

# SAVONIA

ammattikorkeakoulu

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO  
TEKNIIKAN JA LIIKENTEEN ALA

## EVIKAVIHKO

eVikavihkon jatkokehitys

TEKIJÄ/T Tommi Puurunen

Koulutusala Tekniikan ja liikenteen ala	
Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä(t) Tommi Puurunen	
Työn nimi eVikavihko jatkokehitys	
Päiväys 12.04.2023	Sivumäärä/Liitteet 20/0
Toimeksiantaja/Yhteistyökumppani(t) Teija Tilli	
Tiivistelmä <p>Opinnäytetyön aiheena oli jatkokehittää Teija Tillin opinnäytetyönä kehittämää eVikavihkoa. Tavoitteena oli käyttää modernimpia teknologioita ja kehittää uusia ominaisuuksia. Työn tarkoituksena oli myös laajentaa käyttöliittymää.</p> <p>Sovelluksen ideana on perustaa erilaisia alueita. Alueille voidaan lisätä resursseja ja niille voidaan kirjata ilmoituksia, ohjeita, huoltohistoriaa, ongelmatilanteita ja lisätä liitteitä. Näitä tietoja voidaan hakea resurssien alta, jolloin tiedot voidaan löytää helpommin.</p> <p>Opinnäytetyössä tutkittiin myös teknistä velkaa, miten sitä kertyy ja miten korjataan. Aiemmin kehitetyssä eVikavihkossa teknistä velkaa oli hieman, koska ei ollut käytetty moderneja teknologioita, jotka nopeuttavat ja helpottavat kehittämistyötä.</p> <p>Sovelluksen käyttöliittymä toteutettiin teknologioilla: Vue, TypeScript, Bootstrap, RxJs ja FontAwesome. Palvelintekniikoihin kuului Node.js, Express ja TypeScript. Tietokannaksi valittiin relaatiotietokanta MariaDB. Sovellus asennettiin Docker-ohjelmalla kontiksi. Sovellus voidaan asentaa myös suoraan palvelimelle.</p> <p>Sovellus on tällä hetkellä testiympäristössä, missä työn tilaaja ja referenssiyritys voi vapaasti testata ja kokeilla. Tämän opinnäytetyön jälkeen sovelluksen kehitystä jatketaan uusilla tekniikoilla ja ominaisuuksilla, jotka eivät sisälly tähän työhön.</p>	
Avainsanat Jatkokehitys, sovellus, tekninen velka, ohje, muistiinpanot, ongelmatilanne	

Field of Study Technology, Communication and Transport	
Degree Programme Degree Programme in Information Technology	
Author(s) Tommi Puurunen	
Title of Thesis Further Development of eVikavihko	
Date 12 April 2023	Pages/Appendices 20/0
Client Organisation /Partners Teija Tilli	
<b>Abstract</b> <p>The topic of this thesis was to develop further the eVikavihko application which Teija Tilli developed in her thesis. The goal was to use the most modern technologies and develop new features. The purpose of the work was also to expand the user interface.</p> <p>The idea of the application is to establish different areas. Resources can be added to areas and notifications, instructions, maintenance history, problem situations and attachments can be added to them. This information can be searched using resources, which makes the information easier to find.</p> <p>The thesis also examined technical debt, how it accumulates and how to fix it. In the previously developed eVikavihko, there was a bit of technical debt, because modern technologies that speed up and facilitate development work had not been used.</p> <p>The user interface of the application was implemented with technologies: Vue, TypeScript, Bootstrap, RxJs and FontAwesome. Server technologies included Node.js, Express and TypeScript. The relational database MariaDB was chosen as the database.</p> <p>The application was installed as a container using the Docker program, but it can also be installed directly on the server.</p> <p>The application is currently in a test environment, where the client and the reference company can freely test and experiment with it. After this thesis, the development of the application will continue with new technologies and features that are not included in this work.</p>	
<b>Keywords</b> Further development, application, technical debt, notes, problem situation	

## SISÄLTÖ

1	JOHDANTO .....	6
2	TEKNINEN VELKA.....	7
2.1	Mitä on tekninen velka .....	7
2.2	Miten estää teknistä velkaa.....	7
2.3	Miten teknistä velkaa estetään eVikavihkossa .....	7
3	TEKNINEN TOTEUTUS .....	9
3.1	Työkalut.....	9
3.2	Palvelinohjelma.....	9
3.3	Tietokanta .....	10
3.4	Docker .....	10
3.5	Käyttöliittymä .....	11
3.5.1	Kirjastot.....	11
3.5.2	Etusivu.....	12
3.5.3	Alueet-sivu .....	12
3.5.4	Valitun alueen sivu .....	13
3.5.5	Valitun alueen Ilmoitustaulu-välilehti.....	13
3.5.6	Valitun alueen Liitteet-välilehti .....	14
3.5.7	Valitun alueen Resurssit-välilehti.....	14
3.5.8	Valitun resurssin sivu .....	15
3.5.9	Valitun resurssin Yleinen-välilehti .....	15
3.5.10	Valitun resurssin Kirjaukset-välilehti .....	16
3.5.11	Valitun resurssin Huoltohistoria-välilehti .....	16
4	EVIKAVIHKON JATKOKEHITYS.....	17
4.1	Palvelinpuolen kielen vaihto.....	17
4.2	Pilvipalvelut .....	17
4.3	Tietokanta .....	17
4.4	Agile ja CI .....	17
4.5	Tuotteen suunta .....	18
5	YHTEENVETO.....	19
6	LÄHTEET .....	20

## KUVALUETTELO

Kuva 1: Docker ympäristö kaavio .....	11
Kuva 2: eVikavihko käyttöliittymä - Etusivu.....	12
Kuva 3: eVikavihko käyttöliittymä - Alueet-sivu. ....	13
Kuva 4: eVikavihko käyttöliittymä - Valitun alueen sivu.....	13
Kuva 5: eVikavihko käyttöliittymä - Ilmoitustaulu. ....	14
Kuva 6: eVikavihko käyttöliittymä - Liitteet. ....	14
Kuva 7: eVikavihko käyttöliittymä - Resurssit. ....	14
Kuva 8: eVikavihko käyttöliittymä - Resurssin ilmoitustaulu.....	15
Kuva 9: eVikavihko käyttöliittymä - Resurssin Yleinen-välilehti .....	15
Kuva 10: eVikavihko käyttöliittymä - Kirjaukset.....	16
Kuva 11: eVikavihko - Huoltohistoria-välilehti .....	16

## 1 JOHDANTO

Tämän opinnäytetyön aiheena on jatkokehittää eVikavihkoa Teija Tillin aiemmin tekemän opinnäytetyön pohjalta. Referenssi yrityksenä toimii Hydroline Oy. Tavoitteena on modernisoida teknologioita, parantaa joitakin prosesseja ja tehdä suunnitelma jatkoa varten. eVikavihko on sovellus, johon voidaan tallentaa tietyille alueille ja resursseille kirjauksia ja ilmoituksia. Sovellusta voi käyttää useampi käyttäjä samaan aikaan.

Tavoitteena on uudistaa käyttöliittymää ottamalla käyttöön modernit teknologiat mistä on myöhemmin helppo jatkaa edelleen kehitystyötä. Toisena tavoitteena on tutkia ja toteuttaa selvempi arkkitehtuuri palvelinpuolelle. Kolmantena tavoitteena on tutkia niin sanottua teknisestä velkaa, mitä se on ja kuinka sitä ehkäistään.

## 2 TEKNINEN VELKA

### 2.1 Mitä on tekninen velka

Tekninen velka kuvaa ohjelmistoprojekteihin syntyvää vanhaa koodia, joka pitää myöhemmin muokata toimimaan yksinkertaisemmin tai nopeammin. Ohjelmistoprojekteissa teknistä velkaa syntyy esimerkiksi kiireessä tai vanhojen teknologioitten käytöllä. Kiireessä syntyvällä teknisellä velalla tarkoitetaan nopeiden ratkaisujen toteuttamisella, jotka toimivat tietyssä ominaisuudessa, mutta pitää myöhemmin muuttaa soveltumaan muuhun tarkoitukseen. (Niemistö, 2016)

Esimerkkinä voisi olla, että valitaan joku teknologia, joka on yksityisen henkilön ylläpitämä ja hän päättää lopettaa sen kehittämisen. Jos sovelluksella on jo paljon pohjaa kyseiselle teknologialle, voi olla haastavaa päästä tästä eroon. Teknologioiden päivittäminen kuitenkin tuo pääosin uusia ominaisuuksia ja turvallisuuteen liittyviä korjauksia. Näiden asioiden takia on hyvä valita teknologia, joilla on jo esimerkiksi jonkun yrityksen tuki ja aktiivinen kehittäjäyhteisö.

Yhtenä ongelmana on myös esimerkiksi, jos sovellusta ei ole suunniteltu laajaan käyttöön. Myöhemmin, kun sovellustuote saa isoja asiakkaita tai muuta laajemmän tietomäärän käyttöä tulee vastaan teknisiä ongelmia, joita ei olla ajateltu sovellusta suunnitellessa. Mikäli koko ajan halutaan saada uusia ominaisuuksia, niin vanhojen ominaisuuksien ylläpitäminen ja kehittäminen voi jäädä taka-alalle. Mikäli myöhemmin kerääntyy tämänkaltaisia ongelmia niistä voi tulla haasteellisia ratkaista. Sovelluksessa voi ilmetä suorituskyvyn heikkenemistä, virheellisiä tuloksia tai muuta toimintakyvyn vaikuttavaa tekijää.

### 2.2 Miten estää teknistä velkaa

Teknistä velkaa voidaan estää suunnittelemalla sovelluksen arkkitehtuuri hyvin sekä modulaariseksi ja valita oikeat teknologiat tuotteelle. Tällöin voidaan myöhemmin helposti parantaa tai korjata ominaisuutta. Teknologioita ovat esimerkiksi Angular, Vue ja React.

Jo olemassa olevan teknisen velkaan voi puuttua laittamalla resursseja uuden teknologian käyttöönottoon tai uudelleen kirjoittamalla vanhat ja ongelmalliset ohjelman osat. Tätä varten on yksikkötestaus. Yksikkötestauksella varmistetaan, että sovelluksen osat toimivat samalla tavalla, vaikka toteutusta muutettaisiin. Hyvin harvoin päästään tilanteeseen missä kaikki on suunniteltu ja otettu huomioon sekä teknologiat on oikein valittu, vaikka käytettäisiin näitä menetelmiä.

### 2.3 Miten teknistä velkaa estetään eVikavihkossa

eVikavihkossa estetään teknistä velkaa esimerkiksi modernien teknologioiden valinnalla ja käyttämällä sovellusarkkitehtuuria, joka mahdollistaa yksittäisten komponenttien vaihtamisen toiseen. Käyttöliittymäpuolen kirjastoksi on valittu Vue, koska se on erittäin joustava. Vuen lisäksi hyvin soveltuvia vaihtoehtoja olisi ollut muun muassa Angular tai React.

Vue koostuu komponenteista. Niin sanotut peruskomponentit on suunniteltu käytettäväksi laajemmin. Peruskomponentteja hyödyntämällä voidaan johtaa tiettyyn asiaan erikoistuneita komponentteja. Kun peruskomponenttia muokataan tulevat muutokset myös tästä johdettuihin erikoiskomponentteihin.



### 3 TEKNINEN TOTEUTUS

eVikavihko on selainpohjainen sovellus. Käyttöliittymä on toteutettu käyttäen Vueta. Palvelinpuolella on käytetty TypeScriptiä. Palvelinpuoli lähettää ensimmäisellä sivunlatauksella käyttöliittymän tarvittavat tiedostot ja tämän jälkeen käyttöliittymä tekee kutsuja palvelimen rajapintaan.

#### 3.1 Työkalut

Opinnäytetyön toteuttamiseen on käytetty Visual Studio Codea. Visual Studio Code (lyhennettynä VS Code) on kevyt, kuitenkin erittäin tehokas lähdekoodieditori, joka on suunniteltu käytettäväksi työpöytäympäristössä. Se on käytettävissä Windows-, macOS- ja Linux-käyttöjärjestelmille. VS Code sisältää sisäänrakennetun tuen JavaScriptille, TypeScriptille ja Node.js:lle. Lisäksi se tarjoaa laajan valikoiman laajennuksia eri ohjelmointikielille, kuten C++, C#, Java, Python, PHP ja Go, sekä suoriutusympäristöille kuten .NET ja Unity. (Microsoft, 2022)

Tietokannan luomiseen ja suunnitteluun on käytetty ilmaista DBeaver Community Editionia. DBeaver on maksuton monialustainen tietokantatyökalu, joka tarjoaa tuen kaikille suosituimmille tietokannoille. Näihin kuuluvat MySQL, PostgreSQL, MariaDB, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Derby ja monet muut. (Solvusoft, n.d.)

#### 3.2 Palvelinohjelma

Palvelinpuolen teknologiana toimii Express ja TypeScript. Expressin lisäksi on otettu käyttöön lisäosia, kuten express-fileupload, mariadb ja express-cors. Express on HTTP-palvelin kirjasto, jolla voi nopeasti saada pystyyn toimivan HTTP-palvelimen. (Fullstack Open, 2023). Expressiin voi asentaa kirjastoja, jotka tuovat lisäominaisuuksia kuten tuen CORS kutsuille ja tiedoston lataamisille.

Palvelinohjelma noudattaa osin REST API käytäntöä. RESTful API, jota kutsutaan myös nimellä REST API, on sovellusohjelmointirajapinta (API), joka toimii REST-arkkitehtuurin periaatteiden mukaisesti. Se tarjoaa mahdollisuuden kommunikoida RESTful-verkkopalveluiden kanssa. REST on akronyyymi, joka tulee sanoista Representational State Transfer. (TR Plane, n.d.)

Mariadb kirjasto on tietokanta-ajuri, jolla otetaan yhteys tietokantaan. Palvelinohjelma ottaa vastaan pyyntöjä käyttöliittymältä ja vastaa niihin. Palvelinpuoli noudattaa arkkitehtuuria, jossa on itse pyynnön vastaanottava komponentti, sitten pyynnön käsittelevä komponentti, seuraavana itse kantakyselyn luoja. Kantakyselyn luova komponentti vastaa takaisinpäin käsittelevälle komponentille ja siitä sitten vastaanottavalle komponentille, joka välittää tämän vastauksen käyttöliittymän puolelle. Vasteaika tässä koko prosessissa on tässä työssä noin 50ms. Tietokantakyselyt on rakennettu mahdollisimman yksinkertaisiksi ja modulaarisiksi, jotta jatkossa voidaan ottaa uusia tietokantakenttiä käyttöön ilman isompaa kehitystyötä.

### 3.3 Tietokanta

Alue on kokoelma resursseja ja ilmoituksia. Resurssit ovat kokoelmia ilmoituksista ja kirjauksista. Käytännössä alueella voi olla monta resurssia ja ilmoitusta. Resurssilla voi olla monta kirjausta ja ilmoitusta, mutta vain yksi alue.

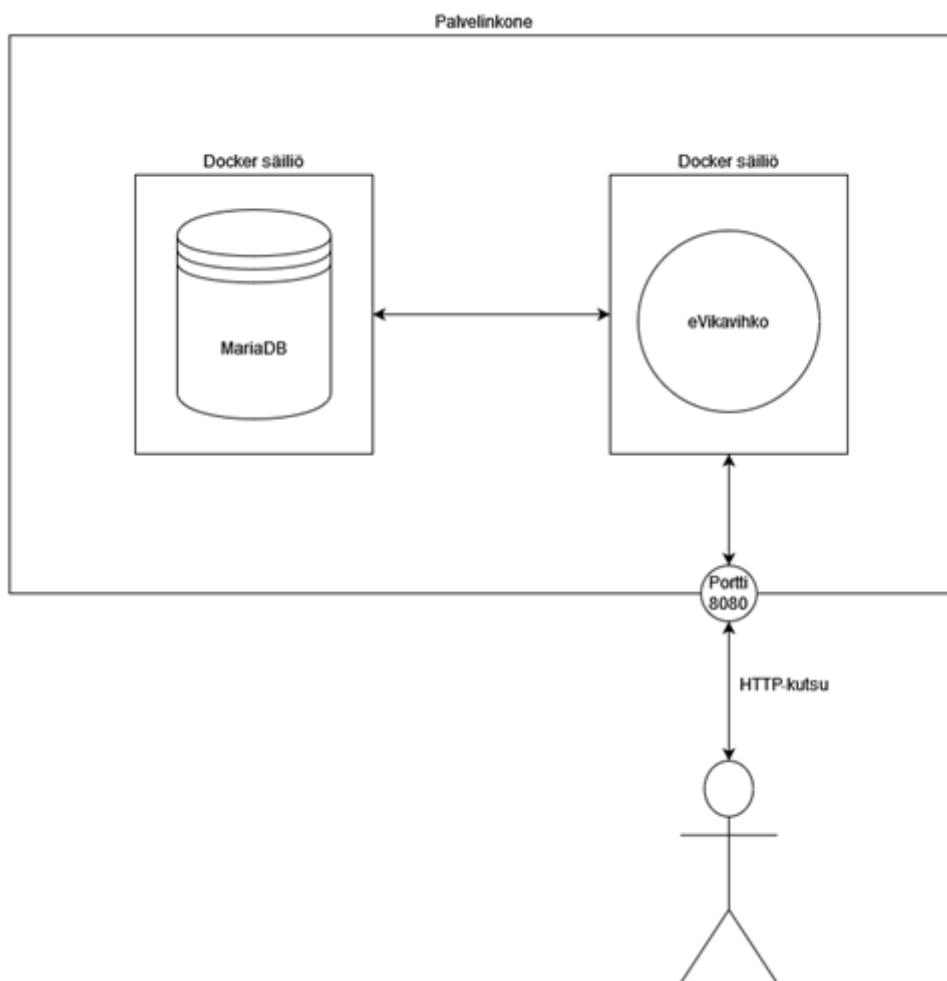
Tietokannaksi tämän opinnäytetyön ajaksi on valittu MariaDB. Rakenne tietokannassa on yksinkertainen. MariaDB on laajalti käytetty tietokantamoottori, joka on osa LAMP-järjestelmää. Tämän kehittäjänä toimii Monty Widenius, joka on tuttu MySQL:n parista. MariaDB on täysin yhteensopiva GPL-lisensoidun MySQL-tietokantamoottorin kanssa. (Linux.fi, 2020)

### 3.4 Docker

Docker on monipuolinen tuoteperhe, joka mahdollistaa sovellusten sijoittamisen kontteihin sekä konttien käynnistämisen ja suorittamisen. Docker tarjoaa tehokkaan ympäristön sovellusten ajamiseen konteissa. (Wallenius, 2022). Docker kontit on rakennettu Linux käyttöjärjestelmän pohjalle. Konttitekniikan avulla luodaan standardisoitu ympäristö, jossa ohjelmistojen tai sovellusten kehittäminen, ylläpito ja ajaminen tapahtuu. Yleisimmät konttien ajoympäristöt ovat Docker ja Kubernetes. (itewiki, n.d.)

Linux on käyttöjärjestelmän ydin, joka sai alkunsa suomalaisen Linus Torvaldsin toimesta. Käyttöjärjestelmän ydin, eli kernel on olennainen osa tietokoneen käyttöjärjestelmää ja mahdollistaa muiden järjestelmän komponenttien toiminnan. Yleisesti termi "Linux" viittaa kokonaiseen käyttöjärjestelmään, jossa Linux toimii ytimenä ja siihen sisältyy myös erilaisia sovellusohjelmia. Linuxin tapauksessa sekä käyttöjärjestelmä että sovellusohjelmistot koostuvat usein GNU-projektin tuottamista työkaluista. (Krimaka, n.d.)

Kontit ovat eräänlaisia säiliöitä, jossa on kaikki mitä sinne asennetut sovellukset tarvitsevat. eVikavihko pyörii yksinkertaisessa kontissa, joka on asennettu Docker sovelluksella. Tietokantaa, sekä sovellusta ajetaan konteissa. Sovellus lisäksi antaa portin palvelinkoneelle, josta palvelinkone voi tarjota eVikavihkon palvelun verkkoon.



Kuva 1: Docker ympäristö kaavio

### 3.5 Käyttöliittymä

Kaikkien selainkäyttöliittymien pohjana on HTML, CSS ja JavaScript. HTML on selaimen ymmärtämä merkintäkieli, jota standardisoi W3C. HTML kieli sisältää paljon erilaisia merkintöjä, joita selain ymmärtää. CSS puolestaan on HTML elementtien tyyliin tarkoitettu merkintä kieli, jolla saadaan itse sivujen rakenne ja tyyli ihmiselle selemmäksi. JavaScript on taas selaimen ymmärtämä ohjelmointikieli, jolla voidaan tehdä toiminnallisuus nettisivulle kuten reaaliaikainen päivitys eräänlaisista ilmoituksista.

#### 3.5.1 Kirjastot

Vue.js on progressiivinen JavaScript-kehys, jota käytetään käyttöliittymien ja yksisivuisten sovellusten (SPA) rakentamiseen. Tämä kehys on kuuluisa nopeatempoisesta oppimiskäyrästä. (Ciksiti, 2021)

TypeScript on ohjelmointikieli, joka perustuu laajalti käytettyyn JavaScript-ohjelmointikielen syntaksiin ja ominaisuuksiin. Se ei kuitenkaan ole täysin erillinen kieli, vaan laajentaa JavaScriptiä tarjoamalla uusia toiminnallisuuksia. Yksi merkittävimmistä lisäyksistä on vahva tyyppijärjestelmä, josta myös kielen nimi "TypeScript" juontuu. (Tarvainen, 2016)

Axios on HTTP pyyntöjä muodostava kirjasto, joka helpottaa huomattavasti HTTP-pyyntöjen lähetyksestä. Tästä on hyötyä myös koodin luettavuuden kannalta.

RxJs, lyhenne sanoista Reactive Extensions for JavaScript, on JavaScript-kirjasto, joka tarjoaa reaktiivisen ohjelmoinnin ratkaisuja Angular-verkkosovellusten kehittämiseen. (Sarlin, 2021)

RxJs kirjastoa voi myös käyttää muissakin kuin Angular sovelluksissa ja se soveltuu hyvin myös Vue pohjaisiin sovelluksiin.

Bootstrap on erittäin suosittu sovelluskehys (framework), joka on suunniteltu responsiivisten verkkosivujen toteuttamiseen. Se tarjoaa kattavan kokoelman CSS-määrittäjiä ja JavaScript-komponentteja, jotka helpottavat responsiivisten sivustojen luomista ja varmistavat käyttäjän selaintoon huomioimisen. (Timonen, n.d.) Font Awesome on kirjasintyyppinä jaettava kokoelma kuvakkeita www. (Viikon Valo, 2014)

Font Awesomen kuvakkeita käytetään käyttöliittymän elementtien selventämiseen.

### 3.5.2 Etusivu



Käyttöliittymä on pyritty helppokäyttöiseksi ja perustoimintoihin pääsee vain yhtä nappia painamalla. Etusivulla on myös pienet ohjetekstit, jotka kertovat mitä eri sivut tekevät.



Kuva 2: eVikavhko käyttöliittymä - Etusivu.

### 3.5.3 Alueet-sivu

Alueet-sivulla voidaan lisätä uusia alueita. Taulukossa näkyy alueen nimi, resurssimäärä, milloin kyseinen alue on luotu ja milloin sitä on muokattu. Alueen pystyy poistamaan roskakorinapista. Jos alueen riviä klikkaa hiirellä, niin päästään alueen omalle sivulle.

eVikavihko Etusivu Alueet Resurssit Roskakori				
Alueet				
+Lisää alue				
Nimi	Resurssimäärä	Luotu	Muokattu	Poista
Putkivarsi	1	12/11/2022	05/12/2022	
OV	1	28/11/2022	30/11/2022	

1

Kuva 3: eVikavihko käyttöliittymä - Alueet-sivu.

### 3.5.4 Valitun alueen sivu

Valitun alueen yleinen-välilehdelle voi kirjoittaa vapaamuotoisen kuvauksen alueesta. Ilmoitustaulu-välilehdelle voi lisätä ilmoituksia, jotka näkyvät alueeseen liitettyillä resursseilla. Liitteet-välilehdellä voi lisätä alueeseen liittyviä liitteitä, kuten kuvia tai dokumentteja. Kuvat-välilehdellä on kooste liitteisiin lisätyistä kuvista. Resurssit-välilehdeltä löytyy taulukko alueen resursseista. Siinä käyttäjä pystyy lisäämään uusia resursseja tai poistamaan niitä.

eVikavihko Etusivu Alueet Resurssit Roskakori				
				<a href="#">Alueet &gt; Putkivarsi</a>
Putkivarsi				
Yleinen	Ilmoitustaulu	Liitteet	Kuvat	Resurssit
<b>Muokkaa</b>				
Luotu: 12/11/2022, 20.11.03				
Muokattu: 05/12/2022, 21.02.45				
<b>Kuvaus / Vapaa sisältö</b>				
Putkivarsi alueella on 12 metallintyöstökonetta, hitsauslaitteita 5 kpl, kitkahitsauslaitteita 1 kpl ja varsisorvi. Esimiehiä Jussi ja Paavo puh.044345434533334.				

Kuva 4: eVikavihko käyttöliittymä - Valitun alueen sivu.

### 3.5.5 Valitun alueen Ilmoitustaulu-välilehti

Valitun alueen Ilmoitustaulu-välilehdellä voidaan luoda ilmoituksia, jotka näkyvät alueen ilmoitustaululla sekä kaikilla alueeseen liitettyjen resurssien ilmoitustauluilla. Ilmoituksen värejä on valittavana valkoinen, vihreä, keltainen ja punainen. Värejä voi hyödyntää esimerkiksi ilmoituksen prioriteettia kuvattessa.

Kuva 5: eVikavikko käyttöliittymä - Ilmoitustaulu.

### 3.5.6 Valitun alueen Liitteet-välilehti

Liitteet-välilehdellä on alueen liitteet, joihin voi kuulua esimerkiksi kuvia, dokumentteja ja videoita. Kuvat välilehdellä näkyvät alueen liitteisiin lisätyt kuvat.

Tiedoston nimi	Tiedosto linkki	Poista
404_image_89a12a7af49c2f544a239b853ed5286a7ae2985.jpg	/uploads/404_image_89a12a7af49c2f544a239b853ed5286a7ae2985.jpg	
favicon_4b584ff655ae58661250f1e4195071fe948dd46e.ico	/uploads/favicon_4b584ff655ae58661250f1e4195071fe948dd46e.ico	
IMG-20230105-WA0000_cf9db348d2fe62242f265c2dc5318deeba02d364.jpeg	/uploads/IMG-20230105-WA0000_cf9db348d2fe62242f265c2dc5318deeba02d364.jpeg	

Kuva 6: eVikavikko käyttöliittymä - Liitteet.

### 3.5.7 Valitun alueen Resurssit-välilehti

Resurssit-välilehdellä näkyy kaikki kyseisen alueen resurssit. Taulukosta näkyy yksilöllinen resurssi-numero, resurssin nimi, milloin resurssia on muokattu, milloin resurssi on luotu, ja poisto nappi. Resurssin riviä klikkaamalla voidaan päästä tämän resurssin sivulle.

Nimi	Resurssinumero	Luotu	Muokattu	Poista
test	77	30/11/2022	30/11/2022	

Kuva 7: eVikavikko käyttöliittymä - Resurssit.

### 3.5.8 Valitun resurssin sivu

Tietulle resurssille pääsee kahdella eri tavalla. Ensimmäinen tapa on Resurssit-sivua klikkaamalla navigaatiopalkista ja siellä olevasta taulukosta klikkaamalla tiettyä resurssia. Toinen tapa on tiettyltä alueelta Resurssit-välilehdeltä resurssia klikkaamalla. Ensimmäisenä tulee näkyviin Ilmoitustaulu-välilehti, johon tulee kaikki alueen ja resurssin omat ilmoitukset.

The screenshot shows the eVikavikko website interface. At the top, there is a navigation bar with links: eVikavikko, Etusivu, Alueet, Resurssit, and Roskakori. The main heading is 'Kitamura 1 (112)' with a breadcrumb trail 'Alueet > Putkivarsi > Kitamura 1'. Below the heading are tabs for 'Ilmoitustaulu', 'Yleinen', 'Liitteet', 'Kuvat', 'Kirjaukset', and 'Huoltohistoria'. The 'Ilmoitustaulu' tab is active, showing 'Alueen ilmoitukset' (Area announcements) and 'Resurssin ilmoitukset' (Resource announcements). Under 'Alueen ilmoitukset', there are two cards: a green one titled 'Moikka kaikki, iloisia uutisia!' and a red one titled 'Kuitaukset kuntoon'. Under 'Resurssin ilmoitukset', there is a green card titled 'Resurssi spesifinen ilmoitus' with a '+Lisää ilmoitus' button above it.

Kuva 8: eVikavikko käyttöliittymä - Resurssin ilmoitustaulu.

### 3.5.9 Valitun resurssin Yleinen-välilehti

Resurssin Yleinen-välilehdellä voidaan muokata resurssin kuvausta/vapaata sisältöä, resurssinumeroa ja nimeä. Resurssi voidaan tässä myös siirtää alueelta toiselle. Liitteet ja kuvat toimivat samalla tavalla kuin alueen kohdalla.

The screenshot shows the eVikavikko website interface for the 'Yleinen' (General) tab of the resource 'test (77)'. The navigation bar is the same as in the previous screenshot. The main heading is 'test (77)' with a breadcrumb trail 'Alueet > Putkivarsi'. Below the heading are tabs for 'Ilmoitustaulu', 'Yleinen', 'Liitteet', 'Kuvat', 'Kirjaukset', and 'Huoltohistoria'. The 'Yleinen' tab is active, showing a 'Muokkaa' (Edit) button, the creation date 'Luotu: 30/11/2022, 20.18.57', and the update date 'Muokattu: 30/11/2022, 20.20.22'. Below this is the section 'Kuvaus / Vapaa sisältö' with the text '777777777'. At the bottom, there is a section 'Siirrä resurssi alueelle' with a dropdown menu showing 'Putkivarsi (Valittu alue)' and a 'Siirrä' (Move) button.

Kuva 9: eVikavikko käyttöliittymä - Resurssin Yleinen-välilehti

### 3.5.10 Valitun resurssin Kirjaukset-välilehti

Kirjaukset-välilehdellä voidaan lisätä taulukkoon erilaisia kirjauksia. Nämä voivat liittyä resurssin peruskäyttöön, kuten käynnistää kyseessä oleva kone tai mitä tehdä erilaisissa virhetilanteissa.

eVikavihko Etusivu Alueet Resurssit Roskakori

test (77) [Alueet > Putkivarsi > test](#)

Ilmoitustaulu Yleinen Liitteet Kuvat Kirjaukset Huoltohistoria

Merkinnät

+Lisää merkintä

Haku

Otsikko	Hälytyskoodi	Sisältö	Muokattu	Poista
No data				

Kuva 10: eVikavihko käyttöliittymä - Kirjaukset.

### 3.5.11 Valitun resurssin Huoltohistoria-välilehti

Huoltohistoria-välilehdelle voi lisätä kyseisen resurssin huoltoon liittyviä kirjauksia ja ilmoituksia. Esimerkiksi ilmoituksilla voidaan viestiä, milloin huolto tulee tai onko huolto tilattu jo virhetilannetta varten. Kirjauksista on tarkoitus ilmetä resurssille tehdyt huollot.

eVikavihko Etusivu Alueet Resurssit Roskakori

test (77) [Alueet > Putkivarsi > test](#)

Ilmoitustaulu Yleinen Liitteet Kuvat Kirjaukset Huoltohistoria

Huoltoilmoitukset

+Lisää ilmoitus

Huolto tulee viikon välein!

Ensimmäinen huolto on viikolla 2 perjantaina!

Muokattu: 29/01/2023, 12.01.42 by internal

Huotokirjaukset

+Lisää merkintä

Haku

Otsikko	Sisältö	Muokattu	Poista
No data			

Kuva 11: eVikavihko - Huoltohistoria-välilehti



## 4 EVIKAVIHKON JATKOKEHITYS

### 4.1 Palvelinpuolen kielen vaihto

Suunnitelmissa jatkoa varten on vaihtaa palvelinpuolen ohjelmointikieli C#-ohjelmointikieleen. Ohjelmointikieli C# vahvuutena verrattuna TypeScriptiin on sen ylläpidettävyys, koska kehitys on helpompaa ja C# on käännetty ohjelmointikieli. Käännetty ohjelmointikieli on tehokkaampi, koska se on jo muunnettu tietokoneelle ymmärrettäväksi kieleksi. TypeScript taas kuuluu tulkittuihin kieliin. Tulkitut kielet muunnetaan tietokoneelle ymmärrettäväksi kieleksi ohjelmaa ajaessa. C#-ohjelmointikielen etuna on sen hyvä dokumentaatio ja kirjastot. C#-ohjelmointikieli sisältää kirjastoja esimerkiksi Azuren pilviympäristöön ja erilaisiin HTTP-palvelinohjelmiin.

### 4.2 Pilvipalvelut

Palveluntarjoajia ovat muun muassa Microsoft Azure Cloud, Google Cloud Platform ja Amazon Web Services. eVikavihko voidaan asentaa mihin tahansa näistä palveluntarjoajista. Kaikki nämä palveluntarjoajat tarjoavat eVikavihkole palveluita, joilla voidaan myös kehittää uusia ominaisuuksia.

### 4.3 Tietokanta

Tietokantavaihtoehtoina relaatiokannat ovat luotettavimpia, mutta MariaDB tietokanta voi vaihtua PostgreSQL tai SQL Server tietokantaan.

PostgreSQL on vanha relaatiomallinen avoimen lähdekoodin tietokantajärjestelmä ja siten myös testattu. PostgreSQL tietokannan toimintaa voi muokata omilla laajennuksilla. PostgreSQL on suosittu ja. (Linux.fi, 2016)

SQL Server on Microsoftin julkaisema relaatiomallinen tietokantajärjestelmä (Microsoft, 2023). Käyttäjät Azuren AD järjestelmässä voivat saada oikeudet kirjautua joihinkin tietokantoihin ja joihinkin eivät, riippuen käyttöoikeuksista. Tästä voi olla hyötyä eVikavihkole integraatioitten kehittämiseen.

### 4.4 Agile ja CI

Kehitystapa tulevaisuudessa eVikavihkole on Agile-mallinen. Agile-mallin avulla voidaan nopeammin toimittaa ominaisuuksia. Jos käytettäisiin kehitystapana vesiputousmallia, pitäisi olla jo suunniteltu kaikki ominaisuudet ennen kuin toteutusta voidaan aloittaa. Tämä taas ei ole nyky maailman tarpeilla mahdollista. Asiakkaalta saattaa esimerkiksi tulla pyyntö tiettyyn ominaisuuteen liittyen. Tässä saattaa perinteisellä vesiputousmallilla kestää kauankin aikaa, koska ennen toteutusta on päätettävä, mitkä ominaisuudet otetaan käyttöön, sekä niiden toiminta, toteutus ja testaus. Agile on tämän tyyppisissä ongelmissa ketterämpi kehitysmenetelmä, josta sen nimikin tulee. Siinä voidaan toteuttaa yksi ominaisuus kerrallaan, testata ja julkaista asiakkaille. Lisäksi Agilessä ominaisuuksia voidaan suunnitella ja toteuttaa samanaikaisesti.

CI (Continuous Integration) on tällä hetkellä jo pienessä mallissa kehitteillä, mutta ei ole täysin automaattinen. CI on käytäntö, jolla automatisoidaan useiden kehittäjien koodimuutosten integrointi yhdeksi ohjelmistoprojektiksi. (Atlassian, 2023). CI käyttötarkoituksia voi olla esimerkiksi automaattisesti testiympäristöön ohjelman asentaminen, automaattinen testienajo ja tuotantoversioitten testienajo.

#### 4.5 Tuotteen suunta

Tuotetta on tarkoitus tarjota useammalla toimialalla ja kehittää sovellusta asiakkaiden tarpeiden mukaan. Tarkoituksena on myös ylläpitää ja uudistaa teknologioita tarpeen mukaan. Tässä opinnäytetyössä kehitetty sovelluksen tilanne mahdollistaa sovelluksen käytön useammalla eri toimialalla. Jatkossa myös käyttöoikeuksien hallinta olisi tärkeä, jotta voidaan rajata oikeutta tiettyihin ominaisuuksiin.

## 5 YHTEENVETO

Opinnäytetyönä eVikavihkon jatkokehitys oli antoisaa. Työn aiheena oli saada laajennettua käyttöliittymää ja luoda uusia ominaisuuksia. Samalla uudistettiin sovellusarkkitehtuuria ja käytettiin uusia teknologioita. Oli mielenkiintoista uudistaa vanha teknologia uuteen.

Työssä myös tutkittiin teknistä velkaa, miten sitä kertyy ja miten sitä korjataan. Aiemmin kehitetyssä eVikavihkossa teknistä velkaa oli hieman. Tämän työn aikana nämä velat korjattiin. Uudet teknologiat mitkä otettiin tämän työn aikana käyttöön, mahdollistavat helpomman jatkokehityksen ja uusien ominaisuuksien luonnin.

Työn kehitys sujui hyvin ja aikataulussa. Käytännön toteutus tehtiin 2022 syksyllä ja kirjallinen osuus kevään 2023 aikana. Työtä ollaan testaamassa testausympäristössä, missä työn tilaaja ja referenssiyriytys Hydroline Oy voivat testata ja kokeilla työtä. Tämän opinnäytetyön jälkeen sovelluksen kehitystä jatketaan uusilla tekniikoilla ja ominaisuuksilla, jotka eivät sisälly tähän työhön.

Työhön ei saatu kaikkia toivottuja ominaisuuksia johtuen teknillisistä ja aikataulu rajoitteista. Kehitystyöstä suurin osa ajasta kului uusien teknologioitten käyttöönottoon. Tätä olisi voinut suunnitella enemmän, jotta palvelisi tulevaisuuden tarpeita. Työhön otettujen uusien ominaisuuksien kehitys onnistui hyvin. Lisäksi uudet teknologiat nopeuttavat tämän työn ulkopuolelle jääneiden ominaisuuksien kehittämistä ja uusien ominaisuuksien kehittämistä. Yhteenvetona sovelluksen jatkokehitys onnistui hyvin. eVikavihkon kehitys jatkuu työn tilaajan kanssa.

## 6 LÄHTEET

Ciksiti. (30. 7. 2021). *What is Vue.js and why is it cool*. Haettu 23. 12. 2022 osoitteesta Ciksiti: <https://ciksiti.com/fi/chapters/2050-what-is-vue-js-and-why-is-it-cool---linux-hint>

itewiki. (ei pvm). *Konttitekologia*. Haettu 29. 3. 2023 osoitteesta itewiki: <https://www.itewiki.fi/opas/konttitekologia/>

Krimaka. (ei pvm). Haettu 29. 1. 2023 osoitteesta krimaka.net: <https://www.krimaka.net/linux/alkuun/mik%C3%A4-on-linux.html>

Linux.fi. (13. 8. 2016). *PostgreSQL - Linux.fi*. Haettu 20. 3 2023 osoitteesta PostgreSQL: <https://linux.fi/wiki/PostgreSQL>

Linux.fi. (16. 11. 2020). *MariaDB*. Haettu 23. 12. 2022 osoitteesta Linux.fi: <https://www.linux.fi/wiki/MariaDB>

Microsoft. (20. 12. 2022). *Visual Studio Code*. (Microsoft) Haettu 29. 1. 2023 osoitteesta Microsoft: <https://learn.microsoft.com/fi-fi/power-apps/maker/portals/vs-code-extension>

Microsoft. (22. 3. 2023). *Microsoft: SQL Server*. Noudettu osoitteesta SQL Server: <https://www.microsoft.com/fi-fi/sql-server/>

Niemistö, T. (7. 11. 2016). *Tekninen velka - organisaatiosi pahin myrkky*. Haettu 30.. 3. 2023 osoitteesta digia: <https://blog.digia.com/tekninen-velka-yrityksesi-suurin-digitalisaation-jarru>

Sarlin, S. (2. 2021). *Angular-sovelluksen keskitetty tilanhallinta NgRx-kirjastolla*. Haettu 23. 12. 2022 osoitteesta Theseus.fi: <https://urn.fi/URN:NBN:fi:amk-202103193587>

Solvusoft. (ei pvm). *DBeaver*. Haettu 29. 1. 2023 osoitteesta Solvusoft: <https://www.solvusoft.com/fi/file-extensions/software/open-source/dbeaver/>

Tarvainen, J. (30. 7. 2016). *Mikä on typescript*. Haettu 23. 12. 2022 osoitteesta Symfony: <https://symfony.fi/artikkeli/mika-on-typescript>

Tilli, T. (2022). *eVikavihko : sovellus koneistajan ja huoltohenkilökunnan työn tukemiseksi konepajaympäristöön*. Haettu 28. 3. 2023 osoitteesta Theseus.fi: <https://urn.fi/URN:NBN:fi:amk-202203314284>

Timonen, J. (ei pvm). *Bootstrap sivupohjan tekeminen*. Haettu 23. 12. 2022 osoitteesta Jyrki Timonen: [http://jyti.net/dat16/bs\\_1.html](http://jyti.net/dat16/bs_1.html)

TR Plane. (ei pvm). *Mitä REST API tarkoittaa*. Haettu 29. 1. 2023 osoitteesta TRPlane.com: <https://www.trplane.com/fi/mit%C3%A4-rest-api-tarkoittaa/>

Viikon Valo. (16. 8. 2014). *FontAwesome*. Haettu 23. 12. 2022 osoitteesta Viikon valo: [https://viikonvalo.fi/font\\_awesome/](https://viikonvalo.fi/font_awesome/)

Wallenius, N. (23. 2. 2022). *Mikä on Docker ja mitä hyötyä siitä on?* Haettu 23. 12. 2022 osoitteesta Wallenius: <https://niklaswallenius.fi/mika-on-docker/>