös paljon tekemisissä muiden osastojen kanssa.

Ulkoisia sidosryhmiä ovat POP Vakuutuksen ja Säästöpankin vakuutusten asiakkaat ja sivustojen vierailijat. Ulkoisiin sidosryhmiin kuuluvat myös yhteistyökumppanit, ulkopuoliset palveluntarjoajat ja -toimittajat. Ulkoiselta palveluntarjoajalta saamme valmisohjelmistojärjestelmän, jolla voimme hallinnoida heidän tarjoamaansa vakuutusjärjestelmää ja tietokantaa. Lisäksi keskeisenä ulkoisena toimijana on infrastruktuurin toimittaja, joka vastaa muun muassa yrityksemme palvelimista ja verkkoyhteyksistä.



Kuvio 3. Sidosryhmäkaavio

ICT-osastomme on vahvasti tekemisissä kaikkien sisäisten sidosryhmien kanssa, sekä ulkoisista sidosryhmistä ulkoisten palveluntarjoajien kanssa (kuvio 3). Vakuutusjärjestelmä ja infrastruktuuritoimittajiin ovat yhteyksissä lähinnä ICT-osasto ja aktuaariosaston data-analyytikot. Asiakkaisiin ovat pääasiallisesti yhteyksissä, asiakas-, ja korvauspalvelu, sekä markkinointiosasto.

2.3 Vuorovaikutustaidot työpaikalla

Työssäni vaaditaan hyviä vuorovaikutustaitoja, sillä työskentely tapahtuu avokonttorissa, ja olen paljon tekemisissä kollegoideni sekä muiden osastojen työntekijöiden kanssa. Yrityksessämme on hyvin tuttavallinen ilmapiiri, ja yleisesti keskustelemme toistemme kanssa epämuodollisesti. Työpaikallamme kommunikointia tapahtuu monin eri tavoin, ja käytämme eri keskustelu- ja viestipalveluita eri tarpeisiin.

Yrityksen sisäinen viestintä ja ilmoitusluontoiset asiat välitetään sähköpostilla, johon käytämme Microsoftin Outlook-sovellusta. Sähköpostia käytetään myös eri osastojen välisiin keskusteluihin silloin, kun käsitellyt asiat ovat tarpeen saada kirjattua muistiin. Sähköpostiviestin sävy on usein muodollisempi kuin kasvokkain käydyissä keskusteluissa. Väärinymmärrysten välttämiseksi sähköpostiviestinnässä on tärkeää muistaa, että viesteistä ei ole mahdollista tulkita äänensävyä. Pyrin sähköpostiviesteissäni käyttämään neutraaleja ilmaisuja ja kirjoitan viestini kirjakielillä.

Toinen yrityksessä yleisesti käytetty keskustelusovellus on Skype, josta käytämme yritysversiona Skype for Business. Skypen välityksellä voi lähettää viestejä, soittaa puheluita, videopuheluita, tai konferenssipuheluita. Itse käytän Skypeä lähinnä vain, jos joku on ottanut sillä minuun yhteyttä. Työtovereiden kanssa keskustelen epämuodollisesti joko kasvotusten tai Slack-keskusteluohjelman välityksellä. Slackissa voi viestiä suoraan toiselle henkilölle, tai perustaa keskustelukanvia, joihin on mahdollista kutsua haluamansa määrän henkilöitä. Slack tukee muun muassa kuvien, tiedostojen ja koodileikkeiden jakamista.

Mikäli minulla on kysyttävää työstettävistä Jira tai HelpDesk tiketeistä niiden luojien kanssa, esitän kysymykseni kasvotusten niin usein kuin mahdollista. Kasvotusten saan tarvitsemani vastaukset usein nopeammin kuin sähköpostiviestillä, ja vastavuoroisesti minun on nopeampi tarvittaessa antaa vastauksia. Tikettien luojat ovat usein muilta osastoilta, jolloin on hyvä puhua käyttäen mahdollisimman vähän teknistä sanastoa. Jira- ja HelpDesk-tiketeissä on viesti- ja kommenttikentät, joiden kautta on myös mahdollista kysyä lisätietoja ja keskustella työtehtävästä.

Koen olevani sosiaalinen persoona, eivätkä sosiaaliset kanssakäymiset työpaikalla tuota minulle ongelmia. Koen, että vuorovaikutustaitoni ovat hyvät, ja suoriudun sosiaalisissa kanssakäymisissä ongelmitta.

3 Päiväkirjaraportointi

3.1 Seurantaviikko 1

Maanantai 14.1.2019

Suunnittelen tänään tekeväni POP Vakuutuksen Oma Palvelun kalenteripainikkeiden korjauksia. Tarkoituksena on muokata kalenteria siten, että vakuutusta ei ole mahdollista päättää menneellä päivämäärällä. Tavoitteenani on saada mahdollisimman paljon tehtyä, mutta en usko saavani tehtyä kaikkia korjauksia tämän päivän aikana.

Asiakkaan Oma Palvelussa asiakkaan on mahdollista tarkastella omia vakuutuksiaan, vaihtaa mahdollisen autovakuutuksensa turvatasoa, ilmoittaa vahingoista ja irtisanoa vakuutuksia.

Päivä kului suunnitellusti ongelmaa ratkoessa. Suurin haaste oli saada syötekenttä näky-mään oikeassa tilassa asiakkaalle ja myyntisovelluksen käyttäjälle. Lähdin työstämään ratkaisua Coffeescriptin kautta, missä ongelmana oli saada molemmille kentille sama readonly-tila. Lopulta sain muutoksen tehtyä PHP:llä suhteellisen helposti

Tiistai 15.1.2019

Tänään tarkoituksenani on saattaa loppuun keskeneräinen tiketti päivämäärän valitsemi-seen käytettävästä datepickeristä, sekä ottaa työn alle muut Oma Palvelun tiketit, mikäli niihin liittyvät ongelmat eivät korjaannu muiden korjausten vaikutuksesta. Aion tarkistaa, mikä on muiden tikettien tilanne, saadakseni paremman kuvan tulevista työtehtävistä. Ta-voitteenani on saattaa datepicker-tiketti resolved-tilaan.

Päivä kului tiketin viimeistelyssä, ja koodin laadun varmistamisessa. Tiketti ovat yhä kes-ken, sillä en ole vielä kyennyt testaamaan muutoksia mobiililaitte-simulaattorilla.

Keskiviikko 16.1.2019

Päivän tehtävälistalla on viimeistellä keskeneräiset tiketit. Aamupäivän aikana lataan xcode-sovelluksen, minkä simulaattoria käyttäen tarkistan, että tehdyt muutokset toimivat odotetusti mobiililaitteilla. Korjaan mahdolliset virheet ja vien muutokset testiserverille, ja merkkeaan tiketit resolved-tilaan. Toivon ehtiväni tehdä nämä ennen kello 13 alkavaa sprint-palaveria. Palaverin jälkeen tutkin lisää, mitä tikettejä jatkan, tai mitä uusia alan

työstämään. Tavoitteenani saada datepickeriin liittyvä tiketti valmiiksi. Loppupäivän käytän muiden tikettien työstöön.

Saadakseni Xcode-koodieditorin ladattua, minun täytyy saada palautettua Apple Id:n unohtunut salasana. Aloitin salasanan palautuksen eilen keskiviikkona työpäivän loppupuolella. Salasanan vaihtamiseksi minun tarvitsee odottaa vielä seitsemän tuntia, eli työpäivän päättymiseen saakka, ennen kuin saan salasanan palautettua. Unohtunut salasana esti tehtyjen muutosten testaamista. Vaihtoehtoinani on siirtää muutokset testiserverille, ja toivoo, että luomani koodi toimii oikein, tai voin odottaa päivän loppuun, ja työstää muita tikettejä sillä välin, ja siirtää muutokset testiserverille testauksen jälkeen. Koen, että odottaminen olisi ajan haaskausta, sillä testiserverillä joku testaajistamme voi koostaa muutokset puolestani nopeasti. Muutokset ovat myös helposti korjattavissa, mikäli ongelmia ilmenee. Tarpeen vaatiessa voin jopa peruuttaa muutokseni, tekemällä uuden puskun develop-haaraan, jolla kumoan tehdyt muutokset.

Siirtää muutokset testiserverille, ja ohjeistin testaajaa, mistä muutokset löytyvät, ja miten tehtyjen muutosten on tarkoitus vaikuttaa sivustoon eri laitteilla. Kokonaisuudessaan Oma Palvelun kalenterivalinnan korjaamiseen käytin 12 tuntia kolmen päivän ajanjaksolla.

Sprint-palaverissa kävimme läpi tulevan sprintin tehtäviä. Omaksi tehtäväkseni valitsin uuden murupolkunavigaation luonnin. Murupolkunavigaatiota on suunniteltu useamman viikon ajan SEO-auditoinnista saatujen tulosten pohjalta. Ryhdyin kollegani kanssa tutkimaan miten murupolun saisi luotua helpoiten ja kestävimmin.

Työpäiväni ei mennyt suunnitellusti. En kyennyt testaamaan muutoksia mobiililaitteilla, minkä odotan aiheuttavan ongelmia jatkossa. En odottanut, että Apple Id:n salasanan palauttaminen tulisi kestämään niin pitkään. Olin tyytyväinen, että olin saanut vihdoinkin kalenterivalikon vietyä testiserverille, eli develop-haaraan testattavaksi.

Torstai 17.1.2019

Tänään jatkan murupolun prototyypin luontia, ja tutkin, miten helppoa tai vaikeaa se on tehdä Contentfulista kautta. Tavoitteenani on saada kehiteltyä murupolkunavigaation alustava prototyyppi. Tarkoituksena on tässä vaiheessa tutkia, miten se on viisainta luoda.

Aloitin prototyypin työstämisen paperilla piirtäen ja suunnitellen. Alustavan ideana oli, että Contentfuliin luotaisiin alasveto- eli dropdown-valikko, josta voidaan valita vanhempisivu, eli minkä sivun alle sivu kuuluu, mikä kertoo, millä tasolla sivu sijaitsee sivuhierarkiassa.

Loin suunnitelmieni perusteella Contentfuliin uuden murupolku sisältötyypin, johon loin listan eri sivuja. Yritin monia eri ratkaisuja, kuten useampaa dropdown-valikkoa, joista rakentaa murupolkua ja asettaa vanhempisivua. Totesin, ettei tämän tyyppinen ratkaisu tule olemaan helppo toteuttaa.

Kollegaltani sain viestin, että 404-sivu oli alkanut tuottamaan virheviestejä lokeihin testiserverillä sekä QA:lla. Kollegani ohjasi minut oikeaan suuntaan, missä kohtaa koodissa virhe esiintyy. Tein korjauksen sivun rakenteeseen sekä controlleriin, josta uusi ulkoasu piirretään. Virhesivun ulkoasu määrytyy nyt aikaisemmassa vaiheessa, mikä samalla korjasi "status code" eli tilakoodin 404:ksi. Aiemmin status code oli täytynyt asettaa erikseen koodissa, sillä uusi virhesivu ei ollut käynyt koodissa läpi kaikkia samoja vaiheita kuin vanha sivu, mikä johti siihen, että uusi sivu ei automaattisesti saanut status code 404.

Sain myös testaajilta palautetta Oma Palvelun kalenterikorjauksiin liittyen. Kalenterin ominaisuuksista löytyi useita ongelmia. Onneksi ongelmat oli raportoitu selkeästi ja jäsennellysti, mikä teki uusien korjauksien luomisesta helpompaa. Tämän ongelman ratkaisemisen siirrän myöhemmälle.

Päivä meni lähes kokonaisuudessaan murupolkua työstäessä. Lopputuloksena minulla ei ollut juuri mitään näytettävää. Opin paljon, mitä Contentfulilla on mahdollista tehdä, ja mitä taas ei ole. Koen, että en epäonnistunut päivän tavoitteessani. Loin prototyyppiä murupoluista, vaikkakaan en saanut luotua toimivaa ratkaisua. En koe, että suunnitelmani olisi täysin epäonnistunut, sillä sain tärkeää tietoa ja opin, ettei suunnittelemani ratkaisu ollut oikea ratkaisu. Minun tuleekin alkaa rakentamaan murupolkuja jollain toisella tapaa.

Perjantai 18.1.2019

Suunnitelmissani on tänään korjata ongelmat ja virheet Oma Palvelun kalenterissa, josta sain tietoa eilen. Sen jälkeen jatkan murupolkujen kehittämistä. Murupolku-tiketti on mielestäni tärkeä saada valmiiksi, koska sillä on vaikutusta hakukonenäkyvyyden parantamisessa. Murupolku-tiketti on iso ja hankala, enkä usko ehtiväni saada sitä valmiiksi ennen ensi keskiviikkoa, joten tiketti siirtynee vielä seuraavalle sprintille. Tavoitteenani on kehittää murupolkua mahdollisimman pitkälle tämän päivän aikana.

Aloitin päivän Oma Palvelun kalenterin ongelmien korjaamisella, sillä se oli prioriteetissa korkeampi kuin murupolkujen luominen. Testaajat olivat luoneet kolme uutta tikettiä kalenteripainikkeen ongelmista. Ne olivat otsikoitu "Several issues with ending insurance and reporting an accident", " Problems with calendar and date fields (Asiakkaan Oma

Palvelu)”, sekä “Mobile iOS: User can insert past date when ending insurancy”. Kaikki kolme tikettiä liittyvät vahvasti toisiinsa, ja ovat miltei samanlaisia. Ongelmana on sivulla käytettävät kaksi eri kalenteria. Kalenteri esitetään sen perusteella, onko asiakas selaamassa sivua mobiililaitteella vai tietokoneella.

Useista yrityksistä huolimatta kalenteria ei ollut mahdollista muokata haluttuun muotoon. Kalenterikentästä olisi täytynyt poistaa päivämäärän vaihtamiseen ja kentän tyhjentämiseen käytettäviä painikkeita. Painikkeiden piilottaminen ei ole mahdollista kaikilla selaimilla, minkä vuoksi syötekentän tyyppi `type="date"` oli vaihdettava tekstityypiksi `type="text"`. Tekstityypiseen kenttään ei tule selaimen kalenteritoimintoja, jolloin on helpompaa muokata kentän ulkoasua ja toiminnallisuutta haluttuun muotoon. Asiakkaalta on poistettu mahdollisuus kirjoittaa kenttään käsin, ja on sen sijaan pakotettu käyttämään kalenteria. Näin ehkäistään virheellisten päivämäärien syöttäminen kalenterikenttään. Sopimuksen päättäminen menneellä päivämäärällä, tai vahinkoilmoituksen luominen tulevalle päivämäärällä tulee saada estettyä.

Kalenteri vaikuttaa nyt toimivan halutulla tavalla. Uutena ongelmana nousi datepickerin päivämäärän väärä muoto. Päivämäärä tulee esittää ensin päivä, sitten kuukausi, ja vuosi, mutta datepickerissä muoto on ensin vuosi, sitten kuukausi, ja päivä. Järjestyksen voisi vaihtaa näennäisesti helposti, mutta päivämäärän formaatti vaikuttaa kalenterin toiminnallisuuteen. Tämä ongelma jää ratkaistavaksi ensi viikolle.

Viikkoanalyysi

Viikon aikana työstin POP Vakuutuksen Oma Palvelun kalenterikenttien ominaisuuksia, sekä suunnittelin uuden murupolkunavigaation toteuttamista. Oma Palvelun kalenterin muutokset tuntuivat helpoilta, mutta osoittautuivat monimutkaisiksi. Suurimmat ongelmani viikon aikana olivat Oma Palvelun kalenterissa, sen valikoissa ja ulkoasussa, sekä päivämäärän validoinnissa. Kalenterin päivämäärän valitseminen oli aiemmin onnistunut ja toiminnut selaimen oman kalenterivalikon avulla. Ongelmana oli kuitenkin, että asiakkaan saattoi olla mahdollista asettaa vakuutuksen päättämispäivämäärä menneisyyteen. Tikeissä tahdottiin estää Oma Palvelun kalenterissa virheellisten päivämäärien syöttö.

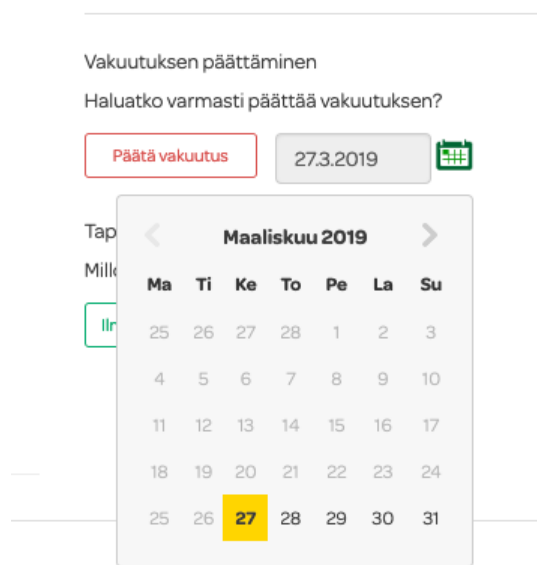
Vuonna 2008 julkaistu HTML 5 (W3C 2008), mahdollisti selaimien omat sisäänrakennetut kalenterivalikot. Kaikki selaimet, kuten Internet Explorer, eivät tue tätä ominaisuutta (Go Make Things 2018), vaikka HTML 5 julkaistiin jo yli kymmenen vuotta sitten. Vasta useiden yritysten ja turhautumisten jälkeen huomasin tarkistaa, että mitkä kaikki selaimet tukevat date-tyyppistä syöttökenttää eli `input:ia`. Natiivi-kalenterivalikkoa ei tuntunut olevan

helppoa tai edes mahdollista muokata tyyliä. Käytin paljon aikaa saadakseni kalenterin näkymään oikein, mutta lopulta jouduin vaihtamaan syöttökentän tyylin tekstimuotoon.

```
<input type="text"
```

Kuva 1. Tekstityyppinen syöttökenttä

Tekstityyppinen kenttä (kuva 1) ottaa vastaan kaikkia kirjaimia, numeroita ja merkkejä. Kentän tiedot on mahdollista lähettää eteenpäin taustalla pyörivän koodin käsiteltäväksi. Kirjoittamisen sijaan Oma Palvelussa tekstikenttää klikattaessa avataan itsetehty päivämäärän valitsin.



Kuva 2. Oma Palvelun vakuutuksen päättämisen kalenteri

Päivämäärän valitsin on käytännössä kalenteri, josta käyttäjä voi valita päivämäärän (kuva 2). Oikean päivämäärän raja-arvojen asettaminen oli yksi haasteista. Ratkaisin ongelman asettamalla inputille minimi- ja maksimiarvot, joiden väliltä päivämäärä on valittavissa. Käytin raja-arvoina koodiin haetun vakuutuksen tietojen perusteella. Vakuutuksen tiedoista sain muun muassa alkamis- ja päättymispäivämäärät. Vakuutuksen irtisanomisen minimiarvoksi asetin käyttämään senhetkistä päivämäärää, eli vakuutusta ei pysty irtisanomaan menneellä päivämäärällä. Maksimiarvoksi asetin vakuutuksen viimeisen voimassaolopäivän. Välitin koodissa nämä tiedot kalenterielementille, jossa minimi- ja maksimipäivämäärien välissä olevat sallitut päivämäärät näkyivät normaalisti vaaleina ja olivat valittavissa, siinä missä kielletyt päivämäärät olivat tummennettuina, eikä niitä voinut valita. Vahinkoilmoituksessa minimipäivämääräksi asetin vakuutuksen alkamispäivän, sillä vahinkoilmoitusta ei voi tehdä ajalta ennen vakuutuksen voimaantuloa. Maksipäivämäärän

asetin senhetkisen päivämäärän, koska vahinkoilmoitusta ei ole mahdollista tehdä vahingolle, joka sattuu tulevaisuudessa.

Päivämäärän pystyisi yhä asettamaan menneisyyteen, jos tekstikenttään kirjoittaisi menneen päivämäärän. Tekstikenttään kirjoittamisen estin asettamalla teksti-inputille attribuutin `readonly`, mikä estää tekstikentän muokkaamisen ja siihen kirjoittamisen (w3schools.com 2019a). `Readonly` muuttaa hiiren kursorin kieltomerkiksi useimmilla selaimilla, mikä ei ollut haluttu ominaisuus. Tein tyylimuutoksen, jotta kursori näyttää tavalliselta nuolelta.

Korjasin 404-virhesivun aiheuttaman virhetilanteen. Virhekoodilla 404 ilmaistaan, että selaimella haettua sivua ei löytynyt (Fielding & Reschke 2014, 59). Tällainen virhe voi tapahtua, jos käyttäjä kirjoittaa selaimeen väärän osoitteen, tai jos sivustolla oleva linkki ohjaa virheellisesti sivulle, jota ei ole olemassa. Sivustollamme kohdatut 404-virheet ohjataan omalle tyylitellylle 404-sivulle, jossa käyttäjälle kerrotaan, että haettua sivua ei löytynyt. Testiserverillä virhesivun PHP-koodi oli alkanut tuottaa virheviestiä, mikä johtui funktiokutsusta, jolla piirrettävä ulkoasu poistettiin käytöstä. Korjasin ongelman, ja uusia virheviestejä ei enää syntynyt.



Pahoittelut, sivua ei löydy

Etsimme sivuasi kissojen ja koirien kanssa, tuloksetta. Olisiko jokin alla olevista se mitä haet?

[Laske autovakuutuksen hinta](#)

[Laske kotivakuutuksen hinta](#)

[Kaikki vakuutuksemme](#)

[Kirjaudu Oma Palveluun](#)

[POP Vakuutuksen etusivulle](#)

Kuva 3. Uuden virhesivun sisältö

Uudella virhesivulla pahoitellaan, että sivua ei löydy, sekä ohjataan sivuston muihin osioihin. Aiemmin käytetty 404-sivu oli rakennettu layout helperillä, joka luo koko sivun ylä- ja alanavigaatioineen, sekä sivun virheviestin. Uusi sivu sen sijaan tuo sivulle pelkän

virhesivun sisällön (kuva 3). Ylä- ja alanavigaation poistamiseksi vanha ulkoasu oli otettava pois käytöstä funktiolla `disableLayout`, mikä oli alkanut tuottamaan virheviestiä. Poistin funktion koodista, ja sen sijaan, että sivun ulkoasu valittaisiin sivulle tultaessa, ulkoasu valitaan, kun käyttäjää ohjataan virhesivulle. Tällä muutoksella sain korjattua myös tilakoodin eli `status code:n`, mikä kertoo sivun tilan (Fisher 2019). Aiemmin koodi oli täytynyt asettaa manuaalisesti tilaan 404, mutta tekemilläni muokkauksilla, koodi tuli sivulle oikein ilman erillistä asettamista.

Viikon aikana opin paljon selaimien ominaisuuksista, ja niiden rajoitteista. Caniuse.com-verkkosivulla on mahdollista tarkistaa muun muassa eri HTML-attribuuttien, CSS-tyylien ja JavaScript-funktioiden toimivuutta ja tukea eri selaimilla. Huomasin hyödylliseksi tutkia sivulta muun muassa `readonly`-attribuutin toiminnallisuutta ja ominaisuuksia. `Readonly` asettaa kentän tilaan, jossa se on nimensä mukaisesti vain luettavissa, mutta ei muokattavissa. Osaamiseni on kehittynyt ongelmanratkaisussa, ja kokonaisuuksien hallinnassa. Huomasin löytäväni ongelmakohtia, ja niiden ratkaisun entistä nopeammin, vaikkakin koin vielä tarvitsevani oman työni oikeellisuuden varmentamista kollegoiltani. Pienellä kollegan avustuksella 404-sivun ongelmissa kykenin tekemään sivuston rakenteesta eheämpää ja latautumisesta nopeampaa.

3.2 Seurantaviikko 2

Maanantai 21.1.2019

Päivän suunnitelmissa on viimeistellä tekemäni korjaukset kalenteripainikkeisiin. Tämän viikon viikoittainen tiimipalaveri on kalenterin mukaan peruttu, ja palaverissa läpi käytävät asiat käsitellään keskiviikon sprint-palaverissa. Tavoitteenani on saada tämän päivän aikana luotua jokin toimiva ratkaisu Oma Palvelun kalenteriin ja sen päivämäärävalintaan.

Viime viikolla sain kalenteripainikkeen muilta osin toimimaan, mutta päivämäärän esitysmuoto oli vielä väärä. Tein muutokset päivämäärän esitysmuotoon, ja päivämäärä näkyi nyt oikein kalenterivalikossa. Päivämäärän muokkaaminen vaikutti rikkoneen kalenterin, sillä vakuutusta ei enää voinut päättää.

Päivä ei sujunut suunnitelmien mukaisesti. Lopputuloksena oli aiempaa huonommin toimiva kalenteri. Vakuutusta ei nyt voi päättää edes vakuutuksen ensimmäisellä päivämäärällä. En päässyt tavoitteeseeni saada Oma Palvelun kalenterivalikko toimimaan oikein. Suurimmat ongelmat olivat päivämäärien muotoilu oikeaan formaattiin.

Tiistai 22.1.2019

Tänään tavoitteenani saattavani loppuun muutokset Oma Palvelun kalentereihin. Suunnittelen käyttäväni päivän valtaosin kalenterivalikoiden ja päivämäärävalintojen työstämiseen. Toiveenani on pystyä ottamaan työn alle kalenterivalikon korjauksen lisäksi muitakin tehtäviä. Kalenteriin liittyviin tiketteihin on mielestäni kulunut jo liian kauan aikaa, ja luulen, että tarvitsen apua kollegoiltani.

Tein eilen muutoksia päivämäärän syöttöihin ja sen oikeellisuuden tarkistukseen, mitkä aiheuttivat bugin, minkä vuoksi vakuutusta ole mahdollista irtisanoa. Löysin ongelmaan syyn kirjaamalla lokiin koodin dataa.

Tein muutoksentyylitiedostoon, ja muutin cursorin ulkoasun osoittavaksi sormeksi, kun hiiri viedään kalenterikentän päälle. Osoittava cursori auttaneen nuolen sijasta asiakasta paremmin hahmottamaan, että kenttä on klikattavissa. Kenttää klikattaessa aukeaa päivämäärän valitsemiseen käytettävä kalenteri.

Päivän lopuksi huomasin, että Oma Palvelun vakuutukset välilehdellä on ongelma, että sivulla ei näytetä kaikkea dataa sivun latautuessa. Olen huomannut tämän jo aiemmin, mutta ongelma ei esiinny aina, mikä tekee sen tutkimisesta ja ratkaisemisesta hankalaa. En ryhdy nyt tutkimaan lisää, sillä en löytänyt aiheeseen liittyvää Jira-tikettiä. On siis mahdollista, että kyseinen ongelma on johtunut omasta selaimestani.

Pääsin päivän tavoitteeseeni; sain Oma Palvelun kalenterikentät pitkän työstämisen jälkeen toimimaan oikein. Tämän päivän ongelmat johtuivat valtaosin kalenterivalikon monimutkaisista päivämäärän muotoiluista, ja tarkistuksesta.

Keskiviikko 23.1.2019

Suunnittelen tälle päivälle tehtäväksi päivän alussa käydä läpi minulle osoitettuja avoimia tikettejä, ja tarkistaa vielä kertaalleen, että testiserverillä aiemmin tekemäni muutokset kalenteriin ovat kunnossa. Suunnittelen jatkavani viime viikolla työstämäni sivuston murupolkua. Kello 10.30 alkaa kokous, jossa käydään läpi hakukoneoptimointiin tehtyjä muutoksia ja niiden tuloksia. Viikoittainen sprint-palaverimme alkaa kello 13.

Aloitin päivän käymällä läpi avoimia tikettejä, ja merkkasin osan ratkaistuksi, ja luovutin niitä Proven testattavaksi. Seuraavaksi jatkoin murupolkunavigaation työstämistä. Yhdessä kollegani kanssa keskustelimme ja pohdimme mahdollisia vaihtoehtoja

murupolkujen luomiseksi. Murupolut olisi luotava dynaamisesti Contentfulin syötteiden kautta. Jokainen sivu saisi oman syötteen, jonka nimi on murupolku, johon kirjoitetaan, millä nimellä kyseenomainen sivu esitetään navigaatoriossa, ja kentän, joka kertoo minkä sivun alla kyseenomainen sivu sijaitsee sivuhierarkiassa. Jokaiselle sivulle täytyi lisätä oma murupolku-moduuli. Murupolku-navigaatio täytyy luoda JavaScriptillä.

Contentfulissa on hankala luoda hierarkkisuutta sivujen välille. Sivuja voitaneen luoda muiden sivujen sisälle, mutta niiden ylläpito ja löytäminen Contentfulissa olisi tällöin liian vaikeaa ja vaivalloista. Contentfulissa jokainen yksittäinen sivu on hierarkkisesti samalla tasolla; mikään sivu ei ole minkään toisen sivun sisällä, ala- eikä yläpuolella. Contentfulissa on mahdollista linkittää sivuja toisiinsa, mutta se ei itsessään luo hierarkisuutta sivujen välille.

Torstai 24.1.2019

Sairaana

Perjantai 24.1.2019

Sairaana

Viikkoanalyysi

Toisella seurantaviikolla työstin pääasiallisesti Oma Palvelun kalenteripainikkeita. Olin aloittanut kalenteripainikkeiden työstämisen jo edellisellä viikolla. Jiraan oli tullut bugitietti, jossa kerrottiin, että käyttäjän oli mahdollista päättää vakuutus menneellä päivämäärällä. Ongelma haluttiin ratkaista mahdollisimman pian, sillä asiakkaan ei tule olla mahdollista päättää vakuutustaan menneellä päivämäärällä. Tiesin edellisellä viikolla jo työtä aloittaessani, että kalenterivalikoiden rakentuminen ja niiden kenttien validoiminen olisi hankalaa, mutta en odottanut kohtaavani niin paljon ongelmia kuin kohtasinkaan.

Oma Palvelussa on kaksi samankaltaista kalenterikenttää, yksi vakuutuksen päättämiselle, ja toinen vahinkoilmoitukselle (kuva 4). Kentät ja niiden kalenterivalikot rakentuivat erikseen omista tiedostoistaan, jotka ovat rakenteeltaan hyvin samanlaisia. Molemmat avaavat oman kalenterivalikon, josta aukeaa kalenteri, jolla valita päivämäärä joko vakuutuksen päättämiselle tai vahinkoilmoituksen tekemiselle.

Vakuutuksen päättäminen

Haluatko varmasti päättää vakuutuksen?

Päättä vakuutus

27.3.2019



Tapahtuiko vahinko?

Milloin vahinko tapahtui?

Ilmoita vahingosta

27.3.2019



Kuva 4. POP Vakuutuksen Oma Palvelun päivämääräkentät

Oma Palvelun kalenterivalikko oli käyttäytynyt väärin ja sallinut asiakkaan päättää vakuutuksensa menneellä päivämäärällä, minkä lisäksi päivämäärät olivat esitetty väärin. Olin kalenterivalikkoa työstäessäni muuttanut kalenterikentässä näytettävän päivämäärän muotoon j.n.Y. Pieni j-kirjain osoittaa päiviä, pieni n-kirjain osoittaa kuukausia, ja iso Y-kirjain osoittaa vuosia. Tarkistin PHP:n dokumentaatiosta (PHP 2019), että pieni j-, ja n-kirjain voivat sisältää joko yhden tai kaksi päivämäärää osoittavaa lukua, eli päivän numeroa ei ole pakko kirjoittaa kahdella numerolla. Pieni j-, ja n- kirjain mahdollistavat esimerkiksi päivämäärän 01.02.2019 kirjoittamisen myös muodossa 1.2.2019. Tiesin entuudestaan, että Y-kirjain kirjoitettuna isolla tarkoittaa vuotta neljällä numerolla esitettynä. PHP dokumentaation mukaan pieni y-kirjain sen sijaan tulostaisi vuosiluvusta vain kaksi viimeistä numeroa, mikä ei tässä tapauksessa ole haluttu lopputulos. Tekemilläni muokkauksilla sain kalenterin päivämäärän muotoilun näyttämään oikealta.

Taustalla ajettavaan koodin oli vielä tehtävä muutoksia päivämäärän käsittelyyn. Tällä hetkellä koodi odottaa vastaanottavansa päivämäärän siinä muodossa, mitä se oli ennen tekemiäni muokkauksia. Koodi näin ei toimisi, ja syntyisi virhe, mikäli päivämäärä on väärässä muodossa. Tein koodiin numeroiden muotoiluun muutoksia, mutta kävi ilmi, että jotain näytti menneen pieleen. Tekemäni muutokset eivät korjanneetkaan mitään, vaan aiheuttivat ongelman, ettei vakuutusta pystynyt päättämään laisinkaan. Perehdyin tekemiini muutoksiin, mutten huomannut, mikä olisi vialla. Päätin tutkia koodin dataa, ja tulostin lokeihin, mitä kaikkea tietoa koodissa oli käytössä. Lokista etsin päivämääriä, mutta hakuni oli niin laaja, että sain liian paljon tietoa, ja merkityksellisen tiedon löytäminen oli vaikeaa. Rajasin lokitusta koskemaan vain kalenterivalikon vaihtoehtojen sisältöihin, ja löysin lokeista poikkeavalta vaikuttavaa dataa. Kävi ilmi, että päivämäärädata, jolla vertauksia tehtiin, oli negatiivinen, mutten keksinyt mistä se taas johtui. Kollegani kanssa tutkimme

koodia, ja huomasimme, että negatiivinen luku johtui väärästä päivämääräformaatista taustalla ajettavassa koodissa. Ongelman ydin oli se, että validoinnissa oli yritetty käyttää negatiivista päivämäärä. Raja-arvoihin sopimaton päivämäärä esti vakuutuksen irtisanomisen. Ongelma johtui siis siitä, etten yksinkertaisesti ollut huomannut tehdä muutoksia kaikkiin tarvittaviin koodin osiin. Yksi funktioista odotti yhä virheellisesti vastaanottavansa päivämäärän vanhassa muodossa, kun sille lähetettävän päivämäärän muotoilu oli muuttunut. Muutin koodissa funktion formatoon oikeaan muotoon, ja kalenteri toimi taas toivotusti.

Onnistuin saamaan Oma Palvelun kalenterin toimimaan mielestäni kokonaisuudessaan oikein. Testailin työni tulosta, ja yritin saada syötettyä kenttään virheellisiä päivämääriä siinä onnistumatta. Kenttään ei ollut mahdollista kirjoittaa, sillä sille oli asetettu readonly-attribuutti. Poistin attribuutin käyttäen selaimen kehitystyökaluja, minkä jälkeen kenttään oli mahdollista kirjoittaa mennyt päivämäärä. Onnistuin päättämään vakuutuksen menneellä päivämäärällä, mikä oli ongelma. Tavallinen käyttäjä tuskin keksisi muokata kenttää kuten olin muokannut, mutta ongelma oli silti ehdottomasti korjattava.

Tein koodiin tarkistuksen, että vakuutusta päättäessä kalenterista tulevan päivämäärän on oltava vähintään nykyinen päivämäärä. Vakuutusta ei päätettäisi, mikäli saatu päivämäärä ei läpäisisi tarkistusta. Huomasin vahinkoilmoitusta tehtäessä saman ongelman kuin vakuutuksen päättämisessä. Tein korjauksen samaan tapaan tarkistamalla kalenterista tulevan päivämäärän oikeellisuuden. Kalenterista saatu päivämäärä ei saa olla nykyistä päivää suurempi, eli vahinkoilmoitusta ei voi tehdä tulevaisuuteen. Tekemäni muutokset estivät virheellisten päivämäärien lähettämisen, vaikka kalenterikenttiä, maksimiarvoja, päivämääriä, tai muuta kentän sisältöä manipuloisi selaimen kehitystyökalujen avulla.

3.3 Seurantaviikko 3

Maanantai 28.1.2019

Tehtäväni on tänään jatkaa viime viikoilla aloittamaani murupolkunavigaation luontia ja suunnittelen rakentamani murupolkujen hierarkiaa. Minulle ei ole suunniteltu muita tekemisiä. Tavoitteenani on saattaa murupolkunavigaatio alkuun. En kuitenkaan usko saavani aikaan vielä tämän päivän mitään merkittäviä tuloksia.

Minulla ei ole vertailukohtaa, jonka perusteella osaisin arvioida kauanko murupolkunavigaation luomisessa tulisi kestämään, mutta kollegani arvion mukaan siinä voi helposti kestää yli viikon. Aloitin murupolkunavigaation työstämisen luomalla uuden Handlebars-

sivupohjan, johon tietoja tuotaisiin. Seuraavaksi koitin tuoda sivupohjaan Contentfuliin luotuja tietoja. Työstäminen oli hidasta ja hankalaa.

Pääsin hieman eteenpäin, mutta murupolku on kaukana valmiista. En osaa arvioida pääsinkö tavoitteeseeni, sillä odotin saavani aikaan paljon enemmän kuin päivän aikana sain tehtyä.

Torstai 31.1.2019

Murupolkunavigaation luominen on niin haastavaa, etten päässyt työssäni käytännössä yhtään eteenpäin. Tänäpä suunnittelen jatkavani murupolkunavigaatiota, keskiviikkona kollegaltani saamieni ohjeiden perusteella. Tavoitteenani on saada murupolkunavigaation luovaan JavaScript-tiedostoon tuotua Contentfulista dataa ja oikeassa muodossa.

Minulla kesti pitkään luoda monimutkainen silmukkasääntö, jolla kävin läpi kaikki olemassa olevat tiedostot, joiden sisältä etsin yhtä tiettyä tiedostoa. En kuitenkaan saanut vielä dataa tulostettua oikein konsolilokille.

Päiväni kului suunnitellusti ja työstin koko päivän murupolkunavigaatiota. En päässyt päivän tavoitteeseeni, sillä Contentfulin rakenne ja tietojen tuominen Contentfulista oli minulle vielä uutta.

Perjantai 1.4.2019

Suunnittelen tänään jatkavani murupolkunavigaatiota ja tavoitteenani on saada tuotua Contentfuliin lisäämäni tiedot murupolkujen tietoja käsittelevään JavaScript-tiedostoon.

Sain useiden korjausten jälkeen tulostettua konsolilokiin oikeanlaisia tietoja. Seuraavaksi minun pitäisi saada vietyä tiedot sivupohjille.

Saavutin päivälle asettamani tavoitteen. En kuitenkaan ollut tyytyväinen työni aikaansaamiseen, sillä työn teossa oli kulunut mielestäni kohtuuttoman paljon aikaa. Ymmärsin ja tiesin, että työssä tulisi kulumaan paljon aikaa, ja se olisi haastava tehtävä, mutta lukuisat epäonnistumiset lannistivat minua. Huomasin oppineeni valtavasti asioita Contentfulin tietojen hakemisesta ja niiden käsittelystä. Koen, että minun olisi pitänyt pyytää enemmän apua kollegaltani, sen sijaan, että olisin käyttänyt niin paljon aikaa asian tutkimiseen itsestäni.

Olin käyttänyt koko viikon pelkästään murupolkunavigaation luomiseen. Murupolkujen luominen oli yksi osa sivustomme hakukonenäkyvyyden parantamiseen liittyvistä tiketeistä. Hakukoneoptimoinnin parantamiseen ryhdyttiin sivuston kävijämäärien lisäämiseksi ja myynnin parantamiseksi. Hakukoneoptimointi on tärkeää, sillä valtaosa hakukoneen tuloksien klikkauksista kohdistuu ensimmäisiin viiteen esitettyyn tulokseen (Chris 2019). Saa-daksemme enemmän klikkauksia, sivustomme kannattaisi siis olla viiden ensimmäisen hakutuloksen joukossa. Hakukonenäkyvyyden parantaminen on tällä hetkellä yksi kehittä-misen kohteistamme. Hakukonenäkyvyys perustuu hakukonerobottien suorittamaan sivu-jen kahlaamiseen, jolla robotit käyvät läpi sivuja, niiden sisältöä ja linkkejä, ja hakemistoi-vat eli indeksoivat sivut. Vain indeksoidut sivut esitetään hakukoneen hakutuloksissa. (Os-man 23.2.2018; Terenteva 1.11.2016.)

Murupolkunavigaatio on toissijainen navigaatio, joka näyttää käyttäjälle millä sivulla hän on sivuhierarkiassa. Navigaatiossa näytetään, minkä sivujen alla kulloinenkin sivu on, mikä auttaa havainnollistamaan sivuston rakennetta käyttäjälle. Murupolkunavigaation nimi juontuu Hannu ja Kerttu -tarinasta, jossa sadun hahmot jättivät jälkeensä kiviä ja lei-vänmuruja löytääkseen takaisin kotiinsa. (Babich 2019; Forsey 20.9.2018; Gube 2009.) Murupolkunavigaation muruilla tarkoitetaan selailtavan sivun edeltäviä sivuja.



Kuva 5. Murupolkunavigaation suunnitelmakuva

Uudesta navigaatiosta oli luotu Jiraan tiketti, johon käyttöliittymäsuunnittelijamme oli lisän-nyt malleja murupolkunavigaation ulkoasusta (kuva 5). Sivustolle tahdottiin luoda uusi mu-rupolkunavigaatio, koska sillä olisi positiivinen vaikutus hakukonenäkyvyyteen. Uuden na-vigaation luonti perusteltiin SEO-auditoinnin raportin tuloksilla. Auditointi oli laaja sivuston löydettävyysoanalyysi, jossa käytiin yksityiskohtaisesti läpi sivustoltamme löytyviä ongelma-kohtia. Raportissa kerrottiin sivustomme puutteista, sekä esitettiin lista muutosehdotuk-sista, joilla hakukonenäkyvyyttä voitaisiin parantaa.

Raportissa mainitut muutosehdotukset olivat listattu taulukkoon, ja niille oli annettu prioriteetti asteikolla yhdestä viiteen, jossa luku viisi oli kaikkein korkein, ja luku yksi kaikkein matalin prioriteetti. Prioriteettiluvulla kuvattiin muutoksien tärkeyttä ja merkittävyyttä hakukoneoptimoinnin parantamisessa. Murupolkunavigaatio oli prioriteetiltaan tasolla neljä.

Murupolkunavigaatio tahdottiin luoda käyttäjien sivunavigoinnin helpottamiseksi. Murupolkunavigaatiolla olisi myös positiivinen vaikutus sivujen indeksoinnissa. SEO-auditoinnin raportissa kerrottiin, että hakukoneille ei selvästi käynyt ilmi sivujemme nykyisistä osoitteista, minkä sivun alle mikäkin sivu kuuluu. Useimmille sivustomme sivuista pääsi muuttamalla klikkauksella. Googlen hakukone antaa enemmän painoarvoa, ja sen myötä näkyvyyttä, niille sivuille, joille pääsee etusivulta pienimmällä määrällä klikkauksia (Southern 9.6.2018). Sivustomme kaupallisesti tärkeimpien sivujen painoarvoa pystyttäisiin hakutoksissa nostamaan, mikäli vähemmän tärkeitä sivuja siirrettäisiin useamman klikkauksen taakse etusivulta laskettuna. Google antaa pienemmän painoarvon sivuille, joille päästäkseen käyttäjän on klikkailtava useampaa linkkiä (Southern 9.6.2018).

Sivustomme sivuhierarkia ei ollut optimaalinen, ja sivuston rakennetta tultaisiin muuttamaan auditoinnin suosituksen mukaan. Tehtävät sivurakenteen muutokset vaativat monien sivujen URL-osoitteiden muutattamista. Osoitteiden muuttamista on kuitenkin vältettävä, koska se johtaisi merkittävään hakukonenäkyvyyden hetkelliseen heikkenemiseen, johtuen hakukoneiden tarpeesta indexoida kaikki muutetut URL-osoitteet uudelleen (Lincoln 2019). Hyvä sivustorakenne ja sivuhierarkia kuitenkin edesauttavat sivujen indexointia (Sitechecker 2018). SEO-auditoinnin raportissa oli otettu hakukonenäkyvyyden heikkeneminen huomioon, ja sen arvioitiin kestävän useita viikkoja. Tehtävät muutokset olisivat raportin mukaan kuitenkin niin merkittäviä, että ne parantaisivat hakukonenäkyvyyttä pitkällä aikavälillä huomattavasti.

SEO-auditoinnin löydettävyyksianalyysissä oli suositeltu murupolkunavigaation lisäämistä kaikille POP Vakuutuksen verkkosivuille. Hakukonenäkyvyyden optimoimiseksi murupolkunavigaatio lisätään kaikille sivuille, mutta se näytettäisiin käyttäjälle vain tiketissä määritellyissä tapauksissa. Uudistusta koskevassa tiketissä oli määritelty, että navigaation näytettäisiin kaikilla niillä sivuilla, joilla murupolussa on vähintään kaksi aiempaa murua.

3.4 Seurantaviikko 4

Maanantai 4.2.2019

Tämän päivän tavoitteena on saada jokaiselle sivulle sivukohtaisen murupolun tiedot, joiden perusteella voin rakentaa murupolun näkyviin sivulle.

Viikonlopun aikana mielestäni keksin ratkaisun ongelmaan, miksi murupolkunavigaatio käyttäytyi oudosti. Ratkaisin ongelman siirtämällä breadcrumb-muuttujan alustuksen tasoa ylemmäs, sivujen läpi käyvän silmukan sisään, jolloin jokaisella iteraatiolla, eli silmukan kierroksella sivu alustetaan uudelleen alkuperäisillä arvoilla, sen sijaan, että kirjoitettaisiin jo tehtyjen muutosten päälle. Luomani currentPage-muuttuja ei näkynyt oikein, enkä keksinyt sille syytä. En ollut tehnyt sivujen silmuttamiseen muutoksia, joten ongelma hämmensi minua. Tutkin, mitä tietoja sivu sisälsi. Jostain syystä currentPageia ei löytynyt olenkaan. Siirsin tiedon haun koodissa ennen breadcrumbien alustusta, ja tiedot tulivat taas normaalisti läpi. Ongelma siis johtui siitä, että breadcrumb sisälsi jo osan datasta, jota parentPageen oltiin hakemassa, mikä aiheutti virheen. Hetken pohtimisen ja korjausten jälkeen sain kaikki tarvittavat sivukohtaiset tiedot kasaan. Seuraavaksi täytyisi saada vietyä nämä tiedot jokaiselle sivulle.

Breadcrumbs-komponentti ei saanut sille lähetettävää dataa. Minun täytyi sisällyttää staattisille sivuille tuotaviin komponentteihin murupolku-komponentti. Sain tämän ongelman korjattua helposti. Breadcrumbs.hbt-tiedostossa kävin läpi kaikki murut läpi, ja tulostin ne listasta peräkkäin, luoden murupolun. Sivun murupolku näkyi nyt niillä staattisilla sivuilla, joille oli asetettu murupolku Contentfulissa. Murupolkujen otsikoiden tulisi toimia linkkeinä edellisille sivuille. Kohtasin nyt ongelman, että otsikoista koostuvaa listaa ei ole mahdollista käyttää linkkien luonnissa. Otsikoissa on välilyöntejä, mitä ei voi käyttää linkeissä. Otsikoista koostuvan listan luontia on nyt muokattava siten, että se sisältää myös kutakin otsikkoa vastaavan sivun osoitteen nimen. Loin uuden muuttujan nimeltään currentPage, jota on tulisin käyttämään linkeissä. Nykyinen lista täytyi muuttaa objektiksi, jotta jokainen listan otsikko saa avaimekseen sivunsa osoitteen nimen. Objektissa saisin avain-arvo - parilla tuotua sivulle murujen otsikot ja osoitteen nimet.

Taulukon muuttaminen objektiksi jäi kesken, mutta suunnittelen jatkavani niiden luomista huomenna. Päiväni meni odotusteni mukaan. Sain aikaan sen mihin pyrin; murupolut näkyivät oikein. Murupolkunavigaation linkitykset jäivät pahasti kesken. On odotettavissa, että staattisilla sivuilla näkyvät murupolut tulevat menemään hetkellisesti rikki, kun muokkaan tauluja lisää.

Tiistai 5.2.2019

Suunnitelmissani on jatkaa eilen kesken jäänyttä muokkausta murupolkunavigaation tietojen siirtoon. Teen muutoksen objektiin, minkä jälkeen tutkin toimisiko se vielä sivustolla. Tavoitteenani on muuttaa koodia siten, että sivulle tuleva data sisältää kutakin murupolun vaihetta vastaavan avain-arvo -parin, jolla pystyn koostamaan linkit oikein murupolkuun.

Muutin listan objektiksi, ja muokkasin koodia tarpeen mukaan. Lista voidaan lisätä arvoja käyttämällä push-metodia (w3schools.com 2019b), objektiin lisättäessä käytetään kommentia Object.assign. Lista puskettaessa, listan aiempi sisältö säilyy, mutta assign vaati, että lisättävän datan lisäksi mukaan oli tuotava aiempi data, muutoin aiempi data yli- kirjoittuu.

Sain luotua objektin, jonka sisältää sivun nimen, minkä sisällä on toinen objekti, jossa murupolkujen jokaisen tason otsikko ja sitä vastaava osoitteen nimi. Seuraavaksi olisi korjattava silmukka, jossa valitaan kutakin sivua vastaava murupolku jokaiselle sivulle. Työni keskeytyivät hetkeksi pienimuotoisen palaverin vuoksi. Kävimme läpi hakukoneoptimointiin liittyviä valmiita ja tekemättömiä tikettejä, ja niiden aikatauluja.

Palaverin jälkeen jatkoin korjaustöitä. Sain koodin toimimaan, ja onneksi se ei ollut vaikea; piti vain tarkistaa mitä tietoja yritettiin syöttää, ja vaihtaa virheelliset tiedot oikeiksi. Pääsin päivän tavoitteeseeni, ja sain murupolkujen linkit toimimaan. Murupolun korjaaminen jäi vielä kesken, sillä ulkoasussa oli vielä muokattavaa.

Keskiviikko 6.2.2019

Sairaana.

Torstai 7.2.2019

Sairaana.

Perjantai 8.2.2019

Sairaana.

Viikkoanalyysi

Lyhyeksi jääneen työviikon aikana jatkoin murupolkunavigaation työstämistä. Murupolkunavigaation luonti on ollut nyt työn alla jo yli kolme viikkoa. Olin tähän mennessä kirjannut Jiraan murupolkunavigaatio-tiketilte jo yli 90 työtuntia. Olin saanut murupolkunavigaation

koodin nyt niin pitkälle, että staattisille sivuille pystyi Contentfulissa luomaan murupolku-moduulia käyttäen oman murupolun. Käyttäjän tarvitsee lisätä murupolku-moduuliin kyseisen sivun nimi, murupolussa esitettävä otsikko, ja minkä sivun alle nykyinen sivu kuuluu.

Murupolkunavigaation rakentava JavaScript-koodi tuodaan staattisille tuotesivuille Metalsmith-liitännäisellä (engl. plugin). Murupolkunavigaation rakentavan tiedoston nimi on breadcrumb.js, ja se on yksi monista sivulle tuotavista liitännäisistä. Liitännäiset ovat toisistaan riippumattomia kokonaisuuksia, jotka yleensä keskittyvät vain yhden sivuston osan alueen muokkaamiseen tai manipulointiin.

Kaikki Contentfuliin lisätyt staattiset tuotesivut eli landing paget ovat yhdessä files-muuttujassa, joka sisältää massiivisen määrän dataa. Files-muuttuja koostuu lukuisista objekteista, joiden sisällä on lisää lukuisia muita objekteja, joiden seasta on poimittava tarvittavat tiedot. En ollut aiemmin tehnyt kokonaista uutta liitännäistä alusta alkaen, joten kaikki tämä oli minulle uutta. Minun tuli dataa logittamalla löytää Contentfuliin lisätty murupolkuun liittyvä tieto. JavaScript-funktiolla console.log on mahdollista tuoda lokiin mitä tahansa aiemmin määriteltyä dataa (GeeksforGeeks 2019a). Ainut tapa saada console.log-funktiolla tulostettava data näkyviin, on ajaa terminaalissa komentosarja build-static-site.sh, joka rakentaa staattiset sivut uudelleen. Staattisten sivujen rakentaminen kestää noin puoli minuuttia.

Olin luonut breadcrumbBuilder-nimisen funktion, jossa lähdin etsimään murupolku-moduulia files-muuttujan sisältä. Aloitin tekemällä silmukan, jossa käydään läpi kaikki files-muuttujan objektit, jotka tulostin lokiin. Silmukassani kävin läpi jokaisen tiedoston, ja tulostin niiltä kaiken tiedon. Sain todella ison määrän dataa, josta koitin etsiä murupolkua löytämättä sitä. Tarkensin konsolilokituksiani, mikä vaatii, että staattiset sivut rakennetaan aina uudelleen. Jouduin tekemään useita eri lokitulostuksia, ja aloin turhautumaan komentosarjan ajon hitauteen. Muutin staattisten sivut rakentavaa komentosarjaa, siten että kaikkien staattisten sivujen sijaan rakennan vain landing page -sivut. Tällä muutoksella nopeutin komennon ajoa, minkä ansiosta ajo kesti enää kymmenen sekuntia. Lukuisilla uusillaa console.log-hauilla sain lokitulosteeksi kaikki kulloisenkin sivun moduulit. Konsolilokista löysin myös murupolku-moduulin, jonka olin tehnyt yhdelle sivuista. Seuraavana tehtävänäni oli rajata sivujen silmukointia koskemaan vain sivuja, joilla on murupolku-moduuli.

Tein uuden silmukan sivujen silmukoinnin sisään, jossa kävin läpi kaikki moduulit. Uuden silmukan sisällä etsin sitä moduulia, jonka sisältötyypin id oli murupolku. Onnistuin löytämään sivun, jolle olin luonut murupolku-moduulin. Yhtenä suurena ongelmana

silmutuksessa oli se, että kaikilla sivuilla ei ollut kaikkia kenttiä tai moduuleita, joita silmu-
kan sisällä yritin etsiä.

```
if (
  typeof files[file].data !== 'undefined' &&
  typeof files[file].data.fields !== 'undefined' &&
  typeof files[file].data.fields.modules !== 'undefined'
) {
```

Kuva 6. Moduulin etsinnässä käytettäviä ehtolauseita

Koodin täytyi lisätä paljon ehtolauseita (kuva 6), joilla rajasin niitä sivuja pois, joilla haettavaa tietoa ei ollut. Kollegani ehdotti, että ehtolauseessa voisi käyttää lodash-kirjaston has-funktiota. Lodash on Node.js-kirjasto, jolla voi muun muassa muokata objekteja ja tauluja, ja jonka funktioiden avulla voi yksinkertaistaa koodia, ja tehdä siitä helppolukuisempaa (Meenakshi 13.7.2017; Saraogi 2018). Muokkasin aiempaa ehtolauseitani (kuva 6), ja korvasin ehtolauseet has-funktiolla (kuva 7), joka ajaa saman asian.

```
if (_.has(files[file], 'data.fields.modules')) {
```

Kuva 7. Moduulin etsinnän ehtolause has-funktiota käyttäen

Tutustuin hieman lodashiin, josta minulla ei ollut aiempaa kokemusta. Lodashin dokumentaatiota tutkimalla huomasin, että useista monimutkaisesti toteutettavista funktioista on mahdollista yksinkertaistaa todella paljon. Korjasin toisenkin ehtolauseen lodash-muotoon. Minulla oli kestänyt kokonainen päivä etsiä, ja luoda funktio, jolla päästään käsiksi kaikkien tiedostojen kaikkien moduulien joukosta murupolku-moduuliin (kuva 8).

```
function breadcrumbBuilder() {
  const pages = []
  // go through files and locate the ones with murupolku module
  for (let file in files) {
    if (_.has(files[file], 'data.fields.modules')) {
      for (let m in files[file].data.fields.modules) {
        if (_.has(files[file].data.fields.modules[m], 'sys.contentType.sys.id')
          && files[file].data.fields.modules[m].sys.contentType.sys.id == 'murupolku'
        ) {
```

Kuva 8. Murupolku-moduulin hakeminen

Oikean moduulin löydyttyä, moduulista poimitaan siihen Contentfulissa asetetut tiedot, ja tallennetaan listaan objektina. Olin nyt saanut haettua moduulista sivun osoitteen, sivun otsikon näytettäväksi murupolussa, sekä vanhempisivun eli sivun, jonka alla murupolku sijaitsee. Sivun tietojen kerääminen, ja niiden asettaminen objektiin oli helppoa. Tämä ei

vielä riittänyt, sillä Contentfulin tietojen perusteella ei vielä pystynyt päättämään koko linkkipolkua, jota tarvittaisiin linkkipainikkeissa.

Kollegani oli huomannut, että murupolkunavigaatio oli minulle haastava, ja oli luonut työni helpottamiseksi esimerkkinä funktioita, joilla sivujen hierarkian pystyisi luomaan, ja miten sivujen linkit voisi koostaa. Minusta tuntui, että collegani avustus edisti työn valmistumista ainakin viikolla, sillä uskon, että minulla olisi ongelmien ratkaisemisessa mennyt todella pitkään. Pystyin luopumaan osasta luomaani koodia, sillä en saanut asioita toimimaan, vaikkakin luomukseni muistuttivat collegani luomia esimerkkejä. Tein joihinkin collegani esimerkeistä pieniä muutoksia, jotta luomani muuttujat täsmäisivät collegani funktioissa käytettäviä muuttujia. Konsolilokien perusteella koodi näyttäisi luovan murupolon, jossa on linkit, lapsisivut, ja sivuhierarkia.

Seuraavaksi kunkin staattisen sivun dataan tuli saada asetettua tiedot, jotta saan vietyä tarvittavat tiedot murupolon piirtävään Handlebars-tiedostoon. Huomasin, että saadakseni vietyä sivulle tiedot, pystyin uudelleenkäyttämään luomaani sivujen silmukointia, jolla tiedostosta etsitään murupolku-moduuli.

```
// go through all files to provide each landing page containing breadcrumbs
for (let file in files) {
  if (_.has(files[file], 'data.fields.modules')) {
    for (let m in files[file].data.fields.modules) {
      if (_.has(files[file].data.fields.modules[m], 'sys.contentType.sys.id')
        && files[file].data.fields.modules[m].sys.contentType.sys.id == 'murupolku'
      ) {
        const currentPage = files[file].data.fields.modules[m].fields.currentPage
        files[file].data.fields.breadcrumbs = getBreadcrumb(currentPage)
      }
    }
  }
}
```

Kuva 9. Sivun murupoluntietojen asetus sivun dataan

Etsin taas kaikkien tiedostojen joukosta ne tiedostot, joilla oli murupolku-moduuli. Moduulista poimin sille Contentfulissa asetetun nykyisen sivun arvon. Nykyisen sivun arvoa käytetään murupolon rakentamiseen getBreadcrumb-funktiossa. Edellä mainittu funktio koostaa yhteen kaiken tiedostossa käsiteltävän tiedon, ja lisää murupolkuun lapsisivut ja niiden lapsisivut ja näiden URL-osoitteet. Sivun murupoluksi asetin getBreadcrumb-funktion lopputuloksen (kuva 9). Konsoliin lokittamalla varmistin tietojen asettuneen oikein, ja jokainen sivu sai nyt oikean sille kuuluvan murupolkunavigaation datan, jonka perusteella murupolkunavigaation pystyisi piirtämään.

3.5 Seurantaviikko 5

Maanantai 11.2.2019

Päivän suunnitelmissa on aloittaa lukemalla viime viikolla tulleita sähköposteja, ja vastailla niihin, jos tarve vaatii. Sähköpostista on myös tarkistettava kalenterikutsut, ja tulevat palaverit. Tämän jälkeen tarkistan missä vaiheessa murupolut ovat. Kollegani työsti murupolkuja sillä aikaa, kun olin sairauslomalla. Tavoitteenani on saada murupolut näyttämään isolla ruudulla oikein.

Kävin läpi sähköpostit, mutta kaikki uudet viestit olivat suunnattu yleiseen ICT-info -sähköpostiosoitteeseen, eivätkä vaatineet minulta vastausta. Kalenteriin oli merkattu tiistaille tapaaminen, johon osallistunee lisäksi kolme muuta henkilöä. Tapaamisessa käymme läpi, missä tilassa hakukoneoptimointiin liittyvät tiketit ovat.

Murupolkunavigaatio oli tässä vaiheessa vielä tyylittelemätön. Käytännössä sivulle ilmestyi murujen otsikot, jotka linkkasivat otsikkoa vastaavalle sivulle, ja lista lapsisivujen linkeistä. Tein tarvittavat tyylimuutokset ulkoasuun. Unohdin melkein, että tiketin suunnitelmissa murujen välillä oli nuoliosoitimet, jotka täytyi myös saada aseteltua tyylimuutoksilla oikeille paikoilleen. Nuolina käytettiin tiketin mukana toimitettua kuvaa.

Pääsin päivän tavoitteeseen; tein tyylimuutoksia ja sain murupolun murut vierekkäin, ja murujen välille lisättyä nuoliosoitimet. Tekstien asettelussa oli vielä hienosäätämistä, sillä ne eivät asetu aina oikealla korkeudelle.

Tiistai 12.2.2019

Tänään suunnittelen jatkavani aikaansaannoksieni työstöä, ja hienosäädän tyylejä. Kaikki murupolkunavigaation osaset tulevat näkyviin oikein. Seuraava haaste olisi saada mobiilinäkymä vastaamaan suunnitelmaa. Tavoitteenani on saada murupolkunavigaation mobiilinäkymään alasvetovalikko toimimaan ja vastaamaan tiketin suunnitelmakuvaa.

Mobiililaitteilla murupolkunavigaatio ja alasvetovalikko tulevat olemaan koko ruudun levyisiä. Murupolussa näytettäisiin vain nykyisen sivun nimi, ja aiemmat murut olisivat piilotettuna. Tyylimuutosten tekeminen ei olisi vaikeaa. Minun täytyi tehdä murupolkuun avauksen toiminnollisuus JavaScriptillä. Mobiilinäkymään tahdottiin ylimääräinen avauspainike ruudun oikeaan laitaan.

Tein uuden JavaScript-tiedoston, johon työstin skriptejä, joilla lisäisin murupolkuun luokan, mitä tyylimäärittelemällä mobiilinäkymässä navigaatio näkyisi avattuna. Koestin koodini toimivuutta selaimen konsolin avulla. Navigaation avaaminen ei ollut helppoa, mutta onnistuin luomaan koodin, jolla luokka lisätään navigaatiota tai avauspainiketta painettaessa.

En ensin meinannut löytää tapaa, jolla saisin kirjoittamani JavaScript-tiedoston koodin tuotua staattisille sivuille. Etsin, ja löysin tavan, jolla tiedoston koodin sai sivulle. Otin mallia siitä, miten muita JavaScript-tiedostoja oli liitetty sivulle. Koodi tuli sivustolle ja navigaatio aukesi, mutta ei sulkeutunut.

Päivä sujui hyvin ja suunnitellusti ja mutten aivan päässyt tavoitteeseeni. Sain murupolkunavigaation näkymään osittain oikein mobiiliruuduilla, sillä toiminnollisuus ei vielä ollut kunnossa. En onnistunut saamaan uutta avauspainiketta avaamaan ja sulkemaan alasvetovalikkoa oikein. Ongelman korjaaminen jää huomiseksi.

Keskiviikko 13.2.2019

Suunnittelen jatkavani murupolun työstämistä. Tavoitteenani on korjata eilen aloittamani mobiilinäkymän alasvetovalikon avaus ja sulkeminen. Uskon tänään pääseväni päivän tavoitteeseen, sillä koen eilen onnistuneeni itsenäisesti luomaan paljon toimivaakin koodia. Olin tehnyt paljon toimimatonta koodia, mutta onnistuin korjaamaan virheet.

Aloitin päivän korjaamalla eilisen ongelman alasvetovalikon toiminnoissa kirjoittamalla skriptin, jolla kuuntelin, milloin sivulla tehty painallus ei kohdistunut navigaatioon. Muualle kuin murupolkuun kohdistettu klikkaus sulkee valikon.

Tein lisää tyylimuutoksia, joilla asettelin mobiilinäkymän murut oikeille paikoilleen. Olin koodatessani luonut uuden bugin, ja alasvetovalikko ei enää auennut laisinkaan. Ongelma ei johtunut virheellisestä JavaScriptistä, vaan bootstrapin dropdown-luokkaan liittyvistä ominaisuuksista, mitkä piilottivat murupolun lapsisivut.

Mobiilinäkymän navigaatio aukeaa ja sulkeutuu oikein, mutta jostain syystä sivun päivittämisen jälkeen se ei enää sulkeutunutkaan oikein. Hetken päästä alasvetovalikko taas toimi. Koitin etsiä syytä ongelmalle, mutten sille keksinyt, mistä ongelma olisi johtunut. Pohdiskelin, olisiko koodissani virhe, vai oliko ongelma peräisin selaimen konsoliin tekemistä kokeiluista, joilla manipuloisin sivua. Alkuperäinen koodi toimi koko ajan, mutta sivut olivat todennäköisesti rakentuneet jossain vaiheessa väärin. Ongelma jäänee mysteeriksi,

sillä edes kolleganikaan ei keksinyt, mistä nämä ongelmani olisivat johtuneet. Kollegani vitsaili, että ehkäpä planeetat olivat nyt oikeassa asennossa, ja koodi toimi.

Koodissa lisäsin murupolkunavigaatiolle opened-luokan addClass- ja removeClass-meto-
deilla, joilla lisätään ja poistetaan luokkia. Syystä tai toisesta removeClass ei toiminut odo-
tetusti. Sain luokan poiston toimimaan kollegan avulla, kun katsoi bootstrapin dokumen-
teista, että mitä kaikkea taustalla tapahtuu, kun dropdown avataan. Alasvetovalikko au-
keaa ja sulkeutuu taas oikein mobiiliruuduilla. Esiin nousi uusi ongelma: isolla ruutukoolla
navigaatio ei enää auennut.

Pääsin päivän tavoitteeseeni, sillä sain mobiilinäkymän murupolkunavigaation näkymään
ja toimimaan oikein. Harmikseni olin aiheuttanut samalla uuden bugin, minkä johdosta
isolla ruudulla eli työpöytänäkymässä murupolku ei enää avautunut. En ollut suunnitellut
rikkovani työpöytänäkymän navigaatiota mobiiliruudun korjauksia tehdessäni, mutta muu-
toin arvioisin päivän sujuneen suunnitellusti.

Torstai 14.2.2019

Tänään jatkan taas murupolun työstämistä. Toivoisin tänään saavani navigaation toimi-
maan oikein työpöytänäkymässä, mutta edellisten päivien virheistä oppineena pelkään,
että tulen vielä kohtaamaan paljon ongelmia. Päivän tavoitteenani on kuitenkin saada työ-
pöytänäkymän murupolkunavigaatio avautumaan ja sulkeutumaan oikein.

Aloitin päivän tekemällä tyylien muokkauksia, ja mielestäni tyylit ovat nyt viimeistelty. Te-
kemäni koodimuutokset navigaation avaukseen koskivat vain mobiilinäkymän avauspaini-
ketta. Huomasin tarvitsevani mobiilinavigaatioissa lisättävää opened-luokkaa myös työpöy-
tänäkymän murupolussa.

Sain myös työpöytänäkymässä lisättyä murupolkua klikatessa sille opened-luokan, mutta
se ei toistu, kun navigaatiota yrittää avata useamman kerran. Tämän korjaaminen jäi kes-
ken.

En päässyt tänään päivän tavoitteeseen. Olin tietoisesti asettanut päivälle mielestäni kor-
kean tavoitteen, enkä lopulta saavuttanut sitä. En silti koe epäonnistuneeni, vaan päinvas-
toin koen onnistuneeni, sillä opin taas paljon. Opin muun muassa kuinka paljon työtä yk-
sinkertaiselta vaikuttavan valikon avaamiseen tarvitaan koodia ja ennen kaikkea pikku-
tarkkuutta.

Perjantai 15.2.2019

Suunnitelmissa on jatkaa murupolku tänään loppuun ja tavoitteenani on saada murupolku toimimaan niin työpöytä- kuin mobiilinäkömässä. Olin saanut murupolkunavigaation jo niin valmiiksi, että uskon saavani kaiken toimimaan.

Uutena ongelmana oli, että murupolkunavigaatio aukeaa kerran, mutta ei toiste. Pohdiskelein, olinko tehnyt jotain väärin. Päätin hetken tutkia koodiani, ja huomasin, että olisin voinut tehdä asiat paremmin. Tein Google-hakuja, ja löysin ratkaisun, jolla sain kuunneltua alasvetovalikon aukeamista, jolla lisäsin opened-luokan. Tein saman muutoksen JavaScriptiin myös valikon sulkeutumiselle. Sain koodin toimimaan, ja navigaatio aukesi oikein.

Seuraavana haasteena oli korjata sivun päälle tuleva peite-elementti, jolla on luokka breadcrumbOverlay. Peite-elementin avulla sivun muut osiot piilotetaan navigaation ollessa auki. Koitin erilaisilla tyylimuutoksilla saada peitteen asettumaan linkkien alle. Etsin Internetistä vinkkejä ja ohjeita, ja sain lopulta murupolkunavigaation näkymään ja toimimaan oikein jokaisella ruutukoolla.

Koen oppineeni ohjelmoimaan JavaScript-koodilla paljon aikaisempaa paremmin. Minulla ei enää mene yhtä paljon aikaa miettiessä, mitä minun tulee kirjoittaa, saadakseni tehtyä johonkin elementtiin muutoksia.

Viikkoanalyysi

Tein koko seurantaviikon murupolkunavigaatiota, mitä olin työstänyt jo kolmena edellisenä viikkona. Navigaatio oli viikon alussa vielä todella keskeneräinen. Viikon edetessä kohtasin lukuisia haasteita valikoiden avaamisissa, jotka sain ratkaistua. Murupolkunavigaatio on ollut todella haastava kokonaisuus, joka viimein näyttää lähestyvän valmista.

Viikon alussa murupolun murut eli linkkielementit täytyi asettaa toistensa vierelle, sen sijaan, että ne olisivat allekkain. Murupoluille täytyi luoda paljon uusia tyylejä. Navigaatiolle täytyi asettaa maksimikorkeus, jonka sisällä murut asetettiin vierekkäin. Sain kaikki murupolun osat näkyviin oikein muutamilla tyyliuokkauksilla. Tyylien luomista helpotti se, että minun ei tarvinnut välittää olemassa olevista tyyleistä, sillä niitä ei juurikaan ollut. Tehtyäni tyylimuutokset työpöytänäkömän ulkoasuun, ryhdyin miettimään, mitä seuraavaksi tulisi tehdä.

Murupolkua ei vielä ollut lisätty kaikille Contentfulista löytyville tuotesivuille, mutta en kokenut sen olevan vielä tärkeää, vaan keskityin koodin työstöön. Pystyisin lisäämään murupolut helposti Contentfulin myöhemmin.

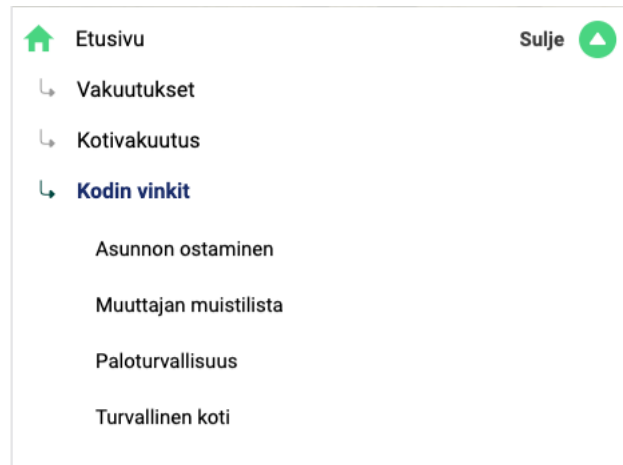
Työpöytänäkymän ulkoasun jälkeen aloin työstämään mobiilinäkymää. Murupolkujen tikeissä vaadittiin, että mobiilinäkymän valikko tulisi saada aukeamaan otsikkoa painamalla, tai avauspainikkeesta. Mobiilinäkymässä avauspainike tulisi sijoittaa murupolun oikeaan laitaan, mikä vaatisi erillisen avauspainikkeen. Tein murupolkunavigaation rakentamaan breadcrumbs.hbt-tiedostoon uuden avauspainikkeen mobiilinäkymää varten. Muokkasin painikkeen oikealle paikalleen. Molempia painikkeita ei tule esittää samanaikaisesti. Työpöytänäkymässä mobiilinavigaation avauspainike on tyylimäärityillä poistettu näkyvistä, kun taas ruudun leveyden ollessa tablettikokoa pienempi, alkuperäinen avauspainike on poistettu näkyvistä. Seuraavaksi oli ohjelmoitava valikon avauksen toiminnollisuus uudesta mobiilinäkymän painikkeesta.

Tiesin jo ennen ensimmäisenkään skriptin kirjoittamista, että olisi hankala saada murupolut aukeamaan ja sulkeutumaan otsikkoa tai avauspainiketta painamalla, sillä en ollut aiemmin tehnyt vastaavaa. Haasteena oli löytää keino, jolla saisin kaiken tämän tehtyä. Mietiskelin tehtävän haasteita, ja tutkin Internetistä mahdollisia ratkaisumalleja. Sain ideoita, joiden avulla jatkaa eteenpäin. Päätin tehdä siten, että JavaScriptillä lisätään navigaatiolle opened-niminen luokka, minkä perusteella murupolun aiemmat vaiheet näytettäisiin aukinaisena. Ajatuksenani oli, että murupolun lapsisivut asetetaan mobiilinäkymässä CSS-tyylimäärityksillä oletusarvoisesti piiloon. Nämä tyylimääritykset yliajettaisiin toisilla tyylimäärityksillä, kun navigaatiolla on opened-luokka, jolloin navigaatio näkyisi avattuna.

Koodistamme löysin jQueryn toggleClass-funktion, jolla on mahdollista lisätä ja poistaa luokkia (Motto 2.2.2013), mikä soveltui tarpeeseeni täydellisesti. Kirjoitin funktion, jolla lisättäisiin ja poistettaisiin luokka. Minä ja kollegani olimme tyytyväisiä ratkaisumalliini, mutta kollegani esitti toiveen, että toggleClass-funktion muuttaisi addClass-funktioksi, jolla lisätään luokka, ja removeClass-funktio, jolla poistetaan luokka (Motto 2.2.2013). Kollegani toivoi muutosta, jotta koodi olisi helpommin luettavaa. Muutoksen myötä koodissa oli nyt eri funktio luokan lisäämiselle ja luokan poistamiselle, sen sijaan että yksi funktio tekisi molemmat tehtävät. Selaimen kehitystyökaluista löytyvään konsolin avulla pystyin koestamaan koodin käyttäytymistä sivulla. Sain koodin toimimaan konsolilla, ja kirjoitin tarvittavat skriptit JavaScript-tiedostoon.

Löysin koodista esimerkkejä, joilla sain tekemäni koodimuutokset vietyä staattisille tuotesivuille. Rakensin sivut uudelleen ja koestin koodia omassa kehitysympäristössä. Luomani

JavaScript toimi odotetusti, ja mobiilinäkymässä murupolkua tai sen avauspainiketta klikkaamalla murupolkunavigaatiolle tuli opened-luokka. Loin uudet tyylit, joilla navigaation vanhempisivut esitetään avattuna allekkain, minkä alla näkyvät lapsisivut allekkain (kuva 10).



Kuva 10. Murupolku avattuna mobiilinäkymässä

Saatuani mobiilinavigaation näyttämään oikealta ja aukeamaan oikein, huomasin uutena ongelmana, että työpöytänäkymässä murupolku ei ensimmäisen avaamisen ja sulkemisen jälkeen enää avautunut uudelleen. Tarkastelin selaimen kehitystyökaluilla elementtejä, ja pystyin varmistamaan, että koodini näytti toimivan oikein ensimmäisillä klikkauksilla, ja lapsisivut olivat olemassa.

Mietin, lieneekö työpöytänäkymän ongelmana koodissa käytetty on-metodi. On-metodilla on mahdollista kuunnella klikkauksia, jotka kohdistuivat murupolkunavigaatioon (jQuery 2019a). Koodissa murupolkunavigaatiolle lisättiin tai siltä poistettiin showable-luokka, kun murupolkunavigaatiota oli klikattu. Tutkin koodia, ja sain pääteltyä, ettei ongelma ollut metodissa. Koodiin piti vain erikseen lisätä kuuntelu sille, kun dropdown-elementti avautuu. Tarkistin bootstrapin dokumentaatiosta, mitä tapahtumia koodissa laukeaa, kun alasvetovalikko avataan. Dokumentaatiosta (Bootstrap 2019a) löysin, että kun alasvetovalikko on avattu, taustalla laukeaa tapahtuma (engl. event) nimeltään `shown.bs.dropdown`. HTML-sivujen manipulointi tapahtuu `event`:llä eli tapahtumalla, joita laukeaa aina kun JavaScript-koodi tai selain manipuloi eli muokkaa tai käsittelee sivua (TutorialsPoint 2019). Korjasin on-metodin kuuntelemaan navigaatioon kohdistuvan klikkauksen sijasta `shown.bs.dropdown` -tapahtumaa. Tein vastaavan korjauksen myös navigaation sulkemiselle, jossa kuunnellaan navigaation klikkauksien sijaan `hidden.bs.dropdown` -tapahtumaa, joka puolestaan laukeaa kun bootstrapin dropdown-elementti suljetaan.

Koodin selkeyttämiseksi asetin murupolkunavigaatio-elementin parent-nimiseen muuttu-
jaan, ja lapsisivut sisältävän dropdown-elementin asetin dropdown-nimiseen muuttujaan.

```
if (!parent.hasClass('opened') && !dropdown.hasClass('open')) {  
    parent.addClass('opened');  
    dropdown.find('.dropdown-toggle').dropdown('toggle');  
}
```

Kuva 11. Murupolkunavigaation avaavan funktion ehtolause

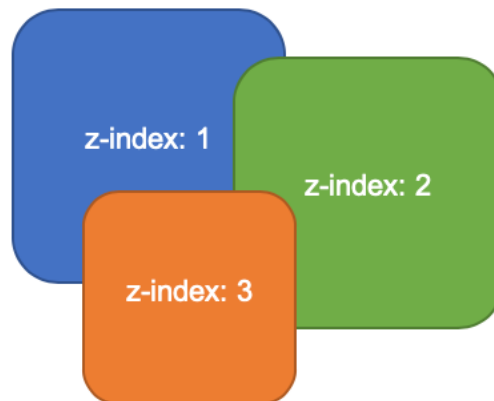
Loin koodiin ehtolauseen (kuva 11), jolla tarkistetaan, että mikäli murupolkunavigaatiolla ei vielä ole opened-luokkaa, eikä dropdown-elementti ole auki, murupolkunavigaatiolle lisätään opened-luokka, ja dropdown-elementti avataan. Dropdown('toggle') -funktiolla voidaan kutsua dropdown-elementin avausta (Bootstrap 2019a). Loin koodiin vastaavan, mutta käänteisen ehtolauseen navigaation sulkemiselle, jossa tarkistetaan, että opened-luokka löytyy, ja dropdown-elementti on auki, ennen kuin navigaatio voidaan sulkea.

Alasvetovalikko aukesi nyt oikein, ja koestin murupolkua muilla sivuilla. Mobiilinnavigaatio ei kuitenkaan sulkeutunut kaikilla sivuilla oikein. Huomasin ongelman koodissa, että siinä etsittiin dropdown-elementtiä, jota ei ollut olemassa. Ongelma esiintyi silloin, jos sivu oli murupolkunsa viimeinen sivu, jolloin sivulla ei ollut lapsisivuja. Olin luonut koodiin muiden ehtolauseiden lisäksi myös ehdon, jolla alasvetovalikko suljetaan, mikäli se oli määrittelemätön, eli tyhjä. Tyhjää alasvetovalikkoa ei olisi tarkoitus olla mahdollista avata työpöytäkymässä, mutta mobiiliruudulla valikko oli saatava auki, vaikkei sivulla ollut lapsisivuja. Olin siinä luulossa, että valikon ollessa tyhjä se on myös määrittelemätön, mutta valikko itseasiassa ei ollut määrittelemätön, vaan sen pituus oli 0. Pituudella tarkoitetaan elementin sisältämien elementtien määrää (jQuery 2019b). Korjasin ehtolauseen, jotta mobiilinnavigaatio aukesi oikein. Yhä oli ongelma, että valikko ei sulkeudu työpöydän näky-
mässä.

Paikallistin ongelman breadcrumbOverlay-nimiseen elementtiin, joka nousee esiin navigaation avautuessa. Olin itse luonut kyseen omaisen elementin sivupohjaan, ja sen tarkoituksena on peittää sivun muut osiot haalealla peitevärillä, kun navigaatio on avattuna. Päälle nouseva peite-elementti asettui väärin, ja se piilotti alleen linkkielementit, estäen niiden painamisen. Todellisuudessa murupolku siis avautuu myös isolla ruudulla, mutta se jää piiloon. Ongelma itsessään ei ole virhe, vaan ominaisuus, joka minun tapauksessani aiheutti ongelman. Mietiskelin, että navigaatio sulkeutuisi, jos valikon taakse tuleva breadcrumbOverlay-elementin tyylimääritteli piiloon käyttäen "display: none", joilla se poistettaisiin näkyvistä, tai "visibility: hidden", jolla se piilotettaisiin näkyvistä (Eygi

7.12.2018). Peite-elementin poistaminen tai piilottaminen ei kuitenkaan olisi oikea ratkaisu.

Yritin korjata ongelmaani antamalla valikolle tyylitiedostossa z-index arvoa, jolla siirtäisin valikon taustan taaemmaksi, jotta linkit olisivat painettavissa. Päällekkäisten elementtien sijaintia ”syvyysuunnassa” on mahdollista tyylimääritellä z-index -arvolla (Cope 2011a). Syvyysuunnalla tarkoitetaan elementtien asettamista toistensa päälle, ikään kuin toiset elementeistä olisivat lähempänä ruutua (kuvio 4).



Kuvio 4. Elementit asetettuina päällekkäin käyttäen z-indexiä

Yritin asetella elementit paikoilleen antamalla niille z-index arvoja, mutten saanut alasvetovalikon linkkejä klikattavaksi. Irtauduin hetkeksi työstäni miettimään, mitä tehdä seuraavaksi.

Etsin Internetistä, miten z-index oikeasti toimii. Löysin kattavasti tietoa Philip Waltonin (2013) verkkosivujen z-index -artikkelista, josta opin, että jokaisella elementillä on oma ”pinoamiskonteksti” (engl. stacking context), jossa on yksi juurielementti. Juurielementin sisällä olevat elementit ovat rajattu yhteen kontekstiin. BreadcrumbOverlay-elementti sijaitsee sivurakenteessa hierarkkisesti viimeisen linkin alla. Ongelmana oli se, että annoin kaikille valikon muruille, eli linkeille z-index -arvon 1000, jolloin myös viimeinen linkki sai arvon 1000. Koska overlay-elementti sijaitsee samassa pinoamiskontekstissa, sillä myös oli teoriassa z-index 1000, minkä lisäksi sillä oli entuudestaan oma arvo 990, eli yhteensä 1990. Artikkelista kävi ilmi, että minun ei ollut mahdollista saada siirrettyä breadcrumbOverlay-elementtiä linkkipainikkeiden taakse sen nykyisessä kontekstissa.

Muokkasin murupolkunavigaation rakennetta, ja siirsin breadcrumbOverlay-elementin pois murupolkunavigaation sisältä, ja asetin sen hierarkkisesti samalle tasolle murupolkunavigaation kanssa. Näin sivun peite on navigaation rinnalla omassa pinoamiskontekstissaan,

ja peite olisi mahdollista asettaa ruudulla murupolun linkkien taakse. Asetin murupolulle z-index -arvoksi 2, ja peitteelle arvoksi 1, jolloin pienempiarvoinen peite-elementti jää murupolkujen taakse.

Murupolkunavigaatio oli nyt lähes valmis. Kirjoitin viikon aikana paljon JavaScript-koodia, ja koen, että olen kehittänyt koodin kirjoittamisessa, ja erityisen paljon käyttöliittymän manipuloinnissa. Kehitystyö ei mielestäni aina tuntunut etenevän, sillä navigaation toiminnollisuuksien luomisessa, ja korjaamisessa oli toisinaan suuria vaikeuksia. Vaikeudet johtuivat vielä vähäisestä JavaScript-osaamisestani. Usein kävi niin, että onnistuttuani korjaamaan jonkin navigaation osan, aiheutin uuden virheen, minkä jälkeen murupolkunavigaatio ei enää toiminutkaan. Hyväksyn vastoinikäymisten ja ongelmien olevan osa ohjelmistokehitystä. Enkä koe epäonnistuneeni, vaan pidän epäonnistumisiani oppimiskokemuksina. Uskon, että ajan saatossa ja taitojeni kehittyessä onnistun luomaan koodia, ilman että samaan aikaan rikkoisin jotain muuta.

3.6 Seurantaviikko 6

Maanantai 18.2.2019

Suunnitelmissani on koestaa murupolkunavigaation toimivuutta ja viimeistellä tyyliä. Tavoitteenani on viedä nyt jo lähes valmis murupolkunavigaatio kehitys-, eli develop-haaraan, jossa jatkan tyylien viimeistelyä. Uskoisin saavani tyyliä viimeisteltyä lopulliseen muotoonsa. Päästäkseni päivän tavoitteeseen, minun täytyy ensin varmistaa, että navigaatio toimii, minkä jälkeen voin siirtää muutokset kehityshaaraan. Uskoisin saavani tyyliä viimeisteltyä lopulliseen muotoonsa.

Päivän aluksi tarkistin vielä, että murupolkunavigaatio todella toimi oikein. Työstin tähän asti murupolkunavigaatio erillisessä feature/breadcrumbs -haarassa, ja nyt tehtävänäni on siirtää muutokset pääasialliseen kehityshaaraan eli develop-haaraan.

Jouduin pyytämään kollegaltani apua, jotta tekisi jotain väärin, ja pahimmassa tapauksessa rikkoisi jotain. Tallensin eli kommitoin muutokseni "git commit" -komennolla, ja ajoin vaadittavat komennot kollegani avustamana. Olin liian pikainen ajamaan komentoja, ja aiheutin virheen, jonka sain onneksi korjattua. Kollegani auttoi minua aiheuttamani ongelman korjaamisessa, ja lopulta sain tuotua murupolun muutokset develop-haaraan. Seuraavaksi laitoin testiserverin rakentamaan koodin uudelleen, jotta muutokset tulisivat näkyviin.

Siirryin selaimella testipalvelimelle, ja huomasin, että muutokset eivät vielä tulleet näkyviin, joten päivittelin sivua ja kävin muilla sivuilla, joille murupolut oli lisätty. En löytänyt murupolkuja, joten rakensin uudelleen staattisten sivujen tyylitiedostot, mutta muutokset eivät vieläkään tulleet näkyviin. En tiennyt johtuiko ongelma siitä, että muut sivut olivat olleet ajossa, kun laitoin sivujen uudelleenrakennuksen ajoon. Seuraavaksi päätin ajaa uudelleen staattisten sivujen rakentamisen. Myös tämä sivujen uudelleenrakennus oli jo työjonossa rakentumaan. Ajon valmistuttua murupolkunavigaatio näkyi testiserverillä.

Testiserverillä murupolkunavigaatioissa ensimmäisessä askeleessa on kotipainike, mutta kuva ei ollut näkyvissä. Laitoin testipalvelimen sivut vielä kertaalleen rakentumaan.

Murupolkujen tyylit toimivat lukuun ottamatta kotipainiketta, joka ei toistaiseksi tuntemattomasta syystä näy sivustolla. Päiväni sujui suunnitellusti murupolkunavigaation testiserverille viennissä. Päivän aikana tapahtui odottamaton virhe, kun muutoksia develop-haaraan tuodessani ajoin väärän komennon. Virheen korjaaminen vei aikaa muulta kehitystyöltä. En päässyt päivälle asettamaani tavoitteeseen, sillä murupoluista puuttui etusivulle ohjaava kotipainike.

Tiistai 19.2.2019

Päivän alun suunnittelin käyttäväni murupolkujen tyylikorjausten parissa, sillä minun oli saatava kotipainike näkyviin. Suunnitelmissani on työstää tehtävää liittyen sivun ylänavigaatioon saatuaani murupolon tyylit korjattua. Olin aiemmin käyttöliittymäsuunnittelijalta kuullut, että ylänavigaatioissa oli vielä joitain avonaisia asioita. Päivän tavoitteena on saattaa ylänavigaation muutokset alkuun, ja kenties saada ne valmiiksi. Ennen kuin voisin aloittaa työstämisen. Minun olisi keskustella tehtävistä muutoksista käyttöliittymäsuunnittelijan ja tuoteomistajan kanssa, jotta tekisi väärää tai turhaa työtä.

Aloitin päivän selvittämällä, mistä murupolkujen kotipainikkeessa ongelma johtui. Paikansin ongelman tutkimalla tyylitiedostoa. Murupolon kotipainike ei näkynyt testiserverillä, koska kuvan nimessä oli kirjoitusvirhe. Olin kirjoittanut yhden kirjaimista isolla, kun todellisuudessa se oli kirjoitettu pienellä kirjaimella. Koin ärsyttäväksi sen, että ongelma johtui niinkin pienestä asiasta, mutta hyvää asiassa oli se, että ongelma selvisi helposti.

Murupolkunavigaatioissa oli noussut esiin uusi ongelma ulkoasussa, sillä breadcrumbOverlay-peite ei noussut tarpeeksi korkealle. Ongelma saattoi osittain johtua tekemistäni tyyli muutoksista, sillä olin muokannut murupolon elementtien korkeuksia. Tutkin peitteen asettelun määrittelevää koodia, joka nykyisellään ei enää näyttänyt toimivan. Työstin

ongelman parissa pitkään, kunnes lopulta löysin ratkaisun, ja peite-elementti asettui oikealle korkeudelle.

Päiväni ei sujunut suunnitellusti, enkä päässyt päivälle asetettuun tavoitteeseen, sillä sivun peite-elementti oli haastava korjata, minkä johdosta en ehtinyt työstämään ylänavigaatiota. Peite-elementin korjaamisessa oli hankalaa löytää oikea tapa määritellä sen paikka sivulla. En ehtinyt keskustelemaan käyttöliittymäsuunnittelijan kanssa ylänavigaatioon tehtävistä muutoksista. Tulisin työstämään ylänavigaatiota huomenna.

Keskiviikko 20.2.2019

Tänään selvitän, mitä muita hakukoneoptimointiin liittyviä tikettejä ylänavigaation korjauksen lisäksi on tehtävä. Ylänavigaation muutokset olivat yksi hakukoneoptimoinnin tikeistä, joista oli puhetta aiemmissa aiheeseen liittyvissä palaverissa. Tänään on viikoittainen sprint-palaverimme, jossa alustavasti suunnittelen ottavani työstöön lisää hakukoneoptimointiin liittyviä tikettejä. Tämän päivän suunnittelen käyttäväni ylänavigaatiota työssä, ja tavoitteenani on saada se valmiiksi päivän loppuun mennessä.

Aloitin päivän tutkimalla ylänavigaatiosta luotua tikettiä. Keskustelin tehtävistä muutoksista käyttöliittymäsuunnittelijan kanssa, joka oli tehnyt tikettiin suunnitelman uudesta ulkoasusta. Tarkoituksena on muokata ylänavigaatiosta dropdown-valikko pois, sekä poistaa työpöytäkymästä Oma Palvelu -linkki.

Sprint-palaverissa keskustelimme edellisen sprintin onnistumisista ja tulevan sprintin tikeistä. Otin itselleni tehtäväksi kolme uutta hakukoneoptimointiin liittyvää tikettiä. Ennen kuin ryhtyisin työstämään uusia tikettejä, minun oli saatettava päättyneen sprintin tehtävät loppuun.

Palaverin jälkeen viimeistelin ylänavigaation ulkoasun tyylimäärityksiä. Tehtävänannon suunnitelmakuvissa ylänavigaation asettelu tahdottiin siirtää enemmän sivun oikeaan laitaan, kuin mitä se oli aiemmin ollut. Tyylimuokkausten jälkeen ryhdyin muuttamaan sivuston URL-osoitteita niiden oikeisiin muotoihin. Osoitteiden korjaus ei ole vaikeaa, mutta se on työlästä ja hidasta. Loput päivästä muokkasinkin linkkien osoitteita.

Saavutin päivän tavoitteen, ja sain ylänavigaation valmiiksi. Ehdin myös aloittamaan uutta tehtävää, jossa sisäisten linkkien osoitteita tulisi muuttaa. Pääsin päivän tavoitteeseen, ja koen päivän sujuneen odotetusti. Ylänavigaation korjaus oli sopivan haastava tehtävä, sillä opin uutta sivustomme rakentumisesta, mutta minun ei ollut vaikeaa toteuttaa

muutoksia. Ehdin myös ottaa työn alle uuden hakukoneoptimointiin liittyvän työtehtävän, ja pääsin siinä hyvin alkuun.

Torstai 21.2.2019

Tänään jatkan eilen aloittamaani sisäisten linkkien korjausta, ja tavoitteenani on saada mahdollisimman moni linkki korjattua. Tälle päivälle minulla ei ole suunnitteilla tehdä mitään muuta, sillä linkkien URL-osoitteiden korjaaminen on hidasta.

Aloitin aamun käymällä läpi sivuja yksi kerrallaan, etsien sivuilta linkkielementtejä. Linkeistä tutkin URL-osoitetta, ja korjasin virheelliset oikeaan muotoon. Sivustollamme on paljon sivuja, joita käydä läpi, ja mietin, voisinko saada konsoliin tulostettua kaikki ne sivut, joilla oli virheellinen linkki. Yritin saada lokitettua koodista kaikki ne linkkielementit, joissa URL-osoite oli väärä siinä onnistumatta.

Jatkoin lounaan jälkeen linkkien korjaamista käymällä sivut läpi yksitellen etsien virheellisiä URL-osoitteita. Tein muutostöitä aina päivän loppuun saakka.

Päivä meni suunnitellusti linkejä muokaten, enkä kohdannut haasteita työssäni.

Perjantai 22.2.2019

Tälle päivälle suunnittelen URL-osoitteiden korjaamista ja aloittan alanavigaation korjaamisen, mikä on yksi tähän sprinttiin valitsemistani tiketeistä. URL-osoitteiden korjaamisessa tulee menemään vielä pitkään, sillä niiden työstäminen on todella hidasta. Linkkien korjaaminen ei ole kaikkein suurin prioriteetti, minkä vuoksi en tule käyttämään sen parissa yhtä paljon aikaa kuin eilen olin käyttänyt. Tavoitteenani on tänään saada etsittyä ja korjattua loput virheelliset URL-osoitteet, ja aloittaa alanavigaation muutostyöt.

Nopeuttaakseni URL-osoitteiden etsintää, yritin tänään uudelleen saada konsolilokitettua kaikki ne sivut, joilla osoite oli virheellinen. Kysyin kollegaltani neuvoa, ja hän osasi antaa vinkkinä grep-komennon, jolla pystyin etsimään dataa kaikista rakennetuista sivuista. Komennossa käytin regular expression -hakukaavaa, millä tein hakuun rajauksen, johon tuloksen on täsmättävä (Regular-Expressions.info 2019). Ruudulle tulostui vain ne tulokset, joissa osoite sisälsi sanan laske-hinta, mutta ei sisältänyt sanaa perustiedot. Lopullisessa tulosteessa oli ne linkit, joissa oli virheellinen osoite, ja sen sivun nimi millä linkki sijaitsee. Kaikki löytyneet osoitteet eivät olleet virheellisiä, sillä kaikkien ostoputkien URL-osoitteissa ei ole perustiedot-loppuosaa.

En korjannut kaikkia linkkejä, vaan siirryin sen sijaan työstämään alanavigaatioon liittyvä tehtävää. Tutkin tikettiä ja aloitin korjausten tekemisen. Tein lukuisia muutoksia Contentfuliin. Työ jäi kuitenkin kesken, mutta työ jatkan alanavigaation ensi viikon maanantaina loppuun.

Viikkoanalyysi

Kuudennella seurantaviikolla pääsin viimein murupolkunavigaation lisäksi työstämään muitakin tehtäviä. Viikon aikana tein ylänavigaatioon muutoksia, muokkasin sisäisiä linkkejä, sekä aloitin sivuston alanavigaation työstämisen. Pääsin aloittamaan muut työtehtävät saatuaani murupolkunavigaation lähes valmiiksi, ja vietyä muutokset develop-haaraan. Muutosten tuominen feature-haarasta develop-haaraan oli jotain, mistä minulla ei ollut aiempaa kokemusta, ja huomasin oppineeni paljon. Parhaiten mieleen viikosta jäi itselleni aiheuttamat ongelmat, kun etenin tekemällä asioita liian hätäisesti viedessäni murupolkunavigaatiota develop-haaraan.

Viikkoni alkoi saattamalla murupolkunavigaatio testiserverille, eli develop-haaraan. Git-versionhallinnassa työstämme pääasiallisesti develop-haarassa. Gitissä voi luoda uusia haaroja, jotka voidaan nimetä tahtomallaan tavalla. Haarasta voi vaihtaa toiseen, lähes milloin tahansa, mutta Git ei salli vaihtaa haaraa, mikäli haarassa on muutoksia, jotka eivät ole joko piilotettu tai kommitoitu (Klughertz 10.2.2016). Murupolkunavigaation työstäminen omassa feature-haarassaan mahdollisti muutosten jakamisen muiden kanssa ilman, että muutokset vaikuttivat muiden työhön tai kehityshaaran toimintaan (Software Engineering 2014). Juuri feature-haaran ansiosta kollegani oli pystynyt työstämään murupolkua minun ollessa sairaana neljännellä seurantaviikolla.

Minulla ei juurikaan ollut kokemusta, kuinka tuoda muutokset haarasta toiseen. Minun täytyi tehdä Git-komentoja, joista en ollut aivan varma, joten pyysin kollegaltani apua. Ensin vielä työskennellessäni feature/breadcrumbs-haarassa puskin muutokset omasta ympäristöstäni versionhallintaan. Sen jälkeen ajoin komennon "git checkout develop", jolla siirryin develop- eli kehityshaaraan. Seuraavaksi ajoin komentosarjan "git merge --squash feature/breadcrumbs". Komennolla "merge" voi yhdistää muutoksia eri haaroista (Atlassian 2019a), lisäargumentti "--squash" yhdistää kaikki yhdistettävän haaran kommitit yhdeksi kommitiksi (Linnerud 3.9.2018), ja lopussa "feature/breadcrumbs" kertoo, mikä haara nykyisen haaran kanssa tahdotaan yhdistää. Näin sain kaikki feature/breadcrumbs -haaran muutokset tuotua käsittelyyn yhdistettäväksi develop-haaran kanssa.

Pääsin kollegani avustuksella tilanteeseen, jossa olin saanut tuotua murupolun muutokset feature/breadcrumbs -haarasta vietäväksi develop-haaraan. Muutokset eivät vielä olleet puskettu kehityshaaraan. Ajoin komennon "git fetch", mitä ei olisi tarvinnut ajaa, ja minkä johdosta en kyennyt tehdä push-komentoa. Kollegani neuvoilla ajoin reset-komennon, jolla pääsin edelliseen develop-haaran kommittiin, tilaan ennen merge-komentoa. Tällöin sain tilan näyttämään oikein, eli tyhjää. Ajoin komennon "git pull", jolla hain muutokset takaisin ja "git status" -komennolla tarkistin, että sain feature/breadcrumbs:n muutokset takaisin käsittelyyn. Olin taas päässyt tilaan, jossa pystyisin viemään muutokset develop-haaraan. Puskin muutokset develop-haaraan, ja laitoin testiserverin rakentamaan koodin uudelleen, jotta muutokset tulisivat näkyviin.

En ensin löytänyt murupolkuja, joten rakensin staattisten sivut uudelleen, mutta muutokset eivät vieläkään tulleet näkyviin. En ollut varma, mutta uskoin ongelman johtuneen siitä, että olin ajanut sivujen päivitykset väärässä järjestyksessä, kun laitoin sivujen uudelleenrakennuksen ajoin. Rakensin sivut vielä kerran uudelleen, ja muutokset tulivat näkyviin.

Jatkoin murupolkujen parissa, ja uutena ongelmana oli, että breadcrumbOverlay-peite asettui sivulla liian alas. Koodissa peitteen asemointi tapahtui aina, kun navigaatio avattiin. Koodissamme peitteen asettelu tapahtui laskemalla kaikkien peitteen yläpuolella sijaitsevien elementin korkeuksien summa. Tutkin selaimen kehitystyökalulla elementtejä, ja niiden ominaisuuksia, kuten korkeutta ja sijaintia sivulla. Ongelman korjaaminen vaikutti alkuun vaikealta, mutta keksin mahdollisen ratkaisun käyttämällä jQueryn offset-metodia, joka palauttaa valitun elementin koordinaatit sivulla (GeeksforGeeks 2019b). Muokkasin laskutoimitusta ja koodia, jolla peite-elementin sijainti asetetaan.

```
$('.breadcrumbOverlay').css('top', (crumbPosition.top + height) + 'px');
```

Kuva 12. Peite-elementin sijainnin asettaminen

Peitteen oikea paikka löytyi hakemalla murupolkunavigaation etäisyys ruudun ylälaidasta eli "crumbPosition.top", mihin lisättiin murupolun oma korkeus eli "height" (kuva 12). Tyylimäärittelyllä "top" kerrotaan elementin ylälaidan sijainti (Tech on the Net 2019). Tyylien asettamiseen käytin jQueryn css-metodia, jolla on mahdollista asettaa tyylimääriyksiä valituille elementeille (Arnold 26.4.2017). Ongelma ei lopulta tuntunut hankalalta ratkaista. Murupolkunavigaatiota avattaessa breadcrumbOverlay-elementti asettui nyt oikein. Olin kirjannut murupolkunavigaation työstämiseen jo 130 työtuntia, ja olin iloinen, että sain murupolkunavigaation jo lähes valmiiksi, ja pääsisin työstämään muitakin tehtäviä.

Viikon aikana olin tehnyt ylänavigaatioon muutoksia. Ylänavigaatio rakentuu eri tiedostoista mobiililaitteille ja työpöytänäkymälle. Aloitin muutokset poistamalla työpöydän navigaation luovasta tiedostosta ylimääräisen koodin, jolla luotiin alasvetovalikko ja sain ulkoasun näyttämään halutulta. Toisaalta mietin, että olisiko järkevää poistaa kymmeniä rivejä koodeja, mikäli dropdown-ominaisuutta haluttaisiin käyttää joskus tulevaisuudessa. Kysyin mielipidettä kollegalta, joka myös oli sitä mieltä, että koodin poistamisen sijaan, voisin muokata navigaation luontiin käytettävää dataa. Palautin koodin ennalleen.

Sen sijaan, että muuttaisin koodia, muutin Contentfulista dataa, minkä perusteella murupolku rakennetaan. Contentfulista löytyy kenttä, johon on kirjoitettu ulkoasun rakentamiseen vaadittava data JSON-muodossa. JSON, eli JavaScript Object Notation, on datan siirtämiseen käytettävä formaatti (Squarespace 2019). Muokkasin JSON-datan rakennetta siten, että yhdelläkään ylänavigaation linkillä ei ollut lapsisivuja. Muutosten jälkeen sivut rakentava tiedosto ei enää luo dropdown-elementtiä, koska ylänavigaatiom linkeillä ei enää ollut lapsisivuja. Tein myös mobiilinäkymän Contentful-sivulle navigaation dataan samat muutokset, joissa poistin lapsisivut, jolloin myös mobiilinnavigaation ulkoasu näkyi oikein.

Seurantaviikolla onnistuin viemään murupolkunavigaation omasta kehityshaarastaan develop-haaraan, mistä muutokset siirtyvät testausviikon jälkeen tuotantoon. Murupolun viimeistelyn jälkeen työstin ylänavigaation loppuun, ja sain sen uuden suunnitelman mukaiseksi. Viikon merkittävimmät ongelmat kohtasin tuodessani murupolun muutokset feature-haarasta develop-haaraan. Ongelmat eivät olleet suuria, ja onnistuin kollegani avustuksella korjaamaan tilanteen helposti, mutta en usko, että olisin saanut korjattua ongelmani itsenäisesti. Ylänavigaatiota työstäessäni opin yhä paremmin, miten sivustomme elementit rakentuvat, ja uskon osaavani jatkossa paremmin ja nopeammin korjata ongelmatilanteita, mitkä johtuvat Contentfuliin lisätyistä tietueista ja niiden ongelmista.

3.7 Seurantaviikko 7

Maanantai 25.2.2019

Päivän suunnitelmissani on jatkaa alanavigaation muutoksia, jotka jäivät viime viikolla kesken. Tavoitteenani on saada alanavigaation muutokset valmiiksi ja vietyä develop-haaraan.

Ryhdyin päivän aluksi työstämään alanavigaatiota tutkimalla tiketin suunnitelmia. Navigaatiosta tahdottiin suppeampi, mikä toteutettaisiin poistamalla linkkejä ja tiivistämällä

elementtejä lähemmäs toisiaan. Alanavigaatio rakentui ylänavigaation tapaan Contentfuliin kirjoitetun JSON-tiedon perusteella. Asetin Jenkinsistä tuotannon päivityksen disabled-tilaan, jolloin QA:lla olisi mahdollista tehdä Contentfulin muutoksia turvallisesti, ilman että muutokset valuisivat vahingossa tuotantoon. Muokkasin JSON-tietueita ja julkaisin niitä QA-ympäristöön, mutta osa linkeistä ei toiminut oikein, sillä niitä klikattaessa linkit eivät johtaneet mihinkään.

Navigaation työstö keskeytyi pian aloittamisen jälkeen, kun sain kollegalta viestin, että Oma Palvelun ja myyntisovelluksen painikkeet olivat lakanneet toimimasta. Käytin hetken tutkiakseni ongelmaa, ja kerroin kollegalleni, mistä ongelma johtui. Kollegani korjasi ongelman, ja sivujen uudelleenrakennuksen jälkeen painikkeet toimivat taas normaalisti. Selvitystyön jälkeen pystyin palaamaan takaisin muihin tehtäviin.

Sen sijaan, että olisin jatkanut alanavigaatiota, siirryin korjaamaan murupolkunavigaatiota. Huomasin, että murupolussa kaikki linkit eivät toimineet oikein. Murupolku olisi tärkeää saada toimimaan tämän päivän aikana oikein, sillä sen on tarkoitus siirtyä huomenna tiistaina tuotantoon. Tutkiakseni ongelmaa julkaisin QA:lle murupolut ja päivitin staattiset sivut.

Murupolkunavigaation linkit eivät toimineet, koska QA-ympäristössä koodi ajetaan minifierin läpi, mikä lisäsi linkkeihin kauttaviivan, mikä rikkoi linkit. Kollegani korjasi ongelman, mutta päivä oli jo iltapäivän puolella, ja päädyimme lykkäämään tuotantopäivitystä tiistailta keskiviikkoon.

Päivä ei sujunut suunnitellusti; ehdin työstämään alanavigaatiota vain hetken, ja se jäi kesken. Murupolkunavigaation ongelma lykkäsi tuotantopäivitystä tiistailta keskiviikkoon, mitä en osannut odottaa. Päivityksen lykkäys ei onneksi aiheuttaisi ongelmia, vaan se antaisi meille lisää aikaa testata, että murupolku toimisi oikein.

Tiistai 26.2.2019

Suunnittelen käyttäväni suuren osan päivästäni murupolkujen laadun varmistamiseen. En voi viedä alanavigaation muutoksia vielä tuotantoon, sillä sen työstäminen on vielä kesken, eikä sitä ole vielä testattu. Loppupäivän käytän alanavigaatioon tehtyjen muutosten tallentamiseen omalle koneelleni, jotta voin poistaa ne ennen tuotantopäivitystä. Kesken-eräiset navigaation muutokset menisivät tuotantoon, ellei niitä poistaisi ennen sitä. Päivityksen jälkeen palautan muutokset takaisin Contentfuliin.

Aloitin tekemällä lisää muokkauksia murupolkunavigaatioon, ja puskin muutokset testi-serverille. Sivujen rakentaminen epäonnistui, sillä Jenkinsin ajama node.js versio on liian vanha osataksaan käsitellä ja rakentaa yhtä funktiosta. Kollegan avustuksella korjasin koodin toimimaan oikein, ja sivut rakentuivat ilman virhettä. Kolleganikin harmitteli, miten monet eri ohjelmistojen versiot ovat usein vanhoja tai muuten epäyhteensopivia muiden versioiden kanssa.

Pyysin Slackin kautta testaajaa tutkimaan murupolkua QA-ympäristössä, jotta saisimme vielä viimeisen varmistuksen, että kaikki toimii oikein. Testauksen jälkeen ilmoitin tuotemistajalle korjausten valmistumisesta.

Murupolkunavigaatioissa oli enää korjattava Internet Explorer 11 -selaimella esiin noussut ongelma alavetovalikossa. Samaan aikaan kuin mietin asiaa, käyttöliittymäkehittäjäamme kertoi kohtaamastaan bugista iPad 4:llä. Tässä kohtaa pyysin esimieheiltäni luvan hankkia BrowserStack-lisenssin, jotta voisin testata sivujamme eri selaimilla. BrowserStackin lisenssin ostaminen ja siihen liittyvän selainliitännäisen asentaminen onnistui helposti, ja pääsin jo noin varttia myöhemmin tutkimaan, miksi IE11 ei toiminut oikein. Virhe ei ollut z-indexissä, kuten olin aiemmin luullut, vaan virheellisessä tyylimäärityksen syntaksissa. Olin käyttänyt väärää syntaksia eli sääntömuotoa asettaessani dropdown-elementin näkyvyyttä. En ollut edes huomannut ongelmaa, sillä Chrome ja Firefox-selaimet olivat osanneet piirtää sivun väärästä syntaksista huolimatta. IE11 sen sijaan ei osannut. Huomasin, että tekstieditorini tarjoaa myös virheellisiä arvoja joillekin syntakseille, kuten "min-width: unset", vaikka oikea syntaksi olisi "min-width: 0". En ollut kiinnittänyt tähän huomiota. Kerroessani ongelmastani kollegalleni, hän sanoi, että on harmillista, että kaikki editorit eivät osaa tulkita kieliä yhtä hyvin kuin toiset. Itselläni on käytössä Visual Studio Coden ilmaisversio, kun kollegallani on JetBrainsin maksullinen editori. On mahdollista, että tulevaisuudessa kysyisin esimieheiltäni, voisinko saada itsellenikin vastaavanlaisen editorin työni helpottamiseksi, ja laadun varmistamiseksi.

Päiväni sujui kuten olin odottanut; työstin tehtäviä, joita olin päivän alussa suunnitellut tekäväni. Sain korjattua murupolut toimiviksi, palautin Contentfuliin tekemäni muutokset takaisin alkuperäiseen tilaan, ja sain vielä korjattua IE11 liittyvän bugin. Tästä päivästä opin entistäkin paremmin kiinnittämään huomiota kirjoittamani koodin oikeellisuuteen.

Keskiviikko 27.2.2019

Suunnittelen varmistavani aamun tuotantopäivityksessä tehdyt muutokset murupolkuihin, ja että sivujen sisäisiin linkityksiin tehdyt muutokset toimivat oikein tuotannossa. En pysty

jatkamaan alanavigaation muutoksia tänään, joten tavoitteenani on luoda A/B-testi, joka liittyy VWO-tikettiin.

Aloitin päivän tarkistamalla, että murupolkunavigaatio toimii oikein tuotannossa. Sekä murupolun ulkoasu ja toiminnollisuudet että sisäisten linkkien korjaukset olivat kunnossa, ja pystyin alkaa työstämään VWO-testiä.

VWO-testiin liittyvä tiketti oli kulkenut sprintistä toiseen jo kolme kertaa. Olin ollut niin kiireinen murupolkunavigaation työstämisessä, etten ehtinyt edes katsoa VWO-tikettiä. Tike-tissä tahdottiin luoda A/B-testi ostoputkeen. A/B-testissä käyttäjille näytetään sattumanva-raisesti, joko alkuperäistä versiota A, tai testiversiota B, minkä tarkoituksena on selvittää, kumpi versioista johtaa parempaan tulokseen (VWO 2019). Tike-tin testissä tahdotaan tutkia, onko avattavalla turvatasotaulukolla vaikutusta käyttäjiin, mikäli se olisi avattuna sivun latautuessa.

Testin luominen oli hankalaa, koska en kyennyt näkemään työn tulosta koodia työstäes-säni. Koin, että VWO:n dokumentaatiosta oli hankala saada selville, voinko käyttää jQueryä testissä vai en. Sain lopulta selville, että aikaisemmissa testeissämme oli käytetty jQueryä, sillä löysin koodista asetuksen, joka ei poista jQueryä käytöstä, eli tulkitsin, että sitä voi siis käyttää. Testi itsessään ei ollut vaikea luoda, sillä tiesin, mitä elementtiä minun tulisi manipuloida sivun auetessa. Sain luotua koodin valmiiksi, jolla A/B-testin pystyisi ajaa. Työstöäni vaikeutti se, etten voinut selaimen konsolista kunnolla varmistaa koodissa ajettavien skriptien toimivuutta.

Onnistuin luomaan A/B-testin, ja pääsin päivälle asetettuun tavoitteeseen. Päiväni kului suunnitellusti, ja sain itsenäisesti luotua koodin, A/B-testiä varten. Testi ei vielä ollut ajossa, ja se oli testattava ja viimeisteltävä. Koen kehittyneeni JavaScriptin kirjoittami-nessa, sillä kykenin itsenäisesti luomaan koodin, joka tulisi ajettavaksi VWO-testissä. Koodi olisi vielä testattava, mutta uskon sen toimivan halutusti.

Torstai 28.2.2019

Suunnittelen viimeisteleväni VWO-tike-tin, ja tutkin, mitä kaikkea muuta voisin alkaa työstä-mään ennen kuin jatkan alanavigaation loppuun. Tavoitteenani on tehdä VWO-testi val-miiksi, sekä aloittaa työstämään jotain hakukoneoptimointiin liittyvää tehtävää.

Eilen testaamatta jääneen A/B-testin viimeistelyyn käytin noin puoli tuntia. Varmistelin, että koodi toimii odotetusti, minkä jälkeen tallensin sen VWO-editorin skripteihin. Asetin

testin finished-tilaan, jolloin pystyin esikatselemaan testiä, ja se toimi odotetusti, eikä mikään näyttänyt menneen rikki. Eräs kollegoistani tarkisti testin asetukset, ja sanoi testin olevan valmis käynnistettäväksi. Kollegani laittoikin testin käyntiin tuotannossa, ja minä sekä kollegani tarkistimme selaimella, että testi toimi odotetusti. Sivulla vieraillessa testi laukaisee 50% todennäköisyydellä kävijälle version, jossa turvatasotaulukko on auki. Minun koestaessani sivua taulukko aukesi heti sivun latautuessa, eli testi toimi kuten sen oli tarkoitus.

Varmistettuani VWO-testin toimivuuden siirryin tekemään hakukoneoptimointiin liittyvää tehtävää. Otin työstääkseni sivuston tuotelinkkien muutokset. Tuotelinkeistä tahdottiin poistaa ylimääräisiä painikkeita, minkä tein helposti tyylimäärityksillä. En muokannut koodia, sillä samalla koodilla rakennetaan vastaavanlaisia painikkeita muille sivuille. Vein tekemäni muutokset develop-haaraan, minkä jälkeen päätin jatkaa sivuston sisäisten linkkien korjaamista.

Sisäisten linkkien korjaus oli yksi hakukoneoptimointiin liittyvistä tehtävistä, jossa tarkoituksena on vähentää sivustolla tapahtuvia uudelleenohjauksia. Uudelleenohjauksien vähentämisen pitäisi SEO-auditoinnin mukaan vaikuttaa positiivisesti hakukonenäkyvyyteen. Korjailin linkkejä, ja sain valtaosan staattisten sivujen linkeistä muokattua oikeaan muotoon päivän loppuun mennessä.

Päivä kului suunnitellusti, ja pääsin päivän tavoitteeseen; luomani A/B-testi toimi odotetusti, ja ehdin työstämään hakukoneoptimointiin liittyvän tiketin. VWO:n testiä koestaessani sain itselleni vahvistuksen, että olen tosiaan kehittynyt JavaScriptin kirjoittamisessa. Koen osaavani paljon aikaisempaa paremmin ja itsevarmemmin luoda koodia, ja tiedän, mitä minun täytyy tehdä, jotta saan koodin toimimaan.

Perjantai 1.3.2019

Suunnittelen työstäväni sisäisten linkkien korjausta, ja jatkan alanavigaation muokkamista. Tavoitteenani on tänään saada alanavigaatio valmiiksi.

Aloitin jatkamalla sisäisten linkkien korjaamista, minkä olin aloittanut eilen. Luulin saaneeni valtaosan linkeistä jo korjattua, mutta sain terminaalin grep-komennolla selville, että sivustolla oli vielä paljon linkkejä, jotka tulisi korjata. Työskentelin URL-osoitteiden korjaamista koko päivän. Sain korjattua sivustolta kaikki koti- ja ajoneuvovakuutuksen tuotesivujen linkit, ja tehtäväkseni jäi vain autovakuutus sivujen linkkien korjaaminen. Olin niin

uppoutunut työhöni, että en huomannut ajan kulua, enkä ehtinyt jatkamaan alanavigaation työstämistä.

Päivälle asetettu tavoite jäi saavuttamatta, mutta ehdin sen sijaan tehdä paljon sisäisten linkkien korjauksia.

Viikkoanalyysi

Ehdin seurantaviikon aikana työstää monia eri hakukoneoptimointiin liittyviä tehtäviä. Hakukonenäkyvyyden parantaminen on monien asioiden summa (Search Engine Land 2019), minkä vuoksi SEO-tikettejä on useita. Tehtävistä merkittävin oli murupolkunavigaation korjaustehtävät, ja siinä kohdatut ongelmat.

Viikon alussa tutkin Oma Palvelussa esiin noussutta ongelmaa, josta kollegani kertoi minulla Slackin kautta. Oma Palvelussa ja myyntisovelluksessa vahinkoilmoituksen painike oli rikki. Olin työstänyt painikkeita ensimmäisellä ja toisella seurantaviikolla. Painikkeiden korjaaminen oli tärkeä saada korjattua nopeasti, ja se oli ensisijaisen tärkeä tehtävä.

Siirryin tutkimaan myyntisovellusta, ja huomasin, että kalenteri ei ole rakentunut oikein, vaan päivämäärien valitsemiseen käytettävä kenttä puuttui. Kentän puuttumisen vuoksi taustalla pyörivä koodi ei saanut tarvittavaa dataa, jotta vahinkoilmoituksen olisi voinut tehdä tai sopimuksen päättää. Kerroin ongelmasta kollegalleni, joka oli tutkinut ongelmaa samanaikaisesti, ja löysi syyn, miksi sivuja rakentaessa yhtä tiedostoista ei rakennettu lainkaan.

Linkkien toimimattomuuden syy oli siinä, että koodi ajettiin minifierin eli koodin minimoijan läpi. Minimoinnin tarkoituksena on pienentää datan määrää, minkä käyttäjä joutuu lataamaan. Minimoinnissa on ominaisuus lisätä linkkien alkuun kauttaviiva, mikäli se puuttuu. Linkkien URL-osoitteet olivat rakennettu relatiivisiksi osoitteen nykyisen polun suhteen (kuva 13), mikä tarkoittaa, että linkin osoite lisätään nykyisen sivun osoitteen perään (Smith 11.7.2014). Mikäli linkkielementti sijaitsee sivulla "etusivu/vakuutukset/ajoneuvovakuutukset/", ja linkkielementin osoitteessa on "mopovakuutus", linkki johtaisi klikattaessa osoitteeseen "etusivu/vakuutukset/ajoneuvovakuutukset/mopovakuutus".

``

Kuva 13. Polkurelatiivinen URL-osoite

Koodin minifier lisäsi osoitteen alkuun kauttaviivan, eli linkissä pelkkä mopovakuutus olisi ”/mopovakuutus”, ja linkki johtaisi osoitteeseen ”etusivu/mopovakuutus”. Osoitteen alussa kauttaviiva tarkoittaa osoitteen olevan relatiivinen osoitteen juuren eli etusivun suhteen (Smith 11.7.2014).

Murupolkujen linkit olivat rakennettu polkurelatiivisiksi, eli osoitteet eivät sisältäneet osoitteen koko polkua. Tämä itseasiassa oli perustavan laatuinen virhe. Murupolkujen luomisen tavoitteena oli parantaa hakukonenäkyvyyttä, mitä ei välttämättä saataisi aikaan, mikäli linkit ovat relatiivisia. Otimme kollegani ja käyttöliittymäsuunnittelijan kanssa yhteyttä tuoteomistajaan, ja kerroimme tälle murupolkujen ongelmasta. Oli hyvä, että huomasimme ongelman ennen tuotantoon vientiä, mutta oli harmillista, että ongelma havaittiin näin myöhään. Murupolkuja ei voitu viedä tuotantoon suunnitellusti tiistaiaamuna. Keskustellessamme totesimme, että tuotantopäivityksessä ei olisi mitään niin kriittistä, etteikö se voisi odottaa keskiviikkoon. Päädyimmekin lykkäämään päivitystä yhdellä päivällä, saadaksemme tarpeelliset korjaukset tehtyä.

Kollegani ryhtyi tekemään korjauksia koodiin, ja hän sai tehtyä muutokset, joilla linkit rakentuivat siten, että ne sisälsivät kaikki ne sivut, joita pitkin kullekin sivulle on navigoitu (kuva 14).

```
<a href="/vakuutukset/ajoneuvovakuutukset/mopovakuutus">
```

Kuva 14. Juurirelatiivinen URL-osoite

Murupolkunavigaation ongelma oli saatu korjattua, ja kollegani antoi tehtäväkseni tutkia koodia lisää, ja siistiä sitä. Koodin siistimisellä kollegani tarkoitti, että poistaisin koodista sinne mahdollisesti unohtuneet konsolilokitukset.

Tuotantopäivitys oli poikkeuksellisesti siirretty tiistaista keskiviikkoon mikä mahdollisti murupolkunavigaation toimivuuden testaamisen. Tarkistin, että navigaatio toimi oikein, minkä jälkeen ajoin cherry-pick komennon, jolla toin muutokseni QA-ympäristöön. Gitin cherry-pick on komento, jolla yksittäinen kommitointi eli ”git commit” -komennolla tallennettu muutos voidaan tuoda haarasta toiseen (Atlassian 2019b). Ihmettelin mikseivät muutokset tulleet vielä testiserverillä näkyviin. Tarkistin Jenkinsin testiserverin ajosta syntyneen lokitiedoston, joka ilmoitti yhden tiedoston rakentamisen epäonnistuneen, sillä se sisälsi tuntemattoman funktion. Ongelmaa Googlettamalla löysin vastauksen, mikä todennäköisimmin vastaa tilannetta, että Jenkinsin ajama node.js versio on liian vanha osatakseen käsitellä

ja rakentaa kyseistä funktiota. Minun täytyykin vaihtaa aiemmin käytetty funktio vastaavaan lodash-funktioon.

Tarkistin tehdyt korjaukset omassa kehitysympäristössä, minkä jälkeen vein muutokset develop-haaraan, ja siitä edelleen cherry-pick -komennolla QA-haaraan. Ongelmaksi muutoksia tuodessani tuli aikaisemmin poimimani kommitointi, joka oli vielä tallessa. Minun täytyikin palata taaksepäin komennolla "git reset HEAD~1", jolloin päästiin tilanteeseen ennen aikaisempaa cherry-pick -komentoa. Tarkistuksen yhteydessä huomasimme kollegan kanssa, että tarvitsemme vielä kaksi muuta muokattua tiedostoa develop-haarasta. Sen sijaan että poimisin yhden kommitin kuten tein aiemmin, hain kaksi tiedostoa QA-haaraan, jotka kommitoin, ja puskin QA-haaraan komennolla "git push origin qa:qa". Kollegani suositteli, että komennon "push origin qa" sijaan käyttäisin komentoa "push origin qa:qa", mikä erityisesti kertoo, mistä haarasta pusketaan ja minne pusketaan (Git 2019). Tällä menetelmällä voidaan välttää mahdolliset virheet, mikäli nykyinen haara olisi-kin eri, kuin missä luultiin olevan. QA:lle tehtyjä muutoksia ei pysty rakentamaan Jenkinsin kautta, vaan sivustojen rakentamiseen käytettävät skriptit oli ajettava käsin kirjautumalla QA:n palvelimelle. Sivujen uudelleenrakennuksen valmistuttua muutokset näkyivät oikein sivustolla. Murupolkunavigaatio oli saatu korjattua, ja se oli valmis tuotantopäivitystä varten.

Murupolkunavigaatioissa linkkien URL-osoitteet olivat alusta alkaen rakennettu siten, että ne olivat hakukoneoptimoinnin kannalta väärässä muodossa. Murupolkujen tiketissä ei ollut otettu kantaa linkkien relatiivisuuteen, ja linkkien URL-osoitteiden ongelma nousi esiin vasta QA-ympäristössä. Kollegani sai ongelman korjattua, mutta linkkien virheellisyys johti tuotantopäivityksen lykkäämiseen. Poikkeuksellinen lykkäys oli mielenkiintoinen kokemus, josta opin, että tarpeen vaatiessa on mahdollista saada lisää aikaa. Opin paljon hakukoneoptimoinnista tutkittuani aihetta Internetistä, ja lukemalla auditoinnin tuloksia.

3.8 Seurantaviikko 8

Maanantai 4.3.2019

Nykyisessä sprintissä minulla on vielä loppuunsaattamatta sivujen URL-osoitteiden korjaaminen sekä alanavigaation tyylimääritykset. Suunnittelen tänään tekeväni alanavigaation muutoksia koko päivän. Tavoitteenani on saada alanavigaatio valmiiksi ja tyylimääritykset näkymään oikein.

Aloitin päivän tarkistamalla, mihin asti olin saanut työstettyä alanavigaation datan Contentfuliin. Olin aiemmilla viikoilla ehtinyt aloittamaan alanavigaation työstön, mutta Contentfulin data oli vielä keskeneräinen. Muokkasin JSON-dataa, kunnes navigaation näytti oikealta, ja pystyin alkaa työstämään tyylimuutoksia. Tyylimuutoksilla pienensin linkkien rivien välejä, ja asetin elementtejä oikeille paikoilleen. Käytin lähes koko päivän tyylimuutosten luontiin.

Päivän lopulla huomasin, että osa linkeistä ei toiminut oikein. Ongelma johtui siitä, että koodissa otsikot muutettiin alasvetovalikoiksi. Työstin alanavigaation kuntoon tekemällä koodiin muutoksia, joilla sain näyttämään tiketin suunnitelman mukaiselta.

Päivä kului suunnitellusti, ja pääsin päivälle asetettuun tavoitteeseen ilman suuria ongelmia. Sain alanavigaation tyyllitetyä oikein. Päivän loppupuolella huomasin ongelman linkeissä, minkä sain korjattua helposti. Huomasin osaavani tunnistaa helpommin, mistä ongelmat johtuvat, ja kykenen korjaamaan virheet nopeasti.

Tiistai 5.3.2019

Terveystenhoitoa koko päivä

Keskiviikko 6.3.2019

Suunnittelen käyväni läpi SEO-tiketit, ja katson, mitkä niistä ovat sellaisia, joita voisin ryhtyä työstämään seuraavassa sprintissä. Päivän tavoitteenani on korjata alanavigaation virheet, ja viimeistellä alanavigaatio täysin valmiiksi.

Kollegani oli kirjoittanut tiistaina Slackissa viestin, että hän oli huomannut selaimellaan "undefined" -virheen alanavigaatiossa. Aloitin päivän tutkimalla alanavigaation ongelmia. Kollegani kertoi, että osalla sivuista alanavigaatio näkyi väärin. Ongelma johtui siitä, että sivut rakennettiin PHP:llä, toisin kuin staattiset sivut, jotka rakentuivat JavaScriptillä. En ollut ajatellut tätä navigaatiota työstäessäni. Minun tuli muokata PHP:lla rakentuva alanavigaatio vastaamaan tiketin mallia. Uudessa alanavigaatiossa oli yhdistetty vanha alanavigaatio, sekä sivun alalaidassa sijaitseva footer-elementti.

Korjasin alanavigaation ulkoasun samaan muotoon, kuin se oli staattisilla sivuilla. Huomasin, että PHP-pohjaiset sivut eivät syystä tai toisesta saaneet samoja tyylitiedostoja kuin JavaScriptillä rakennettavat sivut. En ehtinyt tutkia ongelmaa ennen sprint-palaveria.

Sprint-palaverissa otin työstääkseni vain yhden uuden tiketin tulevaan sprinttiin, sillä halusin varata resursseja mahdollisiin uusiin brändiuudistukseen liittyviin tiketteihin. Palaverin jälkeen jatkaisin sisäisten linkitysten korjauksien työstämistä, sen sijaan että olisin jatkanut alanavigaation korjaamista. Sisäisten linkkien korjaus oli tärkeämpi tehtävä kuin PHP-sivujen alanavigaation korjaaminen.

Jatkoin linkkien URL-osoitteiden korjaamista. En ehtinyt korjaamaan kaikkea, ja osalla sivuista oli vielä linkkejä, joiden osoite ei ollut linkin lopullisen sivun osoite. Jatkoin linkkien korjaamista päivän loppuun saakka.

Torstai 7.3.2019

Päivän suunnitelmana on jatkaa linkkien URL-osoitteiden korjaamista, sekä käydä läpi avoimien tikettien puutteet. Tavoitteenani on tänään saada alanavigaation tyylit toimimaan oikein myös PHP-sivuilla.

Aloitin korjaamalla linkkejä, uudelleenohjauksien poistamiseksi. Tutkin sivustoa, ja hain grep-komennolla virheellisiä linkkejä. Käytin linkkien korjaamiseen suuren osan päivästä, mutta lopulta sain kaikki jäljelle jääneet virheelliset linkit korjattua.

Linkkien korjaamisen jälkeen siirryin työstämään alanavigaatiota, johon oli tehtävä vielä tyyli muutoksia. Paikansin ja korjasin ongelman, minkä vuoksi tyylit eivät tulleet PHP-sivuille. Muutokset eivät jostain syystä tulleet näkyviin omassa työympäristössäni. Puskin muutokset testiserverille, missä muutokset toimivat oikein.

Perjantai 8.3.2019

Suunnittelen tekäväni vakuutuslaskuri-sivulle alustavia luonnoksia uudelle vakuutuslaskuri-widgetille. Tavoitteenani on tutkia, kuinka saan luotua vakuutuslaskurin, minkä lisäksi tavoitteena on luoda alustava versio vakuutuslaskurista.

Vakuutuslaskurin tarkoituksena on auttaa asiakasta löytämään oikeaan ostoputkeen. Sivustollemme ei ollut vielä luotu vastaavan tyyppistä ominaisuutta, eikä minulla ollut mitään mistä ottaa mallia, joten jouduin aloittamaan työni käytännössä tyhjästä. Ryhdyin työstämään vakuutuslaskuria kokeilemalla, voisiko sen luoda käyttäen Contentfulista löytyvää dropdown-tietotyyppiä. Huomasin pian, että dropdown-tietotyyppi ei tulisi toimimaan. Mietin, että ylläpidettävyyden kannalta helpoin ratkaisu olisi, jos laskuri rakentuisi Contentfulista JSON-datan perusteella, sillä se olisi myöhemmin helposti muokattavissa.

Muokkasin Contentfulin vakuutuslaskuri-tietotyyppiä, ja lisäsin tarvittavan datan. Tarkoitukseni olisi, että JSON:ssa olisi lista vakuutuksen tyypeistä, ja lista vakuutuksen kohteista. Vakuutuksen kohteet esitettäisiin valitun vakuutustyyppin perusteella.

Sain datan siirrettyä Contentfulista staattiselle sivulle tulostumaan kahtena listana. Muutin listat select-elementeiksi, joista valinnat voisi tehdä. Seuraavaksi minun olisi luotava JavaScript-koodi, jolla tehdyt valinnat muokkaisivat näytettäviä vaihtoehtoja.

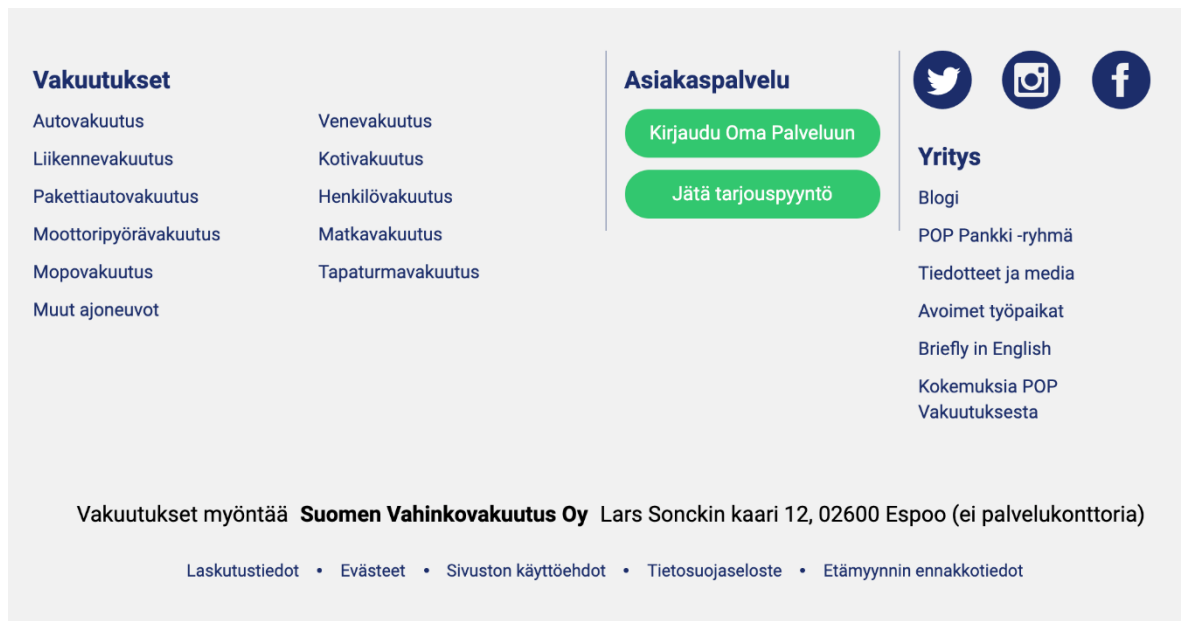
Viikkoanalyysi

Viikon aikana ehdin työstämään monia toisistaan hyvin erilaisia työtehtäviä, jotka kaikki liittyivät hakukoneoptimointiin. Tein sivuston linkkien URL-osoitteiden uudelleenohjausten muokkauksia, joilla korjasin linkit ohjaamaan niihin URL-osoitteisiin, joihin ne lopulta tulisivat johtamaan, sen sijaan, että palvelimemme tarvitsisi tehdä osoitteille uudelleenohjauksia. Työstin viikolla myös alanavigaatiota, johon tein rakenteellisia muutoksia, ja suuren määrän tyylimuutoksia. Saatuaani sivuston linkit korjattua ja alanavigaation valmiiksi, ryhdyin työstämään sivustollemme täysin uutta vakuutuslaskuria.

Seurantaviikolla työstämässäni alanavigaation tiketissä oli tarkat ohjeet, miten linkkien tulisi asettua työpöytänäkyymässä, tableteilla, ja puhelimien ruuduilla. Alanavigaation tiketissä tahdottiin yhdistää footer-elementti ja alanavigaatio yhdeksi kokonaisuudeksi. Footer-elementti sijaitsee sivun alalaidassa ja se sisältää yleistä sivuston tietoa kuten tietoa yrityksestä ja sivuston käyttöehdoista (Computer Hope 2018). Saadakseni navigaation toimimaan minun täytyi tehdä muutoksia Contentfulin dataan, rakenteellisia muutoksia navigaation koostavaan Handlebars-tiedostoon, sekä luoda uusia tyylimäärittelyjä.

Alanavigaatio rakentuu staattisille sivuille Handlebars-tiedostosta, ja sen luomiseen käytetään Contentfuliin lisättyä dataa kolmesta eri JSON-kentästä. Poistin JSON-datasta turhat linkit ja järjestelin loput oikeille paikoilleen. Seuraavaksi muokkasin Handlebars-tiedoston rakennetta siten, että siirsin footer-elementin piirtymään samassa tiedostossa kuin alanavigaation. Muutoksen myötä minun olisi mahdollista saada elementit asettumaan sivulla oikein, minkä lisäksi oli muokattava alanavigaation tyylimäärittelyt, joilla asettelisin elementit.

Uusien tyylimäärittelyjen luominen oli alanavigaation työstön vaikein osuus. Elementtien tuli asettua vierekkäin isolla ruudulla, kahdelle riville tablettikoossa, ja allekkain yhdeksi pylvääksi pienillä ruuduilla.



Kuva 15. Uusi alanavigaatio työpöytä näkymässä

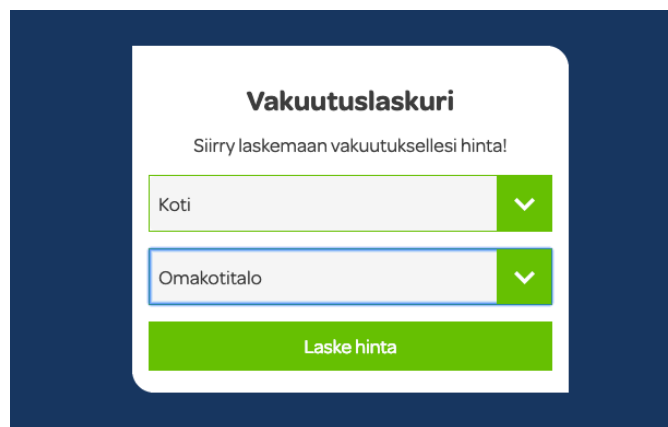
Uudessa alanavigaatioissa (kuva 15) vasemman laidan vakuutukset-otsikon alla olevat linkit tuli tyylimäärittellä asettumaan kahteen pylvääseen, sen sijaan, että ne olisivat olleet allekkain yhdessä rivissä. Pystyin jakamaan linkkejä ympäröivän elementin yhdestä pylväästä kahteen pylvääseen column-määrittelyllä (MDN web docs 2019b). Linkit asettuivat muilta osin oikein, mutta toisen pylvään linkit olivat hieman väärällä korkeudella. Tein korjauksen ensimmäisen pylvään kuudenteen linkkielementtiin "nth-of-type" -määrittelyllä, jolla tyyli voidaan kohdistaa elementin järjestysnumeron perusteella (Cope 2011b). Määrittelyn myötä linkit asettuivat samaan tasoon.

Saatuani tyylimuutokset valmiiksi huomasin, että muun muassa asiakaspalveluun johtava linkki ei toiminut. Päättelin syyn olevan navigaation rakentavassa tiedostossa, jossa otsikolinkeistä luodaan alasvetovalikoita. Tein korjauksen JSON-dataan muuttamalla otsikot tavallisiksi tekstilinkeiksi, mutta se aiheutti ongelmia sivurakenteessa. Palautin otsikot takaisin, ja lähdin etsimään ratkaisua navigaation rakentavasta Handlebars-tiedostosta. Muokkasin koodia siten, että otsikoista tehtäisiin alasvetovalikoita vain silloin, jos otsikolla oli JSON-datassa asetettu lapsisivuja.

Olin saanut tehtyä alanavigaation rakenteelliset muutokset ja tyylimäärittelyt staattisille tuotesivuille sekä etusivulle. En ollut työstäessäni huomannut, että alanavigaatio ei toiminut PHP:lla rakentuvilla sivuilla. Minun tuli muokata koodissa PHP:lla rakentuva alanavigaatio vastaamaan tiketin mallia. Uudessa alanavigaatioissa oli yhdistetty vanha navigaatio, sekä sivun alalaidassa sijaitseva footer-elementti. Staattisilla sivuilla alanavigaatio ja

footer rakentuivat eri tiedostoista, mutta PHP:llä ne rakentuvat samassa tiedostossa, minkä vuoksi ongelma oli helppo korjata.

Perjantaina pääsin aloittamaan vakuutuslaskurin työstön. Ulkoasultaan vakuutuslaskuri muistuttaa widgettiä, sillä se on muusta sivuston sisällöstä poikkeava elementti. Widgetillä tarkoitetaan pientä ohjelmaa, jonka voi asettaa sivustolle tai blogiin (Nations 19.4.2019). Vakuutuslaskuri ei siis todellisuudessa ole widget, sillä vakuutuslaskuri rakennetaan sivun sisälle, eikä se ole erillinen ohjelma. Vakuutuslaskuri ei myös nimestään huolimatta laske mitään, vaan sen tarkoituksena on auttaa käyttäjää löytämään oikeaan os-toputkeen.



Kuva 16. Prototyyppi vakuutuslaskurista

Vakuutuslaskurista oli tiketti, johon käyttöliittymäsuunnittelijamme oli luonut luonnoksia vakuutuslaskurin ulkoasusta ja toiminnallisuudesta (kuva 16). Vakuutuslaskurissa valitaan ensin vakuutuksen tyyppi vaihtoehtoista ajoneuvo-, koti-, henkilö-, ja venevakuutus. Seuraavaksi ilmestyy toinen alasvetovalikko, josta valitaan vakuutuksen kohde. Kotivakuutuksella vakuutuksen kohteita ovat esimerkiksi omakotitalo, kerrostalo, rivitalo ja ulkorakennus.

Pohdin, millä tavoin vakuutuslaskuri tulisi rakentaa, ja kokeilin, voisiko vakuutuslaskurin luoda Contentfulin dropdown-sisältötyypin avulla. Loin uuden vakuutuslaskuri-moduulin, johon lisäsin dropdown-sisältötyypin. En alkuun saanut Contentfulin kenttiä näkyviin, sillä en ollut huomannut lisätä tarvittavia tietoja koodiin, joka kertoisi mistä vakuutuslaskuri-moduuli piirrettäisiin. Kokeilin miten dropdown soveltuisi vakuutuslaskurin luomiseen, mutta se ei minusta vaikuttanut toimivalta ratkaistulta, joten hylkäsin idean.

Päädyin kokeiluni tuloksena luomaan Contentfuliin JSON-kentän, johon vakuutuksen tyyppin ja kohteen tiedot lisätään. JSON-datan muokkaaminen myöhemmin olisi helppoa, ja

muutokset tulisivat näkyviin nopeasti. Ideanani on, että sivustolla valitaan jokin vakuutus-tyyppi, minkä jälkeen näytetään valitun vakuutustyyppin lapset, eli vakuutuksen kohteet. Kirjoitin JSON-dataan oman objektin jokaiselle vakuutustyyppille. Vakuutustyyppille annoin otsikon ja sen id:n, sekä sitä vastaavat lapsisivut, johon oli listattu vakuutuksen kohteet. Vakuutuksen kohteille lisäsin otsikon ja id:n sekä URL-osoitteen, joka johtaisi oikeaan os-toputkeen. Saatuaani datan lisättyä ryhdyin tekemään koodimuutoksia, joilla saisin datan tuotua ruudulle. Tarkistin konsolilokilla saavani datan sivustolle, ja muokkasin Handlebars-tiedostoa vastaamaan tekemiäni muutoksia. Sain ruudulle listautumaan kaikki vakuutuksen kohteet ja niitä vastaavat lapsisivut.

Ehdin muuttamaan koodissa vakuutuksen kohteet listautumaan yhteen select-elementtiin, ja lapsisivut omaan select-elementtiinsä. Select-elementillä voidaan luoda lista vaihtoehtoista (engl. option), jotka selectiä klikatessa aukeavat listana, joista käyttäjän on mahdollista valita jokin vaihtoehtoista (W3docs 2019). Vakuutuslaskurissa oli vielä tekemättä otsikko, otsikon alle vaihtuvat tekstit, alasvetovalikot ja laske hinta -painike.

Viikon työtehtävät olivat mielenkiintoisia, ja huomasin oppineeni käyttämään Contentfulia paremmin. Opin muun muassa Contentfulin sisällön tuottamisesta, JSON-datan luomisesta, käsittelystä, ja tulostamisesta. Viikko oli kokonaisuudessaan todella opettavainen, ja arvioin osaavani nyt käyttää Contentfulia paljon paremmin. Tiedän nyt kykeneväni luomaan Contentfulin sisältöä ja korjaamaan sisältöön liittyviä ongelmia.

3.9 Seurantaviikko 9

Maanantai 11.3.2019

Viikon ensimmäisen päivän suunnitelmissani on jatkaa vakuutuslaskurin työstämistä, ja saada sen toiminnallisuus kuntoon. Käytän koko työpäivän alasvetovalikoiden korjaamisessa. Kalenterissani on vain yksi peruttu kokousmerkintä, joten uskoisin voivani käyttää koko päivän vakuutuslaskurin työstämiseen. Tavoitteenani on saada vakuutuslaskurin tekstit ja painikkeet näkymään oikein, sekä saada lisättyä URL-osoite laske hinta -painikkeelle.

Vakuutuslaskuri tulisi koostumaan otsikosta, vaihtuvasta leipätekstistä, kahdesta valikosta ja yhdestä linkkipainikkeesta. Loin JavaScript-koodia, jolla select-valikon vaihtoehtoa muuttamalla lisätään vanhempielementille luokaksi select-valikosta valitun vaihtoehdon id. Olin täysin unohtanut, että meillä aloittaa tänään ainakin yksi konsultti, joka avustaa meitä frontend-tiketeissä. Meille on tulossa yhteensä kaksi konsulttia, mutta toinen heistä

aloittaa vasta keskiviikkona. Minun on tarkoitus avustaa heitä molempia työympäristön pystyttämisesssä ja käyttöönotossa. Avustin konsulttia kertomalla hänelle, mistä löytää ohjeet työympäristön pystytykseen. Ongelmatapauksissa kehotin rohkeasti kysymään minulta lisää.

Jatkoin vakuutuslaskurin työstöä. Ongelmanani on saada alempi dropdown nollaantumaan, kun ylempää dropdownia muutetaan. Tein erilaisia kokeiluja, selvittääkseni miten saisin valikot toimimaan halutusti. Sain korjattua ongelman muuttamalla JavaScriptiä siten, että vanhempielementin kuuntelun sijasta kuunneltiin molempia alasvetovalikoita erikseen. Seuraavaksi minun oli saatava laske hinta -linkkipainike näkymään oikein.

Laske hinta -painike ei saanut näkyä käyttäjälle, ennen kuin alemmasta valikosta oli valittu vakuutuksen tyyppi. Sain luotua funktion, jolla tarkistettiin, oliko vakuutuksen kohde valittu, minkä jälkeen laske hinta -painike muutettiin näkyväksi. Seuraava haaste on saada linkkiin oikea URL-osoite, jota en vielä saanut Contentfulista. En ehtinyt tehdä linkkejä Contentfuliin, joten niiden luominen jää huomiseksi.

Koen päivän kuluneen suunnitellusti, vaikken saavuttanutkaan päivälle asetettua tavoitetta. Sain vakuutuslaskurin toimimaan oikein alasvetovalikoiden osalta. Minulta jäi kesken linkin muodostaminen painikkeeseen, sekä tekstien esittäminen oikein. Vakuutuslaskurissa minulla on vielä tekemättä linkkien muodostaminen, laskurin ulkoasun tyyllittäminen, sekä muut mahdolliset sivupohjan muokkaukset. Olen oppinut paljon JavaScriptiä lyhyessä ajassa, ja osaankin jo paljon paremmin kirjoittaa sitä itsenäisesti. Tiedän nyt paremmin, mitä minun on kuunneltava, muokattava, tai sisällytettävä funktioihini saadakseni halutun lopputuloksen aikaan.

Tiistai 12.3.2019

Tänään suunnittelen jatkavani loppuun vakuutuslaskurin laske hinta -painikkeen URL-osoitteen lisäämisen, sekä vakuutuslaskurin tyyliuokkaukset. Tavoitteenani on tänään saada vakuutuslaskuri kokonaan valmiiksi.

Sain vakuutuslaskurin toimimaan, kuten se oli prototyypissä suunniteltu: Ensin valitaan vakuutuksen tyyppi, sen jälkeen vakuutuksen kohde, minkä jälkeen ilmestyy painike, josta pääsee tuotteen ostoputkeen. Loin tekstit jokaiselle vaiheelle, jossa käyttäjälle kerrotaan, mitä seuraavaksi tehdä. Sen sijaan, että olisin kirjoittanut tekstit koodiin, tein Contentfuliin kentät teksteille, jotta ne olisivat helpompi jatkossa muuttaa, mikäli sille olisi tarvetta.

Safari-selaimella oli ongelma, että valikoiden vaihtoehdot eivät näkyneet oikein, sillä option-elementtejä ei ole mahdollista muokata tyylimäärittelyillä. Vakuutuslaskuri jäi taas kesken, mutta uskoisin ehtiväni tehdä tarvittavat korjaukset ennen huomista sprint-palaveria.

En tänään harmikseni päässyt päivän tavoitteeseen. Vakuutuslaskurin aiempi rakenne ei ollutkaan toimiva, vaan se vaati muokkauksia. Koin olevani tyytyväinen työn tulokseen, sillä olin ehtinyt mielestäni saamaan paljon aikaan, vaikkei ratkaisu ollutkaan ensimmäisellä yrittämällä oikea. Huomasin tehdyistä työstäni, että olen jo oppinut työskentelemään paljon itsenäisemmin, ja osaan paremmin löytää tarvittavat palaset koodin luomiseksi. Huomaan tunnistavani nopeammin väärät lähestymistavat, ja tajuan pysähtyä ajattelemaan muita ratkaisumalleja. Toisinaan tahdon tarkistaa ratkaisuni kollegaltani koodin laadun ja ylläpidettävyyden varmistamiseksi.

Keskiviikko 13.3.2019

Päivän suunnitelmissa on viimeistellä muutokset vakuutuslaskuriin. Minun tarvitsee myös hieman testata sitä itse ennen kuin ohjaan tiketin Provelle. Aiemmin mainittu toinen konsultti saapuu vasta torstaina. Tänään sprint-palaverissa tulen ottamaan uusia brändin uudistukseen liittyviä tikettejä. Aiempina päivinä nousseiden ongelmien vuoksi, en aseta tavoitteitani tälle päivälle liian korkealle. On mahdollista, että kohtaisin vakuutuslaskurissa ongelmia, joiden korjaaminen saattaisi viedä pitkään. Tälle päivälle asetan tavoitteekseni saada vakuutuslaskurin kokonaan valmiiksi, ja vietyä sen testiserverille.

Aloitan päivän vakuutuslaskurin select-valikoiden korjaamisella. Olin jo työstänyt vakuutuslaskuria niin paljon, että minun oli jo hieman hankalaa ajatella, mitä kaikkea tulee vielä tehdä. Mietin hetken aikaa, mitä vakuutuslaskurin tahdotaan tekevän ja missä tapauksissa mitään tulee tapahtua. Koodissani on nyt jo niin paljon muutoksia ja sääntöjä, että minun täytyy käyttää aikaa oman koodin tutkimiseen, jotta saan karsittua siitä mahdolliset turhat osat pois. Hienosäätämisen jälkeen onnistuin saamaan ehtolauseet toimimaan, ja tekstit sekä linkkipainike tulivat näkyviin oikein. Vein korjaukset testiserverille, ja siirsin tiketin Provelle. Olin saavuttanut päivälle asettamani tavoitteen.

Seuraavaksi siirryn tekemään korkean prioriteetin brändiuudistukseen liittyviä tikettejä backlogilta. Aloin työstämään haitareiden tyylimuutoksia brändikirjan mukaiseksi. Tutustuin BEM-nimeämiskäytäntöihin ja sen perusperiaatteisiin, minkä ansiosta osaan nyt paremmin lukea tyylimäärittelyitä. BEM tulee englanninkielien sanoista Block, Element ja Modifier. Nimeämiskäytännön tarkoituksena on tehdä luokkien nimeämisistä

mahdollisimman helposti ymmärrettäviä (BEM 2019). Haitarivalikoiden tyylimuutokset eivät olleet ongelmallisia, vaan sain tyylit muokattua prototyypin mukaisesti vaivattomasti. Ainoa puute tällä hetkellä haitareissa on fontit, jotka korjaantuvat tämän sprintin aikana toisen tiketin yhteydessä.

Sprint-palaverissa kävimme läpi tehtyjä tikettejä. Jiran Portfolio-työkalulla oleva SEO-epic on saatu jo lähes valmiiksi. Kaikista 29 tiketistä vain yksi oli tekemättä. Olin jo ennen palaveria etsinyt itselleni kaksi uutta brändin muutoksiin liittyvää tikettiä. Haitarivalikko -tiketin lisäksi otin ensi sprinttiin työstettäväkseni turvataulukoiden tyylimuutokset.

Palaverin jälkeen jatkoin haitarivalikoiden viimeistelyä, ja sain ne mielestäni lähes valmiiksi. Siirryin seuraavaksi turvataulukoiden tyylimuutosten tekemiseen. En ehtinyt tehdä niitä pitkään, sillä en tahtonut jäädä ylitöihin. Vielä ennen lopettamista tarkistutin tyylimuutokset UI-designerilla, ja hän löysi vielä kaksi puutetta, joita en itse ollut huomannut. Kirjasin puutteet muistiin, jotta minun on huomenna helpompi jatkaa työskentelyä. Koin oppineeni päivän aikana paljon tyylimäärittämisestä lukemalla BEM-nimeämiskäytännöistä. Opin lukemaan koodin tyylimäärittäystä paremmin, sillä nyt tiedän, miten luokat nimitään, ja miten niille tulee tehdä tyylit. Saavutin päivälle asetetun tavoitteen jo ennen puolta päivää. Tavoitteeni saada vakuutuslaskuri valmiiksi oli mielestäni realistinen, muttei liian helppo. Koin, että oli suuri mahdollisuus, että jokin vakuutuslaskurin osa tai ominaisuus ei olisi-kaan toiminut, kuten olin suunnitellut.

Torstai 14.3.2019

Tämän päivän suunnitelmissa on jatkaa eilen aloittamani turvataulukoiden tyylimuutoksia, ja saattaa ne valmiiksi. Otan lisää tikettejä työn alle, mikäli ehdin. Kaikki uudet brändiuudistukseen liittyvät muutokset pitäisi tästä edes viedä uuteen feature-haaraan.

Aloitin päivän turvataulukoiden tyylimuutoksen työstämisellä. Niissä oli paljon pientä viilailua, mutta ne eivät olleet kovin haastavia. Hankalampi osuus oli muokata taulukon rakennetta mobiilinäkymään sopivaksi. Taulukko on rakennettu käyttäen flexboxia ja grid-asettelua. Tein funktion, jolla seurataan, mikä on taulukon leveys kullakin hetkellä. Funktiossa asetin tyylejä, joiden avulla haitarien leveydet muuttuivat. Tein vielä viimeisiä viilauksia taulukon tyyleihin, kuten hiiren hover-efekti, jossa kursorin viettäessä otsikon päälle, otsikko vaihtaa hieman väriä.

Kirjasin ylös työstämiäni tehtäviä, ja huomasin samalla, että olin unohtanut tehdä loppuun eilen aloittamani korjaukset haitarivalikoihin. Tein tarvittavat korjaukset haitarin

fonttikokoon, padding-arvoon, sekä leveyteen, jolloin haitari näkyi prototyypin mukaisesti kaikilla ruudunleveyksillä.

Olin nyt saanut valmiiksi sprinttiin valitsemani tiketit. Enää tehtäväkseni jää puskea muutokset feature-haaraan. Kysyin varmistukseksi neuvoa kollegaltani, minkä jälkeen puskin muutokset onnistuneesti eteenpäin. Tämän jälkeen siirryin tutkimaan vakuutuslaskuri-tikettiä, jonka Prove oli uudelleenavannut testaamisen jälkeen. Ongelmana oli, että kaikki tekstit eivät kaikilla mobiiliselaimilla näkyneet oikein.

Työpäivän lopulla ehdin vain aloittaa vakuutuslaskurista löytyneen bugin tutkimista. Ongelman ratkaisu jää huomiselle. Päivän tehtävät olivat valtaosin tyylimuutoksia ja niiden pikkuviihauksia. Huomasin, että olen kehittynyt ongelmakohtien paikantamisessa ja korjausten luomisessa. Pystyn lähes välittömästi päättämään, mitä tyylimuutoksia on tehtävä, jotta haluttu lopputulos saadaan aikaan. Minulla on vielä opittavaa joidenkin tyylimäärittelyjen toimivuuden ymmärtämisestä. Osaan käyttää oikein monia eri tyylimäärittelyjä, mutta en aina osaa sanoa, onko kaikkia elementtejä mahdollista muokata.

Perjantai 15.3.2019

Suunnitelmissani on tänään korjata vakuutuslaskurin ongelma, joka jäi eilen kesken. Sen jälkeen aion työstää brändiuudistukseen liittyviä tikettejä. Tälle päivälle ei ole merkitty palaveria, eikä muitakaan tapaamisia.

Testaaja oli löytänyt vakuutuslaskurista ongelman: laskurin kentät eivät aina nollaantuneet, mikäli käyttäjä palasi ostoputkesta takaisin vakuutuslaskuriin. Mielestäni oudointa oli se, että osa selaimista palautti arvot eri tavoin kuin toiset. Joillakin selaimista molemmat kentät tyhjenivät, kun toiset selaimet pitivät valinnat muistissa. Sain ongelman korjattua sillä, että sivun latautuessa tarkistettiin, oliko vakuutuksen tyyppi valittu. Mikäli tyyppi oli valittu, vakuutuslaskurille lisättiin tyyppiä vastaava luokka, jolloin toinen select sekä laske hinta -painike muuttuivat näkyviksi. Vein muutokset testiserverille, ja annoin tiketin takaisin Proven testattavaksi.

Seuraavaksi siirryin tekemään staattisten sivujen sisältönostojen ilmeen muutosta. Sisältönostojen tekninen nimi on landingpagebanner. Nimi tulee elementin sijoituskohteesta landing page eli etusivu, ja elementin tyylistä banner. Landingpagebannereita käytetään pääasiallisesti etusivulla, mutta niitä on mahdollista lisätä muillekin sivuille. Sisältönostojen tarkoituksena on tuoda esille jotain sivuston kohdetta tai tuotetta. Tiketissä oli kerrottu, minkä näköinen ulkoasun tulisi olla milläkin näytön leveydellä. Uusin sisältönostoihin

tahdottiin yksi linkkipainike lisää. Landingpagebannerin väriteemaa täytyi myös pystyä muuttamaan Contentfulista erikseen jokaiselle bannerille.

Sisältönostoihin tarvittavat muutokset olivat tyylimuutoksia. Aloitin vaihtamalla bannerin leveyden koko ruudun levyiseksi, ja vaihdoin värit vaihtamalla vanhoja tyylejä, samalla poistaen tarpeettomia tyylejä. Sain kaiken näyttämään hyvältä. Seuraavaksi oli tehtävä väripalettien luominen. Sisältönostoilla voi olla kolmea eri väripalettia: mintunvihreä, tummansininen, tai valkoinen. Tein Contentfuliin valikon, josta voi valita minkä värin tahtoo asettaa, ja valitun värin nimi tulee elementille luokkana, minkä perusteella landingpagebanner saa tyylejä.

Kaikki tarvittavat muutokset olivat tyylimuutoksia. Aloitin vaihtamalla bannerin leveyden koko ruudun levyiseksi, ja vaihdoin värit vaihtamalla vanhoja tyylejä, samalla poistaen tarpeettomia tyylejä. Sain kaiken näyttämään hyvältä. Seuraavaksi oli luotava väripalettien valitseminen. Sisältönostoilla voi olla kolmea eri väripalettia: mintunvihreä, tummansininen, tai valkoinen. Olikin mietittävä, että mikä on kestävin ratkaisu asettaa tuotenostoille väripaletti Contentfulista käsin. Koin, että yksinkertaisin ja varmin ratkaisu on lisätä sisältönoston templateen luokka, jonka perusteella värit vaihtuvat. Sain kuulla UI-designerilta, että myös elementeissä olevien painikkeiden väriä tulisi pystyä vaihtamaan Contentfulista, mikä ei ollut tullut esille tiketissä. Kollegaltani sain vinkkiä, että hän oli jo ehtinyt luoda toiseen Handlebars-tiedostoon ehtolauseen, jolla tällaisen toiminnallisuuden voi luoda helposti. En ehtinyt luomaan linkkipainikkeiden tyylien valintaa Contentfuliin, ja jatkaisin sitä ensi viikolla.

Landingpagebannerin jälkeen ryhdyin työstämään turvataulukoita. Contentfulissa taulukon nimi on feature table. Taulukko on rakennettu käyttäen flexbox- ja grid-asettelua. Bootstrapin grid koostuu containerista, riveistä ja kahdentoista pylvään mallista, joiden avulla saadaan luotua responsiivia rakenteita” (Bootstrap 2019b). Kahdentoista pylvään malli tarkoittaa sitä, että elementin voi kuvitella olevan jaettu kahteentoista yhtä suureen osaan, jotka voidaan jakaa muiden elementtien kesken haluamallaan tavalla. Pylväiden yhteenlasketun summan täytyy olla 12. Taulukon kaikkien neljän pylvään leveydet oli määritelty ennalta. Ensimmäiselle pylväälle oli määritelty luokka col-xs-6, mikä vastaa 6/12 eli puolet container-elementistä, ja kolmelle seuraavalle col-xs-2 eli 2/12, mikä vastaa yhtä kuudesosaa ruudusta kullekin.

Autovakuutuksen turvatasot

Turvat	XL	L	S
✓ Rengasturva	✓		
✓ Matkan jatkuminen	✓		
✓ Parkkiturva	✓		
✓ Lunastusturva	✓		
✓ Rahoitusturva	✓	✓	
✓ Luonnonilmiöturva	✓	✓	

Kuva 17. Turvataulukko pienellä ruudulla ennen korjauksia

Haitarivalikoiden ongelma johtui siitä, että mobiililaitteilla kolmen viimeisen pylvään leveys oli liian suuri, jolloin ensimmäiselle pylväälle jäi liian vähän tilaa, eivätkä kaikki tekstit mahtuneet sille varattuun tilaan (kuva 17). Ongelman korjaamiseksi tein funktion, jolla seurataan ruudun leveyden muutoksia. Funktiossa tarkistettiin, mikä on taulukon leveys kullakin hetkellä, ja mikäli ruutu oli alle 520 pikseliä leveä, pylväiden luokille tehtiin muutoksia. Olin kokeillut selaimen kehitystyökaluilla, että kaikki tekstit mahtuvat paremmin ruudulle, jos ensimmäisen pylvään leveyttä kasvattaa, ja kolmen viimeisen pylvään leveyttä pienentää. Loin funktion, jolla seurasin ruudun koon muutosta. Funktio poisti ensimmäisen pylvään col-xs-6, ja lisäsin sen tilalle col-xs-9, kun ruudun leveys oli alle 520 pikseliä. Näin ensimmäisen pylvään leveys oli isompi. Samassa funktiossa poistettiin kolmen viimeisen pylvään col-xs-2 luokat, ja niiden tilalle lisättiin col-xs-1, jolloin viimeisten pylväiden leveys ruudulla pieneni.

Autovakuutuksen turvatasot

Turvat	XL	L	S
✓ Rengasturva	✓		
✓ Matkan jatkuminen	✓		
✓ Parkkiturva	✓		
✓ Lunastusturva	✓		
✓ Rahoitusturva	✓	✓	
✓ Luonnonilmiöturva	✓	✓	

Kuva 18. Turvataulukko pienellä ruudulla korjausten jälkeen

Korjausten jälkeen ruudun leveyden ollessa tasan tai alle 520 pikseliä, taulukon pylväiden leveydet muuttuivat, jotta linkkitekstit mahtuisivat yhdelle riville. Onnistuin saamaan taulukon pienennyksen toimimaan oikein (kuva 18). Totesin, että taulukoiden olisi hyvä myös palata niiden alkuperäiseen leveyteen, mikäli ruudun leveyttä säätää pienemmäksi ja suuremmaksi. Olin tehnyt koodiin alkuperäisten luokkien poistot, mutta minun täytyi tehdä myös alkuperäisten luokkien palautukset. Käytin aiemmin luomaani koodia uudelleen, vaihtaen vain lisättävien luokkien nimet. Pylväille lisätään alkuperäiset luokat ainoastaan, jos ruudun leveys on yli 520 pikseliä, ensimmäisellä pylväällä on luokka col-xs-9, ja muilla pylväillä on luokat col-xs-1. Pienellä lisäyksellä sain turvataulukon muuttamaan leveyttä ruudun kokoa pienentäessä tai kasvattaessa. On hyvin epätodennäköistä, että sivustolla käyvät asiakkaat suurentaisivat tai pienentäisivät ruutua, mutta mielestäni on tärkeää varmistaa, että taulukko näyttää kaikissa tilanteissa oikealta.

Huomasin oppineeni valtavasti ympäristömme käyttäytymistä. Olen oppinut paremmin huomaamaan, mitä kaikkea minun tarvitsee tehdä, jotta saisin haluamani muutokset aikaan ja toimimaan oikein. Huomasin, että olen oppinut tekemään töitä itsenäisesti, mutta toisinaan tarkistan ratkaisuni ja suunnitelmani kollegallani.

Viikkoanalyysi

Valtaosa työviikosta kului vakuutuslaskuria työstäessä ja siihen liittyvien ongelmien korjaamisessa. Vakuutuslaskurin lisäksi ehdin työstämään Landing Page Bannereita, jotka olivat mielestäni näyttävyydeltään merkittävämpiä kuin vakuutuslaskuri, mutta teknisesti huomattavasti helpompia toteuttaa. Viikon aikana työstin myös turvataulukoiden ulkoasua. Turvataulukoiden toiminnallisuuteen minun ei tarvinnut tehdä muutoksia.

Viikon aikana kohtaamani merkittävä ongelma vakuutuslaskurissa oli, että vakuutuksen kohteen valinta ei näkynyt oikein. Koodissa kuuntelin, eli seurasin kaikkia klikkauksia, jotka kohdistuivat vakuutuslaskurin vanhempielementtiin, jonka alla kaksi alasvetovalikkoa sijaitsivat. JavaScriptissä hyödynsin murupolkunavigaatioissakin käytettyä jQuery:n on-metodia, jolla oli mahdollista kuunnella kaikkia muutoksia valitun elementin sisällä. Koodi seurasi siis kaikkia klikkejä vakuutuslaskurissa. Ongelmana oli, että ylemmän valikon klikkauksia lisäksi myös alemman dropdown-valikon klikkauksia kuunneltiin samassa funktiossa. Tämä johti siihen, että osa koodia, joka "nollasi" ensimmäisen valikon eli vakuutuksen kohteen, laukesi myös niissä tapauksissa, kun alempaa valikkoa klikattiin. Alempaa dropdown-valikkoa, eli vakuutuksen kohdetta valittaessa koodi huomasi, että vanhempielementissä on tapahtunut muutos, ja vakuutuksen kohteen alasvetovalikko oli nollattava.



Kuva 19. Ongelma vakuutuksen kohteen valinnassa vakuutuslaskurissa

Vakuutuksen kohteen valitseminen sai aikaan sen, että alemmassa dropdownissa tuli aina näkymään teksti "Valitse kohde", sen sijaan että siinä olisi valitun kohteen nimi (kuva 19). Huomasin, että olin tehnyt liian paljon muutoksia yhden funktion sisään, joten minun oli jaettava valikkojen muutokset omiksi funktioikseen. Korjasin ongelman sillä, että siirsin alemman listan nollauksen pois vakuutuslaskurin muutoksia kuuntelevasta funktiosta. Tein uuden funktion, jolla kuunneltiin vain ylemmän listan muutoksia. Loin toisen erillisen funktion, jolla alempi valikko nollattiin. Nyt koodissa alemman valikon nollausta kutsuttiin vain silloin, kun ylemmässä valikossa oli tapahtunut muutoksia. Näin varmistin, ettei alempi lista nollaannu väärään aikaan. Seuraavaksi minun oli saatava laske hinta -linkkipainike näkymään oikein.

Aiemmin tekemäni JavaScript-koodit olin tehnyt vakuutuslaskuri.js-tiedostoon, jonka nimesin uudelleen vakuutuslaskuri-scripts.js. Tein uuden tiedoston, jonka nimesin vakuutuslaskuri.js, jossa tarkoitukseni on hakea vakuutuslaskuriWidgetille kunkin

vakuutusvaihtoehtoon URL-osoite. VakuutuslaskuriWidget on moduulin nimi Contentfulissa. Kävin koodissa läpi kaikki tiedostot, ja etsin niiden sisällä kaikki moduulit, joista etsin ne, joiden id oli vakuutuslaskuriWidget. VakuutuslaskuriWidget pitäisi löytyä vain yhden sivun sisältä, mutta harmikseni ei ole muuta tapaa löytää moduulia, kuin käymällä läpi jokainen tiedosto perin pohjin. Löysin tarvittavan datan nopeasti ja helposti, sillä pystyin onnekseni uudelleenkäyttämään murupolkunavigaatiossa käytettyä tiedostojen hakua vaihtamalla vain aiemmin haetun murupolku id:n vakuutuslaskuriWidgetiksi. Seuraavaksi pohdiskelin, miten parhaiten saisin tiedot tuotua ruudulle.

Ajattelin, että on tarpeettoman monimutkaista ensin tuoda osa tiedoista vakuutuslaskuri.hbt -sivupohjan mukana staattiselle sivulle, minkä jälkeen vielä tuotaisiin erikseen JSON-muodossa URL-osoitteet Contentfulista. Keskustelin kollegani kanssa, ja hän vahvasti ajatukseni, että on parempi tuoda URL-osoitteen tiedot suoraan templatessa. Poistin uuden vakuutuslaskuri.js-tiedoston, ja uudelleen nimesin aiemmin tehdyn tiedoston takaisin alkuperäiseen. Handlebars-tiedostossa lisäsin vakuutuksen kohde -select-elementin vaihtoehtoille data-url -attribuutin, jolle annettaisiin arvoksi kunkin tuotteen laskurin osoite. Sain jokaiselle option-elementille lisättyä vakuutuksen kohdetta vastaavan URL-osoitteen, joka vie ostoputkeen.

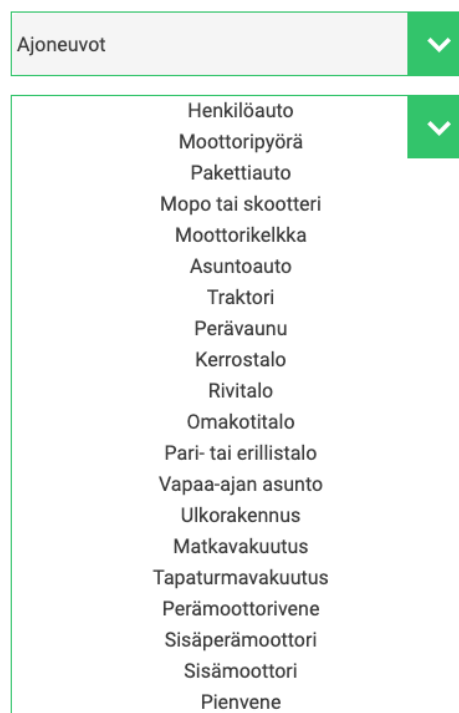
Seuraavaksi työstin koodia, jolla valitun kohteen data-url -attribuuttiin lisätty osoite saadaan lisättyä Laske hinta -painikkeelle. Kokeilujen ja Googletuksen avulla sain lopulta luotua ratkaisun, joka toimi. Kävin läpi vakuutuksen kohde -alasvetovalikon elementit silmukalla, ja poimin niistä data-urlin, jos kyseisen elementin value-arvo vastasi dropdown-listan valinnan arvoa, jolloin sain oikean URL-osoitteen. Tallensin tämän tiedon url-nimiseen muuttujaan, ja lisäsin sen seuraavalla rivillä linkkipainikkeen href-attribuutille, jolloin Laske hinta -painiketta painamalla siirtyi oikeaan ostoputkeen.

Olin saanut vakuutuslaskurin toiminnallisuuden toimimaan, mutta vielä oli luotava teksti jokaiselle vaiheelle, ja saada ne näkymään oikein. Tekstin tulee vaihtua sen perusteella, mitä alasvetovalikoista on valittu. Lisäsin Contentfuliin kolme tekstikenttää, joihin voidaan syöttää teksti näytettäväksi otsikon alla. Tein tyylimuutokset sekä JavaScript muutokset, jotta tekstikentät näytetään vain yksi kerrallaan. Tässä kohtaa oli pieniä haasteita saada teksti näkymään oikein, mutta ongelma ei onneksi ollut vaikea korjata.

Pienen testailun jälkeen totesin, että on parempi, mikäli tekstikentät syötetäänkin markdown-tyylisesti Contentfulin editorikenttään. Markdown-kenttä ajetaan Handlebars-tiedostossa erillisen koodin läpi, jolloin tekstin sekaan on mahdollista lisätä html-elementtejä. Näin pitkien sanojen väliin pystyi lisätä *soft hyphenin* eli pehmeän tavutuksen, joka ei

näy käyttäjälle. Soft hyphen luo tekstiin paikan rivinvaihdolle ja tavuttaa sanan, mikäli se ei mahtuisi sille varattuun tilaan (Korpela 2014). Tein korjauksia tyyleihin; muutin valikoiden värejä, sekä mobiilinäkylässä elementtien leveyksiä ja korkeuksia.

Testauksessa ilmeni ongelma Safari-selaimella, jolla option-elementtejä ei ole mahdollista muokata tyylimäärityillä. Ongelma muistutti paljon Oma Palvelun kalenterin kanssa kohtaamiani ongelmia. Kalenteripainikkeessa minulla oli haasteita saada tehtyä tyylimuutoksia inputille. Aiemmasta oppineena tarkistin heti netistä, mitä kaikkea option-elementille on mahdollista tehdä. Kalenterikenttien tapaan vakuutuslaskurin select-elementin optioneita ei ole mahdollista muokata tyyleillä. ”Select-elementit piirtyvät eri tavoin riippuen selaimesta ja käyttöjärjestelmästä, mikä johtuu siitä, että elementin näkymä periytyy käyttöjärjestelmästä, sen sijaan, että se olisi selaimen välittämä (Roberts 21.8.2014). Ongelmani Safari-selaimen kanssa ilmeni siten, että minkä tahansa vakuutuksen tyyppin valitessaan, aukesi lista kaikista vakuutuksista.



Kuva 20. Vakuutuslaskurin ongelma vakuutuksen kohteiden esittämisessä

Tarkoituksena oli saada näkyviin vain valittua vakuutuksen tyyppiä vastaavat vakuutuksen kohteet, sen sijaan ruudulle aukeaa lista kaikista vakuutuksista (kuva 20). Tein ongelmaan korjauksen, jossa kaikki muut kuin valitun tyyppin vakuutukset muutetaan disabled-tilaan, jolloin ne ole klikattavissa. Tämä ei kuitenkaan tuntunut järkevältä ratkaisulta. Kysyinkin kollegaltani neuvoa, ja hän oli samaa mieltä kanssani, että olisi parempi, jos jokaisen vakuutustyyppin vakuutukset olisivat omissa valikoissaan. Jokainen valikko piilotettaisiin, tai

asetettaisiin näkyväksi yksitellen, sen perustella mikä vakuutuksen tyyppi oli valittu. Ratkaisu oli onnekseni suhteellisen helppo toteuttaa. Kollegani osoitti templatesta kohdan, johon lisäsin select-elementin, jolloin jokainen vakuutustyyppi sai oman select-valikkonsa.

Muutin tyylejä siten, että kaikki option-elementit ovat oletuksena näkyvillä. Aiemmin optionit olivat piilotettu näkyvistä tyylimuutoksilla. Tein JavaScriptillä muutokset selectien piilotukseen, mutta huomasin, että saman asian voi tehdä myös tyylitiedostossa. Sain tyylitiedoston muokkauksilla näytettyä oikean select-valikon, mutta jos vakuutuksen tyyppiä vaihtoi, niin laske hinta -painike ei enää tullut uudelleen näkyviin. Minulla kesti pitkään löytää keino, jolla saada valitun vakuutuksen kohteen index-numeron. Select-elementeilla on oma index-numero, mikä kertoo, minkä avulla voidaan selvittää, mikä option-elementeista on valittuna (MDN web docs 2019c). Tarvitsen valitun vakuutuksen index-numeroa, jotta pystyin liittämään oikean linkin linkkipainikkeeseen, sekä esittämään oikeat tekstit oikeaan aikaan. Sain korjaukset tehtyä, ja vein muutokset testiin. Testiserverillä huomasin, että teksti ei vaihtunut aina oikein. Tein tarvittavat muutokset koodiin, ja sain tekstit taas näkymään oikein lokaalisti.

Vakuutuslaskurissa oli testauksen myötä noussut esiin ongelmia. Vakuutuslaskuri ei osannut näyttää oikein tehtyjä valintoja, kun sivulle palasi takaisin ostoputkesta. Ongelma johtui siitä, että sivulle palattaessa vakuutuslaskurilla ei enää ollutkaan sitä luokkaa, joka sillä sivulta poistuessa oli. Ratkaisuksi mietin, että kaikki valinnat poistetaan, aina kun sivulta poistuu, tai sivulle palaa selaimen paluupainikkeella. Kenttien tyhjentäminen ei ole optimaalinen ratkaisu, sillä olisi asiakkaan näkökulmasta mielekkäämpää, jos valinnat eivät olisi kadonneet sivulle palattaessa. Sen sijaan tyylit tulee saada toimimaan oikein, mikäli sivulle palataan, jotta valinnat pysyisivät näkyvissä.

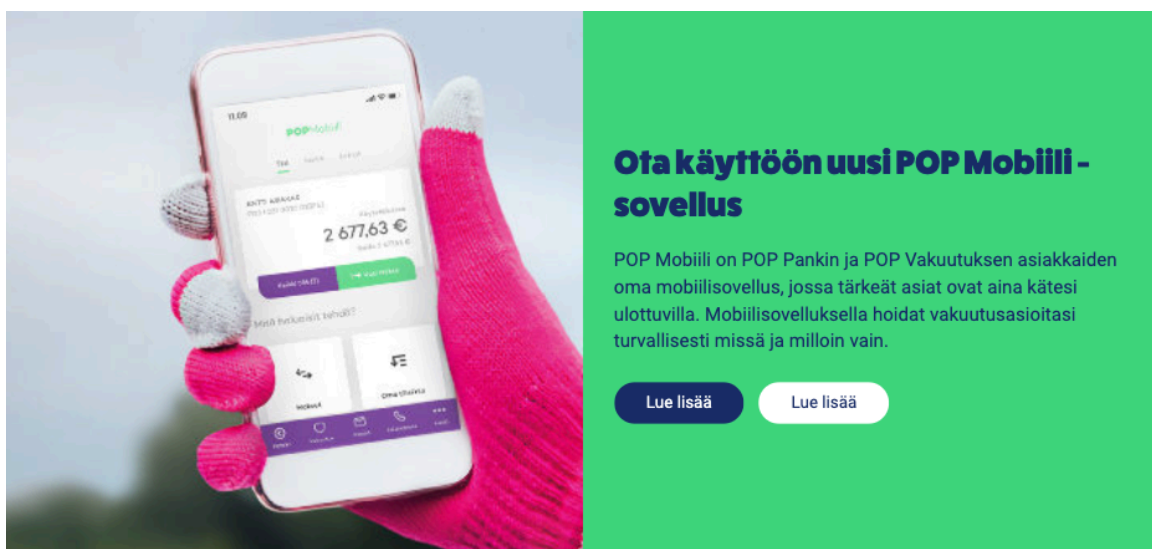
Vakuutuslaskurin kenttien palautuminen alkuperäiseen oli hieman hämmentävää, sillä se tapahtui eri selaimilla eri tavalla. Toiset selaimista tyhjensivät molemmat kentät, toiset selaimista eivät tyhjentäneet mitään. Käytin aikaa ongelman tutkimiseen, ja mietiskelin, mikä ongelmat saa aikaan. Huomasin, että select-elementtien valinnat olivat jääneet voimaan, kun sivulle palasi. Tämä itsessään ei ollut ongelma, mutta ongelma oli siinä, että valikot eivät tulleet näkyviin oikein, koska sivulle palattaessa mitään elementtiä ei ollut vielä klikattu. Vakuutuslaskuri saa eri luokan sen perusteella mikä kohde on klikattu, ja ennen kuin klikkaus on tehty, mitään tyylejä ei ole lisätty. Korjasin tämän ongelman siten, että mikäli jokin kohde on valittu, vaikka vakuutuslaskurin luokka puuttuisikin, lisätään vakuutuslaskurille oikea luokka.

3.10 Seurantaviikko 10

Maanantai 18.3.2019

Tänään suunnittelen jatkavani viime viikolla kesken jäänyttä tikettiä liittyen sisältönostojen eli landing page bannereiden päivityksiin. Tavoitteekseni asetan saada sisältönostojen ulkoasun näyttämään oikealta kaikilla selaimilla ja jokaisella tuetulla ruutukoolla.

Sisältönostojen linkkipainikkeisiin tahdottiin Contentfuliin valikko, josta on mahdollista vaihtaa painikkeen väriä. Loin Contentfuliin lisäkentän, ja sain linkkipainikkeen värivalinnan toimimaan oikein. Contentfulissa on nyt mahdollista valita painikkeelle väri alasvetovalikosta. Kaikilla sisältönostoille ei välttämättä tule toista nappulaa, mutta sellainen täytyy olla mahdollista lisätä Contentfulista. Minun täytyi seuraavaksi tehdä samanlaiset muutokset Contentfuliin, kuin olin tehnyt ensimmäiselle painikkeelle, mikä ei ollut vaikeaa.



Kuva 21. Sisältönosto kahdella painikkeella

Loin helposti ensimmäisen painikkeen värivalinnan lisäksi myös toiselle painikkeelle Contentfuliin ominaisuuden, jolla voi valita kakkospainikkeen värin. Nyt sivuston sisältönostoihin voi lisätä toisen painikkeen valitsemallaan tekstillä, ja jonka painikkeen värin voi vaihtaa (kuva 21).

Seuraavana tehtävänäni oli tekstien ja kuvien asettelu oikeille paikoilleen kaikilla selaimilla. Yritin saada ulkoasun kuntoon jokaisella ruutukoolla ja selaimella, mutta kohtasin suuria ongelmia IE11 eli Internet Explorer 11 -selaimella. Tähän kaikkeen meni paljon enemmän aikaa, kuin mitä olin päivän alkaessa kuvitellut. Kaikkein työläintä oli saada

IE11 näyttämään oikealta. Yksi merkittävä syy työn pitkään kestoon oli BrowserStackin hitaus. Selain toimi todella hitaasti, ja jokainen hiiren liikautus ja tekstin lisäys tuli viiveellä.

Saavutin päivän tavoitteet mielestäni hyvin. En odottanut IE11 aiheuttavan niin paljon ongelmia, kuin se lopulta aiheuttikaan. Olin tietoinen, että kyseinen selain tukee huonosti flex-tyyppistä ulkoasua, mutta korjaukseen vaadittava työmäärä yllätti.

Tiistai 19.3.2019

Suunnittelen tänään tekeväni ainakin yhden brändiin liittyvän tiketin kokonaisuudessaan, ja vien muutokset lab-serverille. Aloitan työt alanavigaatioon liittyvällä tiketillä, joka oli osoitettu minulle. Käyn läpi uudet sähköpostiviestit, sekä tarkistan, onko tälle päivälle sovittu tapaamisia. Osallistun tänään palaveriin, jossa käymme läpi brändiuudistuksen aikaansaannoksia. Tavoitteenani on saada alanavigaation korjaukset valmiiksi tämän päivän aikana.

Tuntui, että tuskin kerkesin aloittamaan työskentelyä, sillä tulin vasta yhdeksältä töihin, ja brändiin liittyvä tapaaminen oli sovittu kello 10.30. Palaverissa kävimme läpi aikaansaannoksia, ja mitä kaikkea olisi vielä tehtävä, ennen kuin muutokset pystytään viemään tuotantoon. Kollegani antoi arvion, että työhön menisi vielä ainakin yksi viikko lisää.

Palaverin jälkeen ryhdyin työstämään alanavigaation muutoksia. Olin muutamia viikkoja sitten tehnyt alanavigaatioon muutoksia rakenteeseen ja tyyliin. Nyt tehtävänä on asettaa värit ja kuvakkeet oikein. Kaikki tämä vaikutti helpolta. Aloitin helpoimmasta, eli väleistä. Aiemmin tekemäni värimuutokset olivat valtaosin turhia, ja pystyin poistamaan niitä. Poistin ylimääräiset värimuutokset. Sisäänkirjautumispainikkeen kanssa oli pieniä vaikeuksia, sillä sen tyylit olivat sidoksissa sivun rakenteeseen, mutta sain senkin korjattua pienen väännön jälkeen. Olin nyt poistanut ylimääräiset värityylit teksteiltä ja painikkeilta. Tarkistin vielä, miltä PHP-sivut näyttävät. Kävi ilmi, että kaikki värit olivat väärin. Kaikki tyylitiedostot eivät näemmä toimineet odotetusti. Palautin kaikki poistamani värimuutokset, ja varmistin, että ne olivat brändikirjan mukaisia.

En saavuttanut päivän tavoitettani, sillä en saanut alanavigaatiota täysin toimimaan. Sain sen toimimaan osittain; alanavigaatio näkyi oikein staattisilla sivuilla, muttei PHP-sivuilla. Pysyin suunnitelmassani työstää yhtä tikettiä, enkä harhautunut tekemään jotain, mitä en ollut suunnitellut tekeväni. Koin päivän palaverin hyödylliseksi, sillä sain itsekkin paremman käsityksen siitä, mitä olimme saaneet aikaan, ja mitä oli vielä tehtävä.

Keskiviikko 20.3.2019

Suunnitelmani on tänään saattaa valmiiksi eilen työstämäni alanavigaatio-tikettiä, ja ottaa työn alle jotain uutta. En vielä osaa sanoa tarkalleen, mitä kaikkea tulen alanavigaation lisäksi työstämään, sillä olen jo tehnyt valmiiksi kaikki minulle osoitetut tiketit. Tämänpäiväisessä sprint-palaverissa tulen ottamaan jonkin jäljelle jääneistä brändiin liittyvistä tiketeistä. Kaikkien bränditickettien tulisi olla valmiita tänään alkavan sprintin puolivälissä, eli viikon päästä. Tavoitteenani on korjata eilen aloittamani alanavigaatio kuntoon, minkä lisäksi saada ainakin jokin toinen bränditicketti valmiiksi.

En ollut muistanut eilen viimeistellä alanavigaation tikettiä. Muutin alanavigaation tiketin tilan ratkaistuksi, ja annoin sen Provelle testattavaksi. Huomasin, että ulkoasussa oli joitain pieniä poikkeamia, joita en ollut eilen huomannut. Leipätekstien fonttikoot ja värit eivät olleet oikein kaikilla elementillä, ja listojen rivivälit olivat mobiilinäkymissä liian pieniä. Tein muutoksia tyyliihin, mutta se oli hidasta, sillä tyyli tiedostossa on jo yli neljäsataa riviä määrittäviä, mikä tekee toisinaan oikean säännön löytämisestä hankalaa.

Tyylimuutoksissa kesti odotettua pidempään, ja työni keskeytyivät sprint-palaverin vuoksi. Uuteen alkavaan sprinttiin en itse valinnut mitään, vaan kaikki brändiuudistuksiin liittyvät tiketit otettiin kerralla. Tehtävät jaamme kehittäjäkollegani kanssa sen mukaan, miten aikaa riittää.

Palaverin jälkeen jatkoin alanavigaation tyylien korjaukset valmiiksi. Puskin tehdyt tyylien viimeistelyt lab-haaraan ja rakensin sivustot uudelleen. Seuraavaksi työstän linkkipatteristoa, joka on tekniseltä nimeltään action-navigation. Käyttöliittymäsuunnittelijamme oli piirtänyt uudet brändin mukaiset mallit, minkä näköisiksi nykyiset palikat olisi muutettava. Nykyisin koodimme luo kuvakkeet sen pohjalta, mikä luokka elementillä on. Pohdiskelin hetken parasta ratkaisua ongelmaani, ja päädyin siihen lopputulokseen, että teen muutokset Contentfuliin, jottei elementit enää saisi aiempia tyylejä, sen sijaan että muokkaisin ole-massa olevia tyylejä. Tein tarvittavat muutokset tyyli tiedostoon, sekä Contentfuliin ajoneuvojen action-navigationin osalta. Seuraavina tehtävinä on tehdä viidelle jäljelle jääneelle action-navigationille vastaavanlaiset muutokset.

Torstai 21.3.2019

Suunnittelen tänään tekeväni linkkipatteriston tiketin mukaiseksi. Tavoitteena tänään on tehdä Action-navigation linkkipatteristo valmiiksi.

Linkkipatteristosta tahdottiin helpommin muokattava, eli HTML-koodin määrää tulisi vähentää Contentfulissa. Ongelman ratkaisu on rakentaa koodi Handlebars-tiedostosta. Action-navigationin rakentavasta Handlebars-pohjasta oli tehtävä laajempi lisäämällä siihen koodia uusille kentille, joita tulen lisäämään Contentfuliin. Tavoitteena on, että Contentfulissa ei tarvitsisi kirjoittaa niin paljoa HTML-koodia, sillä se on hankalasti ylläpidettävää ja muokattavaa. Linkkipatteriston luovan tiedoston nimi on action-nav.hbt, johon tein pienen muutoksen, jolla valitaan, piirretäänkö sivu käyttäen vanhaa HTML-koodilla kirjoitettua sisältöä, vai piirretäänkö sivu käyttäen uusia Action Navigation Item -tietotyyppiä.

Linkkipatteriston asettelussa kohtasin ongelmia, joiden ratkaisuun tarvitsin hieman apua kollegaltani. Käärin linkkilaatikot uuden div-elementin sisään, mikä mahdollisti linkkilaatikoiden sijoittamisen tiketissä esitetyllä tavalla.

Päiväni kului suunnitellusti linkkipatteristoa työstäessä, ja saavutin päivälle asettamani tavoitteen saada linkkipatteristo valmiiksi.

Perjantai 22.3.2019

Päivän suunnitelmissa on tehdä bränditikettejä. Otan työstääkseni vakuutusasiakirjat esittävän elementin, eli page-navigationin tiketin. Tehtävä ei vaikuta vaikealta, ja päivän tavoitteenani on saada vakuutusasiakirjojen tyylimuutoksia koskeva tiketti tänään valmiiksi.

Aloitin tekemällä muutoksia rakenteeseen lisäämällä sivun rakentavaan pohjan koodiin uuden otsikkoelementin, johon otsikko tullaan piirtämään, sekä Contentfuliin sitä vastaavan kentän. Otsikko on helpompi ja nopeampi vaihtaa myöhemmin Contentfulista, kuin koodista. Lisäsin elementeille luokkia, jotta niiden tyylimäärittäminen olisi helpompaa. Page-nav.scss -tyylitiedosto, jolla tyylit määritellään, oli poikkeuksellisen vähärivinen. Tein uusia tyylejä BEM-nimeämiskäytännön mukaan. Yritin ensin elementtien asettelua Bootstrapin column-luokilla, mutta ne eivät itsessään riittäneet. Pienten lisäysten jälkeen sain linkit asettumaan oikeille paikoille jokaisella ruutukoolla tikettiin linkatun prototyypin mukaisesti. Puskin muutokset testiin, ja siirryin muihin tehtäviin.

Jirassa oli avattu bugitiketti, jossa ilmoitettiin, että aiemmin tekemäni muutokset tuotenoitoin, eli landing page bannereihin näkyivät väärin. Testaajalle ei ollut kerrottu, että aiemmin käytetty brand2019-haara ei enää ollut käytössä, vaan muutokset tehtiin suoraan develop-haaraan. Kerroin tästä testaajalle, annoin linkin oikeaan osoitteeseen, ja asetin tiketin resolved-tilaan, ja osoitin tiketin takaisin testaajalle.

Sama testaja oli löytänyt mobiilinäkymässä virheen alanvigaatiosta, jossa haitarivalikko sulkeutui välittömästi, mikäli ruutua liikutti. Koitin etsiä Internetistä vastauksia ongelmaan, mutta en löytänyt suoraa vastausta. JavaScript-koodissa seurataan ruudun koon muutoksia, minkä arvelin aiheuttavan ongelman. En kyennyt toistamaan ongelmaa, mutta tein koodiin muutoksen, jossa ruudun koon muutoksen sijaan seurataan pelkästään ruudun leveyden muutosta. Ratkaisin tiketin, ja siirryin seuraaviin tehtäviin.

Kävin läpi sivustoa, ja etsin poikkeamia. Huomasin, että action-nav -elementtien linkeille oli jäänyt alleviivaus, kun cursorin vei linkin päälle. Poistin alleviivauksen. Vakuutuslaskurin Laske hinta -painike oli vielä vanhoilla tyyleillä, mutta käyttöliittymäsuunnittelijamme sanoi, että vakuutuslaskuriin oli tulossa lisää muutoksia, joten en vielä tehnyt siihen muutoksia.

Viikkoanalyysi

Olin työstänyt viikon aikana monia brändiin liittyviä tikettejä. Haasteellisimmiksi ja työläimmäksi koin uudet sisältönostojen eli Landing Page Bannerin ulkoasumuutokset, ja action-navigation -linkkipatteriston. Koen onnistuneeni työssäni hyvin, sillä haasteista huolimatta sain luotua tiketeissä vaaditun lopputuloksen, ja korjattua mahdolliset esiin nousseet virheet. Koen, että olen oppinut sivuston rakenteista ja ympäristömme kokonaisuuksista paljon, minkä ansiosta tehtävät tuntuvat alati helpommilta.

Haasteeni sisältönostojen eli landing page bannereiden kanssa tuntuivat ärsyttäviltä, sillä osa ongelmista oli selainperäisiä. Tiketissä oli tarkkaan kerrottu, miten teksti tuli asetella sivulle, mutta kohtasin ongelmia Internet Explorer 11 -selaimella. Osassa ominaisuuksista lienee olevan ongelmia, joita tuntuu olevan hankala kiertää. Internet Explorer 11 on todettu ongelma, että tekstien keskittäminen saa sen valumaan vanhempielementin ulkopuolelle (Walton 2015).

Pyrin tekemään sisältönostojen ulkoasun tiketissä esitettyjen suunnitelmien mukaisesti, mutta pitkien tekstien asettelussa kohtasin ongelmia. Minun ei ollut mahdollista saada tekstiä keskitettyä, ja samanaikaisesti pitää elementtiä dynaamisena siten, että vanhempielementin olisi voinut kasvaa pitkän tekstin määrän mukaan.



Kuva 22. Sisältönostojen tekstit pienellä ruudulla

Ongelma ilmeni pienillä ruudunleveyksillä, joilla elementtien koot olivat pienempiä, ja teksti valui sille varatusta alueesta yli (kuva 22). Sisältönostolle oli tiketissä annettu maksimikorkeus ja leveys, joten elementtien korkeuden kasvattaminen ei tuntunut järkevältä. Keskustelin käyttöliittymäsuunnittelijan kanssa, ja päädyimme keskittämään elementin tekstit, jolloin vanhempielementin ei ollut mahdollista kasvattaa korkeutta. Päätös rajaa hieman tekstien pituutta, mutta elementeissä ei suunnittelijan mukaan tulisi olla pitkiä tekstejä.

Työstin tyylejä, ja onnistuin saamaan sisältönostot näyttämään oikealta niin Internet Explorer 11 kuin Chrome-selaimilla suurilla ruudunleveyksillä. Seuraavaksi tehtävänä on saada sivu toimimaan myös mobilinäköymällä.

Tutkin BrowserStackin avulla sivua IE11 eli Internet Explorer 11 -selaimella, ja muokkasin sivua selaimen sisäisten kehitystyökalujen avulla saadakseni sivun näyttämään oikealta. Selaimen kehitystyökaluilla tein väliaikaisia muokkauksia sivun elementteihin tyyleihin saadakseni sivun näyttämään oikealta. Tämän jälkeen tein samat väliaikaiset muutokset Chromeen sen kehitystyökaluilla, ja tarkistin, että sivu näytti samalta, ja vasta sen jälkeen tein muutokset tyylitiedostoon. Rakensin staattiset sivut uudelleen, ja päivitin selaimien sivut, nähdäkseni olivatko tulokset odotetut. Usein kävi niin, että tyylitiedostojen

muokkausten jälkeen sivu ei näyttänytkään samalta kuin aiemmin, ja jouduin tekemään muutokset molemmilla selaimilla uudelleen. Ongelma johtui siitä, että saatoin toisinaan unohtaa tehdä kaikki tarvittavat muutokset tyylitiedostoon. Tein muutokset uudelleen, kunnes sain kaikki tekstit ja kuvat oikeille paikoilleen jokaisella ruutukoolla. Pitkään kestäneen työstämisen jälkeen onnistuin saamaan pienten ruutukokojen näkymän toimimaan oikein.

Sisältönostojen lisäksi yksi hankalimmista viikon aikana työstämistäni brändiuudistuksen tiketeistä oli Action Navigation -linkkipatteristojen uudistus. Sivustollamme on ostoputkiin ohjaavia linkkipatteristoja, jotka tuli muokata uuden brändin mukaiseksi. Linkkipatteriston värimaailma, sekä linkkilaatikoiden asettelu tulisivat muuttumaan uuden brändin myötä.



Kuva 23. Linkkipatteristo ennen brändiuudistusta

Linkkipatteriston vaalean vihreät taustat, sekä vihreät tekstit ja kuvat tulisivat muuttumaan (kuva 23). Muutoksiin liittyvässä Jira-tiketissä linkkilaatikoiden taustaväri oli määritelty valkoiseksi, ja kuvat sekä teksti oli määritelty tumman sinisiksi. Contentfulissa linkkipatteriston nimi on action-navigation. Linkkipatteristo rakentui Contentfuliin kirjoitetun HTML-koodin perusteella. Tiketissä nostettiin esille tarve tehdä linkkipatteriston muokkaamisesta helpompaa Contentfulissa. Tavoitteena oli, että patteriston jokainen laatikko voitaisiin lisätä tai poistaa erikseen Contentfulin käyttöliittymästä, ilman että tarvitsisi kirjoittaa HTML-koodia.

Ohjelmistokehittäjä-kollegani oli viikolla luonut uuden sisältötyypin Product Teaser, joka vaikutti vastaavan tarpeitani, joten minun ei tarvinnut luoda kaikkea itse alusta alkaen. Kollegani luomassa Product Teaser -sisältötyypissä jokaisen laatikon otsikko ja teksti voidaan lisätä omiin tekstikenttiin. Kopioin kollegani luoman sisältötyypin ja nimesin sen Action Navigation Items:ksi. Poistin uudesta sisältötyypistä muutaman ylimääräisen kentän, joita ei tarvittu Navigation Item -sisältötyypissä. Lisäsin Contentfuliin yhden uuden kentän,

johon on mahdollista lisätä otsikko linkkielementtien yläpuolelle. Uusia linkkilaatikon elementtejä ja otsikoita ei näytetä, mikäli linkkipatteristolle ei ole asetettu Contentfulissa Action Navigation Item -sisältöä.

Loin uuden Handlebars-tiedoston, johon ryhdyin luomaan uutta pohjaa, jonka perusteella laatikot rakentuivat. Pohjan luominen oli alkuun hankalaa, sillä en meinannut onnistua saamaan Contentfulin sisältöä piirtymään sivustolle. Muokkasin Handlebars-tiedostossa sivun rakennetta saadakseni kuvat ja linkit oikeille paikoilleen. Kollegani oli jo lisännyt Contentfuliin kuvat, joita linkkipatteristossa tullaan käyttämään.

Koin, että linkkipatteriston uuden rakenteen luominen onnistui lopulta suhteellisen helposti. Tyylien korjaus sen sijaan kesti todella pitkään.



Kuva 24. Uudet linkkilaatikat keskitettynä

Muokkasin tyylejä siten, että linkkilaatikat ovat aina ruudun keskellä (kuva 24). Tavoitteena tosin olisi saada viimeisen rivin palikat sijoitettuna vasempaan laitaan sen sijaan, että ne olisivat keskitettyinä. Ongelmana oli, että vasempaan laitaan keskitettynä, kaikki linkkilaatikat sijaitsisivat sivun vasemmassa laidassa. Sain luotua tyylimuutoksen, jolla vain viimeisen rivin laatikot siirtyvät vasempaan laitaan. Muutokseni ei toiminut, sillä elementin siirtyivät liian paljon vasempaan, ohi niiden yläpuolella olevista keskitetyistä linkkilaatikoista. Sain kollegaltani neuvon, että voisin luoda linkkilaatikoille uuden ympäröivän kääre-elementin, jonka sisässä kaikki palikat olisivat sijoitettuna vasempaan laitaan.



Kuva 25. Linkkilaatikot sijoitettuna vasempaan laitaan

Linkkilaatikoita ympäröivä kääre-elementti mahdollisti linkkilaatikoiden sijoittamisen vasempaan laitaan, samalla kun itse ympäröivä kääre-elementti oli sivupohjassa keskitettynä (kuva 25). Laatikot olivat nyt asetettu oikeille paikoilleen, ja oli kokonaisuudessaan valmis.

Käytin hetken testatakseni, että linkit toimivat, ja ohjaavat oikeaan paikkaan. Tarkastelin ulkoasua vielä usealla eri selaimella ja mobiililaitteella. Tarkistusten jälkeen siirsin luomani muutokset develop-haaraan. Viimeistelin linkkipatteriston tiketin asettamalla sen tilan ratkaistuksi, ja luovutin tiketin testattavaksi.

4 Pohdinta ja päätelmät

Aloittaessani päiväkirjamuotoisen opinnäytetyön minulla oli ohjelmistokehityksen työkokemusta hieman yli seitsemän kuukautta. Arvioin tuolloin osaamiseni tason aloittelevaksi toimijaksi, ja koin usein tarvitsevani kollegaltani vahvistuksen tekemälleni työlle. Seurantaviikkojen aikana pääsin tekemään lähes joka viikko uusia haastavia tehtäviä. Tehtävien haasteiden ratkaiseminen ja niiden kirjaaminen seurantaviikkojen päiväkirjamerkintöihin auttoi oman kehittymisen arvioinnissa. Päiväkirjamerkintöjen viikoittaisten analyysien avulla opin tunnistamaan osa-alueita, joissa olin kehittynyt ja missä minulla oli vielä kehitettävää.

Opinnäytetyön ensimmäisellä ja toisella seurantaviikolla keskityin lähes yksinomaan Oma Palvelun kalenterin uudistamiseen, mitä aloittaessani tunsin epävarmuutta oman osaamiseni suhteen. Onnistuin ratkaisemaan työhön liittyvät ongelmat, ja työssä onnistuminen paransi luottamustani omiin taitoihini. Koin usein työhöni kuluvan aikaa enemmän, kuin olin siihen kuvitellut kuluvan. Tästä opin paremmin arvioimaan töiden suuruutta ja kokoluokkaa.

Kolmannella viikolla ryhdyin tekemään murupolkunavigaatiota, mikä oli opinnäytetyön aikana kohdatuista haasteistani kaikkein vaikein. Murupolkunavigaation valmiiksi saamiseen kului yli satakolmekymmentä työtuntia yli kolmen viikon ajalta. Työn laajuus oli yllättänyt minut, ja työ oli todella haastava. Työn valmistumiseen vaadittavan työmäärän ja kuluvan ajan arvioiminen oli ollut minulle vaikeaa, sillä en ollut aiemmin työstänyt vastaavan laajuista kokonaisuutta. Tuolloin opin, miten Contentfulin datamassaa on jäsenneltävä, sekä opin luomaan koodia, jolla pystyn etsimään datan seasta työhön tarvittavat olennaiset tietueet. Viikon kolme analyysissä tutkin hakukoneoptimoinnin merkitystä hakukonenäkyvyyteen, ja miten murupolkunavigaation muutokset tulisivat edesauttamaan näkyvyyden parantamisessa.

Kuudennella seurantaviikolla viimeistelin murupolkunavigaation, ja siirsin muutokset develop-haaraan. Kohtasin muutosten siirtämisessä ongelmia, joista opin Git-versionhallinnan haarojen käsittelyä ja hallinnointia. Koen oppineeni virheistäni, ja osaan nyt hallinnoida muutosten viemistä ja tuomista, sekä osaan siirtää tarvittaessa joko yhden tai useamman muutoksen haarasta toiseen. Murupolkunavigaation muutosten viimeistelyn jälkeen pureuduin ylänavigaation muokkaamiseen, josta sain entistäkin paremman käsityksen sivuston rakenteista.

Seitsemännellä viikolla uutena oppimiskokemuksena oli luoda A/B-vertailutesti verkkosivuillemme. VWO on tehokas työkalu A/B-vertailutestien luomiseen. Testien avulla on helppo saada tietoa asiakkaiden käyttäytymisestä sivustolla, minkä perusteella sivustosta voidaan luoda entistä käyttäjäystävällisempi. Onnistuin luomaan sivustollemme A/B-testin itsenäisesti JavaScriptillä, mikä entisestään kohotti luottamustani omiin kykyihini.

Poikkeuksellista seitsemännessä seurantaviikossa oli tiistailta keskiviikkoon siirretty tuotantopäivitys, mikä johtui murupolkunavigaatioissa löydetyistä ongelmista. Murupolkuun luomani linkkien rakennetta jouduttiin muuttamaan. Murupolun linkkien URL-osoitteiden virheellisyys oli oman oppimiseni kannalta edullinen sattuma, sillä opin linkkien relativisuudesta, niiden vaikutuksesta hakukoneoptimointiin. Tietämys hakukoneiden toiminnasta, ja hakukoneoptimoinnista on ehdottomasti yksi asioista, joita tulen tästä lähin ajattelemaan uutta sisältöä sekä muutoksia suunnitellessani ja luodessani.

Seurantaviikolla kahdeksan jatkoin loppuun sivuston URL-osoitteiden korjaukset sekä al navigaation muutokset. Koin viikon merkittävimmän työn olleen vakuutuslaskurin luominen, mikä vaati Contentfuliin uutta JSON-dataa, joka piti tuoda Handlebars-tiedostoon sivustolle rakennettavaksi. Vakuutuslaskurin työstö jatkui viikolla yhdeksän, jolloin loin vakuutuslaskurin toiminnallisuuden JavaScriptillä, jolla vakuutuslaskurin valinnat muokkasivat ulkoasua ja valikoiden vaihtoehtoja. Viimeisellä seurantaviikolla työtehtävät liittyivät brändiuudistukseen. Uudistukset olivat tyylimäärittelyjen muokkauksia ja olemassa olevien rakenteiden muokkauksia. Työstäessäni tutustuin BEM-nimeämiskäytänteisiin, jota tulen hyödyntämään luodessani uusia luokkia ja niiden tyylimäärittelyjä.

Kymmenen seurantaviikon aikana tekniset taitoni kehittyivät merkittävästi. Lähes kaikissa työstämissäni tehtävissä loin uusia tyylimäärittelyjä ja kirjoitin koodia PHP-, tai JavaScript-ohjelmointikielillä. Osassa tehtävistäni tein sisällönhallintajärjestelmään muutoksia ja lisäyksiä, joita olin tehnyt vain vähän ennen opinnäytetyön aloittamista.

Opinnäytetyön kymmenen seurantaviikon aikana kehityin paljon enemmän kuin olisin aloittaessani uskonut. Koin seurantaviikkojen aikana saaneeni alati haastavampia työtehtäviä, minkä huomaa auttaneen minua parantamaan työssä vaadittavaa teknistä osaamista. Huomattavimman kehityksen havaitsen tapahtuneen JavaScript-ohjelmoinnissa, minkä pystyn todentamaan onnistuessani seurantaviikoilla kahdeksan ja yhdeksän työstämään vakuutuslaskurin itsenäisesti omin avuin. Vielä viittä viikkoa aiemmin murupolknavigaatiota työstäessäni minulla oli haasteita luoda JavaScript-koodia. Nyt kykenen lukemaan ja kirjoittamaan koodia hyvin, ja paikannan nopeasti JavaScript-koodissa ilmentyvien virheiden alkuperän, ja osaan korjata ne.

Olen oppinut ylläpitämään Contentfulin olemassa olevia tietueita, luomaan uusia tietotyypejä ja niille vaadittavia kenttiä. Contentfuliin luodut staattiset sivut rakennetaan käyttäen Handlebars.js -kirjastoa, jota olen oppinut käyttämään hyvin. Tietämykseni tiedostojen eri komponenteista, niiden luomista kokonaisuuksista ja sivustorakenteista on parantunut merkittävästi. Huomaan entistä nopeammin löytäväni oikeat tiedostot ja koodin osat, joita minun tulee työstää saadakseni työtehtäväni suoritettua.

Oman työn analysointi on auttanut minua huomaamaan, että minulla on toisinaan tapana aliarvioida taitojani, ja epäillä ratkaisujani. Analyysien perusteella voin todeta, että onnistuneen ratkaisemaan kohtaamani haastavatkin ongelmat, mikä on kohottanut itseluottamustani ja luottamustani omiin taitoihini. Koen seurantaviikkojen aikana tulleeti itsevarmumaksi, ja sen myötä itsenäisemmäksi. Toisin kuin opinnäytetyötä aloittaessani, en enää koe tarvetta varmistaa jokaista luomaani muutosta tai uuden ominaisuuden ratkaisua kollegaltani, vaan työskentelen paljon aikaisempaa itsenäisemmän, ja luotan enemmän tekemäni työn laatuun. Olen oppinut paremmin tunnistamaan taitoni, ja sitä vastoin myös kehityskohteeni, minkä ansiosta osaan arvioida, milloin minun on hyödyllistä kysyä neuvoja.

Aloittaessani opinnäytetyön olin arvioinut itseni aloittelevan toimijan tasolla, sillä tarvitsin työssäni ohjeistusta, ja en itsenäinen suoriutuminen oli vajavaista. Kymmenen viikon jälkeen olen kehittynyt valtavasti, ja kykenen toimimaan itsenäisesti huomattavasti paremmin. Arvioin nyt suoriutuvani lähes taitavan suoriutujan tasoisesti, sillä minulla on syvällinen ymmärrys työtehtävistä, ja suoriutuminen on työtehtävien vaatimusten tasolla. Toisinaan on yhä tilanteita, joissa huomaan, että en vielä tiedä tarpeeksi syvällisesti kaikkea saadakseni työn itsenäisesti valmiiksi.

Opinnäytetyön aloittaessani olin itselleni kriittinen ja hetkittäin liian ankara ja tavoitteenani oli saada työt mahdollisimman nopeasti valmiiksi ilman virheitä. Seurantaviikkojen päiväkirjamerkinnoissa mainitsin töiden valmistumiseen kuluneen enemmän aikaa kuin olin toivonut. Asennoitumiseni työtehtäviini kuluvaan aikaan ja epäonnistumisiin on muuttunut opinnäytetyön aikana. Olen oppinut olemaan itselleni armollisempi, ja hyväksymään epäonnistumiset osana ohjelmistokehitystä. En enää ajattele, että käytin työhöni liikaa aikaa, vaan mietin, mikä työssä oli aikaa vievää, mikä siinä oli vaikeaa, mikä ei toiminut, mikä puolestaan toimi ja miten lopulta sain työni valmiiksi ja ongelmani ratkaistua. Huomaan tietoisella asenteen muutoksellani olleen paljon positiivisia vaikutuksia: koen nyt oppivani paremmin työn aikana, ja työssä jaksaminen ja työhyvinvointini on parantunut.

On kuitenkin kehitykseni jatkuvuuden kannalta hyvä asia, että työssä on haasteita, sillä ilman haasteita kehitykseni saattaisi hidastua tai pahimmillaan jopa pysähtyä.

Ohjelmistokehityksessä käytettävät tekniikat ja teknologiat kehittyvät ja päivittyvät jatkuvasti, minkä vuoksi kyky ja halu oppia uutta on erityisen tärkeää. Tekemieni työtehtävien päivittäinen kirjaaminen ja viikoittainen analysoiminen on auttanut minua kehittämään omaa oppimistani. Osaan entistä paremmin etsiä luotettavista lähteistä tietoa työssä vaadittavien taitojeni kehittämiseksi. Viikkoanalyysit ovat auttaneet minua sisäistämään paremmin oppimiani asioita, ja koen päiväkirjamuotoisen opinnäytetyön olleen erinomainen keino kehittää omaa oppimista ja työssä vaadittavia ammatillisia taitoja ja edistämään ammatillista kasvuani.

Lähteet

Arnold, J. 26.4.2017. Setting CSS with jQuery. Medium. Luettavissa: <https://medium.com/@thejasonfile/setting-css-with-jquery-bc69558eae19>. Luettu: 26.4.2019.

Atlassian 2019a. Git Merge. Luettavissa: <https://www.atlassian.com/git/tutorials/using-branches/git-merge>. Luettu: 26.4.2019.

Atlassian 2019b. Git cherry-pick. Luettavissa: <https://www.atlassian.com/git/tutorials/cherry-pick>. Luettu: 30.4.2019.

Babich, N. 2019. Breadcrumb Navigation. Usersnap. Luettavissa: <https://usersnap.com/blog/breadcrumbs/>. Luettu: 5.4.2019.

BEM 2019. BEM Naming convention. Luettavissa: <https://en.bem.info/methodology/naming-convention/>. Luettu: 28.3.2019.

Bootstrap 2019a. Dropdowns. Luettavissa: <https://getbootstrap.com/docs/4.0/components/dropdowns/>. Luettu: 24.4.2019.

Bootstrap 2019b. Grid system. Luettavissa: <https://getbootstrap.com/docs/4.0/layout/grid/>. Luettu: 28.3.2019.

Carver, M. 2015. The Responsive Web. Manning Publications. Shelter Island.

Chris, A. 2019. What Is Search Engine Optimization And Why Is It Important. Luettavissa: <https://www.reliablesoft.net/what-is-search-engine-optimization-and-why-is-it-important/>. Luettu: 5.4.2019.

Computer Hope 2018. Footer. Luettavissa: <https://www.computerhope.com/jargon/f/footer.htm>. Luettu: 2.4.2019.

Contentful 2019. About Contentful. Luettavissa: <https://www.contentful.com/faq/about-contentful/>. Luettu: 1.4.2019.

Cope, S. 2011a. z-index. Luettavissa: <https://css-tricks.com/almanac/properties/z/z-index/>. Luettu: 24.4.2019.

Cope S. 2011b. :nth-of-type. Luettavissa: <https://css-tricks.com/almanac/selectors/n/nth-of-type/>. Luettu: 2.5.2019.

Crockford, D. 2008. JavaScript: The Good Parts. O'Reilly Media. Sebastopol.

Eygi, C. 7.12.2018. Understanding CSS Display: None, Block, Inline and Inline-Block. The Startup. Luettavissa: <https://medium.com/swlh/understanding-css-display-none-block-inline-and-inline-block-63f6510df93>. Luettu: 24.4.2019.

Fielding, R.T. & Reschke, J.F. 2014. 404 Not Found. Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. s. 59. Luettavissa: <https://tools.ietf.org/html/rfc7231#section-6.5.4>. Luettu: 25.3.2019.

Fisher, T. 2019. HTTP Status Codes. Luettavissa: <https://www.lifewire.com/http-status-codes-2625907> Luettu: 25.3.2019.

Forsey, C. 20.9.2018. 9 Breadcrumb Tips and Examples to Make Your Site Way Easier to Navigate. HubSpot. Luettavissa: <https://blog.hubspot.com/marketing/navigation-breadcrumbs>. Luettu: 5.4.2018.

GeeksforGeeks 2019a. JavaScript | console.log() with Examples. Luettavissa: <https://www.geeksforgeeks.org/JavaScript-console-log-with-examples/>. Luettu: 11.4.2019.

GeeksforGeeks 2019b. jQuery | offset() with Examples. Luettavissa: <https://www.geeksforgeeks.org/jquery-offset-with-examples/>. Luettu: 26.4.2019.

Git 2019. git push. Luettavissa: <https://git-scm.com/docs/git-push#Documentation/git-push.txt-ltrefspecgt82308203>. Luettu: 29.4.2019.

Go Make Things 2018. How to check if a browser supports native input date pickers. Luettavissa: <https://gomakethings.com/how-to-check-if-a-browser-supports-native-input-date-pickers/>. Luettu: 25.3.2019.

Gube, J. 2009. Breadcrumbs In Web Design: Examples And Best Practices. Luettavissa: <https://www.smashingmagazine.com/2009/03/breadcrumbs-in-web-design-examples-and-best-practices/>. Luettu: 10.5.2019.

Guru99 2019. JIRA Tutorial: A Complete Guide for Beginners. Luettavissa: <https://www.guru99.com/jira-tutorial-a-complete-guide-for-beginners.html>. Luettu: 1.4.2019.

jQuery 2019a. Handling Events. Luettavissa: <https://learn.jquery.com/events/handling-events/>. Luettu: 1.3.2019.

jQuery 2019b. length. Luettavissa: <http://api.jquery.com/length/>. Luettu: 24.4.2019.

jQuery 2019c. What is jQuery? Luettavissa: <https://jquery.com/>. Luettu: 1.4.2019.

Klughertz, J. 10.2.2016. Stash your changes before switching branch with Git. Co-deBlocQ. Luettavissa: <http://www.codeblocq.com/2016/02/Stash-your-changes-before-switching-branch/>. Luettu: 26.4.2019.

Korpela, J. 2014. Soft hyphen (SHY) – a hard problem? Luettavissa: <http://jkor-pela.fi/shy.html>. Luettu: 2.4.2019.

Lincoln, J. 2019. Why changing urls can devastate your seo and traffic. Luettavissa: <https://ignitevisibility.com/why-changing-urls-can-devastate-seo-traffic/>. Luettu: 5.4.2019.

Linnerud, T.E. 2018. git merge –squash. Luettavissa: <https://coderwall.com/p/qkrmjq/git-merge-squash>. Luettu: 26.4.2019.

MDN web docs 2019a. What is JavaScript? Luettavissa: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript. Luettu: 2.4.2019.

MDN web docs 2019b. Columns. Luettavissa: <https://developer.mozilla.org/en-US/docs/Web/CSS/columns>. Luettu: 2.5.2019.

MDN web docs 2019c. HTMLSelectElement.selectedIndex. Luettavissa: <https://developer.mozilla.org/en-US/docs/Web/API/HTMLSelectElement/selectedIndex>. Luettu: 28.3.2019.

Meenakshi, A. 2017. Introduction to Lodash. Luettavissa: <https://medium.com/front-end-weekly/introduction-to-lodash-71dbec093b49>. Luettu: 11.4.2019.

Morris, S. 25.10.2018. Everything You Need to Know About PHP. Skillcrush. Luettavissa: <https://skillcrush.com/2012/04/11/php/>. Luettu: 15.5.2019.

Motto, T. 2.2.2013. How to addClass, removeClass, toggleClass in JavaScript. Ultimate Courses. Luettavissa: https://ultimatecourses.com/blog/JavaScript-hasclass-addclass-removeclass-toggleclass#Adding_a_class_with_addClass. Luettu: 25.4.2019.

Nations, D. 2019. What Are Web Widgets? Luettavissa: <https://www.lifewire.com/what-are-web-widgets-3486686>. Luettu: 15.5.2019.

Osman, M. 2018. How Website Structure Affects SEO. Luettavissa: <https://www.searchenginejournal.com/website-structure-affects-seo/186553/>. Luettu: 5.4.2019.

PHP 2019. Date Formats. Luettavissa: <https://www.php.net/manual/en/datetime.formats.date.php>. Luettu: 9.4.2019.

Regular-Expressions.info 2019. Regular Expression Tutorial - Learn How to Use Regular Expressions. Luettavissa: <https://www.regular-expressions.info/>. Luettu: 28.4.2019.

Search Engine Land 2019. Types Of Search Engine Success Factors. Luettavissa: <https://searchengineland.com/guide/seo/types-of-search-engine-ranking-factors>. Luettu: 30.4.2019.

Sitechecker 2018. Creating a Website Structure to Enhance SEO. Luettavissa: <https://sitechecker.pro/website-structure/>. Luettu: 5.4.2019.

Smith, R. 11.7.2014. Relative vs. Absolute URLs and SEO. The Dirigo Blog. Luettavissa: <http://www.dirigodev.com/blog/seo-web-best-practices/relative-vs-absolute-urls-seo/>. Luettu: 29.4.2019.

Software Engineering 2014. Branch per feature: What are the actual benefits (and risks)? Luettavissa: <https://softwareengineering.stackexchange.com/questions/235504/branch-per-feature-what-are-the-actual-benefits-and-risks>. Luettu: 26.4.2019.

Southern, M. 2018. Google: Click Depth Matters More for SEO than URL Structure. Luettavissa: <https://www.searchenginejournal.com/google-click-depth-matters-seo-url-structure/256779/>. Luettu: 5.4.2019.

Squarespace 2019. What is JSON? Luettavissa: <https://developers.squarespace.com/what-is-json>. Luettu: 26.4.2019.

Suomen Vahinkovakuutus 2019. Luettavissa: <https://www.suomenvahinkovakuutus.fi/>. Luettu: 13.4.2019.

Tech on the Net 2019. CSS: top property. Luettavissa: <https://www.techonthenet.com/css/properties/top.php>. Luettu: 27.4.2019.

Tutorialspoint 2019. What is an Event? Luettavissa: https://www.tutorialspoint.com/JavaScript/JavaScript_events.htm. Luettu: 24.4.2019.

Roberts, A. 2014. select (HTML element). Luettavissa: <https://www.sitepoint.com/select-html-element/>. Luettu: 2.4.2019.

Saraogi, R. 29.3.2018. What is Lodash in Node.js? Quora. Luettavissa: <https://www.quora.com/What-is-Lodash-in-Node-js>. Luettu: 11.4.2019.

Terenteva, E. 1.11.2016. What are Crawlability and Indexability of a Website? SEMrush blog. Luettavissa: <https://www.semrush.com/blog/what-are-crawlability-and-indexability-of-a-website/>. Luettu: 5.4.2019.

VWO 2019. What is A/B Testing? Luettavissa: <https://vwo.com/ab-testing/>. Luettu: 29.4.2019.

W3C 2008. HTML 5. Luettavissa: <https://www.w3.org/TR/2008/WD-html5-20080122/>. Luettu: 25.3.2019.

W3docs 2019. HTML <select> tag. Luettavissa: <https://www.w3docs.com/learn-html/html-select-tag.html>. Luettu: 28.3.2019.

w3schools.com 2019a. HTML <input> readonly Attribute. Luettavissa: https://www.w3schools.com/tags/att_input_readonly.asp. Luettu: 9.4.2019.

w3schools.com 2019b. JavaScript Array push() Method. Luettavissa: https://www.w3schools.com/jsref/jsref_push.asp. Luettu: 10.4.2019.

w3schools.com 2019c. What is CSS? Luettavissa:

https://www.w3schools.com/whatis/whatis_css.asp. Luettu: 1.4.2019.

Walton, P. 2013. What No One Told You About Z-Index. Luettavissa: <https://philipwalton.com/articles/what-no-one-told-you-about-z-index/>. Luettu: 24.4.2019.

Walton, P. 2015. Github. Flexbugs. Luettavissa: <https://github.com/philipwalton/flexbugs#flexbug-2>. Luettu: 4.4.2019.

Whatis.com 2006. Native. Luettavissa: <https://whatis.techtarget.com/definition/native>. Luettu: 25.3.2019.

Liitteet

Liite 1. Lyhenteet

CSS, eli Cascading Sheet Values on HTML-sivujen ulkoasun muokkaamiseen käytettävä merkintäkieli (w3schools.com 2019c).

HTML, Hyper Text Markup Language on verkkosivujen luontiin käytettävä merkintäkieli.

SEO eli Search Engine Optimization tarkoittaa hakukoneoptimointia. Hakukoneoptimoinnilla pyritään hakukonenäkyvyyden parantamiseen.

PHP on ohjelmointikieli, jota käytetään tyypillisesti back-end ohjelmointiin (Morris 2018).

QA, Quality Assurance on laadun varmistukseen käytettävä testiympäristö.

URL (engl. Uniform Resource Locator) eli verkkosivun osoite.

VWO eli Visual Web Optimiser on verkkosivujen ulkoasun optimoinnin testaukseen ja konversion parantamiseen käytettävä verkkotyökalu. VWO luomme testejä, joiden avulla on muun muassa mahdollista selvittää, miten ulkoasun muutokset vaikuttavat asiakkaiden käyttäytymiseen verkkosivulla.

Liite 2. Muut käsitteet sekä käytetyt työkalut ja ohjelmistot

BrowserStack on eri selainten simulointiin käytettävä selainpohjainen työkalu.

Dropdown eli alasvetovalikko. Valikon otsikkoa painettaessa valikko aukeaa alaspäin. Dropdown on myös alasvetivalikon tyylimäärittelyihin käytettävän luokan nimi.

Handlebars.js on JavaScript-kirjasto, minkä avulla rakennetaan HTML-sivuja.

jQuery on JavaScript-kirjasto, jolla on tavallista JavaScriptiä helpompi muun muassa muokata ja animoida sivua, sekä seurata sivulla tapahtuvia klikkauksia (jQuery 2019c).

Natiivi (engl. native) tarkoittaa alkuperäistä, esimerkiksi ohjelmiston, laitteen, tai selaimen mukana tulevaa ominaisuutta (WhatIs.com 2006).

Node.js on JavaScript-koodia taustalla ajava alusta.

React.js on JavaScript-laajennuskirjasto.

Slack on pikaviestisovellus.