



Uunin reseptieditorin kehitys hyödyntäen Vue.js -kirjastoa

Petrus Rantanen

OPINNÄYTETYÖ
Toukokuu 2019

Tieto- ja viestintätekniikka
Ohjelmistotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintätekniikka
Ohjelmistotekniikka

RANTANEN PETRUS:

Uunin reseptieditorin kehitys hyödyntäen Vue.js -kirjastoa

Opinnäytetyö 16 sivua
Toukokuu 2019

Opinnäytetyön tarkoituksena oli kehittää asiakasyritykselle uusi reseptiikkaeditori, joka korvaa heidän vanhan, hankalakäyttöisen reseptieditorin. Reseptieditorin avulla asiakasyritys pystyy ohjelmoimaan omakehitteisiin uuneihin reseptejä helposti verkon yli.

Lisäksi opinnäytetyössä tutustutaan Vue.js -kirjastoon, joka oli tämän opinnäytetyön pohjana käytetty JavaScript -pohjainen käyttöliittymäkomponenttien kehitykseen tarkoitettu ohjelmistokehys. Opinnäytetyössä käydään läpi projektin aloituksesta editorin kehitykseen.

Asiasanat: vuejs, resepti, editori

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Information and Communication Technology
Software engineering

RANTANEN PETRUS:
Oven recipe editor development using Vue.js library

Bachelor's thesis 16 pages
May 2019

This thesis's purpose was to create to new recipe editor for the customer company which would replace their old, hard-to-use recipe editor. With recipe editor customer company can program recipes to their self-made ovens over the network.

Also, in the thesis get also know about Vue.js library which was the base of this thesis developed JavaScript based user interface component framework. In the thesis proceed through from the project start to editor development

Key words: vuejs, recipe, editor

SISÄLLYS

1	JOHDANTO	5
2	PROJEKTIN TOIMEKSIANTO	6
3	VUE.JS	7
3.1	Vue.js:n rakenne	7
3.2	Vue.js:n debuggaus	8
4	PROJEKTIN TOTEUTUS	9
4.1	Projektin työkalut.....	9
4.2	Projektin aloitus.....	9
4.3	Tietokannan suunnittelu ja muutoksien teko.....	11
4.4	Kielituen rakentaminen	12
4.5	Uunihallintaohjelman rajapintojen suunnittelu ja kehitys.....	12
4.6	Reseptieditorin kehitys.....	13
5	PROJEKTIN YHTEENVETO	15
	LÄHTEET	16

1 JOHDANTO

Opinnäytetyön projektina oli kehittää asiakasyritykselle uusi reseptieditori heidän uudelle verkkopohjaiselle uuninhallintaohjelmistoon, joka tulisi korvamaan vanhan uunin omassa ohjelmistossa toimivaan hankalakäyttöiseen reseptieditorin tilalle. Koska vanha reseptieditori on ollut hankalakäyttöinen, oli asiakasyritys ajatellut uudistaa reseptieditoria, jonka sisältäisi muun muassa tuen eri kieliversioille.

Tässä opinnäytetyössä käydään läpi projektin taustaa ja miten asiakasyrityksen toivomaa reseptieditoria kehitettiin. Lisäksi tässä opinnäytetyössä tutustutaan hiukan uuteen JavaScript -ohjelmointikieleen pohjautuvaa ohjelmistokehyskirjastoa nimeltä kuin Vue.js. Tutustutaan mikä on Vue.js, millaisiin projekteihin sitä voi käyttää sekä millainen on Vue.js -tiedoston rakenne.

2 PROJEKTIN TOIMEKSIANTO

Projektin toimeksiantona oli kehittää asiakasyritykselle uusi, helppokäyttöisempi verkkoselainpohjainen reseptieditori korvaamaan edeltävän uunin omassa ohjelmistossa olevan hankalakäyttöisen reseptieditorin. Edellinen reseptieditori, joka on asiakasyrityksen omakehitteinen, on ollut hankala ja vaikeakäyttöinen, jonka myötä sitä käyttäneiden henkilöiden on ollut vaikeaa käyttää edellistä reseptieditoria.

Projektin tuotoksena tuleva reseptieditori tulee käyttöön asiakasyrityksen uusitun verkkopohjaisen uunien hallintajärjestelmään. Uutta hallintajärjestelmää kehitettiin helppokäyttöiseksi ja lisäksi responsiiviseksi, jonka myötä järjestelmän käyttäjä mukautuu käyttävien henkilöiden päätelaitteille sopiviksi.

Projektin vaatimuksina on kehittää reseptieditori, joka korvaisi vanhan uunissa olevan reseptieditorin. Lisäksi uudelta reseptieditorilta vaadittiin tukea monikielisyttä sekä resepteillä olevien vaiheiden siirtämistä käyttäen joko tietokoneen hiirtä tai älypuhelimissa sormenvedolla.

3 VUE.JS

Vue.js on JavaScript -ohjelmointikielellä toimiva uudenlainen ohjelmistokehys. Se on julkaissut entinen Googlen työntekijä Evan You helmikuussa 2014 ja on avoimeen lähdekoodiin perustuva JavaScript -ohjelmistokehys. Hän kertoi haastattelussa, että hän kehitti Vue.js:n, koska halusi Googlen Angularia kevyemmän ohjelmistokehityksen ilman ylimääräisiä Angularissa olevia konsepteja. Vaikka Vue.js on aika uusi JavaScript -maailmassa, niin se on myös yksi kuumimmista nimistä JavaScript -maailmassa.

Se on kehitetty erityisesti erilaisiin verkkosovelluksiin kehitettävien JavaScript -pohjaisten käyttöliittymäkomponenttien kehitykseen, jonka avulla sitä voidaan hyödyntää verkkosovelluksessa erilaisissa kohteissa vaivattomasti. Lisäksi Vue.js:lla voi myös kehittää Single Page Application -tyyppisiä sovelluksia käyttämällä lisäkirjastoa Vue.js:ssä.

Vue.js:n yksi hyvistä ominaisuuksista on sen tuki kahdensuuntaiselle datansidonnalle. Kaksisuuntaisen datansidonnin avulla käyttöliittymäkomponentin ja verkkosovelluksen DOM-elementtien välinen data siirtyy vaivattomasti.

3.1 Vue.js:n rakenne

VueJS -kirjastossa tiedostojen rakenne muodostuu kolmesta osasta kuten kuvassa 1. Tiedostojen osiot ovat template eli pohja, johon kehitetään käyttöliittymäkomponentin HTML-pohjainen rakenne. Toisena osiona on script eli käyttöliittymäkomponentin logiikka. Viimeisenä on style eli tyylimäärittely, johon voidaan määritellä käyttöliittymäkomponenttiin liittyviä ulkonäkömäärittelyjä. Monesti style -osiota ei tarvita, jos verkkosovelluksen omassa CSS-tiedostossa on jo olemassa tarvittavat määrittelyt.

```
1  <template>
2    <p>{{ message }}</p>
3  </template>
4
5  <script>
6    export default {
7      data () {
8        return {
9          message: 'Hello World!'
10        }
11      }
12    }
13  </script>
14
15  <style scoped>
16    p {
17      font-size: 2em;
18      text-align: center;
19    }
20  </style>
```

KUVA 1. Hello world esimerkki Vue.js:nä

3.2 Vue.js:n debuggaus

Vue.js:ssä ei ole sisäisesti rakennettu debuggaus -työkalua, mutta Vue.js:n kehittäjätiimiltä ovat kehittäneet selaimessa toimivan debuggaus -työkalun. Vue.js DevTool -niminen ilmainen lisäosa löytyy Google Chrome ja Mozilla Firefox -selaimille.

Vue.js DevTool -työkalun avulla pystytään tutkimaan, miten tietoja käsitellään käyttöliitymäkomponenttilla.

4 PROJEKTIN TOTEUTUS

4.1 Projektin työkalut

Projektin toteutuksessa ohjelmoinnissa käytettiin avoimeen lähdekoodiin perustuvaa ilmaista Microsoftin Visual Studio Code -ohjelmaa, joka tukee erilaisia käyttöjärjestelmäympäristöjä kuten Windows, macOS ja erilaiset Linux distributiot.

Mahdollisissa tietokannan rakenteellisissa muutoksissa hyödynnettiin Oraclen MySQL Workbench -ohjelmistoa, jonka avulla tietokannasta saadaan luotua EEP-kaavioita. EEP-kaavion avulla nähdään, miten uunien sisällä toiminut tietokanta oli rakennettu.

Projektiin liittyvät ohjelmointitapahtumat kirjattiin Trello -verkkosovellukseen, johon projektin toteutuksen aikana tehtävät ohjelmoinnit voitiin aikatauluttaa. Lisäksi Trello antoi asiakasyritykselle kuvan ohjelmoinnin tilanteesta.

Lopuksi projektin aikana ohjelmakoodi tuotokset ja/tai muutokset tallennettiin BitBucket -versionhallintaohjelmistoon, joka on Australialaisen Atlassianin tuottama Git -versionhallintaan perustuva verkkosovellus. BitBucketin avulla pystyttiin kohdistamaan versiohallinnan muutokset Trellossa oleviin tapahtumiin.

4.2 Projektin aloitus

Projektin aloitus aloitettiin käymällä palaverissa läpi vanhan reseptiikkaeditorin ongelmakohtia sekä millaisia vaatimuksia haluttiin uuteen reseptiikkaeditoriin. Uuden reseptiikkaeditorin vaatimuksiin haluttiin uusi, helppokäyttöisempi käyttöliittymä uunien reseptiikan kehittämiseen tai muokkaamiseen.

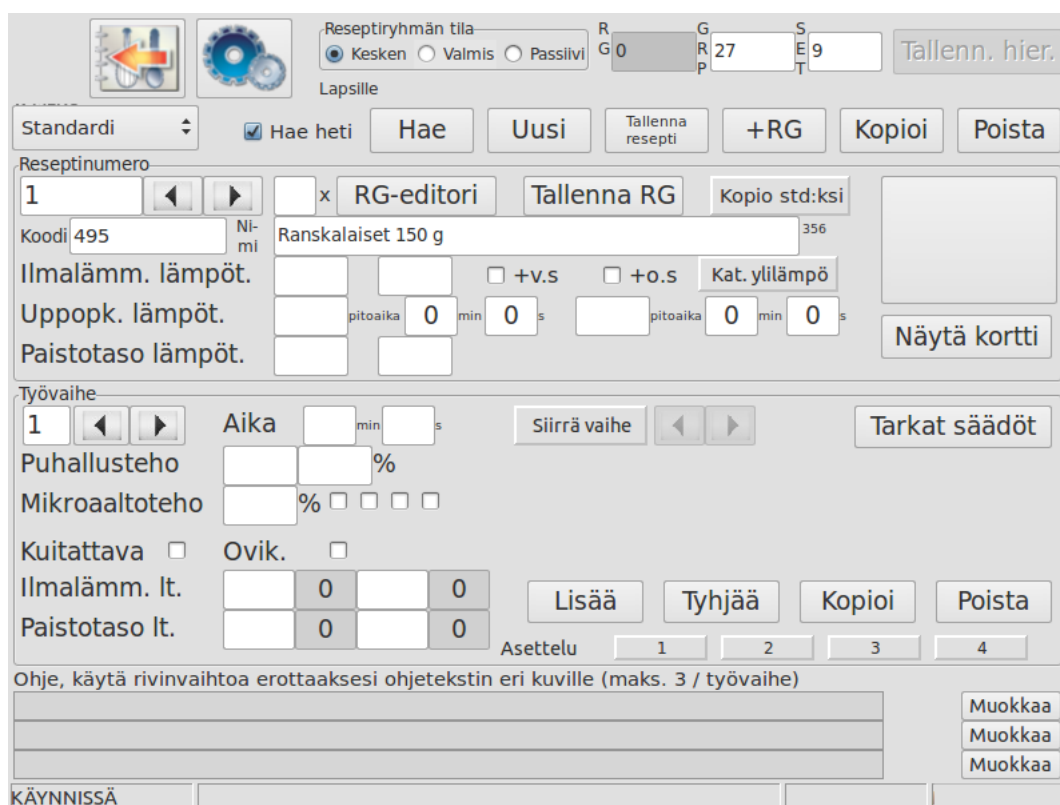
Palaverin aikana käytiin läpi haluttaanko uusi reseptieditorin toimivan uuneissa oleva asiakasyrityksen uuniohjelmistossa yhteydessä vai halutaanko sitä oma erillinen ohjelmisto. Koska asiakasyrityksellä oli uunien ylläpidosta hoitava

verkko-ohjelma, oli projektin pohjalta luontevampaa kehittää uusi reseptiikkaeditori heidän uuteen uunien hallintaohjelmistoon mukaan. Yksi ongelmallisena on ollut asiakasyrityksellä se, että uunien käyttäjien eli asiakkaiden soittaessa reseptiongelmaista oli asiakasyrityksen selvitettävä vanhasta hallintajärjestelmästä reseptiin kohdistuva tunniste, jonka myötä korjattava resepti haettiin uunista. Tähän haluttiin ratkaisuna, että resepteistä vastaava henkilön on helppo siirtyä uunin reseptiajoista suoraan reseptiin uudessa uunin hallintajärjestelmässä.

Uusi uunien hallintajärjestelmä haluttiin tuoda myös resepteistä vastaavan henkilön käyttöön, koska ennen vanhaa järjestelmää käytti vain uunin huolloista vastaava henkilö eli tällöin reseptivastaava joutui selvittämään huoltovastavalta reseptin tunniste.

Lisäksi uuteen reseptieditorin haluttiin tukevan monikielisyyttä, koska asiakasyrityksellä oli tavoitteena tulevaisuudessa laajentaa uunien myyntiä suomen ulkopuolelle. Alussa vaadittiin reseptieditori tukisi kolmea kieltä suomi, ruotsi ja englantia. Mutta reseptieditoria toivottiin tulevaisuudessa helposti laajennettavissa tukemaan uusia kieliä pienellä vaivalla.

Palaverin lopuksi tutustuttiin vanhaan reseptiikkaeditoriin, joka on ollut vaikea-käyttöinen ja lisäksi uusien henkilöiden on ollut vaikeahkoa tottua vanhaa reseptiikkaeditoriin. Reseptiikkaeditorin käyttöliittymä oli erittäin sekava ja erityisesti pienessä 10-tuumaisessa 800x600 resoluutioisessa näytössä katsottuna.



KUVA 2. Vanha reseptiikkaeditori

4.3 Tietokannan suunnittelu ja muutoksien teko

Asiakasyrityksen tietokannat olivat kehitetty monitauluiseksi, jonka myötä resepteihin liittyviä tietoja oli eri tauluissa. Päätauluna käytettiin recipegroup (reseptiryhmä) -taulua, jonka antaa resepteille niiden nimikkeen ja mahdollisen koodin.

Recipe (resepti) -taulussa oli taas reseptiryhmän alle kuuluvat resepti, joiden perustarkoituksena on kertoa uunille aloitukseen liittyvät tiedot, kuten ilman, paistotason tai rasvakeittimen lämpötilat. Resepteillä on myös kertoimille omat rivitiedot, joten yhden reseptiryhmän alla voi useita reseptejä.

Lopuksi on vielä phase (työvaihe) -taulu ja tämän taulun tiedoista löytyy uunin ohjaukseen liittyvät määrittelyt ja uunikäyttäjän toimintaohjeet. Tässä taulussa voi olla paljon rivejä yhtä reseptiä varten, koska reseptin suorituksen aikana voi olla monta erilaista ohjausta uunia varten.

4.4 Kielituen rakentaminen

Asiakasyrityksen sen hetkinen tietokanta tuki vain yhtä kieltä, joka alkoi muodostumaan ongelmalliseksi. Koska asiakasyrityksellä oli mielessä laajentaa uunien tarjontaa ulkomaille, kotimaan lisäksi. Parin asiakkaan vuoksi oltiin hätäratkaisuna luotu samasta reseptistä omia versioita, joissa kieli oli joko ruotsi tai englanti.

Uuniohjelmisto jo alkuvaiheessa toimi kolmella kielellä suomi, ruotsi ja englanti ja tämän myötä lähdettiin selvittämään, olisiko ratkaisua, jonka avulla saataisiin ratkaisu, jossa olisi yksi resepti monilla eri kielivalinnoilla.

Uuniohjelmiston kehitystiimin kanssa mietittyään ratkaisuksi saatiin rakentamalla tietokantaan erilaisia tekstiversioita sisältävä taulu, josta tekstiversiota tunniste vietiin kielten liitântätauluun. Kielten liitântätaulussa on erilaisia kombinaatioita, joiden avulla esimerkiksi työvaiheessa oleva tekstitietue sai numerotunnisteen ja tämän avulla kielten liitântätaulussa oli numerotunnisteelle eri kielille tekstitietueeseen kohdistuva numero. Tämän avulla reseptien nimet, työvaiheessa olevat tekstiohjeet saatiin monelle kielelle ilman, että reseptistä jouduttaisiin tekemään monta versiota jokaiselle kielelle.

Lopuksi uuniohjelmistoon tehdyt muutokset olivat erittäin pienet, koska kielitautut pyrittiin kehittämään uuniohjelmistolle sopivaksi. Ainoana haittapuolena on tässä mallissa se, että samasta ohjeesta voi olla useampi versio. Eli versioita voi olla esimerkiksi pisteen tai pilkun puuttuminen lauseesta, mahdolliset ylimääräiset välilyönnit lauseissa.

4.5 Uunihallintaohjelman rajapintojen suunnittelu ja kehitys

Asiakasyrityksen uunihallintajärjestelmässä on valmiina rakennettuna erilaisia rajapintoja, jonka avulla uunit keskustelevalat uunihallintajärjestelmän kanssa. Koska projektin tavoitteena on korvata uunin päässä tapahtuva editointi uunihallintajärjestelmässä, jouduttiin reseptieditointiin liittyvät rajapinnat poistamaan käytöstä.

Samalla myös kehitettiin rajapinnat uudelle verkkopohjaiselle reseptieditorille. Rajapinta oli lähes sama kuin vanha rajapinta, mutta rajapinnalta poistettiin uunien autentikointi. Uunien autentikoinnin poistamisella uunin päästä tulevat reseptieditoinnit hylätään automaattisesti. Uuteen rajapintaan lisättiin tuki monikielisyttä varten, luomalla uusia kenttiä sanomaan.

4.6 Reseptieditorin kehitys

Reseptieditorin kehittäminen alkoi tutkimalla, miten reseptieditori tulisi rakentaa. Tutkintavaiheessa ajateltiin ensimmäisenä rakentaa staattinen sivu, jossa tiedot olisivat kaikki valmiina käsittelyä varten. Koska reseptieditorissa haluttiin toimintona, jossa vaiheiden järjesteleminen tapahtuisi helposti, oli tällöin staattinen sivu huono ratkaisu.

Tutkittaessa huomattiin, että uunihallintaohjelmisto ohjelmistokehys, joka oli PHP-ohjelmointikielipohjainen ja se oli rakennettu käyttämällä Taylor Otwellin kehittämää Laravel -ohjelmistokehystä. Havaittiin, että Laravelissa oli tehty tuki Vue.js -kirjastolle. Tämän myötä huomattiin, että Vue.js -kirjaston avulla tehty käyttöliittymäkomponentti olisi ratkaisu reseptieditorille.

Internetistä löytyi Vue.js -kirjastolle myös oma kirjasto VueDraggable, jonka avulla tietoa voidaan järjestellä helposti hiiren vedolla tai älypuhelimessa sormen vedolla. Tämän avulla pysyttiin määrittelemään reseptieditorille toiminto, jonka avulla työvaiheiden järjesteleminen kävi helposti ja samalla myös käyttöliittymän komponentin taustalla olevassa datassa tietue tietueryhmässä siirtyi siirron mukana.

Käyttöliittymäkomponentilla tehdyt muutokset tallennettiin käyttämällä API-rajapintoja kuin normaalia http-pyyntöä, koska havaittiin ongelmana virheiden käsittely. Kun tallennuksessa tapahtuvan tietojen tarkastaminen tai mahdollisen rajapinnan ollessa alhaalla, mahdolliset muutokset eivät kadonneet. Tämän myötä käyttöliittymäkomponentti kertoo esimerkiksi muuttamalla rajapinnalta tulevan validointivirheen vuoksi värjäämällä sitä virheellisen tiedon sisältävän syöttökentän, jonka myötä käyttäjä pystyi silmäyksellä näkemään virheellisen syötön.

Reseptieditorissa erilaisten kielten tukeminen tuotiin työvaiheissa antamalla työvaihekomponenttiin jokaiselle kielelle oma syöttökenttä, eli suomen, ruotsin ja englannin kielelle olisi oma kenttä. Mutta reseptin kehitys- tai muuttamisvaiheessa tällainen ratkaisu helpotti käyttäjää, koska eri kielten ohjeistuksien syöttäminen kävisi näppärästi samalla hetkellä.

Perustiedot

Reseptin koodi

107

Koodi / Nimi

9 - [mtest2] - mtest2

Luotu

2015-05-28 00:18:23

Päivitetty

Tila:

☒ Kesken
☐ Valmis
☐ Passiivi

Kerroin

1

Kopioi

Näytä 107

Tallenna

Poista

Takaisin

Reseptin ominaisuudet

Lämpötilat

Vasen

Oikea

Ilma (keski)

220 °C

220 °C

Ilma (sivu)

220 °C

220 °C

☐ Katalysaattorin yllälämpö

Paistotaso

0 °C

0 °C

Rasvankeittimet

Vasen

Oikea

Lämpötila

0 °C

0 °C

Pitoajat

Vasen

0 min

0 sec

Oikea

0 min

0 sec

Vaiheet

Kesto: 2:00

Vaihe 1 [0]

Poista

Vaihe 2 [1]

Poista

KUVA 3. Asiakasyrityksen uusi reseptieditori

5 PROJEKTIN YHTEENVETO

Projektin yhteenvedona toteutettiin asiakasyritykselle uusi, verkkoselaimella toimiva reseptieditori. Testailujen jälkeen asiakasyritys on ottanut reseptieditorinsa käyttöön tuotantoversiossaan.

Asiakasyrityksessä olleet henkilöt, jotka ovat käyttäneet vanhaa reseptieditoria, olivat tyytyväisiä uuteen reseptieditoriin. Kommentteina uudesta reseptieditorista oli, että se oli loistava parannus vanhaan reseptieditoriin. Koska reseptieditori on verkkopohjainen, on sen käyttö mahdollistanut uunien reseptin muokkauksen missä tahansa, milloin tahansa, vaikka tien päällä. Vanha reseptieditori toimi ainoastaan uuneissa, jolloin henkilön piti olla uunin läheisyydessä reseptin muokkauksia varten.

Kokonaisuudessa projektin myötä tuotettu reseptieditori oli onnistunut projekti asiakasyritykselle. Projektin tavoitteet saatiin toteutettua asiakasyritykselle toivotussa ajassa käyttöön.

LÄHTEET

Cromwell V. 2017. Between the Wires: An interview with Vue.js creator Evan You. Luettu 17.08.2018. <https://medium.freecodecamp.org/between-the-wires-an-interview-with-vue-js-creator-evan-you-e383cbf57cc4>

Vue.js – Reaktiivinen – Nopea – Siisti. Luettu 17.08.2018 <https://jan-rikv.fi/blogi/artikkeli/vuejs-reaktiivinen-nopea-siisti>

Visual Studio Code. Luettu 17.08.2018. <https://code.visualstudio.com>

Atlassian BitBucket. Luettu 17.08.2018. <https://fi.atlassian.com/software/bit-bucket>

Atlassian Trello. Luettu 9.11.2018. <https://trello.com/fi>

VueDraggable. Luettu 9.11.2018. <https://github.com/SortableJS/Vue.Draggable>