



Täyttöohjekokonaisuuden kehitys EURA 2021 - järjestelmässä

Lauri Latva-Kyyny

OPINNÄYTETYÖ
Marraskuu 2021

Tietojenkäsittely
Ohjelmistotuotanto

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely
Ohjelmistotuotanto

LATVA-KYYNY, LAURI:

Täyttöohjekokonaisuuden kehitys EURA 2021 -järjestelmässä

Opinnäytetyö 41 sivua
Marraskuu 2021

Opinnäytetyön tavoitteena oli kehittää työ- ja elinkeinoministeriölle EURA 2021 -järjestelmään täyttöohjekokonaisuus, joka olisi mahdollisimman selkeä, ylläpidettävä ja auttaisi lomakkeiden täyttöprosessien kanssa. EURA 2021 -järjestelmä on sähköinen palvelu, jolla hallinnoidaan EU:n alue- ja rakennepolitiikan rahastojen hankkeita. Toimeksiantajana opinnäytetyössä oli Netum Oy. Järjestelmässä voi olla virkailijoita tai hakijoita. Tarkoituksena oli toteuttaa täyttöohjekokonaisuus, johon sisältyi koostettu täyttöohjesivu virkailijoille ja hakijoille, dynaamiset kenttäkohtaiset täyttöohjeet lomakkeille sekä ylläpitotyökalu täyttöohjeiden ylläpitoon. Tehtävänä oli selvittää, miten generoidaan täyttöohjesivut, miten tietty ohje linkitetään kenttään lomakkeella ja näytetään dynaamisesti lomakkeen sivussa. Tämän lisäksi piti selvittää, miten toteutettaisiin täyttöohjeiden ylläpitotyökalu, joka sisältäisi mahdollisimman vähän manuaalivaiheita ja otettaisiin huomioon mahdolliset virhetilanteet.

Tuloksena valmistui täyttöohjekokonaisuus, josta ylläpitotyökalun valmistuminen viivästy, mutta suunnittelu ja kehitys saatiin kuitenkin hyvään vaiheeseen. Täyttöohjesivu toteutettiin virkailijalle ja hakijalle erillisenä sivuna. Tämän lisäksi lomakkeille saatiin toteutettua dynaamiset kenttäkohtaiset ohjeet hakijalle ja virkailijalle.

Ohjekokonaisuudesta pyrittiin tekemään täysin saavutettava. Tabulaattorilla liikkumista lomakkeen kentästä kenttäkohtaiseen ohjeeseen ei saatu täysin saavutettavaksi. Asiakaspalautteesta ilmeni, että kenttäkohtaisen ohjeen nopea vilkkuminen koettiin häiritseväksi lomakkeella liikkuesssa, mutta se saatiin korjattua lisäämällä ajastusta ja sulavuutta ohjeen vaihtumiseen. Täyttöohjeiden päivittäminen osoittautui aikaavieväksi ilman täyttöohjeiden ylläpitotyökalua, joten sen kehittämisen loppuunsaaminen on ohjekokonaisuuden pääprioriteetti. Muuten ohjekokonaisuudesta tuli selkeä ja vaatimuksia vastaava toteutus.

Asiasanat: eura 2021 -järjestelmä, täyttöohjekokonaisuus, kenttäkohtaiset ohjeet, täyttöohjesivu, täyttöohjeiden ylläpitotyökalu

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Option of Software Development

LATVA-KYYNY, LAURI
Development of Filling Instruction Set in the EURA 2021 System

Bachelor's thesis 41 pages
November 2021

The objective of the thesis was to develop clear and maintainable filling instructions to the EURA 2021 system for the Ministry of Economic Affairs and Employment. EURA 2021 is a digital service for the management of EU regional and structural policy funds projects. The client for the thesis was Netum Oy. The purpose was to implement a filling instruction set, including a compiled filling instructions for officials and applicants, dynamic field-specific instructions for the forms, and an upkeep tool for the instructions.

The result was a separate filling instructions page for officials and applicants and dynamic field-specific instructions on the forms, but the upkeep tool was not able to be finished yet.

The aim was to make the instruction set fully accessible. Customer feedback indicated that the rapid flashing of the field-specific instructions was found to be distracting when moving around the form, but this was resolved by adding a delay. Updating the instructions proved to be time consuming without the upkeep tool, so completing its development is a top priority for the instruction set. Otherwise, the instruction set proved to be clear and successful implementation.

Key words: eura 2021-system, filling instruction set, field-specific instructions, filling instruction page, upkeep tool for filling instructions

SISÄLLYS

1	JOHDANTO	5
2	TAUSTAA	8
	2.1 EURA -järjestelmän taustaa	8
	2.2 Mikä on EURA 2021 -järjestelmä?	9
	2.3 Täyttöohjeiden tarpeen taustaa	11
	2.3.1 Täyttöohjeet muissa digitaalisissa järjestelmissä	11
	2.3.2 Täyttöohjeiden tarve EURA 2021 -järjestelmässä	15
3	TÄYTTÖOHJEKOKONAISUUDEN SUUNNITTELUPROSESSI	16
	3.1 Teknologiavalinnat	16
	3.1.1 React ja TypeScript	16
	3.1.2 Node.js ja SQL	19
	3.1.3 Amazon S3 ja Docker	20
	3.2 Täyttöohjeiden koostesivun suunnittelu	21
	3.3 Kenttäkohtaisten ohjeiden suunnittelu	22
	3.4 Täyttöohjeiden ylläpitotyökalun suunnittelu	23
4	TÄYTTÖOHJEKOKONAISUUDEN KEHITYSPROSESSI	25
	4.1 Täyttöohjeiden koostesivun kehitys	25
	4.2 Kenttäkohtaiset täyttöohjeiden kehitys	27
	4.3 Täyttöohjeiden ylläpitotyökalun kehitys	29
5	TOTEUTUKSEN HAASTEET, ONGELMAT JA NIIDEN RATKAISU ..	32
	5.1 Saavutettavuusparannukset kenttäkohtaisissa ohjeissa	32
	5.2 Renderöinti-ongelmat ja täyttöohjeiden välkkyminen	33
	5.3 Täyttöohjeiden päivitys ilman ylläpitotyökalua	34
6	POHDINTA	36
	LÄHTEET	40

1 JOHDANTO

Toimeksiantona on kehittää täyttöohjekokonaisuus EURA 2021 -järjestelmään. Toimeksiantajana tässä opinnäytetyössä on Netum Oy, joka kehittää EURA 2021 -järjestelmää työ- ja elinkeinoministeriölle. Työ- ja elinkeinoministeriö vastaa järjestelmän toiminnallisuuden ja prosessien määrittelystä (Case: EURA 2021 -järjestelmä n.d.). Netumin sivuilla EURA-järjestelmää kuvataan seuraavasti: *”EURA on EU:n edellyttämä ja lakisääteinen palvelu, jonka avulla hallinnoidaan ja valvotaan alue- ja rakennepolitiikan rahastojen ohjelmaan kuuluvaa EU-rahoitusta Suomessa ohjelmakauden aikana. Järjestelmässä jaetaan valtuuksia, hallinnoidaan hankkeita ja niihin liittyviä maksatuksia sekä valvotaan ja tarkastetaan hankkeita.”* (Case: EURA 2021 -järjestelmä n.d.)

Euroopan unionin alue- ja rakennepolitiikalla (koheesiopolitiikalla) lisätään jäsenvaltioiden taloudellista ja sosiaalista yhteenkuuluvuutta sekä vähennetään alueiden välisiä kehityseroja. Ohjelmakaudet ovat seitsemän vuoden mittaisia ja järjestelmän on tarkoitus tulla käyttöön vuonna 2021 alkavalle ohjelmakaudelle. (EU:n alue- ja rakennepolitiikka n.d.)

Netumilla ja työ- ja elinkeinoministeriöllä on yhteistä historiaa jo ennestään, sillä Netum kehitti myös edellisellä ohjelmakaudella toimineen EURA 2014 -järjestelmän. Koko rakennerahasto-ohjelman EU-rahoitus on ollut aikaisemmalla ohjelmakaudella (2014–2020) noin 1,3 miljardia euroa ja kansallisen vastinrahoituksen kanssa käytettävissä olevan julkisen rahoituksen määrä on ollut noin 2,6 miljardia euroa. Uudelle EURA 2021 -järjestelmälle on tiedossa melko pitkä elinikä, sillä se tulee olemaan käytössä 2030-luvulle asti. (Case TEM: EURA 2014 -HANKE n.d.)

Järjestelmässä on karkeasti jaoteltuna kahdenlaisia käyttäjiä – hakijoita ja virkailijoita. Virkailijat julkaisevat hankehakuilmoituksia eli lomakkeita, joiden kautta rekisteröidyt hakijat voivat hakea hankerahoitusta. Hakeakseen hankerahoitusta hakijat täyttävät hankehakemuksen eli lomakkeen, johon täytetään rahoituksen hakua varten vaadittavat tiedot. Tämän jälkeen virkailija käsittelee julkaistun hankehakemuksen, jotta rahoitusta voidaan jakaa harkitusti.

Lomakkeet joita järjestelmässä täytetään voivat sisältää vieraita käsitteitä tai kohtia, joissa käyttäjät voivat jäädä helposti jumiin täyttöprosessissa.

Siksi järjestelmään haluttiin kehittää täyttöohjekokonaisuus, jonka päätavoitteena on olla mahdollisimman käyttäjäystävällinen sekä rahoitusten haku- ja käsittelyprosessia helpottava, jotta aikaa kuluisi mahdollisimman vähän epäselvyyksien selvittelyyn. Täyttöohjekokonaisuus on tärkeä osa järjestelmän käyttöönottoa, sillä virkailijoiden ja hakijoiden alkaessa täyttää hankehakuilmoituksia ja hankehakemuksia on olennaista saada helposti selville, mitä mihinkin kenttään pitää täyttää tai mikä vaihtoehto täytyy missäkin tilanteessa valita.

Tarkoituksena on kehittää lomakkeiden täyttöprosessiin dynaamiset kenttäkohtaiset ohjeet, jotka näkyvät käyttäjille lomakkeiden oikeassa reunassa. Kenttäkohtaisista ohjeista saadaan lomakkeille reaaliaikaista ja kohdistettua ohjeistusta. Tämän lisäksi haluttiin erillinen täyttöohjesivu, joka näyttäisi koostetusti järjestelmän kaikki täyttöohjeet hakija- tai virkailijakohtaisesti. Haluttiin myös erillinen ylläpitotyökalu, jolla täyttöohjeita voisi vaivattomasti päivittää.

Opinnäytetyö tulee hyödyttämään Netumia sekä työ ja elinkeinoministeriötä, sillä täyttöohjekokonaisuus on tärkeä osa järjestelmän käyttöönottoa. Lisäksi työstä saadaan tietoa olemassa olevista muiden julkisten palvelujen täyttöohjeista. Näiden lisäksi opinnäytetyössä käsitellään täyttöohjekokonaisuuden jatkokehitystä ja mahdollisia ongelmia. Lopputuloksena haetaan käyttäjäystävällistä toteutusta, joka on teknisesti järkevä. Haettava toteutus vastaa tarpeita, on vaivatonta ylläpitää ja tuottaa mahdollisimman vähän virheitä sekä auttaa selkeästi opastamaan käyttäjiä lomakkeiden täyttämässä. Hakijoille ja virkailijoille hyöty tulee täyttöohjeiden tuottamasta selkeydestä ja varmuudesta lomakkeiden täyttämiseen.

Hyvät täyttöohjeet ovat elintärkeä osa järjestelmää, ilman niitä muuten toimivan järjestelmän käyttö tuntuu hankalalta ja monimutkaiselta sekä pakottaa käyttäjät tekemään omia muistiinpanoja tai ulkoa muistamista epäselvistä kohdista. Tällöin voi syntyä vääristynyttä informaatiota siitä, miten lomakkeita tulisi täyttää.

Täyttöohjeet siis edesauttavat sitä, että täytetty sisältö vastaisi mahdollisimman hyvin haluttua vastausta.

Tämä kokonaisuus on mielenkiintoinen ja teknisesti haastava. Kehityksessä ja suunnittelussa täytyy ottaa paljon asioita huomioon jo aikaisessa vaiheessa, sekä olla hyvin perillä asiakkaan tarpeista. Käytössä ovat modernit teknologiat, joten kehityksen aikana oppii paljon uutta alan olennaisista teknologioista ja saa käyttöön uusien ominaisuuksien mahdollistamat hyödyt. Tämän lisäksi täyttöohjekokonaisuus tarjoaa mahdollisuuden kehittää isompia kokonaisuuksia ja oppia samalla saavutettavuudesta. Opinnäytetyön tekijä alkaa kehittämään ohjekokonaisuutta, mutta saa apua muilta tiimeiltä, kuten arkkitehtiimiltä, UX-tiimiltä eli käyttäjäkokemustiimiltä (engl. User Experience) ja omalta kehitystiimiltä.

Kehitys suoritetaan kolmessa osassa. Ensin toteutetaan täyttöohjesivut hakijoille ja virkailijoille, tämän jälkeen kenttäkohtaiset ohjeet hankehakemukselle ja hankehakuilmoitukselle. Lopuksi toteutetaan täyttöohjeiden ylläpitotyökalu. Kehitysvaiheessa otetaan huomioon, miten täyttöohjekokonaisuuden osat ovat riippuvaisia toisistaan, jotta ne saataisiin toimimaan mahdollisimman hyvin yhdessä ja helposti ylläpidettäviksi.

Toteutuksessa täytyy ratkaista seuraavia haasteita:

- Miten generoida täyttöohjesivustolle ohjesisältöä?
- Miten toteuttaa dynaamiset kenttäkohtaiset ohjeet niin, että jokainen kenttä on linkitetty tietyn ohjeen yksilölliseen tunnisteeseen eli avainarvoon tai toisin sanoen id:seen?
- Miten päivittää ohjeet onnistuneesti ilman, että manuaalisia vaiheita on liikaa?
- Miten täyttöohjeiden päivityksen virhetilanteet saadaan otettua huomioon?

2 TAUSTAA

2.1 EURA -järjestelmän taustaa

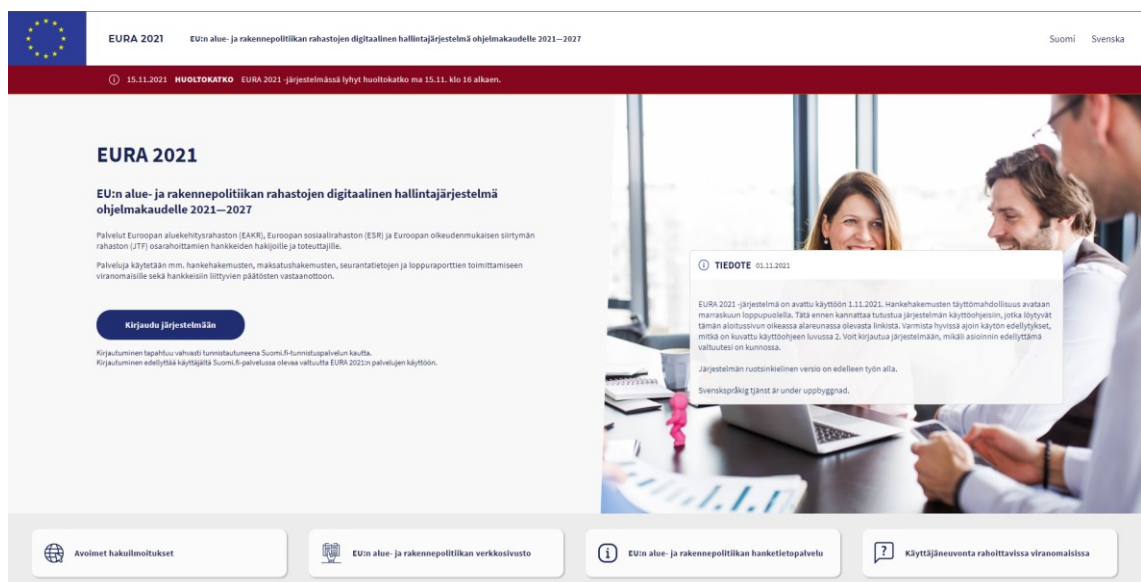
Ennen kuin voidaan selvittää mikä on EURA 2021 -järjestelmä, on syytä perehtyä tarkemmin taustoihin eli, mikä tarve järjestelmälle on, että sellainen täytyy lainmukaisesti kehittää ja minkälaista historiaa aiheeseen liittyy.

Rakennerahastotoiminta käynnistyi kun Suomi liittyi Euroopan Unioniin vuonna 1995. Rakennerahastot ovat EU:n alaisia rahastoja, jotka noudattavat tiettyjä teemoja, esimerkiksi työllisyyden lisäämistä tai hiilineutraaliuutta. Tulevalla ohjelmakaudella (2021–2027) rakennerahastoja ovat Euroopan aluekehitysrahasto (EAKR) ja Euroopan sosiaalirahasto plus (ESR+). EAKR-rahaston tavoite on parantaa työllisyyttä sekä lisätä alueiden kilpailukykyä ja elinvoimaisuutta. ESR+-rahaston tavoite on tukea työllisyyttä ja työllistymisedellytyksiä osaamista ja palvelurakenteita kehittämällä. (EU:n rakennerahastot n.d.) Näiden rahastojen lisäksi on myös myöhemmin ohjelmakaudelle tuleva Oikeudenmukaisen siirtymän rahasto (JTF) (työ- ja elinkeinoministeriö 2021). JTF-rahaston tavoite on tarjota tukea niille alueille, joissa ilmastosiiirtyminen vaikuttaa negatiivisimmin (Rakennerahastot n.d.).

Rahastoista jaetaan rahaa EU-jäsenmaille, jotta pystytään turvaamaan yhtenäinen ja tasavertainen politiikka. Tätä kutsutaan myös koheesiopolitiikaksi. Euroopan unionin alue- ja rakennepolitiikalla (koheesiopolitiikalla) lisätään jäsenvaltioiden taloudellista ja sosiaalista yhteenkuuluvuutta sekä vähennetään alueiden välisiä kehityseroja. (EU:n alue- ja rakennepolitiikka n.d.) Rahoituksia jaetaan yrityksiin ja organisaatioiden hankkeisiin seitsemän vuoden pituisilla ohjelmakausilla, esimerkiksi viimeisin ohjelma Kestävää kasvua ja työtä 2014–2020 (Kestävää kasvua ja työtä 2014–2020 -ohjelma n.d.). Uuden EURA 2021 -järjestelmän taustalla on siis pitkä historia, joka liittyy vahvasti EU:n politiikkaan. Taustan tunteminen vastaa kysymyksiin, kuten, miksi tällaista järjestelmää täytyy kehittää, ja kuinka tärkeä asia tämä on Suomen kannalta.

2.2 Mikä on EURA 2021 -järjestelmä?

EURA 2021 on digitaalinen järjestelmä, jolla hallinnoidaan alue- ja rakennepolitiikan rahastoja, kuten EAKR ja ESR+. Näillä rahastoilla rahoitetaan hankkeita, joita järjestelmässä voidaan hallinnoida, valvoa ja tarkastaa. Hallintointiin sisältyy esimerkiksi tietyn tietyn hankkeen käsittelyprosessi, jonka edistymistä voidaan valvoa ja ennen hyväksymistä tai käsittelyyn ottoa tarkastaa. Järjestelmässä voidaan myös jakaa valtuuksia esimerkiksi muille organisaation henkilöille. EURA 2021 on EU:n edellyttämä ja lakisääteinen palvelu ja sen tulee vastata tulevan ohjelmakauden (2021-2027) vaatimuksiin. (Case: Eura 2021 -järjestelmä n.d.) Alla olevassa kuvassa on EURA 2021 -järjestelmän kirjautumattoman hakijan etusivu, josta voidaan hahmottaa millaista tyyliä järjestelmä edustaa.



KUVA 1. EURA 2021 -järjestelmän kirjautumattoman hakijan etusivu

Netum on päässyt kehittämään myös aikaisempaa EURA 2014 -järjestelmää edelliselle ohjelmakaudelle (Case: Eura 2021 -järjestelmä n.d.). Alla olevassa kuvassa on EURA 2014 -järjestelmän etusivu kirjautumattomalle hakijalle. Kuten voidaan huomata, on EURA 2021 -järjestelmän kirjautumattoman hakijan etusivu alla olevaan kuvaan verrattuna huomattavasti nykyaikaisemman näköinen. Tähän on todennäköisesti syynä modernimmat teknologiat, joilla saadaan kehitettyä tyylikkäämpiä käyttöliittymiä.



The screenshot shows the header of the EURA 2014 website. On the left is the European Union flag. The main header text reads: "EURA 2014 EU:N RAKENNERAHASTOJEN HALLINTAJÄRJESTELMÄ Ohjelmakausi 2014 - 2020 Hakijan ja toteuttajan palvelut". On the right, it says "På svenska" and "Käyttöohje hakijalle". Below the header is a navigation bar with "ETUSIVU" and "RR-TIETOPALVELU". The main content area features a grey box with the title "EURA-järjestelmän tiedotteet" and a news item dated 31.12.2019. Below this is a section titled "EURA 2014 - EU:n rakennerahastohankkeiden sähköisen asiointin palvelut hankkeiden hakijoille ja toteuttajille ohjelmakaudella 2014–2020" with a detailed description of the service.

EURA-järjestelmän tiedotteet

31.12.2019 EURA 2014 -tietojärjestelmän käyttöön liittyvissä ongelmatilanteissa pyydämme teitä ottamaan yhteyttä siihen viranomaiseen, jolle tulitte osoittamaan hakemuksenne. Suomi.fi-palvelun tunnistamiseen sekä valtuuttamiseen liittyvissä asioissa käyttäjätukea antaa Digi- ja väestötietovirasto omilla sivuillaan (<https://www.suomi.fi/ohjeet-ja-tuki> ja <https://dvv.fi/katsosta-suomi.fi-palveluun>). Myös Rakennerahastot.fi-sivuilla (<https://www.rakennerahastot.fi/346>) on ohjeistusta tunnistukseen sekä EURA 2014:ssä tarvittaviin valtuuksiin liittyen. Tutustukaa myös järjestelmän käyttöohjeisiin, jotka löytyvät oikean yläreunan Käyttöohje hakijalle -linkin kautta.

EURA 2014 - EU:n rakennerahastohankkeiden sähköisen asiointin palvelut hankkeiden hakijoille ja toteuttajille ohjelmakaudella 2014–2020

EURA 2014 on Euroopan sosiaalirahaston (ESR) ja Euroopan aluekehitysrahaston (EAKR) osarahoittamien hankkeiden hallintaan tarkoitettu sähköisen asiointin palvelukokonaisuus ohjelmakaudella 2014–2020.

EURA 2014:n palveluilla hoidetaan mm. hankehakemusten, maksatushakemusten ja seurantalomakkeiden sekä loppuraporttien toimittaminen ja käsittely sekä hankehallintaan liittyvien päätösten valmistelu ja vastaanotto. Sähköisen hankehallinnon yhteydessä syntyvistä tiedoista muodostetaan Suomen rakennerahastotoiminnan toteutumista kuvaavat raportointitiedot sekä kansallisiin tarkoituksiin että Euroopan komissiolle.

KUVA 2. EURA 2014 -järjestelmä, kirjautumattoman hakijan etusivua

EURA 2021 -järjestelmä tulee toimintaan uudelle ohjelmakaudelle: Uudistuva ja osaava Suomi 2021–2027. Tällä ohjelmakaudella pyritään tukemaan elinkeino-, energia-, ilmasto-, innovaatio-, koulutus- ja työllisyyspolitiikkaa sekä syrjäytymisen ja köyhyyden vastaista työtä. (työ- ja elinkeinoministeriö 2021.) Uudella ohjelmakaudella on tietyt painopisteet, jotka vaikuttavat siihen minkälainen hankehakuilmoitus virkailijan pitää julkaista haettavaksi tai minkälainen hankehakemus hakijan pitää täyttää rahoituksia haettaessa.

Järjestelmää käyttävät virkailijat ja hakijat. Hakijat voivat olla yrityksiä tai organisaatioita ja virkailijat taas esimerkiksi tietyn kuntayhtymän hyväksytyjä virkailijoita. Virkailijat täyttävät ja julkaisevat hankehakuilmoituksia, joille hakijat voivat hakea hankehakemuksilla hankerahoitusta täyttämällä hakemuksen vaaditut tiedot. Hankehakuilmoitus sisältää tietoja kuten, kuka virkailija sen on luonut, mille alueelle se kohdistuu, mitä kustannusarvioita tai palkkakustannuksia ilmoitukseen kohdistuu. Kun hakijalla on tiedossa hanke ja oikeudet hakea rahoitusta, voi hän täyttää hankehakemuksen. Hankehakemus on hakijan täytettävä lomake, johon täytetään kaikki vaaditut yksityiskohdat hankkeesta.

2.3 Täyttöohjeiden tarpeen taustaa

Mietitään tilannetta, jossa on tarve käyttää jotain julkista järjestelmää. Järjestelmä sisältäisi pitkiä lomakkeita ja minkäänlaista opastusta ei olisi kenttien otsikoiden tai validointitekstien lisäksi. Tällaista järjestelmää käyttäessä törmättäisiin varmasti nopeasti tilanteeseen, jossa joudutaan etsimään internetistä tietoa, mitä mikäkin käsite lomakkeen kentissä tarkoittaa. Toki täyttöprosessissa jumiin jääminen riippuisi paljon käyttäjästä, mutta jos järjestelmää käyttää monta käyttäjää, törmätään virhetilanteisiin useasti ilman minkäänlaista opastusta. Tätä varten on olemassa täyttöohjeet.

Täyttöohjeilla tarkoitetaan kaikkia ohjeita, jotka ohjeistavat sitä minkälaista tietoa lomakkeille tulee täyttää. Täyttöohjeiden tavoite on tuoda järjestelmän käyttäjälle selkeyttä lomakkeiden täyttämiseen, nopeuttaa lomakkeiden täyttöprosessia ja tehdä järjestelmästä käyttäjäystävällisempi sekä saavutettavampi. Täyttöohjeissa avataan usein erikoisempia käsitteitä ja neuvotaan yleisesti lomakkeiden täyttöprosessissa. Ne on usein eroteltuna lomakekohtaisesti ja voivat olla erillisellä sivulla tai ladattavissa PDF-tiedostona.

Jos käyttäjä jää jumiin lomakkeiden täyttöprosessissa ja internetistä ei löydy luotettavaa tai sopivaa tietoa, todennäköisesti hän jättää lomakkeen tekemisen kokonaan kesken tai ottaa yhteyttä järjestelmän mahdolliseen asiakastukeen. Tämä viivästyttää prosessien sujuvaa etenemistä ja kuluttaa molemmilta osapuolilta aikaa. Täyttöohjeet eivät ole pelkästään asiakkaille tai hakijoille vaan myös järjestelmässä asioivat virkailijat voivat tarvita omia ohjeita.

2.3.1 Täyttöohjeet muissa digitaalisissa järjestelmissä

Jos tarkastellaan yleisimpiä digitaalisia järjestelmiä, kuten Veron ja Kelan asiointipalveluja, on niissä samankaltaisuutta EURA 2021 -järjestelmän kanssa. Ne ovat julkisen sektorin digitaalisia järjestelmiä, joiden tarkoituksena on tarjota palveluja käyttäjille. Kaikissa järjestelmissä joudutaan täyttämään lomakkeita,

jotka sisältävät melko yksityiskohtaisia tietoa ja vieraita käsitteitä, joita voi olla useampia riippuen lomakkeen tyypistä ja käyttäjän tietämyksestä.

Täyttöohjeilla on tärkeä rooli järjestelmissä. Jos järjestelmä on julkisen sektorin, se tulee olla jokaisen käyttäjän saatavilla ja käytettävissä. Ei voida olettaa, että osa käyttäjistä osaa täyttää lomakkeita itsenäisesti ilman opastusta, vaan joudutaan ottamaan huomioon myös ne, joille koko täytettävä sisältö tai yksittäiset käsitteet ja valinnat ovat vieraita.

Kelan sivuilla on yleisiä ohjeita pääasiassa sisäiseen käyttöön, joissa neuvotaan miten hakemuksia tulisi käsitellä. Tämän lisäksi Kelan etusivun hakutoiminnosta löytyi hakusanoilla ”täyttöohje” myös erillisiä täyttöohjeita hakemuksille. Alla olevasta kuvasta voidaan havaita, että täyttöohjeet sisältävät väliotsikoita, kuten ”Kustannus- ja toimintatiedot – Ohjaus ja neuvonta”, sekä niiden alla olevia kenttäkohtaisia otsikoita, kuten ”Toimintatiedot” ja siihen liittyvän varsinaisen täyttöohjesisällön. (Kela 2019.) Täyttöohjeissa on siis selkeä hierarkkinen rakenne. Täyttöohjeet sisältävät myös muotoiltua tekstiä, esimerkiksi kuvan alareunassa oleva lihavoitu teksti. Muotoiltua tekstiä kutsutaan myös rikastetuksi tekstiksi (engl. rich text), jonka avulla voidaan tuoda parempia muotoiluja tekstile, esimerkiksi linkit, lihavointi, kursivointi ja luettelomerkit.

KANSANELÄKELAITOS



Kustannus- ja toimintatiedot - Ohjaus ja neuvonta

Toimintatiedot

Ilmoita *Ryhmän neuvonta ja ohjaus* -kohdassa työterveyshuollon ammattihenkilöiden ryhmän neuvontaan ja ohjaukseen käyttämä aika sekä ammattihenkilöiden tarvitsemien asiantuntijoiden neuvontaan ja ohjaukseen käyttämä aika puolen tunnin tarkkuudella. Tieto sisältää suunnittelun sekä neuvonnan ja ohjauksen toteuttamisen matka-aikoiheen sekä seurantaan ja arviointiin käytetyn ajan. Tarvittava ryhmän ohjaus ja neuvonta ja yksilön ohjaus voidaan toteuttaa etäpalveluna.

Ilmoita *Yksilön neuvonta ja ohjaus* -kohdassa ammattihenkilöittäin työntekijöiden neuvonta- ja ohjauskäyntien määrät sekä työntekijöiden neuvonta- ja ohjauskäyntimäärät ammattihenkilöiden tarpeelliseksi katsomien asiantuntijoiden luona. Etäpalveluna annettu yksilön neuvonta ja ohjaus voidaan toteuttaa tarvittaessa etäpalveluna videovälitteisesti.

Kustannukset

Ilmoita ryhmä- ja yksilöneuvontaan osallistuneiden työterveyshuollon ammattihenkilöiden ja asiantuntijoiden toiminnasta aiheutuneet kustannukset. Näitä ovat toiminnan suunnittelusta, toteuttamisesta matka-aikoiheen sekä seurannasta ja arvioinnista aiheutuneet *aikaperusteiset ja käyntikertakohtaiset maksut ja palkkiot*.

Jos kyseessä on **työnantajan oma työterveysasema**, ilmoita kustannuksina työterveyshuollon ammattihenkilöiden ja asiantuntijoiden *palkka- ja palkkasivukustannukset* kyseisen ammattiryhmän kohdalla.

KUVA 3. Kelan työnantajan työterveyshuollon verkkohakemuksen täyttöohjetta (Kela 2019)

Verottajan sivulta löytyy myös täyttöohjeita navigoimalla sivun alhaalla olevasta "Asioi verkossa" -valikossa kohtaan "Lomakkeet" ja valitsemalla ensimmäinen hakutulos "Verokorttihakemus ja ennakkoverohakemus (5010)". Navigoidulta sivulta löytyy "Täyttöohjeet" -väliotsikon alta linkki, josta pääsee erilliselle lomakkeen täyttöohjesivulle. Alla olevasta kuvasta voidaan nähdä osa täyttöohjesivusta. Sivulla on eroteltuna selkeästi eri lomakkeen kohdat numeroiduilla otsikoilla. Näiden alla on täyttöohjeita, mutta myös väliotsikoita, kuten voidaan havaita kohdasta: "7 Henkilökohtaiset ennakonkannon alaiset tulot". (Vero 2020.) Tämän lisäksi tässä nähdään laajemmin muotoillun tekstin ominaisuuksia, kuten luettelomerkit, lihavoitu teksti ja linkki.

6 Opintoraha

Ilmoita tässä verovuoden aikana korkeakouluopintoihin tai keskiasteen opintoihin saamasi opintoraha.

- Täytä laatikoihin nostamasi opintoraha kuukausittain.
- Merkitse muutosajankohtaan mennessä pidätetty vero. Ennakonpidätyksen määrän voit tarkistaa opintorahan maksajan maksuilmoituksesta.

7 Henkilökohtaiset ennakonkannon alaiset tulot

Ilmoita tässä kohdassa tulot, joista maksaja ei tee ennakonpidätystä vaan joista Verohallinto määrää tai on määrännyt erikseen ennakkoveron.

Huomaa, että tässä kohdassa ilmoitetaan vain alla mainitut tulot. **Vuokratulot, luovutusvoitot ja ulkomaantulot ilmoitetaan erillisillä lomakkeilla** (ks. tarkemmin kohta 19, Hakemuksen liitteet). Yritystoiminnasta tai maa- ja metsätaloudesta saadut tulot ilmoitetaan [lomakkeen kohdassa 14](#).

Henkilökohtaiset ansiotulot

Tulonhankkimistoiminnan tulos

Ilmoita ennakonkannon alaiset henkilökohtaisen tulolähteen työkorvaukset. Näillä tarkoitetaan esimerkiksi tulonhankkimistoimintaa, josta ei pidätetä veroa vaan josta maksetaan ennakkovero. Ilmoita tässä kohtaa myös esimerkiksi

KUVA 4. Veron asiointipalvelun verokorttihakemuksen ja/tai ennakkoverohakemuksen täyttöohjetta (Vero 2020)

EURA 2014 -järjestelmässä on myös täyttöohjeita ja näistä voi saada kuvaa hakemalla Rakennerahastot -sivulta hakusanalla "täyttöohje". Sivulla on linkkejä,

joista valitsemalla jälkimmäisen ”ESR-hankkeeseen osallistuvia henkilöitä koskevat tiedot” ja navigoimalla sisällöstä löytyvään ”materiaalipankki” -linkkiin löytyy dokumentteja. Valitsemalla dokumentti ”FI ESR Henkilö -järjestelmän täyttöohje” päästään sivulle, josta täyttöohjeet voidaan ladata PDF-tiedostona. Alla olevasta kuvasta voidaan nähdä, että täyttöohjeissa on vastaavasti selkeä hierarkkinen rakenne. Ylhäällä on sivun otsikko, jolla kuvataan tiettyä lomaketta tai sen välilehteä. Sivun otsikon alla yleistason otsikko ja yleistä sisältöä tähän liittyen. Tämän alla on kenttäkohtaiset ohjeet, eli kenttäkohtainen otsikko ja kentän täyttöohjesisältö. Esimerkiksi otsikolla ”Aloituspäivämäärä” kuvataan päivämääräkenttää ja sen perässä on teksti miten kenttään tulisi täyttää tietoa. (Rakennerahastot 2016.)

TYÖ- JA ELINKEINOMINISTERIÖ



ESR HENKILÖ -JÄRJESTELMÄN TALLENNETTAVIEN LOMAKKEIDEN TÄYTTÖOHJE

ALOITUSTILANNE

Jokainen osallistuja kirjataan hankkeessa aloittaneeksi vain kerran vaikka hän lähdettyään hankkeesta palaisikin hankkeeseen myöhemmin uudelleen. Vaikka hankkeessa olisi erilaisia osioita, joihin henkilö osallistuu, jokainen osallistuja täyttää aloituslomakkeen vain yhden kerran. Osallistujan aloitustilanteen tietojen tulee kuvata hänen tilannettaan silloin, kun hän ensimmäisen kerran aloitti hankkeessa. Lopetustilanteen tietojen tulee sen sijaan kuvata tilannetta silloin, kun hän viimeisen kerran lähti hankkeesta.

Aloituspäivämäärä: Kirjoitetaan muodossa pp.kk.vvvv. Kirjoitetaan se päivämäärä, jolloin henkilö aloitti hankkeessa.

Osallistuja: Osallistujalla tarkoitetaan henkilöä, joka saa suoraa etua ESR:n toimesta, joka voidaan tunnistaa, jolta voidaan kysyä hänen ominaisuuksistaan ja jota varten on varattu määrärahoja. Hankkeeseen osallistuja hyötyy osallistumisestaan esimerkiksi parantuneiden työllistymisvalmiuksien, osaamisen lisääntymisen tai parantuneen työ- tai toimintakyvyn muodossa. Tietojen keruu kohdistetaan siten henkilöihin, jotka kuuluvat hankkeen määriteltyyn kohderyhmään. Osallistujia eivät ole henkilöt, joiden palkka maksetaan hankkeen rahoista (kyseisen hankkeen henkilöstö ja palveluntuottajat) eivätkä henkilöt, jotka osallistuvat esimerkiksi hankkeen järjestämiin tiedotustilaisuuksiin tai jotka saavat kaikille avointa palvelua puhelimitse tai verkossa.

Osallistujan syntymäaika: Ilmoitetaan muodossa pp.kk.vvvv. Syntymäaikatieto on tarpeen, jotta voidaan raportoida osallistujat ikäryhmittäin. Järjestelmä laskee osallistujan iän automaattisesti syntymäajan ja osallistujan hankkeessa aloittamispäivän perusteella.

KUVA 5. Rakennerahastot sivun FI ESR Henkilö -järjestelmän täyttöohjetta EURA 2014 -järjestelmän lomakkeelle (Rakennerahastot 2016)

2.3.2 Täyttöohjeiden tarve EURA 2021 -järjestelmässä

EURA 2021 -järjestelmässä on lomakkeita, joiden sisältö on hyvin vahvasti sidoksissa lainsäädäntöön. Tämä tarkoittaa sitä, että tietyt lomakkeen kentät on pakko täyttää ja täytetyt kohdat tulee miettiä tarkkaan. Hakijat ja virkailijat joutuvat siis täyttämään ainakin vaaditut tiedot omille lomakkeilleen, jotta ne saataisiin eteenpäin. Vaaditut kentät eivät aina ole täysin itsestäänselviä, joten järjestelmässä täytyy olla jokin ohjeistus, jossa kuvataan minkälaista tietoa kenttiin tulee täyttää tai milloin pitää valita mikäkin vaihtoehto. Tämän lisäksi, kun sisältö muuttuu ohjelmakausittain ja lomakkeille tulee uusia, ehkäpä tuntemattomia kenttiä täytyy näille saada selkeät ohjeet, jotta sekä virkailijat että hakijat voivat hoitaa täyttöprosessin nopeammin ja vaivattomammin.

Täyttöohjeet vähentävät todennäköisyyttä sellaiselle kommunikoinnille virkailijoiden ja hakijoiden välillä, jota muuten aiheutuisi epäselvyyksistä täyttöohjeiden puuttuessa. Jos virkailija täyttää virheellistä tietoa puuttuvien täyttöohjeiden takia ja julkaisee hankehakuilmoituksen, seuraa tästä epäselvyyksiä hakijoille ja hankehakuilmoitusta joudutaan jälkikäteen muokkaamaan oikeanlaiseksi. Jos taas hakija täyttää virheellisiä tietoja hankehakemukselle ja jättää sen virkailijalle käsittelyyn, tulee tästä viivästyksiä hakemuksen käsittelyyn ja turhaa ylimääräistä työtä molemmille osapuolille. Tämän takia ohjeet lisäävät käyttäjäystävällisyyttä järjestelmässä sekä hakijoille että virkailijoille, mutta myös edesauttavat prosesseja säästämällä aikaa molemmilta osapuolilta.

3 TÄYTTÖOHJEKOKONAISUUDEN SUUNNITTELUPROSESSI

3.1 Teknologiavalinnat

Täyttöohjekokonaisuuden valitut teknologiat olivat jo järjestelmässä valmiiksi käytössä olevia. Näistä teknologioista käyttöön arvioitiin tarvittavan React-kirjastoa ja Node.js-ajoympäristöä, joissa molemmissa olisi ohjelmointikielenä TypeScript. Reactia tulitisiin käyttämään jokaisessa ohjekokonaisuuden osassa, sillä jokainen osa tulisi sisältämään omat käyttöliittymäkomponentit. Nodea tarvittaisiin ylläpitotyökalua varten täyttöohjeiden hakemiseen ja päivittämiseen. Ylläpitoprosessia varten järjestelmään tulevat ohjeet haluttiin siirtää tietokannasta Amazon S3 -tietosäiliöön. Täyttöohjeiden ylläpitotyökalun teknologioihin arvioitiin vielä tulevan mukaan Docker ja SQL. SQL-kyselyiden avulla voitaisiin hakea lisättyjä tai poistettavia ohjeita tietokannasta. Docker taas auttaisi täyttöohjeiden päivitysprosessin automatisoinnissa.

3.1.1 React ja TypeScript

TypeScript on Microsoftin kehittämä ohjelmointikieli ja kääntäjä. Mikään tietokone ei tulkitse TypeScriptiä suoraan vaan se käännetään ajonaikaisesti JavaScriptiksi, jotta esimerkiksi selaimet ymmärtävät sitä. (Choi 2020.) JavaScript on suosituin ohjelmointikieli ja sitä käytetään yleisesti web-ohjelmoinnissa. JavaScriptiin on kuitenkin kohdistunut kritiikkiä sen dynaamisen tyyppityksen takia, joka varsinkin isommissa ohjelmissa on syy monille virheille. Dynaaminen tyyppitys tarkoittaa sitä, ettei muuttujille tarvitse asettaa tiettyä tyyppiä verrattuna staattiseen tyyppitykseen, jossa muuttujille täytyy asettaa tietty tyyppi. TypeScriptin tärkeimpinä etuina onkin sen tuomat olio-ohjelmoinnin tuki ja staattinen tyyppitys, pakottaen jokaisen kehittäjän kirjoittamaan koodia, jossa on vähemmän virheitä, mahdollistaen ohjelman paremman laadun ja helpomman ylläpidon. (Choi 2020.) TypeScript onkin noussut todella vahvaan suosioon ja onkin Redmonk-sivuston mukaan vuonna 2021 kahdeksanneksi suosituin ohjelmointikieli (Redmonk 2021). Varsinkin isommissa järjestelmissä, kuten EURAssa TypeScript tuo kehitykseen ja toimintaan varmempaa laatua.

React-kirjastosta voidaan hyödyntää sen monipuolisia ominaisuuksia teknisimmissä vaatimuksissa, kuten kenttäkohtaisissa ohjeissa. React on ohjelmistokehittäjä Jordan Walken kehittämä JavaScript-kirjasto ja sillä voidaan luoda nopeasti monimutkaisempiakin käyttöliittymiä, käyttäen niin kutsuttuja komponentteja. Komponentit ovat käyttöliittymän osia, kuten esimerkiksi nappi tai valikko sovelluksessa. Vuonna 2011 React otettiin käyttöön Facebookissa ja on nykyään käytössä suosituissa sovelluksissa, kuten Instagram, Uber Eats, Airbnb ja monessa muussa. Reactiin on tullut ajan mittaan paljon uusia ominaisuuksia, kuten React Router, React Hooks, React Context ja tilanhallintaan liittyviä kirjastoja, kuten Redux ja MobX. (Banks & Porcello 2020.) Reactissa yksittäiset komponentit voivat muodostaa isompia komponentteja, esimerkiksi kokonaisia lomakkeita. Näin Reactissa muodostuu komponenttihierarkia, jossa ylimpänä ovat isommat eli ylätasen komponentit ja alimpana pienemmät yksinkertaisemmat komponentit. (Choi 2020.)

React Router mahdollistaa sivujen välisen navigoinnin (Choi 2020). Tätä tultaisiin tarvitsemaan ainakin täyttöohjesivuille mentäessä ja täyttöohjesivulla olevan valikon navigoinnissa. Tämän lisäksi React Routeria tarvittaisiin myös navigoidessa täyttöohjeiden ylläpitotyökalun sivulle lomakkeelta.

React Hooksien avulla voidaan uudelleenkäyttää vaihtuvaa logiikkaa komponenttien välillä. (Banks & Porcello 2020.) React Hooksien käyttö oli luonnollista, sillä projektin komponentit oli jo tehty käyttäen Hookseja.

Tila on Reactissa se, joka ohjaa sitä, koska komponentin pitäisi päivittyä. Sillä tarkoitetaan komponentin tiettyä tilaa tai tietoa, esimerkiksi onko tietyssä kentässä tällä hetkellä kohdistus vai ei. (Choi 2020.) Tätä kuvattaisiin boolean arvona "true" tai "false". Kun tätä tilaa muutetaan oikeaoppisesti, esimerkiksi käyttäen React Hookseja, React osaa päivittää komponentin tilan oikeaan arvoon ja renderöidä komponentin uudestaan.

Alla olevassa kuvassa korostetulla rivillä on esimerkki, jossa on käytetty Reactin useState- ja useEffect-hookkia komponentin tilan muuttamiseen. Count eli lukuarvoon tallentuu komponentin tila ja lukuarvoa päivitetään napin

painalluksesta kutsumalla `setCount`-toimintoa. `useEffect`-hook tunnistaa, kun komponentin tila muuttuu ja se päivittää dokumentin otsikon. (ReactJS n.d.)

```
import React, { useState, useEffect } from 'react';

function Example() {
  const [count, setCount] = useState(0);

  // Similar to componentDidMount and componentDidUpdate:
  useEffect(() => {
    // Update the document title using the browser API
    document.title = `You clicked ${count} times`;
  });

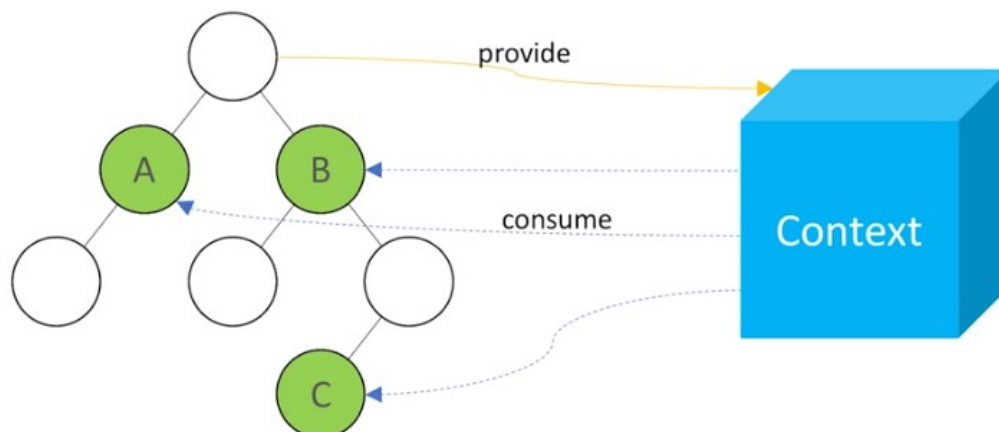
  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

KUVA 6. Yksinkertainen esimerkki React Hooksien käytöstä komponentin tilan manipulointiin (ReactJS n.d.)

Tilanhallintakirjaston kuten Reduxin tai MobX avulla tila voidaan tallettaa erilliseen paikkaan eli komponentin ulkopuolelle ja hakea tarvittaessa. Näin voidaan päästä käsiksi tilaan helposti mistä tahansa sovelluksessa. Tilanhallintakirjastot tulevat olennaisiksi suuremmissa sovelluksissa, kun komponenttien tiloja halutaan käsitellä muissa komponenteissa. Yksi mahdollisuus on kuljettaa halutun komponentin tietoa komponentista toiseen, mutta Reactissa tietojen kuljetus pitäisi tapahtua vain komponenttihierarkiassa ylhäältä alaspäin. Jos otettaisiin käyttöön Redux kirjasto, mahdollistaisi se tiedon jakamisen, päivittämisen ja käytön lokaalisti. (Choi 2020)

React Context on mekanismi, jolla voidaan tallettaa tietoa ja kertoa sitä käsitteleville komponenteille, jos tietoon on tullut muutos (Scott 2020). Alla oleva kuva havainnollista, miten React Context mekanismi toimii. Sen sijaan, että tavanomaisesti siirreltäisiin tietoa komponenttien välillä hierarkiassa ylhäältä alaspäin, voidaan määritellä komponenttihierarkiassa ylätasolle tiedon tarjoaja

(engl. provider) ja tiedon kuluttajat (engl. consumer). Tarjoaja ilmoittaa kun tieto muuttuu ja tallettaa uuden arvon kontekstiin (engl context), jonka jälkeen kuluttajat voivat hakea uuden tiedon. Yhdessä käyttämällä React Hookseja voidaan erikseen määritellä muutokset käyttöliittymässä, kun tieto vaihtuu. (Scott 2020.)



KUVA 7. React Context mekanismia havainnollistettuna. (LaptrinhX 2019)

Kenttäkohtaisiin ohjeisiin tarvittaisiin tiedon talletus paikassa, josta se olisi helposti saatavilla lomakkeilla. React Context vaikutti tähän tilanteeseen hyvin sopivalta, jolloin ei myöskään tarvitse sitoutua toteutuksessa tiettyyn tilanhallintakirjastoon. Redux ei ollut valmiiksi käytössä projektissa ja se toisi lisätyötä React Contextiin verrattuna. Reduxin haittapuolena onkin, että se aiheuttaa paljon ylimääräistä koodin kirjoittamista ja sen opetteleminen hyvin on aluksi vaikeaa. React Context aiheuttaa kuitenkin herkästi isommissa sovelluksissa tai järjestelmissä turhia verkkosivuston toimintaa hidastavia uudelleenrenderöintejä ja vaatii myös jonkin verran ylimääräistä koodin kirjoittamista, joten molemmissa on huonoja ja hyviä puolia. (Choi 2020.)

3.1.2 Node.js ja SQL

Node.js on JavaScript ajoympäristö, jolla voidaan rakentaa full-stack-sovelluksia (Banks & Porcello 2020). Full-stack-sovellukset tai järjestelmät, kuten EURA 2021 koostuvat front end osuudesta, eli käyttöliittymästä sekä back end-osuudesta eli palvelinpuolesta. Node on käytössä palvelinpuolen ohjelmoinnissa

ja ohjekokonaisuudessa sitä tarvitaan täyttöohjeiden hakemiseen ja päivittämiseen. Koska Node käyttää pohjana JavaScriptiä voidaan front end ja back end kehityksessä käyttää TypeScriptiä (Choi 2020). Tämä tekee kehityksen huomattavasti yksinkertaisemmaksi.

Alan Beaulieu kuvaa kirjassaan Learning SQL, 3rd Edition (2020) tietokantaa yksinkertaisesti kokoelmaksi, joka sisältää toisiinsa liittyvää tietoa. Relaatietietokannoissa tiedot on järjestelty tietokantatauluihin, joilla voi olla linkityksiä toistensa välillä. Tietokantataulu koostuu riveistä ja sarakkeista, jossa sarakkeella on kuvaus mitä arvoja sen alla olevat rivit tulevat sisältämään. Sarakkeen arvot voivat olla esimerkiksi täyttöohjeen yksilöllinen avaimia, jolloin sarakkeen nimeksi voisi tulla "avain". Rivit sisältävät arvoja sarakkeittain, ja jokaiselle riville tallettuisi yksi täyttöohje. SQL-kyselykieli on kehitetty tietokannassa olevan tiedon manipulointiin ja hakemiseen. (Beaulieu 2020.) Täyttöohjekokonaisuuden toteutuksessa SQL-kyselykieltä tarvitaan lisättävien ja poistettavien täyttöohjeiden hakemisessa tietokannasta, jotta ne voidaan prosessoida oikein ja tämän jälkeen päivittää nykyiset täyttöohjeet tietosäiliöstä.

3.1.3 Amazon S3 ja Docker

S3 tulee nimestä Simple Storage Server ja palvelu kuuluu Amazon Web Services pakettiin. Nimensä mukaisesti siellä pystyy säilyttämään oliomuotoista tietoa, esimerkiksi tiedostoja. (Amazon AWS n.d.) Täyttöohjeet halutaan säilyttäväksi Amazon S3 -tietosäiliöön, joten tämä oli vaadittu teknologia, joka täytyi ottaa käyttöön. S3-tietosäiliöstä on helppo hakea ja päivittää ohjeita. Tämän lisäksi täyttöohjeet on myös helposti saatavilla. S3 on myös turvallinen ja sillä voidaan helposti rajata, onko tietyt tiedostot julkisia vai ei. (Amazon AWS n.d.)

Docker on "alusta palveluna" -tyyppinen palvelu, jolla voidaan kehittää, kuljettaa ja ajaa sovelluksia (Docker n.d.). Dockeria haluttiin hyödyntää täyttöohjeiden ylläpitotyökalun automatisoinnissa, jotta voitaisiin pitää manuaaliset vaiheet mahdollisimman minimaalisina. Tämä tulisi hyödyttämään jokaista kehittäjää ja helpottamaan täyttöohjeiden ylläpitoprosessia. Täyttöohjekokonaisuutta ajatellen Dockeria tultaisiin käyttämään kirjoittamalla Dockerfile-nimiseen tekstitiedostoon

ohjeita, mitä pitäisi tehdä, kun järjestelmä käynnistetään. (Dockerfile reference n.d.)

3.2 Täyttöohjeiden koostesivun suunnittelu

Ensisijainen tarve oli lomakkeiden koostetuille täyttöohjeille jollain erillisellä sivulla. Tämä tulisi sekä virkailijoiden että hakijoiden puolelle niin, että molemmilla olisi omat täyttöohjesivut. Hakijoilla tulisi olemaan oma täyttöohjesivu hankehakemuksen täyttöohjeita varten ja virkailijoilla oma sivu hankehakuilmoituksen täyttöohjeita varten. Täyttöohjeet luotaisiin hierarkiseen muotoon ja lomakkeilla esiintyvien välilehtien sekä kenttien mukaiseen järjestykseen.

Hakijan täyttöohjesivulle oli valmiiksi suunniteltuna käyttöliittymämalli. Hakijan ja virkailijan sivut tulisivat näyttämään melko samalta, lukuunottamatta tekstejä ja värityksiä. Sivulla olisi yläpalkki, jossa olisi logo ja tekstiä, vasemmassa reunassa sivun sisäinen navigointipalkki ja keskellä varsinaisena sisältönä jokaisen navigointipalkin välilehden generoitu täyttöohjesisältö. Täyttöohjeiden navigointipalkin sisältö ja rakenne tulisi vastata hankehakemuksen ja hankehakuilmoituksen navigointipalkkia. Tämä haluttiin, jotta sivujen navigointipalkin rakenne olisi mahdollisimman samanlainen, jolloin käyttäjä voi helposti löytää lomakkeen tietyn välilehden täyttöohjeet. Alla olevassa kuvassa on UX-asiantuntijoiden luoma käyttöliittymäsuunnitelma eli UI-suunnitelma (engl. User Interface design) hakijan täyttöohjesivusta.

EURO 2021 Hakijan ohjeet hankehakuun

Hankesuunnitelma

Perustiedot

Hanketiedot

Täydentävät tiedot

Seurantatiedot

Kustannusarvio

Rahoitussuunnitelma

Liitteet

Hakemuksen jättäminen

Täydentävien tietojen yleisohje

4.7 Muilta rahoittajilta haettu rahoitus

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

4.8 Yhteydet muihin hankkeisiin

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

4.9 Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat.

KUVA 8. Hakijan täyttöohjesivun UI-suunnitelma

3.3 Kenttäkohtaisten ohjeiden suunnittelu

Hankehakuilmoituksella tai hankehakemuksella on monta eri välilehteä, jossa joillain välilehdillä on lomakkeen osia. Osat sisältävät monta erilaista kenttää, jotka voivat olla esimerkiksi tekstikenttiä tai valintanappeja. Jotta lomakkeen täyttäminen sujusi luontevasti ei riittäisi pelkät koostetut täyttöohjeet, sillä niissä täyttöohje pitää etsiä kentän mukaan ja ne avautuvat aina erilliselle sivulle.

Järjestelmään tarvittiin siis kenttäkohtaiset ohjeet, joilla pystyttäisiin näyttämään ohjeita samalla sivulla, millä hakija tai virkailija täyttää lomaketta. Näin täyttöohjeet näkyisivät yksi kerrallaan, tietylle kentälle, jolloin yksittäisen kenttään kohdistuvan täyttöohjeen erottaminen ja lukeminen olisi paljon helpompaa.

Kenttäkohtaisten täyttöohjeiden mallina oli UI-suunnitelma ja prototyyppi, josta pystyi hahmottamaan täyttöohjeiden toimintaa ja käyttöliittymää eri tilojen aikana. Lomakkeiden jokaisella välilehdellä, joille täyttöohjeita tarvitaan näkyisi sivukohtainen ohje, kun käyttäjä ei olisi minkään kentän kohdalla. Kun käyttäjä liikkuu hiirellä tai tabulaattorilla kentän kohdalle tai valitsee kentän, tulisi näkyviin kenttäkohtainen täyttöohje. Kenttäkohtaisen täyttöohjeen näkyessä ohjeen alareunassa olisi linkki, josta pääsisi valitun kentän ohjeen kohdalle täyttöohjesivulla. Tämä helpottaisi lomakkeen täyttäjää näkemään täyttöohjeet

koostetusti erillisellä välilehdellä tai lisätietoja kenttäkohtaisesta täyttöohjeesta jos sen pituus ylittää tietyn rajan, milloin sitä ei voisi näyttää kokonaisuudessaan lomakkeen sivussa. Täyttöohjetekstin ollessa yli tietyn pituuden, tulisi se näkyä rajoitettuna, eli näytetään esimerkiksi vain kymmenen riviä ja täyttöohjetekstin alla olevasta linkistä pääsisi lukemaan koko täyttöohjeen.

Alla olevassa kuvassa on UX-asiiantuntijoiden tekemä suunnitelma hankehakemukselta, jossa valittuna on vasemmassa navigointipalkissa ”Perustiedot” -välilehti. Keskellä sivua on lomakesisältö, jossa on kohdistettuna ensimmäinen kenttä (kohta 1) ”Hankkeen nimi”. Tälle kentälle on näkyvissä kenttäkohtainen ohje oikeassa reunassa (kohta 2) otsikon ”OHJE Kentän täyttöohje” alla, jossa on myös linkki ”Lue lisää ohjeista”. Tämä linkki veisi hakijan täyttöohjesivulle.

KUVA 9. Hakijan hankehakemuksen UI-suunnitelma

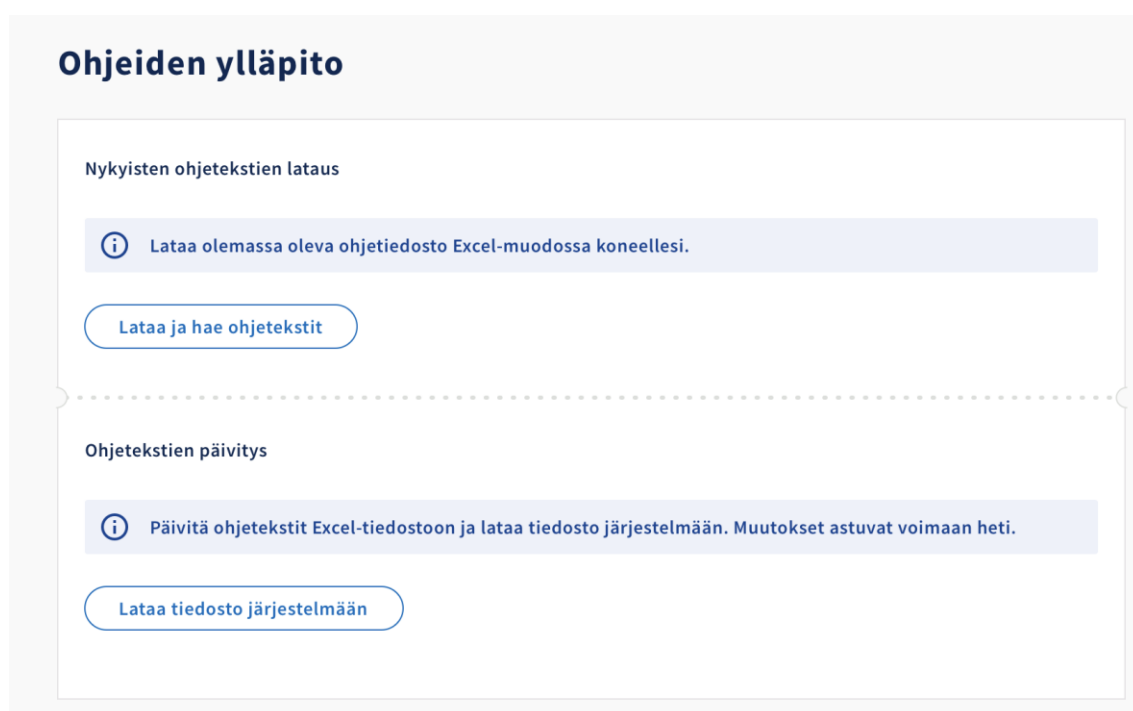
3.4 Täyttöohjeiden ylläpitotyökalun suunnittelu

Ohjeet voivat olla osuvia ja selkeitä silloin kun ne alunperin laitetaan järjestelmään. Ajan kuluessa itse toteutus tai kenttien määrittelyt voivat kuitenkin muuttua. Lisäksi voi tapahtua virheitä ohjeita syöttäessä järjestelmään, esimerkiksi kirjoitusvirheitä tai väärän ohjeen syöttämistä sille kuulumattomaan kohtaan. Kun täyttöohjeissa esiintyy virheitä, puutteita tai tulee uusia muutosehdotuksia, joudutaan ne päivittämään manuaalisesti järjestelmään.

Tämä onnistuu ainoastaan kehittäjän kautta ja välivaiheita tulee asiakkaan ja teknisen toteuttajan välille.

Tämän takia haluttiin myös erillinen täyttöohjeiden ylläpitotyökalu, jolla asiakas pystyisi itse helposti päivittämään ohjeita kirjaamalla päivitettyt ohjetekstit ja lataamaan ne järjestelmään. Tämä auttaisi asiakasta päivittämään ohjeita itsenäisemmin niin, että päivitystarpeista ei tarvitsisi erikseen viestitellä kehittäjien kanssa ja odotella kunnes ohjeet on päivitetty. Tämän lisäksi asiakas halusi, että työkalussa olisi toiminto, jolla myös nykyiset ohjeet voidaan ladata erilliseen Excel-tiedostoon.

Alla olevassa kuvassa on UI-suunnitelmaa täyttöohjeiden ylläpitotyökalun käyttöliittymästä. Sivulla olisi kaksi osiota, joissa molemmissa olisi samanlainen rakenne eli osiokohtainen otsikko, infoteksti ja nappi. ”Lataa ja hae ohjetekstit” -napista pystyisi lataamaan tämänhetkiset täyttöohjeet Excel-tiedostona ja ”Lataa tiedosto järjestelmään” -napista voisi ladata Excel-tiedostona päivitettyt täyttöohjeet järjestelmään. Tiedoston lataamisesta järjestelmään käynnistyisi päivitysprosessi, joka päivittäisi täyttöohjeet ajantasalle.



KUVA 10. Täyttöohjeiden ylläpitotyökalun UI-suunnitelma

4 TÄYTTÖOHJEKOKONAIKUUDEN KEHITYSPROSESSI

4.1 Täyttöohjeiden koostesivun kehitys

Täyttöohjeiden määrittelynä oli luoda generoitava erillinen täyttöohjesivu hankehakemukselle ja hankehakuilmoitukselle. Ennen kehityksen aloittamista tärkeimpänä selvitettävänä ongelmana oli, miten täyttöohjeet generoitaisiin oikeisiin navigointipalkin välilehtiin ja saataisiin ylläpidettyä tulevaisuudessa täyttöohjeiden ylläpitotyökalun avulla. Kenttäkohtaisen täyttöohjeen linkistä haluttiin myös päästä sivun täyttöohjeen tai tietyn kenttäkohtaisen täyttöohjeen kohdalle. Täyttöohjeen toteutus tulee siis linkittää kahteen muuhun tulevaan täyttöohjekokonaisuuden osaan.

Kehitys aloitettiin luomalla käyttöliittymä täyttöohjesivulle. Tähän käytettiin niin paljon olemassa olevia komponentteja kuin oli mahdollista. Sivun yläpalkki ja vasen sivumenu saatiin uudelleenkäyttämällä geneerisiä järjestelmässä olevia UI-komponentteja. Myös sivun keskellä olevan täyttöohjesisältö saatiin helposti laitettua rakennekomponentissa sille kuuluvaan paikkaan. Täyttöohjesivun välilehtien navigoinnissa käytettiin BrowserRouter komponenttia, joka tulee react-routing -kirjastosta. BrowserRouter tarjoaa navigointiin tarkoitettuja apumuuttujia ja sen avulla voidaan määrittää täyttöohjesivun sisällön vaihtuminen, kun klikataan jotain välilehteä navigointimenussa. Välilehden vaihtuminen muuttaa selaimen osoitteen tiettyyn arvoon ja arvolle on määritetty joku vastaava välilehti. (Choi 2020.)

Koska sisältö olisi melkein sama, haluttiin sivun luonnissa käyttää Reactin mahdollistamaa komponenttien uudelleenkäyttöä. Komponenteista tehtiin mahdollisimman geneerisiä ja niiden sisältö luotiin käännosteksteistä, riippuen navigoiko täyttöohjesivulle hakija vai virkailija. Kaikki tieto tulee olemaan täyttöohjesivulla staattista, eli toteutuksessa ei tarvinnut hallita tilojen muutoksia, vaan sivu generoidaan klikatessa linkkiä täyttöohjesivulle ja se muuttuu vasta kun sisältö generoidaan uudestaan. Täyttöohjesivun sisältö tulisi käännostekstien avaimista, mutta sen lisäksi piti selvittää miten käännökset saataisiin haluttuun kohtaan ja tasoon.

Täyttöohjeiden rakenne täytyi siis selvittää sisältöä generoidessa. Tähän tarkoitukseen sopi JSON-tietorakenne. JavaScript Object Notation eli JSON on JavaScript-pohjainen tekstiformaatti, jolla voidaan luoda hierarkista tietoa. JSON-tiedosto koostuu JavaScript olioita muistuttavista rakenteista, jossa on avaimia ja niiden arvoja. Lisäksi JSON-tietoon voidaan myös määritellä olioiden lisäksi taulukoita, joita voidaan ohjelmakoodissa käydä helposti läpi. (Mozilla Developer Network 2021.) JSON-formaatti ratkaisee ongelmat, sillä siinä voidaan määrittää haluttu sisältö tietyille tasoille ja asettaa oikeisiin kohtiin täyttöohjeiden avainarvoja, jonka perusteella käännostekstit haettaisiin. JSON-formaatti mahdollistaa myös sen, että tieto voitaisiin siirtää helposti JSON-tiedostona esimerkiksi AWS S3 -palveluun, jos tarve vaatisi.

Koska JSON-tiedossa käytettäisiin avainarvoja määrittelemään jokainen täyttöohjeteksti helpottaisi se täyttöohjeiden ylläpitoa, sillä muuttamalla tähän avaimen kohdistuvaa käännostä, päivittyisi sisältö täyttöohjesivulta. Tätä avainta voitaisiin myös käyttää suoraan kenttäkohtaisissa ohjeissa linkittämään kenttäkohtainen täyttöohje tietyille ohjetekstille täyttöohjesivulla. Näin voidaan helposti käydä JSON-tietoa ohjelmakoodissa läpi ja määritellä, mikä komponentti tai tyyli näytetään missäkin tasolla. Eri komponenttitasoja generoidussa tiedossa ovat esimerkiksi navigointivalikon välilehden otsikko, välilehden täyttöohjeiden sivun otsikko, täyttöohjeiden kenttäkohtaiset otsikot tai tekstit.

Generoitu sisältö luotiin käymällä JSON-tieto läpi ohjelmakoodissa ja luomalla eri tasoille omia komponentteja esimerkiksi otsikolle oma komponentti ja kenttäkohtaiselle oma komponentti. JSON-tieto laitettiin omaan tiedostoonsa. Täyttöohjesisällön komponentit laitettiin tukemaan rikastettua tekstiä rajoitteilla, eli vain osa tyyleistä, kuten lihavointi, linkit, luettelomerkit sallittiin. Jokaiselle komponentille asetettiin tietyt tyylit komponenttikohtaisesti, jolla saatiin komponentit näyttämään suunnitelman mukaiselta.

Alla olevasta kuvasta voidaan nähdä valmis toteutus hakijan täyttöohjesivusta. Ylhäällä on logo ja sivun tietoa. Vasemmassa sivussa on haluttu hankehakemuksen välilehtiä vastaava valikko, jossa tällä hetkellä valittu sivu on korostettu sinisellä. Sivulla on alussa välilehden yleiset täyttöohjeet ja sen alla

niin sanottu ”lakanakohtainen” otsikko eli lomakkeen osion otsikko. Sen alla on yleistä täyttöohjesisältöä, joka ei liity tiettyyn lomakkeen kenttään. Yleisen täyttöohjesisällön jälkeen näkyy kenttäkohtainen täyttöohjeotsikko ”Mikroyritys” ja sen alla kenttäkohtainen täyttöohjesisältö, joka näkyisi kentän ”Mikroyritys” täyttöohjeen kohdalla hankehakemuksella.

The screenshot shows the EURA 2021 application form interface. The header includes the EURA 2021 logo and the text 'Hakijan ohjeet hankehakuun'. A navigation bar at the top contains several 'TESTI' buttons. On the left, a sidebar lists various sections: Organisaation tiedot, Perustiedot, Hanketiedot, Täydentävät tiedot, **Määrälliset tavoitteet** (highlighted), Kustannusarvio, Rahoitussuunnitelma, Liitteet, and Hakemuksen jättäminen vira... The main content area is titled 'Määrällisten tavoitteiden yleisoheje'. It contains several paragraphs of text, including a note about required fields, a section titled 'Hankkeeseen osallistuvien yritysten arvioitu lukumäärä henkilöstömäärän mukaan (ESR+)', and a section titled 'Mikroyritys'.

KUVA 11. Hakijan valmis täyttöohjesivu

4.2 Kenttäkohtaiset täyttöohjeiden kehitys

Kenttäkohtaisille ohjeille lähtötilanteena oli vaatimus, jossa haluttiin kenttäkohtaisten täyttöohjeiden näkyvän hankehakemuksen ja hankehakuilmoituksen oikeassa sivussa.

Toteutuksessa täytyi valita, millä tavalla täyttöohjeen avain linkitettäisiin mihin tahansa kenttään tai toimintoon, esimerkiksi nappiin lomakkeella. Täytyi myös ottaa huomioon, millä tavalla tarkkaillaan, onko hiiren kursori tietyn kentän päällä tai onko lomakkeen kentässä kohdistus ja vaihdettaisiin sen mukaan kenttäkohtaista ohjetta.

Kenttäkohtaisille täyttöohjeille oli valmiina otsikko- ja ohjetekstikomponentti sijoitettuna lomakkeen oikeaan reunaan. Kehittäminen aloitettiin lisäämällä sivukohtaisen täyttöohjeen otsikko ja ohjeteksti, joka vaihtui lomakkeen avattuna

olevan välilehden mukaan. Komponentti koostui siis kolmesta eri osasta, otsikosta, täyttöohjeteksistä ja linkistä.

Seuraavaksi täytyi kehittää toteutus, jolla saisi vaihdettua sivun täyttöohjeen kenttäkohtaiseen täyttöohjeeseen, kun hiiren kursori olisi kentän päällä tai kentässä olisi kohdistus. Tarvittiin ratkaisu, jolla lomakekomponentti kertoo, milloin täyttöohje pitäisi vaihtaa sivukohtaisesta ohjeesta kenttäkohtaiseen ohjeeseen. Tähän soveltui hyvin toteutukseen valittu React Context -mekanismi ja tästä luotiin lomakkeen ympärille KenttäohjeContextProvider eli kontekstin tarjoajakomponentti. Kenttäkohtaisen täyttöohjeen komponentti asetettiin ContextConsumeriksi eli kontekstin kuluttajaksi, käyttäen siihen soveltuvaa React Hookkia – useContext. Banks ja Porcello (Banks & Porcello 2020) kuvaavat Reactin kontekstin toimintaa hyvin vertaamalla kontekstin tarjoajakomponenttia lähtölentokenttään, jossa tieto eli tilat täyttöohjeiden tapauksessa talletetaan koneen ruumaan. Kontekstin kuluttajakomponentit vastaavat koneen laskeutumisaikoja joissa, tietoa puretaan ruumasta ja käsitellään. (Banks & Porcello 2020.) Tarjoajakomponentti sisältää siis useState-hookkeja, jossa säilötään tiloja, eli onko hiiren kursori kentän päällä, onko kentässä kohdistus ja näihin kohdistuvan kentän avainarvo sekä kaikkien lueteltujen tilojen arvojen muutostoiminnot. Kuluttajakomponentti eli kenttäkohtainen täyttöohje saa arvot useContext-hookista ja muuttaa kenttäkohtaisen tai sivukohtaisen täyttöohjeen saatujen arvojen mukaan.

Tarvittiin siis enää muutostoimintojen kutsu kenttien tilojen mukaan. Tämä ratkaistiin asettamalla jokaiselle lomakkeen kentälle WrapperOhjeet-komponentti, jolle annettiin arvona kentän täyttöohjeen yksilöllinen avain. Tämän avulla tiedetään, mikä kenttä on kyseessä, ja mikä täyttöohje pitää näyttää lomakkeen sivussa. Wrapper-komponentti asetettiin myös kontekstin kuluttajaksi ja sille asetettiin tilojen kuuntelijoita, jotka tarkkailivat tiloja, kuten onko kentässä kohdistus vai ei. Tilojen muutoksista kutsuttiin kontekstin tilojen muutostoimintoja, joiden avulla vaihdettiin kontekstin arvoja ja arvojen muutoksista vaihdettiin kenttäkohtaista täyttöohjetta.

Lopuksi kenttäkohtainen täyttöohjeteksti laitettiin tukemaan rikastettua tekstiä, jonka avulla saatiin kenttäkohtaisiin täyttöohjeisiin saatiin muotoiltua tekstiä

vastaavasti kuin täyttöohjesivulla. Tämän lisäksi kehitettiin kenttäkohtaisen täyttöohjeen alla oleva linkki navigoimaan, joko täyttöohjesivun alkuun tai kenttäkohtaisen ohjeen kohdalle täyttöohjesivulla.

Valmis kenttäkohtaisen täyttöohjeen toteutus on kuvattuna alla olevassa kuvassa. Tässä on näkyvissä hakijan puolen hankehakemus, jossa on valittuna välilehti ”Hanketiedot” ja aktiivisena kenttä (kohta 1) ”Mitkä ovat hankkeen välilliset kohderyhmät?”. Aktiivinen kenttä tarkoittaa myös sitä, että kentässä on kohdistus, jolloin oikeassa reunassa näkyy tämän kentän kenttäkohtainen täyttöohje (kohta 2) otsikolla ”OHJE Kentän täyttöohje” ja sen alla täyttöohjesisältö, jossa tuetaan rikastettua tekstiä, kuten täyttöohjeen seassa olevaa linkkiä. Täyttöohje on tämän lisäksi liian pitkä rivimäärältään, joten siitä on leikattu loput ohjeesta pois ja korvattu kolmella pisteellä. Lopuksi täyttöohjeessa on linkki ”Lue lisää”, josta navigoitaisiin kentän täyttöohjeen kohdalle hakijan täyttöohjesivulla.

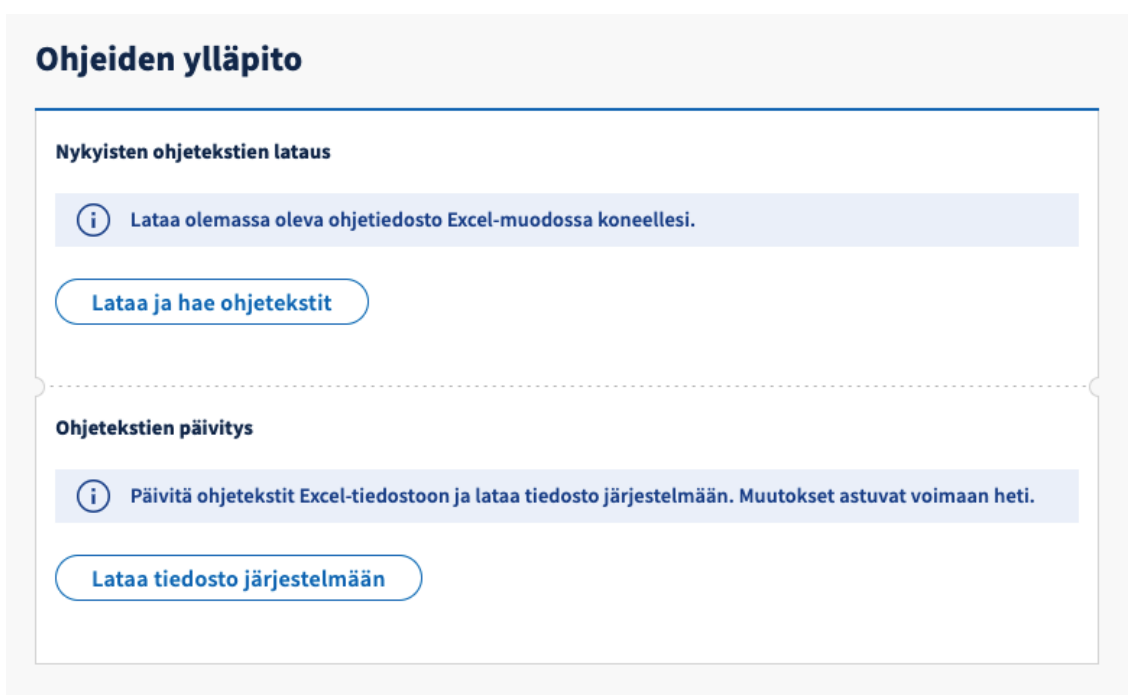
KUVA 12. Hankehakemuksen valmis kenttäkohtainen täyttöohje

4.3 Täyttöohjeiden ylläpitotyökalun kehitys

Täyttöohjeiden ylläpitotyökalua lähdettiin kehittämään viimeiseksi. Alustavaa suunnitelmaa siitä oli jo hahmoteltu ja piirretty yksinkertainen arkkitehtuurikaavio. Täyttöohjesivujen ja kenttäkohtaisten täyttöohjeiden kehityksessä oli otettu huomioon tuleva täyttöohjeiden ylläpitotyökalu, eli laitettiin kaikki osat käyttämään samoja täyttöohjeiden avaimia. Kun täyttöohjeita päivitetään tietyille

täyttöohjeiden avainarvolle, päivittyä täyttöohje molempiin ilman manuaalisia vaiheita.

Ylläpitotyökalun kehitys aloitettiin UI-suunnitelman mukaan kehittämällä yksinkertaista ylläpitokäyttöliittymää täyttöohjeiden ylläpitäjälle. Käyttöliittymä koostui yksinkertaisesta sivusta, jolla oli järjestelmän vastaava tyyli ilman navigointipalkkia vasemmassa reunassa. Käyttöliittymään kehitettiin kaksi osiota, jossa molemmissa oli otsikko, infoteksti ja nappi. Toisesta napista pystyi lataamaan tämänhetkiset täyttöohjeet järjestelmästä ja toisella päivittämään järjestelmän nykyiset täyttöohjeet uusilla.



KUVA 13. Täyttöohjeiden ylläpitotyökalun valmis käyttöliittymä

Seuraavaksi aloitettiin toteuttamaan täyttöohjetekstien latausta Excel-tiedostona. Tätä varten haluttiin siirtää täyttöohjetekstit Amazon S3-tietosäiliöön, josta ne olisi helposti haettavissa. Latausnappiin toteutettiin toiminto, jolla S3-tietosäiliöstä haettiin täyttöohjetekstit ja niistä muodostettiin Excel-tiedosto, jossa oli kaikki järjestelmän täyttöohjeet.

Tämän jälkeen toteutettiin täyttöohjeiden lataaminen järjestelmään Excel-tiedostona. Tätä varten luotiin Excel-sarakkeet alkuperäiselle käännöstekstille ja päivitetylle käännöstekstille, minkä mukaan teksti luettiin latauksessa ja

päivitettiin täyttöohjeet S3-tietosäiliöstä. Validoimalla luettava tieto, saatiin prosessista mahdollisia virheitä minimoitua. Validointi sisälsi esimerkiksi tarkistuksia, että vain halutuista sarakkeista luetaan tietoa.

Seuraavaksi toteutettiin migraatiopalvelu, jossa täyttöohjeita pystyttiin lisäämään, poistamaan tai päivittämään tietokantaan migraatioilla. Migraatioissa voidaan käyttää esimerkiksi SQL-kyselykieltä ja muuttaa tietokannan tilaa pienissä osissa (Prisma 2021). Tietokantaan ajettuja migraatioita varten kehitettiin toinen palvelu, joka prosessoi täyttöohjemuutokset tietokannasta S3-tietosäiliöön. Prosessiin tuotiin automatisointia käyttäen apuna Docker-ajoympäristöä. Jokainen palvelu sisälsi omat Dockerfile-tiedostot, jossa oli ohjeet sille, miten prosessi saadaan ajettua (Dockerfile reference n.d.). Tämä auttoi manuaalivaiheiden minimoimisessa.

Tämän lisäksi kehitettiin rajapinta, josta täyttöohjetekstejä haettaisiin järjestelmän käyttöliittymien käyttöön. Rajapinta hakisi käännostekstit S3-tietosäiliöstä ja sitä kutsuttaisiin käännöspalvelusta.

5 TOTEUTUKSEN HAASTEET, ONGELMAT JA NIIDEN RATKAISU

5.1 Saavutettavuusparannukset kenttäkohtaisissa ohjeissa

Jotta ymmärretään saavutettavuusvaatimuksia paremmin, on syytä avata mihin vaatimukset ja standardit oikein perustuvat. The World Wide Web Consortium (W3C) on kansainvälinen yhteisö, joka ylläpitää ja kehittää standardeja Internetille. Standardien tarkoituksena on kasvattaa ja edistää Internetin jatkuvaa kehitystä. (W3C 2021.) W3C sivuilta löytyy myös julkaistu Web Content Accessibility Guidelines (WCAG) 2.1, eli Internetin standardoidut ohjeistukset saavutettavuudelle (W3C 2018).

Web-kehitysprojekteissa suunnitellaan alussa tarpeiden mukaan, minkälainen saavutettavuustaso sivustolla tai järjestelmällä tulee olla. W3C:n saavutettavuusohjeistuksessa (W3C 2018) mainitaan eri saavutettavuustasot ja Laura Kalbag kuvaakin tasoja kirjassaan Accessibility for Everyone (2017) seuraavasti:

- A – minimitaso, täyttää minimivaatimukset
- AA – keskitaso, täyttää vaatimukset A ja AA tasolla
- AAA – korkein taso, täyttää vaatimukset A, AA ja AAA (Kalbag 2017.)

Julkisten sivustojen tulee lakisääteisesti olla täysin saavutettavia. Tästä hyviä esimerkkejä ovat Suomessa verottajan tai Kelan sivut. Finlexin sivuilta löytyy seuraava laki: Laki digitaalisten palvelujen tarjoamisesta (306/2019), jossa määritellään muun muassa saavutettavuuteen liittyviä vaatimuksia digitaalisten palveluiden suunnittelussa ja kehityksessä. Tämä laki pohjautuu EU-direktiiviin 2016/2102, eli saavutettavuusdirektiiviin. (Finlex 2019.)

Kenttäkohtaisissa ohjeissa ongelmaksi osoittautui saada suunnitelman mukaisesta toteutuksesta täysin saavutettava. Ongelma esiintyi silloin, jos haluttiin navigoida lomakkeen kentästä lomakkeen oikeassa reunassa olevaan kenttäkohtaisen ohjeen linkkiin. Tällöin ei ole mahdollisuutta päästä pelkästään tabulaattorilla navigoimalla kenttäkohtaiselle ohjeelle. Ongelma todettiin status-

palavereissa, jossa käsiteltiin täyttöohjesivujen ja kenttäkohtaisten täyttöohjeiden toteutusta ja käyttäjäkokemusta yhdessä UX-asiantuntijoiden kanssa.

Status-palavereissa esiintyi ratkaisuehdotuksena kenttäkohtaisen ohjeen siirtäminen lomakkeen kentän kohdalle, esimerkiksi kentän oikeaan yläreunaan info-nappina. Tämä ratkaisisi tabulaattorilla navigoimisen ongelman, eli kentän info-nappiin ja kenttäkohtaiseen ohjeeseen pääsisi peräkkäin helposti liikkumaan tabulaattorilla. Tässä oli ongelmana kuitenkin se, että jokaiselle kentälle navigoitaessa tabulaattorilla, tulisi yksi ylimääräinen tabulaattoripainallus lisää. Selain ohjaisi siis automaattisesti aina tietyn kenttäkohtaisen ohjeen kohdalle, vaikkei käyttäjä tätä haluaisikaan. Tästä syystä ajatus jäi lähinnä ideointitasolle.

5.2 Renderöintiongelmat ja täyttöohjeiden välkkyminen

Hankehakemuksella huomattiin renderöintiongelmia kenttäkohtaisissa täyttöohjeissa. Lomakkeelle asetettiin komponentti KenttäohjeContextProvider eli kontekstin tarjoaja. Aina kun lomakkeen kentässä liikuttiin päivittyi kontekstin tarjoajan arvot ja kenttäkohtainen täyttöohje renderöitiin uudestaan. Tämä ei varsinaisesti näkynyt suoraan suorituskyvyssä, sillä kyse on kuitenkin suhteellisen pienestä asiasta, mutta syy oli joka tapauksessa ohjelmointivirhe ja tuli paikata.

Lisäksi asiakkaan omasta testauksesta tuli palautetta, että kenttäkohtaiset täyttöohjeet välkkyivät liian nopeasti lomakkeilla ja tämä näytti sekavalta. Välkkyminen aiheutui siitä, että lomakkeella liikuttiin nopeasti tabulaattorilla tai hiiren kursori osui kentän kohdalle monta kertaa lyhyessä ajassa, tällöin kenttäkohtainen täyttöohje vaihtui useasti ilman erityistä viivettä tai siirtymää. Varsinkin hiirellä nopeasti liikkuesssa kenttäkohtainen täyttöohje vaihtui kenttien välillä sivukohtaiseksi täyttöohjeeksi ja aikaansai lomakkeella "rauhattoman" täyttöohjeiden välkkymisen.

Tämä korjattiin asettamalla rauhallisempi siirtymätyyli kenttäkohtaiselle täyttöohjekomponentille. Tämä saatiin aikaiseksi käyttämällä useState-hookkia tallettamaan kenttäkohtaisen täyttöohjekomponentin tyyli ja vaihtamaan sitä, kun

kenttäkohtainen täyttöohje vaihtui. Täyttöohjeen vaihtumisen tarkistamiseen käytettiin useEffect-hookkia ja siinä tyylin vaihto asetettiin aina kun kontekstissa vaihtui täyttöohjeen avainarvo.

Tämän lisäksi piti vielä muuttaa sitä, kuinka usein kenttäkohtaisia täyttöohjeita uudelleenrenderöitiin. Tässä ratkaisuna oli asettaa puolen sekunnin viive täyttöohjeen vaihtumiseen käyttämällä useCallback-hookkia ja debounce-toimintoa. Debounce-toiminto tuli suoraan lodash -nimisestä JavaScript-kirjastosta, joka sisältää yleishyödyllisiä toimintoja esimerkiksi tiedonkäsittelyyn. Debouncen avulla saatiin asetettua puolen sekunnin viive tilan vaihtamiselle kontekstin tarjoajassa ja se tarjoaa samalla mahdollisuuden peruuttaa tilan vaihto, kutsumalla "cancel"-toimintoa useEffect-hookin sisällä. Cancel-toiminnon kutsuminen on erityisen hyödyllistä silloin kun sivua vaihdetaan ja toiminto muuten jäisi taustalle, jolloin seuraisi virheilmoitus. (Lodash n.d.) UseCallback on käytännöllinen, kun kenttäkohtaisen täyttöohje vaihtuu nopeasti. Sillä varmistetaan, että debounce-toiminto luodaan vain kerran. (ReactJS 2021.)

Ongelman ratkaisi kaksi muuta kehittäjää ja korjaus poisti kenttäkohtaisen täyttöohjekomponentin turhat uudelleenrenderöinnit sekä paransi huomattavasti käyttökokemusta lomakkeella tabulaattorilla sekä hiirellä liikkeessa.

5.3 Täyttöohjeiden päivitys ilman ylläpitotyökalua

Täyttöohjeiden lisääminen ja päivittäminen osoittautui yllättävän työlääksi ilman täyttöohjeiden ylläpitotyökalua. Koska täyttöohjekokonaisuuden kehitysprosessi aloitettiin täyttöohjeisivuista ja sen jälkeen siirryttiin kehittämään kenttäkohtaisia ohjeita, ehti kehitysvaiheessa kulua sen verran aikaa, että täyttöohjeita joutui lisäämään ja päivittämään useampaan kertaan.

Päivittäminen vei paljon aikaa myös asiakkaalta, sillä suoraa täyttöohjeiden ylläpitomahdollisuutta ei ollut. Jokaisesta lisäys- tai päivitystarpeesta asiakas joutui lähettämään tiedostona, mitä ohjeita halutaan lisätä tai päivittää. Varsinkin täyttöohjeiden testaus osoittautui hankalaksi, sillä joutui vertailemaan, mikä on muuttunut, ja mitä ohjeita halutaan päivittää.

Tässä korostui erityisesti se, kuinka tärkeä osa täyttöohjekokonaisuutta ylläpitotyökalu tulee olemaan, ja miksi sen kehitystä halutaan priorisoida täyttöohjekokonaisuuden suhteen.

6 POHDINTA

Tavoitteena oli kehittää suunnitelmia vastaava ja lomakkeiden täyttöprosessia helpottava kokonaisuus. Tuloksena valmistui koostetut täyttöohjesivut virkailijan ja hakijan puolelle. Tämän lisäksi valmistui kenttäkohtaiset ohjeet hankehakemukselle ja hankehakuilmoitukselle. Täyttöohjeiden ylläpitotyökalua kehitettiin suunnitelman mukaan ja arkkitehtuuria hiottiin kehityksen aikana statuspalavereissa. Ylläpitotyökalu saatiin hyvään vaiheeseen, mutta ei vielä täysin valmiiksi.

Täyttöohjesivussa ei otettu mallia muiden palvelujen täyttöohjeista, mutta tulokset sisälsivät samoja piireitä, mitä muissa tutkituissa asiointipalveluissa oli. Tähän sisältyi hierarkinen rakenne, tyylit tekstien tasojen erotteluun ja rikastetun tekstin tukeminen. Samankaltaisuus johtuu todennäköisesti, siitä että saavutettavat täyttöohjeet tulee olla käyttäjälle selkeässä ja hierarkisessa järjestyksessä, sekä kaikki kohdat pitää olla merkattuna omille tasoilleen. Pieni samankaltaisuus on siis hyvä asia, koska tähän käyttäjät ovat todennäköisesti eniten tottuneet ja muiden palvelujen täyttöohjeita on käyttänyt jo useampi käyttäjä, jolloin täyttöohjeita on muokattu käyttäjäystävällisemmäksi.

Täyttöohjeiden ylläpitotyökalu tullaan viimeistelemään lähitulevaisuudessa. Toteutuksessa täytyy viimeistellä palvelujen yhteentoimivuus, lisätä testikattavuutta ja hakea käännöspalvelusta käännökset kehitetyn rajapinnan kautta käyttöliittymien käyttöön. Täyttöohjeen ylläpitotyökalun viivästys aiheutti lisätyötä asiakkaalle ja kehittäjille. Oliskin ollut mielenkiintoista nähdä miten kehitysprosessi olisi helpottunut jos ylläpitotyökalun kehitys olisikin aloitettu samaan aikaan kuin täyttöohjeiden koostesivu. Tällä olisi voitu myös mahdollisesti vähentää täyttöohjeiden päivittämiseen kulunutta aikaa. Testauksesta tulleen palautteen mukaan täyttöohjeissa oli paljon epäselvyyttä sisällön suhteen ainakin kehitysprosessin alussa, sillä täyttöohjeita tuli vähän kerrallaan asiakkaalta. Puuttuva täyttöohjesisältö aiheutti epäselvyyksiä testausprosessissa, sillä oli vaikea verrata toimitettuun täyttöohjedokumenttiin, oliko täyttöohjeita vielä toimitettu vai oliko toteutuksessa virheitä.

Täyttöohjekokonaisuuden kehityshaasteisiin löytyi ratkaisut. Täyttöohjesivulle saatiin generoitua JSON-tiedostosta hierarkinen täyttöohjelistaus, josta kävi selkeästi ilmi, mitkä ohjeet koskevat sivun yleisiä ohjeita ja mitkä ovat kenttäkohtaisia ohjeita. Kenttäkohtaisten täyttöohjeiden kehityksessä saatiin ratkaistua se, miten kenttiin sidotaan yksilöllinen ohjeen avainarvo käyttämällä Wrapper-komponenttia, jolle annettiin tietty avainarvo. React Contextin ja React Hooksien avulla ratkaistiin myös kenttäkohtaisten täyttöohjeiden dynaaminen näyttäminen lomakkeiden oikeassa reunassa. React Context osoittautui toimivaksi ratkaisuksi, mutta on vaikea arvioida olisiko toteutuksesta tullut parempi tai helpommin kehitettävä käyttäen tilanhallintakirjastoa, kuten Reduxia. Renderöintiongelmien esiintyminen oli oletetuin ongelma React Contextia valittaessa, sillä ongelmien mahdollinen esiintyminen oli huomioitu jo vertailuvaiheessa. Täyttöohjeiden ylläpitotyökalussa ratkaistiin, miten ylläpidon virhetilanteilta voidaan välttyä validoimalla päivityslogiikka oikeaoppisesti ja saatiin suunniteltua arkkitehtuuri, joka teki täyttöohjeiden päivitysprosessista mahdollisimman vähän manuaalivaiheita sisältävän.

Täyttöohjekokonaisuudesta saatiin saavutettava, mutta puutteita esiintyi navigoitaessa tabulaattorilla lomakkeen kentästä kenttäkohtaiselle täyttöohjeelle. Tähän esiintyi ratkaisuehdotuksena ainakin yksi ideatasolla harkittava korjaus asettamalla kenttäkohtaisen täyttöohjeen näkymiseen infonappi kentän ylänurkkaan. Tämä oli kuitenkin ratkaisu, joka olisi tuonut mukanaan käyttäjälle vaikeuksia tabulaattorilla navigoidessa. Jatkossa voisi miettiä minkälainen ratkaisu tähän olisi mahdollinen, jotta kenttäkohtaisista ohjeista saataisiin täysin saavutettava. Ongelman ratkaisussa täytyisi ottaa huomioon hyötysuhde, eli paljonko aikaa korjaamiseen haluttaisiin käyttää aikaa, siitä saataviin hyötyihin verrattuna. Olisi kuitenkin hyvä saada myös tabulaattorilla navigoivat myös navigoimaan järkevästi täyttöohjesivulle, varsinkin silloin, kun täyttöohjeteksti ei näy kokonaisuudessaan lomakkeen oikeassa sivussa. Tabulaattorilla navigointi on osa saavutettavuusvaatimuksia, jos halutaan mahdollisimman hyvä luokitus eli esimerkiksi AAA-saavutettavuustaso järjestelmälle (Kalbag 2017).

Kenttäkohtaisissa täyttöohjeissa esiintyi renderöintiongelmia ja nämä korjasi toinen kehittäjä. Tällöin ratkaisun tuloksen raportointi oli tulkintaa ohjelmakoodista, eikä suoraan oman kehityksen raportointia. Tässä saattaa

esiintyä havaintovirheitä eli täysin luotettava ratkaisun raportointi ei ole. Myös sulavuuden ja viiveen lisääminen oli toisen kehittäjän tulosta, joten näiden raportointi oli myös tulkintaa ohjelmakoodista.

Kaikkia tuloksia ei voitu raportoida kovin yksityiskohtaisesti ja esimerkiksi ohjelmakoodia ei voinut suoraan esittää opinnäytetyössä, sillä siihen voi kohdistua tietoturvariskejä. Tämän takia työssä ei esiinny ohjelmakoodeja lainkaan ja on haluttu pitää varmuuden vuoksi sisältö kuvina ja tekstisisältönä. Viranomaisen puolen tuloksia ei myöskään voitu esitellä samasta syystä. Nämä saattavat heikentää tuloksesta saatavaa hyötyä ja selkeyttä varsinkin tulosten esittelyssä ja teknisimmissä teoriaosuuksissa.

Opinnäytetyössä on käytetty monipuolisesti erilaisia luotettavia lähteitä. Lähteissä on mukana jatkuvasti päivitettäviä alalla yleisesti käytettyjä dokumentaatioita, ohjeistuksia, luotettavaa ja kattavaa alan kirjallisuutta monista eri lähteistä ja järjestelmää sekä taustaa kuvaavia lähteitä toimeksiantajan ja asiakkaan sivuilta. Lähteisiin valikoitui paljon täyttöohjekokonaisuuden teknologioihin liittyviä verkkolähteitä ja kirjallisuutta, joita käytettiin tukemaan täyttöohjekokonaisuuden kehitysprosessia.

Opinnäytetyön tuloksia, ongelmia ja kehitysehdotuksia voitaisiin hyödyntää myös EURA 2021 -järjestelmän ulkopuolella. Tuloksista voitaisiin hahmottaa, miten täyttöohjeiden kehitysprosessi kannattaa aloittaa, minkälaisia ongelmia voi tulla vastaan ja miten näitä voitaisiin ratkaista tai ennaltaehkäistä. Työtä voisi hyödyntää esimerkiksi järjestelmän täyttöohjeiden tarpeellisuuden arviointiin, sillä yleensä erillisiä täyttöohjeita ei tarvita jos lomakkeet eivät ole pitkiä tai ne eivät sisällä mahdollisesti vieraita käsitteitä tai monimutkaisia kohtia. Tämän vuoksi työssä kuvattiinkin vain julkisen sektorin muita asiointipalveluja, kuten Vero, Kela ja EURA 2014. Jos täyttöohjeita voitaisiin kuitenkin harkita, suunnitellessa voisi esittää kysymyksiä, kuten tarvitaanko täyttöohjeita tai millaista hyötyä täyttöohjeista voisi olla.

Tulevaisuuden vaatimuksena ilmeni tarve, että täyttöohjeet eivät olisi täysin koostetusti ohjesivulla, vaan ne generoitaisiin dynaamisesti riippuen tietyistä valinnoista lomakkeella, esimerkiksi tietylle rahastolle kohdistetulta

hankehakemukselta pääsisi vain tuon rahaston kyseisiin ohjeisiin. Täyttöohjesivun dynaamisuus tulee olemaan lähitulevaisuudessa melko ajankohtainen aihe ja onkin syytä miettiä, miten ohjeet muunnettaisiin nykyisestä toteutuksesta generoitavaksi lomakkeen tiettyjen valintojen perusteella. Mahdollisuus olisi tehdä linkityksiä JSON-tietoon, jolloin voitaisiin eritellä mitkä täyttöohjeet kuuluvat mihinkin kriteereihin ja näiden perusteella näyttää täyttöohjesivulla.

Täyttöohjekokonaisuudesta olisi hyvä saada palautetta järjestelmän käyttööntovaiheessa hakijoiden ja virkailijoiden kautta. Tämä vaatii toki aikaa, sillä palautteita tulee ajan mittaan lomakkeiden täyttöprosessien käynnistyessä. Täyttöohjeita testasi asiakas ja testaajat, mutta tuotantokäytössä tulisi vasta palaute, onko täyttöohjeista todella saatu käyttäjäystävälliset ja selkeät sekä mitä parannuksia pitäisi tehdä jos näin ei ole. Lisäksi jatkokehityksenä voitaisiinkin miettiä, miten ja mitä osia täyttöohjetyökalun toteutuksesta voitaisiin hyödyntää järjestelmän muiden tekstien päivityksessä, sillä kehityksessä otettiin huomioon myös tuleva tarve tekstien päivitykselle järjestelmässä ylläpitotyökalun avulla.

LÄHTEET

Amazon AWS. n.d. Amazon S3. Luettu 2.11.2020. <https://aws.amazon.com/s3/>

Banks, A. & Porcello, E. 2020. Learning React, 2nd Edition. O'Reilly Media, Inc.

Beaulieu, Alan. 2020. Learning SQL, 3rd Edition. O'Reilly Media, Inc.

Case: EURA 2021 -järjestelmä. n.d. Netum. Luettu 1.11.2021.
<https://www.netum.fi/asiakkaat/case-aura-2021-jarjestelma/>

CASE TEM: EURA 2014 -HANKE. n.d. Netum. Luettu 1.11.2021.
<https://www.netum.fi/asiakkaat/case-tem-aura-2014-hanke/>

Choi, D. 2020. Full-Stack React, TypeScript and Node. O'Reilly Media, Inc.

Docker. n.d. Docker overview. Luettu 2.11.2020. <https://docs.docker.com/get-started/overview/>

Dockerfile Reference. n.d. Docker. Dokumentaatio. Luettu 22.11.2021.
<https://docs.docker.com/engine/reference/builder/>

EU:n alue- ja rakennepolitiikka. n.d. Työ- ja elinkeinoministeriö. Luettu 1.11.2021. <https://tem.fi/eu-alue-ja-rakennepolitiikka>

EU:n rakennerahastot. n.d. Työ- ja elinkeinoministeriö. Luettu 1.11.2021.
<https://tem.fi/suomen-rakennerahasto-ohjelma-2014-2020>

Finlex. 2019. Laki digitaalisten palvelujen tarjoamisesta 15.3.2019/306. Luettu 7.11.2021. <https://www.finlex.fi/fi/laki/alkup/2019/20190306>

Kalbag, L. 2017. Accessibility for Everyone. O'Reilly Media, Inc.

Kela. 2019. Työnantajan työterveyshuollon verkkohakemuksen täyttöohje. Ohjetiedosto. Luettu 14.11.2021.
<https://www.kela.fi/documents/10180/2065374/Ty%C3%B6terveyshuollon+asiointipalvelun+t%C3%A4ytt%C3%B6hje/6208c049-9413-4181-8c40-009b684537e1?version=1.7>

Kestävää kasvua ja työtä 2014-2020 -ohjelma. n.d. Rakennerahastot. Luettu 22.11.2021. <http://www.rakennerahastot.fi/kestavaa-kasvua-ja-tyota-2014-2020-ohjelma>

LaptrinhX. 2019. Advanced Components and Context API for React Learning. Blogi-kirjoitus. Luettu 5.11.2021. <https://laptrinhx.com/advanced-components-and-context-api-for-react-learning-913407994/>

Lodash. n.d. Documentation. Dokumentaatio. Viitattu 20.11.2021.
<https://lodash.com/docs/4.17.15#debounce>

Mozilla Developer Network. 2021. Working with JSON. Luettu 15.11.2021. <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>

O'Grady, S. 2021. The RedMonk Programming Language Rankings: June 2021. RedMonk. Blogi-kirjoitus. Luettu 31.10.2021. <https://redmonk.com/sogrady/2021/08/05/language-rankings-6-21/>

Prisma. n.d. What are database migrations? Luettu 22.11.2021. <https://www.prisma.io/dataguide/types/relational/what-are-database-migrations>

Rakennerahastot. 2016. FI ESR Henkilö -järjestelmän täyttöohje. Luettu 14.11.2021. http://www.rakennerahastot.fi/materiaalipankki?p_p_id=110_INSTANCE_70HfiyzhIXHI&p_p_lifecycle=0&p_p_state=normal&p_p_mode=view&p_p_col_id=column-1&p_p_col_pos=1&p_p_col_count=3&110_INSTANCE_70HfiyzhIXHI_struts_action=%2Fdocument_library_display%2Fview_file_entry&110_INSTANCE_70HfiyzhIXHI_redirect=http%3A%2F%2Fwww.rakennerahastot.fi%2Fmateriaalipankki%2F%2Fdocument_library_display%2F70HfiyzhIXHI%2Fview%2F630592&110_INSTANCE_70HfiyzhIXHI_fileEntryId=630645

Rakennerahastot. n.d. Uudistuva ja osaava Suomi 2021-2027. Luettu 21.11.2021. http://www.rakennerahastot.fi/ohjelmakausi2021_2027

ReactJS. 2021. Introducing Hooks. Dokumentaatio. Luettu 14.11.2021. <https://reactjs.org/docs/hooks-intro.html>

Scott, D. 2020. Redux vs. Context vs. State. Blogi-kirjoitus. Luettu 4.11.2021. <https://betterprogramming.pub/redux-vs-context-vs-state-4202be6d3e54>

Työ- ja elinkeinoministeriö. 2021. Uudistuva ja osaava Suomi 2021–2027 - ohjelma edistää alueiden elinvoimaa, työllisyyttä ja hyvinvointia. Tiedote 10/2021. Viitattu 2.11.2021. <https://tem.fi/-/uudistuva-ja-osaava-suomi-2021-2027-ohjelma-edistaa-alueiden-elinvoimaa-tyollisyytta-ja-hyvinvointia>

Vero. 2020. Verokorttihakemus ja/tai ennakkoverohakemus (lomake 5010), täyttöohje. Täyttöohje. Luettu 14.11.2021. <https://vero.fi/tietoa-verohallinnosta/yhteystiedot-ja-asiointi/lomakkeet/tayttoohjeet/verokorttihakemus-jatai-ennakkoverohakemus-lomake-5010-t%C3%A4ytt%C3%B6hje/>

W3C. n.d. Etusivu. Viitattu 7.11.2021. <https://www.w3.org/>

W3C. 2018. Web Content Accessibility Guidelines (WCAG) 2.1. Ohjeistus. Viitattu 7.11.2021. <https://www.w3.org/TR/WCAG21/>