



# SPA:n prototyypin suunnittelu ja toteutus

Yksisivuinen web-sovellusprototyyppi tuotannon  
infonäytöille

Tuomo Aaltonen

OPINNÄYTETYÖ  
Toukokuu 2023

Tieto- ja viestintäteknikan tutkinto-ohjelma  
Ohjelmistotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tieto- ja viestintätekniikan tutkinto-ohjelma  
Ohjelmistotekniikka

AALTONEN, TUOMO:  
SPA:n prototyypin suunnittelu ja toteutus  
Yksisivuinen web-sovellus tuotannon infonäyttöille

Opinnäytetyö 26 sivua, joista liitteitä 3 sivua  
Toukokuu 2023

---

Tämän opinnäytetyön tavoitteena on suunnitella ja toteuttaa SPA web -sovelluksen prototyyppi yrityksen tuotannon infonäyttöihin, joista katsellaan reaaliajassa työtilausdataa.

Työtilausten on tarkoitus sisältää dataa tuotannon työntekijöille, jota voidaan hyödyntää uutena prosessina työtehtävien nopeuttamiseksi.

Toimeksiantajana työssä toimii Robit Finland Oy Ltd. Toimeksiantajan tarve web-sovellukselle on työtilausdatan käsittelyn automatisointi ja infolaitteiston hyödyntäminen yrityksen tuotannon tiloissa.

SPA eli Single Page Application tarkoittaa yksisivuista applikaatiota, jonka toiminnot ovat kaikki määriteltyinä yhdelle sivulle. SPA:n päätarkoituksena tässä työssä on toimia työtilausdatan esityssivuna, joten se ei sisällä erilaisia interaktiivisia toimintoja käyttöliittymän puolella, paitsi datan selaukseen liittyviä toimintoja.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in ICT Engineering  
Software Engineering

**TUOMO AALTONEN:**

SPA prototype planning and development  
Single page web application prototype for production info screens

Bachelor's thesis 26 pages, appendices 3 pages  
May 2023

---

The purpose of this thesis is to plan and develop a SPA prototype for a company to be used in production info screens, where you can watch work order data in real time.

Work orders are meant to include data for the production workers that can be used as a new process to speed up work goals.

Employer for this job is Robit Finland Oy Ltd. The employer's need for a web application is for handling work order data automatically and using info screens in their production work environment.

SPA aka Single Page Application means a single paged application where all the functions are located on one page. The SPA's main purpose in this is to present work order data, so it doesn't contain interactive functions on the UI side.

---

Key words: spa, single page application

## SISÄLLYS

1	JOHDANTO .....	7
2	TYÖN SUUNNITTELU .....	8
	2.1 Esiselvitys .....	8
	2.2 Projektisuunnitelma .....	8
	2.3 Käyttöliittymäsuunnitelma .....	8
	2.4 Testaussuunnitelma .....	9
3	OHJELMISTOPROJEKTIN ORGANISAATIO .....	10
	3.1 Projektin asettaja ja ohjausryhmä .....	10
	3.2 Projektiryhmä .....	10
	3.3 Organisaatiokaavio .....	10
4	ALUSTAVA AIKATAULU JA VAIHEET .....	11
	4.1 Alustava aikataulu .....	11
	4.2 Vaiheet .....	11
5	TAVOITTEET JA VAATIMUSMÄÄRITTELY .....	12
	5.1 Tavoitteet .....	12
	5.1.1 Alkuperäiset tavoitteet .....	12
	5.1.2 Projektin tulostavoitteet .....	12
	5.1.3 Projektiryhmän tulostavoitteet .....	13
	5.2 Vaatimukset .....	13
	5.2.1 Toiminnalliset vaatimukset .....	13
	5.2.2 Ei-toiminnalliset vaatimukset .....	13
6	KEHITYSYMPÄRISTÖ .....	14
	6.1 Ympäristörakenne .....	14
	6.2 Ympäristökaavio .....	14
7	TYÖMENETELMÄT, VERSIOHALLINTA JA RAJAUS .....	15
	7.1 Työmenetelmät .....	15
	7.2 Versiohallinta .....	15
	7.3 Rajaus .....	16
8	PROJEKTIN PÄÄTTÄMISEN JA KESKEYTTÄMISEN KRITEERIT ...	17
	8.1 Päättämisen kriteerit .....	17
	8.2 Keskeyttämisen kriteerit .....	17
9	LAITTEISTON TOIMINTAKUVAUS .....	18
	9.1 Infolaitteiden toiminta .....	18
	9.2 Infolaite .....	18
10	TIETOKANNAN TOIMINTAKUVAUS .....	19
11	SOVELLUKSEN PROTOTYYPIN TOIMINTAKUVAUS .....	20

11.1 Frontend .....	20
11.2 Backend.....	20
12 TYÖN TULOKSIEN TARKASTELU .....	21
12.1 Sovelluksen prototyypin tulokset .....	21
12.2 Dokumentaation tulokset .....	21
13 POHDINTA .....	22
LÄHTEET.....	23
LIITTEET .....	24
Liite 1. Käyttöliittymäsuunnitelman hyväksytty versio. ....	24
Liite 2. Projektin aikataulu .....	25
Liite 3. Backend testikoodi server.js.....	26

## LYHENTEET JA TERMIT

SPA	Single Page Application tarkoittaa yksisivuista web-sovellusta tai websivua, joka ei lataa uusia sivuja tai päivitä sivua datan käsittelyssä, jotta saadaan nopeampia toimintoja käyttöön.
ERP	Enterprise Resource Planning -järjestelmiä käytetään yrityksissä resurssienhallintaan.
Azure	Microsoftin pilvipalvelut. Tässä työssä on käytössä yritykselle kohdenneet privaatit pilvipalvelut.
Microsoft SQL Express	Microsoftin SQL Serverin ilmaisversio. Käytetään palvelimena ohjelmistokehityksessä kustannuksien vähennykseen.
React JS	JavaScript-kirjasto, joka ratkaisee SPA:n toimintaongelmia. Reactia käytetään käyttöliittymän rakentamiseen.
Node JS	Node JS on asynkroninen JavaScript-tulkki, joka sisältää skaalautuvan arkkitehtuurin ja NPM-paketinhallintatoimintoja.
Express JS	Node JS varten kehitetty minimaalinen ja joustava JavaScript-kirjasto, jota käytetään http-pyyntöissä.
Jest JS	Jest JS on testausta varten luotu JavaScript-kirjasto.
Bootstrap	Bootstrap on tyylikirjasto, joka tekee sovelluksesta responsiivisemmän ja paremmin skaalautuvan eri kokoisille laitteille. Työssä on käytössä React Bootstrap.
Frontend	Frontend on sovelluksen osio, joka on toiminnaltaan keskittynyt käyttöliittymän luomiseen.
Backend	Backend on sovelluksen osio, joka on toiminnaltaan keskittynyt sovelluksen toimintalogiikkaan palvelimella. Backend on siis vastuussa tietokannan yhteyksistä ja tietojen hallinnoimisesta, sekä tietojen käsittelystä.
IP-osoite	IP-osoite eli Internetin protokollaosoite on numerosarja, jolla yksilöidään verkkoon kytkettyjä laitteita.

## 1 JOHDANTO

Työssä käydään läpi SPA-ohjelmistoprojektin vaiheita, web-sovelluksen toteutettuja toimintoja ja tulosten tarkastelua, sekä omaa pohdintaa.

Kappaleessa 2 avataan sovelluksen suunnitteluvaiheen tapahtumia ja sen sisältämää dokumentaatiota.

Kappaleissa 3, 4 ja 5 käydään läpi työn projektin organisaatiota, aikataulua, tavoitteita ja vaatimuksia. Organisaatiosta kuvataan kappaleessa 2 projektin asettajaa, sekä ohjausryhmää ja erikseen projektiryhmää. Kappaleesta 2 löytää myös projektin organisaatiokaavion. Kappale 3 käsittelee työn alustavaa aikataulua ja opinnäytetyön ohjelmistoprojektin vaiheita. Kappale 4 sisältää työn alkuperäiset tavoitteet ja tulostavoitteet projektille, sekä projektiryhmälle. Kappaleesta 4 avataan myös toiminnallisia- ja ei-toiminnallisia vaatimuksia.

Kappaleissa 6 ja 7 esitellään työn kehitysympäristöä, työmenetelmiä, versiohallintaa ja rajausta. Kehitysympäristöstä on kappaleessa 5 kuvaus ympäristörakenteesta ja ympäristökaavio, joka on laadittu esiselvityksen aikana. Kappaleessa 6 on esiteltyä työssä käytetyt yleiset työmenetelmät, versionhallinnan rooli opinnäytetyön kokonaisuudessa ja työn vastuiden ja käyttötarkoituksen rajausta.

Kappaleessa 8 käydään läpi työn päättämisen kriteerit, sekä keskeyttämisen kriteerit. Nämä kriteerit on saatu toimeksiantajalta ja avaavat opinnäytetyön kokonaisuutta yleisellä tasolla.

Kappaleissa 9, 10 ja 11 kuvataan laitteiston, tietokannan ja sovelluksen prototyypin toimintaa yleisesti, sekä kooditasolla.

Kappaleessa 12 tarkastellaan opinnäytetyön lopputuloksia.

Kappaleessa 13 tehdään pohdintaa opinnäytetyön pohjalta.

## **2 TYÖN SUUNNITTELU**

Työn suunnitteluvaiheessa laadittiin esitutkimusraportti, josta luotiin projektisuunnitelma. Työssä suunniteltiin myös sovelluksen käyttöliittymä prototyyppiä varten. Testausta varten luotiin testaussuunnitelma. Dokumenttien tiedot on kuvattu työssä omissa kappaleissaan kokonaisuuden esittelyä varten.

### **2.1 Esiselvitys**

Työn esiselvityksen aikana luotiin esitutkimusraportti, joka sisältää työn taustan, projektin organisaation, nykyisen prosessin kuvauksen, riskiarvioinnin, tavoitteet, vaatimusmäärittelyn, toteutusvaihtoehdot, ympäristön, hyödyt, kustannukset ja alustavan aikataulun.

### **2.2 Projektisuunnitelma**

Työn projektisuunnitelma luotiin esiselvityksen tiedoista ja kuvaa projektin tavoitteita, rajausta, aikataulua, työmenetelmiä, päättämiskriteereitä ja keskeyttämiskriteereitä. Projektisuunnitelmaa ja sen sisältämää projektin aikataulua päivitetään projektin aikana.

### **2.3 Käyttöliittymäsuunnitelma**

Käyttöliittymäsuunnitelmassa on luotu erilaisia mahdollisia versioita käyttöliittymästä. Käyttöliittymän versiot on suunniteltu yksinkertaisesti ajansäästön kannalta ja versioiden korjaus tapahtuu palautteen perusteella. Käyttöliittymäsuunnitelman hyväksynnästä on vastuussa projektin ohjausryhmä. Hyväksytty versio, jonka pohjalta prototyyppi on tehty, löytyy liitteenä 1.



## 2.4 Testaussuunnitelma

Testaussuunnitelmasta löytyvät SPA:n erilaiset testausmetodit ja testitapaukset. Testauskohteisiin kuuluu kaikki, mitä mainitaan testaussuunnitelman testitapauksissa. Testitapaukset pidetään salassa projektin asettajan pyynnöstä.

Sovelluksen frontend -toimintoja ei testata erikseen, koska käyttöliittymällä ei ole interaktiivisuutta tai käyttäjää.

Testaus on suunniteltu automatisoiduksi työn toimeksiantajan toiveesta. Sovelluksen backend-toiminnot testataan Jest JS:n avulla, koska se on yksinkertainen ja luotettava vaihtoehto Node JS -sovelluksien testauksessa.

### 3 OHJELMISTOPROJEKTIN ORGANISAATIO

#### 3.1 Projektin asettaja ja ohjausryhmä

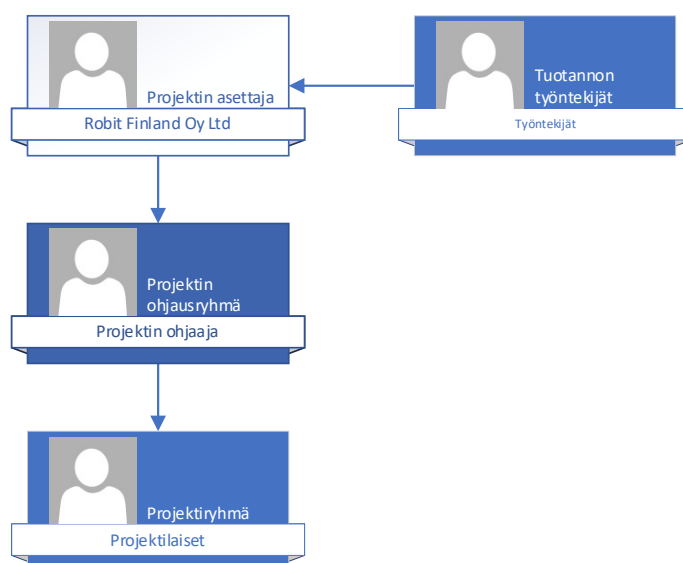
Projektin asettajana ja työnantajana toimii Robit Finland Oy Ltd. Asettajan vastuulla on projektin vaatimusten, tavoitteiden ja päättämiskriteerien laatiminen.

Projektin ohjausryhmään kuuluu yksi ohjaaja, joka ohjaa projektiryhmää ja valvoo projektin etenemistä. Ohjausryhmä selvittää tarvittaessa ongelmia ja toimii keskustelussa siltana projektin asettajan ja projektiryhmän välillä.

#### 3.2 Projektiryhmä

Projektiryhmään kuuluu yksi jäsen, joka on vastuussa työn suunnittelusta, toteutuksesta, testauksesta ja raportoinnista. Projektiryhmän päätavoitteena on työskennellä hyväksytyyn projektisuunnitelman mukaisesti ja keskustella ohjausryhmän kanssa projektin etenemisestä.

#### 3.3 Organisaatiokaavio



Kaavio 1: Organisaatiokaavio

## **4 ALUSTAVA AIKATAULU JA VAIHEET**

### **4.1 Alustava aikataulu**

Projektin aloitus on tammikuussa ja arvioitu lopetus on toukokuun puolella. Eli työn kesto on noin 4–5 kuukautta.

Projekti jakautuu neljään vaiheeseen ja jokaisen vaiheen lopussa luodaan uusi vaiheraportti, joka käydään läpi projektin asettajan ja ohjausryhmän kanssa.

### **4.2 Vaiheet**

Projektin vaiheet ovat aloitusvaihe, toteutusvaihe, testausvaihe ja lopetusvaihe.

Projektin aloitusvaiheessa eli suunnitteluvaiheessa tehdään esiselvitystä ja laaditaan esitutkimusraportti, käyttöliittymäsuunnitelma, testausuunnitelma ja projektisuunnitelma esiselvityksen tiedoista.

Projektin toteutusvaiheessa toteutetaan web-sovelluksen prototyyppi projektisuunnitelman mukaisesti ja tuotantoversio sovelluksesta prototyypin pohjalta.

Testausvaiheessa testataan web-sovelluksen toimintaa testausuunnitelman mukaisesti.

Lopetusvaiheessa tarkistetaan projektin päättämiskriteerien toteutuminen ja luodaan käyttöohjeet, sekä projektin loppuraportti.

Projektin tarkempi aikataulu löytyy liitteestä 2. Projektin aikataulu on luotu Visiolla ja esitetty kahdella GANTT-kaaviolla. Aikataulua on säädetty työn aikana tarpeen mukaisesti.

## **5 TAVOITTEET JA VAATIMUSMÄÄRITTELY**

Projektin tavoitteet ja vaatimusmäärittely kehittyvät projektin aikana. Tavoitteissa on listattuna erikseen projektin alkuperäiset tavoitteet ja lopulliset tavoitteet. Projektin vaatimuksien on oltava virheettömiä, realistisia, ristiriidattomia, tarpeellisia ja todennettavissa.

Projektin tavoitteet ja vaatimukset asettaa projektin asettaja eli tässä tapauksessa Robit Finland Oy Ltd:n tuotannon johtoryhmä.

### **5.1 Tavoitteet**

Projektin tavoitteet voidaan jakaa alkuperäisiin tavoitteisiin ja tulostavoitteisiin. Tulostavoitteet on jaettu erikseen projektin tavoitteisiin ja projektiryhmän tavoitteisiin.

#### **5.1.1 Alkuperäiset tavoitteet**

- Web-ohjelmiston prototyypin toteuttaminen ja testaaminen.
- Web-ohjelmiston tuotantoversion toteuttaminen ja testaaminen.
- Nykyisen prosessin kartoitus ja korvaaminen.
- Infonäyttöjen hyödyntäminen työnteossa tuotannossa.

#### **5.1.2 Projektin tulostavoitteet**

Projektin tulostavoitteena on luoda uusi prosessi työtilausten tarkasteluun yrityksen tuotannon työntekijöille. Projektin tavoitteet voivat kasvaa tulevaisuudessa, mutta ne eivät saa ylittää projektiin asetettuja rajoitteita, jotka ovat mainittu kappaleessa 6.

### **5.1.3 Projektiryhmän tulostavoitteet**

Projektiryhmän tavoitteena on projektin valmistuminen ajallaan, projektin alkuperäisten tavoitteiden saavuttaminen, prototyypin rakentaminen ja käyttökelpoisen ohjelmiston suunnittelu, toteutus ja testaus. Projektilaiset ovat vastuussa projektiryhmän tavoitteiden toteuttamisesta.

## **5.2 Vaatimukset**

Projektin vaatimuksen on jaettu toiminnallisiin ja ei-toiminnallisiin vaatimuksiin. Toiminnalliset vaatimukset kuvaavat vaatimuksia, jotka takaavat projektin toiminnallisuutta. Ei-toiminnalliset vaatimukset sisältävät vaatimuksia, joita ei näy ohjelmiston toiminnallisuudessa, mutta on vaadittu projektin tavoitteita varten.

### **5.2.1 Toiminnalliset vaatimukset**

- Ohjelmisto sisältää todennettua ja hyödyllistä dataa tuotannon työntekijöille.
- Soluissa tulee näkyä valmistumispäivämäärä, työpiste/solu, suunniteltu/valmismäärä ja tuotenimi.
- Työjärjestyksen tulee olla looginen.
- Yrityksen privaattia Azure-alustaa tulee hyödyntää datan noutamisessa.

### **5.2.2 Ei-toiminnalliset vaatimukset**

Projektin ei-toiminnallisiin vaatimuksiin kuuluvat sovelluksen sisällön selkeä esitystapa, skaalautuvuus, helppokäyttöisyys ja reaktiivisuus.

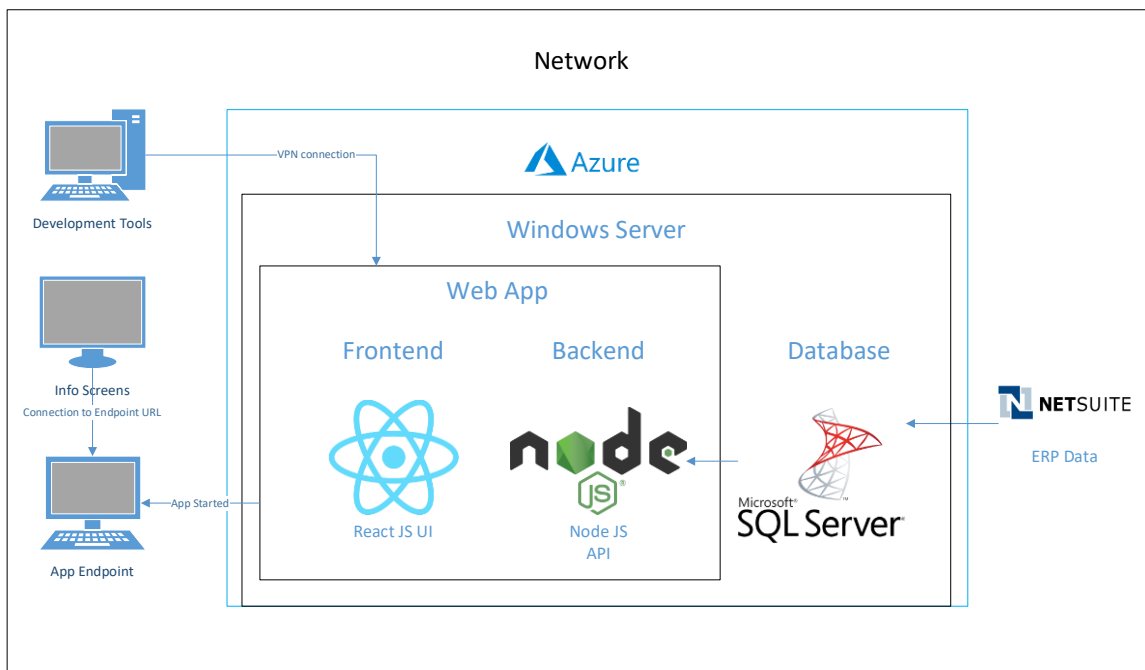
## 6 KEHITYSYMPÄRISTÖ

### 6.1 Ympäristörakenne

Projektin kehitysympäristönä toimii tietty Windows Server -kone yrityksen privaattilla Azure-alustalla. Web-sovelluksen frontend käyttää runkona React JS -kehystä ja bootstrappia tyyliasetuksien asettelussa. Node JS toimii sovelluksen backend API -kutsujen toiminnoissa.

Tietokantana toimii Microsoftin SQL Server Express, joka on ilmainen versio Microsoftin SQL Server RDB:stä. Tietokantaan syötetään työtilausten dataa automaattisesti NetSuitesta. NetSuite toimii yrityksen ERP eli Enterprise Resource Planning -järjestelmänä, joka sisältää tietokantaan lähetettävän datan.

### 6.2 Ympäristökaavio



Kaavio 2: Ympäristökaavio esitutkimusraportista

## **7 TYÖMENETELMÄT, VERSIOHALLINTA JA RAJAUS**

Työssä on määriteltynä projektin työmenetelmät sekä versiohallinta. Työn laajuutta on rajattu projektin esiselvityksen mukaisesti.

### **7.1 Työmenetelmät**

Työnteossa ei sovelleta tiettyä prosessimallia, vaan edetään aikataulun mukaisesti eteenpäin. Aikataulu sisältää vaihekohtaisen työmäärän ja päivittyä projektin edetessä.

Projektin sisäisessä dokumentaatiossa käytetään suomen kieltä ja ulkopuolisessa dokumentaatiossa, sekä koodauksessa käytetään englannin kieltä.

Ohjelmiston koodin keskeisimmät toiminnot kommentoidaan koodissa, esimerkiksi luokat, oliot, metodit, funktiot ja muuttujat. Web-sovellus sisältää myös readme-tiedoston, joka sisältää sovelluksen tärkeimmät toiminnot.

### **7.2 Versiohallinta**

Ohjelmiston versiohallinta automatisoidaan Git-versiohallintajärjestelmällä ja luodaan erikseen varmuuskopiot tiedostoista IT-tiimille tiedostojen palautusta varten.

Kaikki kehityksen aikaiset muutokset ladataan yleisesti omina versioinansa ja vältellään muutosten lähettämistä suoraan tuotannon versioon. Ympäristötietoja ei saa ladata tuotantoversioon tietoturvallisuuden kannalta.

Versiohallinta on taattava ohjelmistokehityksen aikana ja kehitys ei saa jatkua ilman tätä takuuta.

### 7.3 Rajaus

Ohjelmistoprojektissa suunnitellaan ja toteutetaan SPA eli yksisivuinen web-sovellus, joka on rajoitettu opinnäytetyössä prototyypin toteutukseen. Projektissa ei tarvita sovellukseen erillisiä käyttäjiä, joten toimintoja kuten kirjautumista, rekisteröitymistä tai salasanan vaihtoa ei tule toteuttaa.

Sovelluksen ylläpito on yrityksen vastuulla ja kuuluu projektin työtehtävien ulkopuolelle. Tarvittavaa informaatiota ylläpitoon löytyy sovelluksen tiedostoista ja loppuraportista.

Projektilaiset eivät ole vastuussa tietokannan toiminnoista. Projektilaiset eivät myöskään ole vastuussa Infosign-laitteiston jatkuvasta toimivuudesta tai projektin jälkeisistä kehitysympäristössä ilmentyvistä ongelmista.



## **8 PROJEKTIN PÄÄTTÄMISEN JA KESKEYTTÄMISEN KRITEERIT**

Projektin asettaja päättää projektin aloittamisesta, keskeyttämisestä ja päättämisestä. Alla on listattuna kriteerit projektin päättämiseen ja keskeyttämiseen. Projektin päättämisen ja keskeyttämisen kriteerit on liitetty projektisuunnitelmasta.

### **8.1 Päättämisen kriteerit**

Projekti voidaan päättää kun:

- Alkuperäiset tavoitteet on saavutettu.
- Projektin työtehtävät on suoritettu.
- Sovelluksen toiminta on testattu.
- Projektin tiedostot on varmuuskopioitu.
- Loppupalaveri on pidetty.
- Loppuraportti on luotu ja palautettu.

### **8.2 Keskeyttämisen kriteerit**

Projekti voidaan keskeyttää jos:

- Työnteko ei onnistu erikoistilanteiden vuoksi.
- Projektiryhmä on lakkautettu.
- Projekti vaatii liikaa resursseja.
- Ylitse pääsemättömät tekniset ongelmat projektissa.
- Projektin asettaja keskeyttää projektin.

## 9 LAITTEISTON TOIMINTAKUVAUS

### 9.1 Infolaitteiden toiminta

Yritys on hankkinut infolaitteensa alihankkijalta InfoSign Oy. Kyseisen info-tv-järjestelmän laitteet sisältävät Internet Explorer- ja Firefox -selaimet. Tässä työssä hyödynnetään infonäyttöohjelmistoa, joka ohjaa viestintää infolaitteisiin oman viikkokalenterinsa mukaisesti. Infolaitteille on konfiguroitu asetuksissa automaattinen esityksen avaus sovelluksen pääteosoitteeseen ja mahdollistettu animaatioiden näyttäminen Windowsin Performance -asetuksissa.

Infolaitteena toimii pieni laite, joka ottaa selaimella yhteyden yritykselle suunnattuun viikkokalenteriohjelmistoon ja jakaa siellä olevaa sisältöä infolaitteeseen kytkettyyn näyttöön. Näytön tai television hankinta laitetta varten on asiakkaan vastuulla. Infolaitteesta löytyy HDMI ja VGA liitännät, joilla voidaan kytkeä laite näyttöihin kiinni. Laitteessa on myös Ethernet portti ja Wi-Fi -toiminnallisuus. Laitetta voidaan manuaalisesti käsitellä kytkemällä kahteen USB-porttiin näppäimistö ja hiiri.

### 9.2 Infolaitte



Kuva 1: Infolaitte

## 10 TIETOKANNAN TOIMINTAKUVAUS

Työn tietokanta eli Microsoftin SQL Express on ilmaisversio, joten se on rajoitettu pieniin datamääriin. Tietokanta sopii hyvin sovelluksen kehitykseen ja ensimmäiseen käyttöönottoon, sekä vähentää samalla ylimääräisiä kustannuksia.

Tietokantaa voidaan hallinnoida Azuressa sijaitsevalta virtuaalikonepalvelimelta Microsoft SQL Server Management Studion avulla. Tietokantaan on luotu erilliset tunnukset työn sovelluksen yhteyksien ja oikeuksien rajoittamiseksi. Erillisten tunnuksien käyttäminen ei anna oikeuksia muokata tai poistaa dataa, eli ne tukevat työn tietoturvallista toimintaa.

Tietokanta sisältää tuotannon dataa erillisessä taulukossa, joka päivittyy automaattisesti 5 minuutin välein. Kaikkia kenttiä ei hyödynnetä tässä työssä, mutta tarvittavat kentät näkyvät käyttöliittymäsuunnitelmassa, sovelluksen prototyypissä ja viimeistellyssä sovelluksessa.

## 11 SOVELLUKSEN PROTOTYYPIN TOIMINTAKUVAUS

Kaikki tarvittavat paketit sovelluksen toiminnassa on asennettu Noden NPM-paketinhallintatoiminnoilla. Sovelluksen prototyyppi on myös ladattu ja varmuuskopioitu yrityksen verkostoon.

### 11.1 Frontend-toiminnallisuus

Prototyyppi sisältää yhden käyttöliittymän, joka on luotu käyttöliittymäsuunnitelman mukaisesti. Käyttöliittymä sisältää työtilausten taulukon täytettynä mallidatalla, päivitysten aikaleiman tekstikentän ja päivämäärän tekstikentän.

Prototyypissä hyödynnetään React Bootstrappia tyyliasetuksien konfiguraatiossa, koska React Bootstrap on käyttöliittymään yhteensopiva ja hyvin dokumentoitu React -komponentti. Tyyliasetukset säätelevät taulukon kokoa ja datan rivitystä taulukossa.

### 11.2 Backend-toiminnallisuus

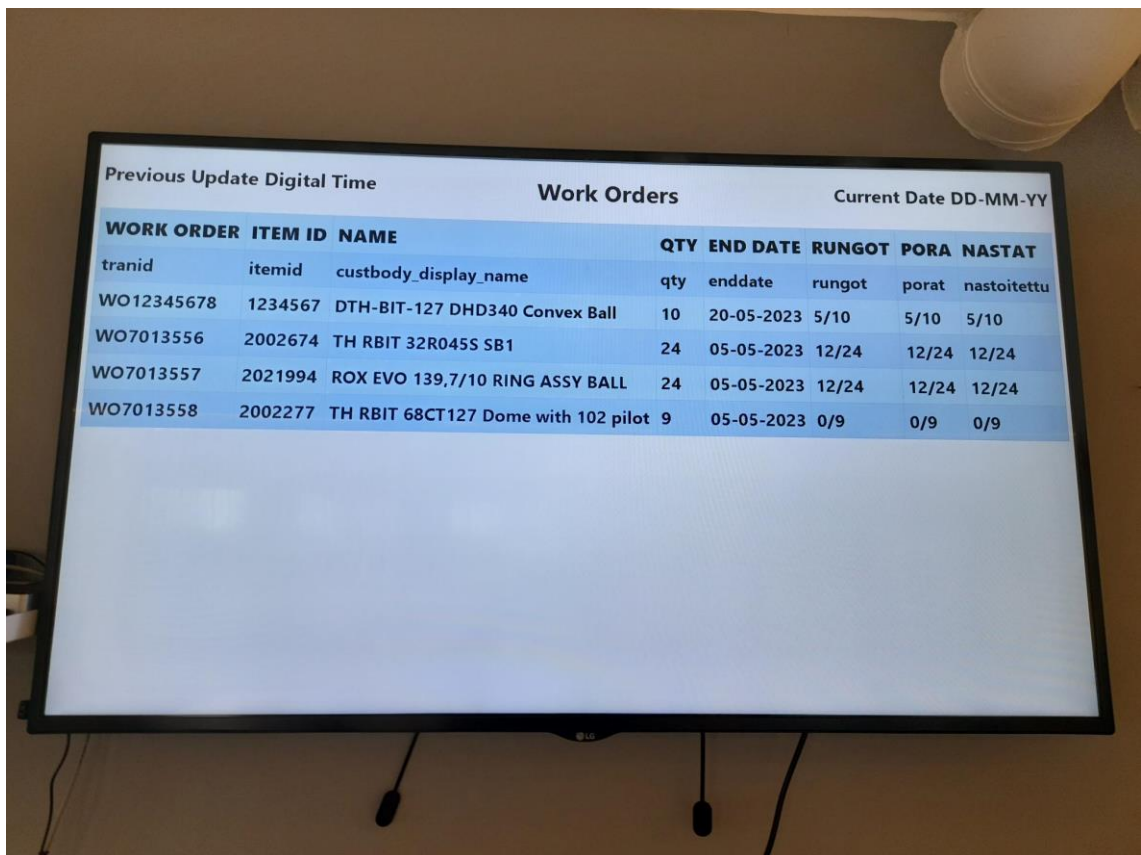
Prototyypin backendillä otetaan yhteys tietokantaan Node JS:n ja Express JS:n avulla. Dataa valitaan tietokannasta käyttäen SQL-autentikointia ja mssql-ajuria yhteyden luomiseen. Yhteyden testaukseen on ajettu Nodella server.js-tiedosto, joka löytyy liitteenä 3. Kaikki salattavat arvot on laitettu erilliseen .env-tiedostoon.

## 12 TYÖN TULOKSIEN TARKASTELU

Alkuperäisistä tavoitteista on saavutettu prototyypin toteutus ja testaus.

### 12.1 Sovelluksen prototyypin tulokset

Prototyyppi toimii infonäytöllä verkolle avatun IP-osoitteen kautta ja näyttää mallidataa taulukossa. Prototyypin backendissä on onnistuneesti otettu yhteys tietokantaan ja noudettu dataa Node Expressin avulla.



Previous Update Digital Time		Work Orders			Current Date DD-MM-YY		
WORK ORDER	ITEM ID	NAME	QTY	END DATE	RUNGOT	PORA	NASTAT
tranid	itemid	custbody_display_name	qty	enddate	rungot	porat	nastoitettu
WO12345678	1234567	DTH-BIT-127 DHD340 Convex Ball	10	20-05-2023	5/10	5/10	5/10
WO7013556	2002674	TH RBIT 32R045S SB1	24	05-05-2023	12/24	12/24	12/24
WO7013557	2021994	ROX EVO 139,7/10 RING ASSY BALL	24	05-05-2023	12/24	12/24	12/24
WO7013558	2002277	TH RBIT 68CT127 Dome with 102 pilot	9	05-05-2023	0/9	0/9	0/9

Kuva 2: Prototyyppi infonäytöllä

### 12.2 Dokumentaation tulokset

Dokumentaation tuloksiin kuuluvat esitutkimusraportti, projektisuunnitelma, käyttöliittymäsuunnitelma ja testausuunnitelma.

## 13 POHDINTA

SPA-sovelluksen prototyyppi sopii yrityksen tarpeisiin, mutta vaatii vielä jatkokehitystä tietoturvan ja toimintojen suhteen ennen tuotannon version laatimista.

Suunnitteluvaiheeseen kului paljon aikaa ja vaati paljon dokumentaatiota työn toteutusvaihetta varten. Toteutusvaihe oli lyhyempi kattavan suunnitteluvaiheen dokumentaation ja käyttöliittymän rajoitettujen toimintojen ansiosta.

Suurimmat haasteet työssä olivat projektin aikataulun hallinta, sekä uusien laitteiden ja järjestelmien käyttöönotto. Aikataulun hallintaa työssä vaikeuttivat ulkopuoliset tekijät ja kehitysympäristön luomiseen kulunut aika.

Työssä opin ohjelmistoprojektin suunnittelua ja toteutusta, sekä käyttämään ja hallinnoimaan Microsoft SQL Server (Express) -tietokantaa.

Työn kokonaisuutta rajoitettiin koko sovelluksen toteutuksesta prototyypin tasolle, mutta sovelluskehitys jatkuu opinnäytetyön jälkeen ja voi laajentua tulevaisuudessa suuremmaksi ohjelmistokokonaisuudeksi.

## LÄHTEET

Tutorialsteacher. n.d. Access SQL Server in Node.js. Verkkosivu. Viitattu 17.4.2023. <https://www.tutorialsteacher.com/nodejs/access-sql-server-in-nodejs>

Microsoft. n.d. SQL Server. Verkkosivu. Viitattu 3.5.2023. <https://www.microsoft.com/en-us/sql-server/>

Wikipedia. 2023. SQL Server Express. Verkkosivu. Viitattu 3.5.2023. [https://en.wikipedia.org/wiki/SQL\\_Server\\_Express](https://en.wikipedia.org/wiki/SQL_Server_Express)

Wikipedia. 2023. IP-osoite. Verkkosivu. Viitattu 2.5.2023. <https://fi.wikipedia.org/wiki/IP-osoite>

InfoSign. N.d. Infonäyttöohjelmisto. Verkkosivu. Viitattu 2.5.2023. <https://infosign.fi/ohjelmisto.php>

Hawthorne, M. 2020. Windows 10: How to Enable/Disable Window Animation. Technipages. Verkkosivu. Viitattu 12.4.2023. <https://www.technipages.com/windows-10-how-to-enable-disable-window-animation/>

React Bootstrap. n.d. Introduction. Verkkosivu. Viitattu 5.4.2023. <https://react-bootstrap.github.io/getting-started/introduction>

Express. n.d. Express. Verkkosivu. Viitattu 4.5.2023. <https://expressjs.com/>

Bekkhuis, S. 2023. Getting Started. Jest. Verkkosivu. Viitattu 4.5.2023. <https://jestjs.io/docs/getting-started>

React. n.d. React. Verkkosivu. Viitattu 4.5.2023. <https://react.dev/>

Papa, J. 2018. Node.js Everywhere with Environment Variables. Medium. Verkkosivu. Viitattu 4.5.2023. <https://medium.com/the-node-js-collection/making-your-node-js-work-everywhere-with-environment-variables-2da8cdf6e786>

## LIITTEET

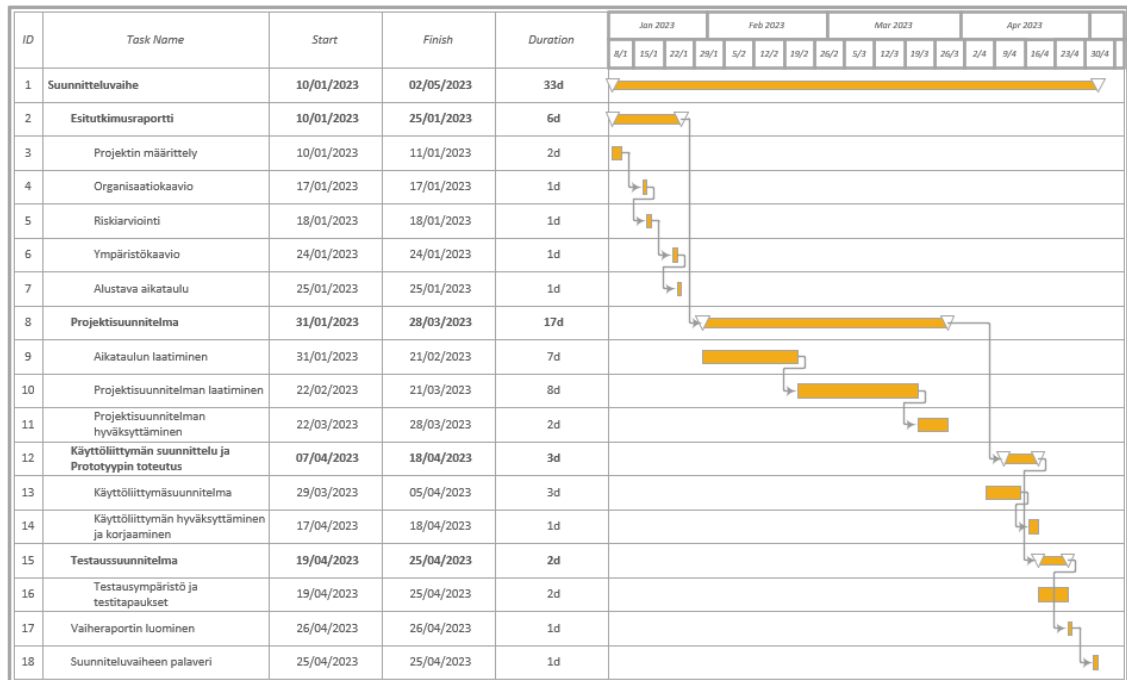
Liite 1. Käyttöliittymäsuunnitelman hyväksytty versio.

PREVIOUS UPDATE DIGITAL TIME			TITLE		CURRENT DATE DD-MM--YY		
WORK ORDER	ITEM ID	NAME	TOTAL QUANTITY	END DATE DD-MM-YY	RUNGOT	PORATTU	NASTOITETTU
<u>tranid</u>	<u>itemid</u>	<u>custbody_displ</u> <u>ay_name</u>	<u>quantity</u>	<u>enddate</u>	<u>rungot_ta</u> <u>lossa</u>	<u>porattu</u>	<u>nastoitettu</u>
WO123456 78	1234567	DTH-BIT 127 DHD340 Convex Ball	10	15-04-2023	5/10	5/10	5/10

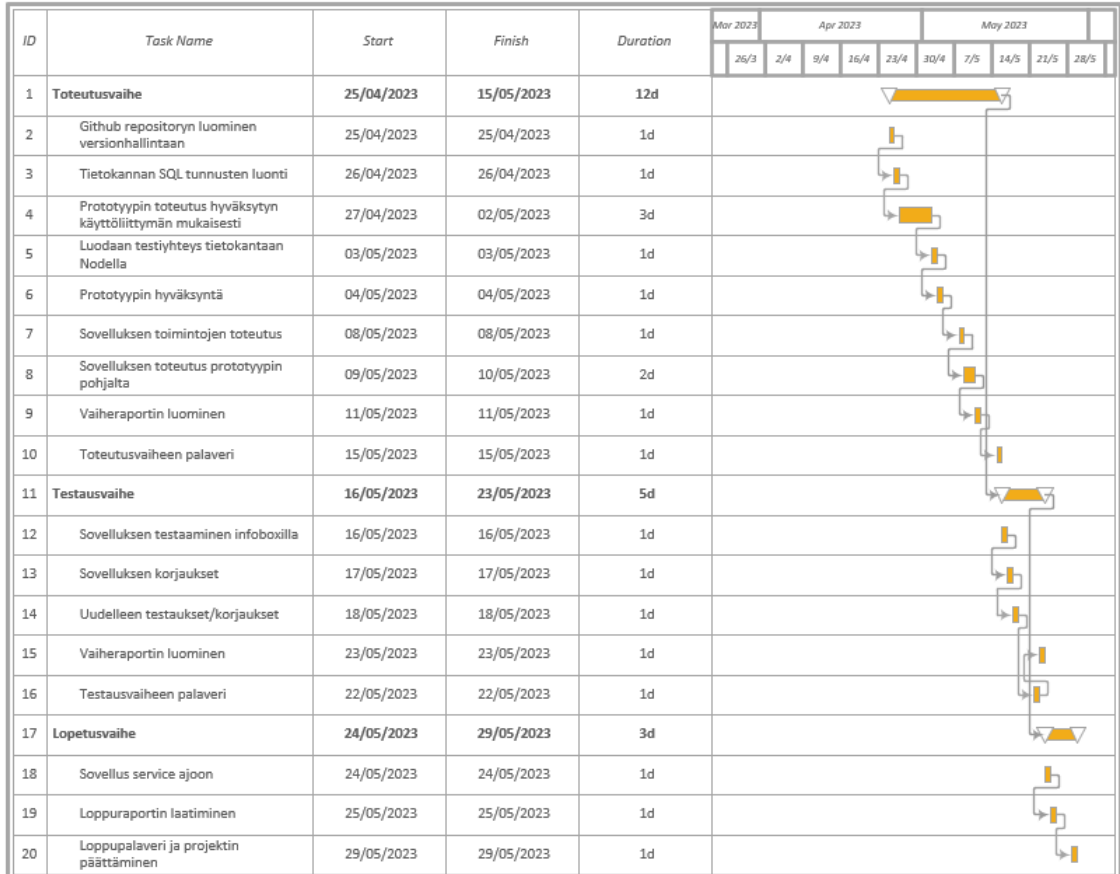


## Liite 2. Projektin aikataulu

1(2)



2(2)



## Liite 3. Backend testikoodi server.js

```
JS server.js > ...
1 // express JS variables
2 var express = require("express");
3 var app = express();
4
5 // app get root
6 app.get("/", function (req, res) {
7 // require mssql to variable sql for use in sql connections
8 var sql = require("mssql");
9
10 // credentials config for database connections
11 var config = {
12 user: process.env.DBUSER, // DB user
13 password: process.env.DBPASSWORD, // DB user password
14 server: process.env.SERVER, // DB host/server address
15 database: process.env.DB, // Database name on the server
16 };
17
18 // connecting to the database using config data
19 sql.connect(config, function (err) {
20 if (err) console.log(err);
21
22 // creating a new request object
23 var request = new sql.Request();
24
25 // SQL query to the database and get the records from a desired table
26 request.query("SELECT * FROM tablename", function (err, recordset) {
27 if (err) console.log(err);
28
29 // send records as a response
30 res.send(recordset);
31
32 // close SQL connection
33 sql.close();
34 });
35 });
36 });
37
38 // listen to port 8080 as default
39 app.listen(8080, function () {
40 console.log("Server is running..");
41 });
```