



# Tuotannonseuranta- ohjelman luominen

Tomi Salminen

Marika Verger

OPINNÄYTETYÖ  
Marraskuu 2024

Tietotekniikan tutkinto-ohjelma  
Ohjelmistotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikan tutkinto-ohjelma  
Ohjelmistotekniikka

SALMINEN, TOMI & VERGER, MARIKA:  
Tuotannonseurantaohjelman luominen

Opinnäytetyö 30 sivua  
Marraskuu 2024

---

Yksi yleisimmistä tehdastoiminnan johtamisen apuvälineistä on tuotantosuunnitelma, joka käsittää aikataulutuksen, työresurssien allokoinnin ja toimitettavat tilaukset. Erityisesti yrityksissä, joissa työvaiheita on useita ja tuotantolinjat ovat monimutkaisia, suurin hyöty saadaan tuotantosuunnitelman ollessa yrityksen toiminnan ympärille rakennettu. Tuotantosuunnitelma tehostaa yrityksen sisäistä koordinointia mahdollistamalla nopean reaktioajan muutoksiin ja minimoimalla virheiden vaikutuksia tuotantoprosessin aikana.

Opinnäytetyön tavoitteena oli tarkastella verkkopohjaisen tuotantosuunnitelman toteuttamiskelpoisuutta suomalaiselle hitsausalan yritykselle Pemamekille konseptitodistuksen muodossa. Konseptitodistuksesta saatujen tuloksien perusteella tehdään päätös projektin jatkotoimenpiteistä ja niiden pohjalta kehitetään paremmin yrityksen nykytoimintaa palveleva tuotantosuunnitelma.

Työssä keskityttiin selvittämään verkkopohjaisen tuotantosuunnitelman integroimismahdollisuuksia yrityksen jo käytössä oleviin järjestelmiin, ja miten oleellimmat tuotantosuunnitelman toiminnot olisi mahdollista toteuttaa valituilla teknologioilla. Konseptitodistuksen tueksi laadittiin havainnollistava versio, joka mallintaa mahdollisen tuotantosuunnitelman toimintaa ja joka voisi toimia runkona lopulliselle ohjelmistoprojektille.

Tuloksien esittämisen jälkeen päätettiin jatkokehityssuunnitelmista projektin osalta. Liitännät yrityksen muihin järjestelmiin todettiin toimiviksi perustoiminnallisuuden osalta ja pystyttiin todentamaan esimerkkisovelluksen avulla. Seuraavina edistysaskelina projektille ovat lisätoiminnallisuuksien kehittäminen ja käyttöönottoon keskittyminen.

---

Asiasanat: verkkosovellus, Vue, Vuetify, TypeScript, MongoDB, tuotantosuunnitelma

## ABSTRACT

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

SALMINEN, TOMI & VERGER, MARIKA:  
The development of a production monitoring program

Bachelor's thesis 30 pages  
November 2024

---

One of the most common tools for leading factory operations is the production plan which consists of scheduling, the allocation of work resources and to-be-delivered orders. Especially in companies that have multiple work phases and complicated production lines, the greatest benefit is gained when the production plan is built around the company's operations. The production plan streamlines the company's inner coordination by enhancing reaction time to changes and minimizing the consequences of mistakes during production process.

The goal of this thesis was to study the viability of a web-based production plan for a Finnish welding company Pemamek as a concept of proof. Based on the results a decision would be made on the future prospects of the project. If deemed viable, a production plan better suited for the current operations of the company would be pursued.

The work was focused on figuring out the integration possibilities with the systems already at place in the company and how the most relevant functions of the production plan would be carried out with the chosen technologies. To support the concept of proof an early version of the production plan was made to further demonstrate the workings of the final product.

After evaluating the results a plan was made to advance the project further. Connections with the company's other systems were proven compatible in terms of basic functionalities and verified through the early version. The next steps for the project include the development of more complex functionalities and deployment.

---

Key words: web application, Vue, Vuetify, TypeScript, MongoDB, production plan

## SISÄLLYS

|   |                                     |    |
|---|-------------------------------------|----|
| 1 | JOHDANTO .....                      | 6  |
| 2 | PROJEKTIN KUVAUS .....              | 7  |
|   | 2.1 Aloitusilanne .....             | 8  |
|   | 2.2 Tavoitteet .....                | 9  |
|   | 2.3 Lopputulema .....               | 10 |
| 3 | PROJEKTIN KULKU .....               | 11 |
|   | 3.1 Suunnittelu .....               | 11 |
|   | 3.2 Toteutus .....                  | 13 |
| 4 | TYÖKALUT JA TEKNOLOGIAT .....       | 18 |
|   | 4.1 Vue.....                        | 18 |
|   | 4.2 Vuetify .....                   | 18 |
|   | 4.3 vue-chartjs .....               | 19 |
|   | 4.4 TypeScript.....                 | 19 |
|   | 4.5 Node.js .....                   | 20 |
|   | 4.6 Express.js .....                | 20 |
|   | 4.7 MongoDB .....                   | 21 |
| 5 | SOVELLUKSEN KUVAUS .....            | 22 |
|   | 5.1 Näkymät.....                    | 22 |
|   | 5.1.1 Etusivu.....                  | 22 |
|   | 5.1.2 Työvaiheet ja aikataulu ..... | 24 |
|   | 5.1.3 Edistymäseuranta.....         | 24 |
|   | 5.2 S-edistymäkäyrä .....           | 25 |
| 6 | POHDINTA .....                      | 27 |
|   | 6.1 Haasteet.....                   | 27 |
|   | 6.2 Jatkokehitysideat.....          | 28 |
|   | LÄHTEET.....                        | 30 |

**ERITYISSANASTO**

|                      |  |
|----------------------|--|
| API                  | Ohjelmointirajapinta (Application Programming Interface)   |
| Backend              | Sovelluksen palvelinpuoli  |
| CRUD                 | Lyhenne sanoista Create, Read, Update ja Delete, jotka kuvaavat tietokantaoperaatioita: tietojen luonti, haku, päivitys ja poisto            |
| ERP                  | Toiminnanohjausjärjestelmä   |
| Frontend             | Sovelluksen selainpuoli  |
| HTTP-metodit         | Verkkoviestinnän perustoimintoja, kuten GET (noudetaan dataa), POST (lähetetään dataa), PUT (päivitetään dataa) ja DELETE (poistetaan dataa) |
| POC                  | Konseptitodistus (Proof Of Concept)  |
| Single Page App, SPA | Verkkosovellus, joka lataa sisällön yhdelle sivulle ja päivittää näkymät dynaamisesti ilman sivun uudelleenlatausta.                         |
| S-käyrä              | Edistymän kuvaajakomponentti   |
| UI                   | Käyttöliittymä (User Interface)  |

## 1 JOHDANTO

Tässä opinnäytetyössä tarkastellaan konseptitodistuksen merkitystä ja toteutusta verkkopohjaisen tuotantosuunnitelman kehittämisessä, joka on tarkoitettu korvaamaan toimeksiantajalla nykyisin käytössä oleva taulukkopohjainen versio. Konseptitodistus on tärkeä vaihe tuotekehitysprosessissa, sillä sen avulla voidaan arvioida idean toteuttamiskelpoisuutta sekä tunnistaa mahdollisia riskejä ennen siirtymistä laajempaan projektiin. Tämä arviointi mahdollistaa varhaisen päätöksenteon, joka voi estää aikaisessa vaiheessa hankkeiden jatkamisen, mikäli niiden toteutettavuus tai taloudellinen kannattavuus ei ole riittävän vakuuttavaa.

Konseptitodistus ja opinnäytetyö tehdään Pemamek Oy:lle, joka tuottaa hitsaus- ja tuotantoautomaattoratkaisuja raskaalle metalliteollisuudelle. Pemamek on suomalainen yritys, jonka tehdas sijaitsee Loimaalla, mutta sillä on useampi tytäryhtiö muissa maissa sekä asiakkaita ympäri maailmaa. Yrityksen liiketoiminta on siis hyvin kansainvälistä ja laajentuu koko ajan.

Opinnäytetyö keskittyy erityisesti siihen, millainen konseptitodistus toimeksiantajalle tuotettiin sekä miten verkkopohjaisen tuotantosuunnitelman kehittäminen voi parantaa nykyisiä käytäntöjä ja tuoda mukanaan enemmän joustavuutta ja hallintaa. Projektin päätavoitteet liittyvät tiiviisti järjestelmän yhteensopivuuden selvittämiseen sekä kolmeen keskeiseen toiminnallisuuteen: projektien ja laitteiden aikataulutukseen, edistymän seurantaan ja tuotantoedellytysten hallintaan. Tavoitteena on, että verkkopohjainen toteutus ei ainoastaan vastaa nykyisiä tarpeita, vaan myös ylittää taulukkopohjaisten ratkaisujen tarjoamat mahdollisuudet.

## 2 PROJEKTIN KUVAUS

Konseptitodistuksen eli POC:n (Proof of Concept) ensisijaisena tehtävänä on selvittää idean toteuttamiskelpoisuus. Se toimii sekä toteutettavuuden että riskien arvioinnin apuvälineenä, kun pohditaan etenemistä tuotantokehityksen seuraavaan vaiheeseen ja kohti täysimittaista projektia. Konseptitodistuksen ei itsessään tarvitse olla toimivuutta todistava demonstraatio, vaan riittää että konseptitodistus antaa vastauksen tutkimuskysymykseen (Virolainen & Yli-Knuutila, n.d.). Lopputulemana voi olla idean osoittautuminen taloudellisesti kannattamattomaksi tai toteutuspuoleltaan mahdottomaksi, jolloin siitä luopuminen varhaisessa vaiheessa säästää yrityksen aikaa, rahaa ja resursseja. (TechTarget, 2023.)

Opinnäytetyön kohdalla konseptitodistus selvittää, onko mahdollista toteuttaa verkkopohjainen tuotantosuunnitelma käytössä olevan taulukkopohjaisen version tilalle. Onnistumisen osalta keskiössä oli selvittää yhteensopivuus Powered-toiminnanohjausjärjestelmän ja tietokantaratkaisun kanssa sekä arvioida kolmen päätavoitteen toteutettavuutta. Aikarajaksi projektille asetettiin kuusi kuukautta, jonka kuluttua saatuja tuloksia arvioitaisiin yhteistuumin ja keskusteltaisiin projektin tulevaisuudennäkymistä. Projekti soveltui aiheensa puolesta hyvin opinnäytetyöksi, mutta työmäärästä ja projektin laajuudesta johtuen kehitystyöhön etsittiin poikkeuksellisesti kahta tekijää.

Lopullinen ohjelmistoprojekti, joka esiteltiin määräajan päätteeksi, ylittää konseptitodistuksen määritelmän rajat ja sisältää siten toteuttamiskelpoisuuden osoittamisen lisäksi prototyypille ominaisia elementtejä. Tällaisia piirteitä ovat muun muassa tuotantosuunnitelman merkittävimpien perustoiminnallisuuksien kehittäminen ja visuaalisen ilmeen suunnitteleminen. Pidemmälle viedyn projektin esitleminen ja lopullista sovellusta havainnollistava versio auttoi paremmin ymmärtämään ohjelmiston toimintaperiaatteita ja helpotti laadukkaamman palautteen antamista.

## 2.1 Aloitustilanne

Yrityksen käytössä oleva tuotantosuunnitelma oli toteutettu Excel-taulukkolaskentaohjelmalla, joka sijaitsi VPN-yhteydellä saavutettavassa tiedonhallintaohjelmassa. Ajankohtaiset tiedot tilauksista päivittyivät tuotantosuunnitelmaan Powerred-toiminnanohjausjärjestelmästä niille määrättyjen solujen kohdille. Nykyisessä järjestelmässä olevat epäkohdat olivat kuitenkin yrityksen toiminnan kasvamisen myötä vaikeuttaneet sen käyttöä huomattavasti.

Käyttäjän tuli manuaalisesti tallentaa tuotantosuunnitelmatiedostoon tehdyt muutokset tiedonhallintaohjelman kautta, jotta päivitetty tieto ja excel-tiedoston viimeisin versio olisi saatavilla muille yrityksen työntekijöille. Avattua tiedostoa tarkastellessa ei siis voinut olla täysin varma, onko toinen työntekijä samaan aikaan päivittämässä tietoa. Automaattisen ja reaaliaikaisen päivityksen puuttuminen johti usein ylimääräiseen koordinointiin ja vuorovaikutukseen yrityksen sisällä.

Tuotantosuunnitelman rakentaminen taulukkolaskentaohjelman päälle antoi omat rajoitteensa sen rakenteeseen, tiedon järjestyyn ja käyttäjäkokemukseen. Järkevän rakenteen ylläpitämiseksi oli tarpeen toistaa samaa tietoa monen rivin verran, joka tilaustietojen lukumäärän tapauksessa paisui suuriin lukemiin. Solut, joiden tiedot käyttäjän tuli täyttää, sijaitsivat usein kaavasolujen vieressä, jolloin koko tiedosto oli varsin haavoittuvainen käyttäjävirheille ja sitä muokatessa tuli olla erityisen varovainen.

Alkuperäisessä tuotantosuunnitelmassa kaikki tuotantoon liittyvä tieto haettiin sovelluksen käynnistymisen yhteydessä, mikä johti huomattavan pitkäkestoiseen latausaikaan. Tärkeimpien tuotantosuunnitelman toiminnallisuuksien lisäksi tiedoston välilehdet sisälsivät paljon yksityiskohtaista tietoa aikatauluihin, työntekijöiden hienokuormitukseen ja yrityksen tilannetta havainnollistaviin kaavioihin. Laaja tietomäärä heijastui tiedoston heikkoon suorituskykyyn, mikä hidasti tiedon selaamista ja päivittämistä. Tarkempaa edistymäseurantaa tilauksille oli kaavailtu, mutta eriytetty omaksi tiedostokseen suorituskyvyn maksimoimiseksi.

## 2.2 Tavoitteet

Projektin luonteeksi määriteltiin konseptitodistus perinteisen ohjelmistoprojektin sijasta sen tutkivan ja selvittävän tehtävän vuoksi. Toteutettavuuden tutkimisen lisäksi projektille kuitenkin annettiin kolme päätavoitetta, jotka keskittyivät tuotantosuunnitelman perustoiminnallisuuksiin. Nämä kolme tavoitetta olivat projektien ja laitteiden aikataulutus, edistymän seuranta ja tuotantoedellytysten seuranta. Toiminnallisuustavoitteiden katsottiin toimivan järjestelmien yhteensopivuuden selvittämisen lisäksi vähimmäisvaatimuksina sovelluksen käyttöönotolle.

Alun perin toiminnallisuustavoitteita oli neljä, joista työntekijöiden hienokuormitus päätettiin rajata projektin ulkopuolelle sen suuresta mittakaavasta johtuen. Hienokuormitus on siitä huolimatta kolmen tavoitteen tavoin välttämätön edellytys tuotantosuunnitelmalle, jos ohjelma halutaan kehittää osaksi yrityksen päivittäistä toimintaa. Tavoitteeseen pätee samat yhteensopivuusselvitykset, jotka kuitenkin saadaan todistettua muiden tavoitteiden myötä.

Projektin epäsuorina tavoitteina voidaan pitää Excel-taulukkolaskentaohjelman päälle rakennetun tuotantosuunnitelman epäkohtien ja puutteiden korjaamista. Verkkopohjaiseen toteutukseen siirtyessä moni näistä ongelmista saadaan ratkottua kohtalaisen luontevasti ja vähällä vaivalla. Yksi suurimmista tekijöistä oli tuotantosuunnitelman käyttöliittymän suunnittelemisen paremman käyttäjäkokemuksen ympärille verkkopohjaisen toteutuksen tarjoamien ratkaisujen avulla.

Kuten edellä on todettu, projektin painopiste säilyi toteutettavuuden todistamisessa, vaikka aloituspalaverissa määriteltiin myös toiminnallisia tavoitteita tuotantosuunnitelmalle ja verkkopohjaisen toteutuksen hyödyt käytiin läpi. Minimitoiminnallisuudet täyttävä ohjelmisto toimisi lisävahvistuksena konseptitodistukselle ja todistaisi valittujen teknologioiden yhteensopivuuden yrityksen toiminnanohjausjärjestelmän kanssa.

## 2.3 Lopputulema

Ohjelmistoprojektin toimintaa demonstroimalla onnistuttiin todistamaan verkkopohjaisen toteutuksen soveltuvuus parempana vaihtoehtona alkuperäiseen verrattuna. Verkkopohjaisessa toteutuksessa useampi kuin yksi henkilö voi muokata ja tarkastella reaaliaikaista tietoa samaan aikaan, jolloin tiedon ajantasaisuudesta ja paikkansapitävyydestä voidaan olla varmoja. Automaattisesti päivittyvä sovellus vapauttaa työntekijöiden aikaa keskittyä muihin työtehtäviin.

Tuotantosuunnitelman käyttöliittymä on selkeämpi, yksinkertaisempi ja helppokäyttöisempi, kun se ei rajoitu isoon taulukkonäkymään ja taulukkolaskentaohjelman toiminnallisuuksiin. Sovellus on tehokkaammin suojattu käyttäjävirheiltä, kun muokkausmahdollisuuksien hallintaan ja rajaamiseen voidaan vaikuttaa eivätkä logiikan toteuttavat kaavat ole käyttäjän täytettävien solujen läheisyydessä.

Erillinen tietokantaratkaisu verkkopohjaisessa toteutuksessa mahdollistaa tietomäärän jaottelun ja noutamisen tarpeen. Uudessa versiossa tiedon lataaminen ja hakeminen voidaan paloitella pienempiin kokonaisuuksiin siten, että tieto latautuu käyttäjän navigoinnin mukaisesti. Tiedon tarpeeton toistuvuus voidaan myös estää, mikä vapauttaa ohjelmiston resursseja muihin tehtäviin.

### 3 PROJEKTIN KULKU

Projekti eteni melkein koko kehityksen ajan niin, että suunnittelu ja toteutus kulivat käsi kädessä. Aluksi ei siis tehty varsinaista suunnitelmaa koko projektin ajalle vaan toteutettiin aina sen verran, mitä tiedossa oli ja suunniteltiin projektia eteenpäin siinä toteutuksen ohella.

#### 3.1 Suunnittelu

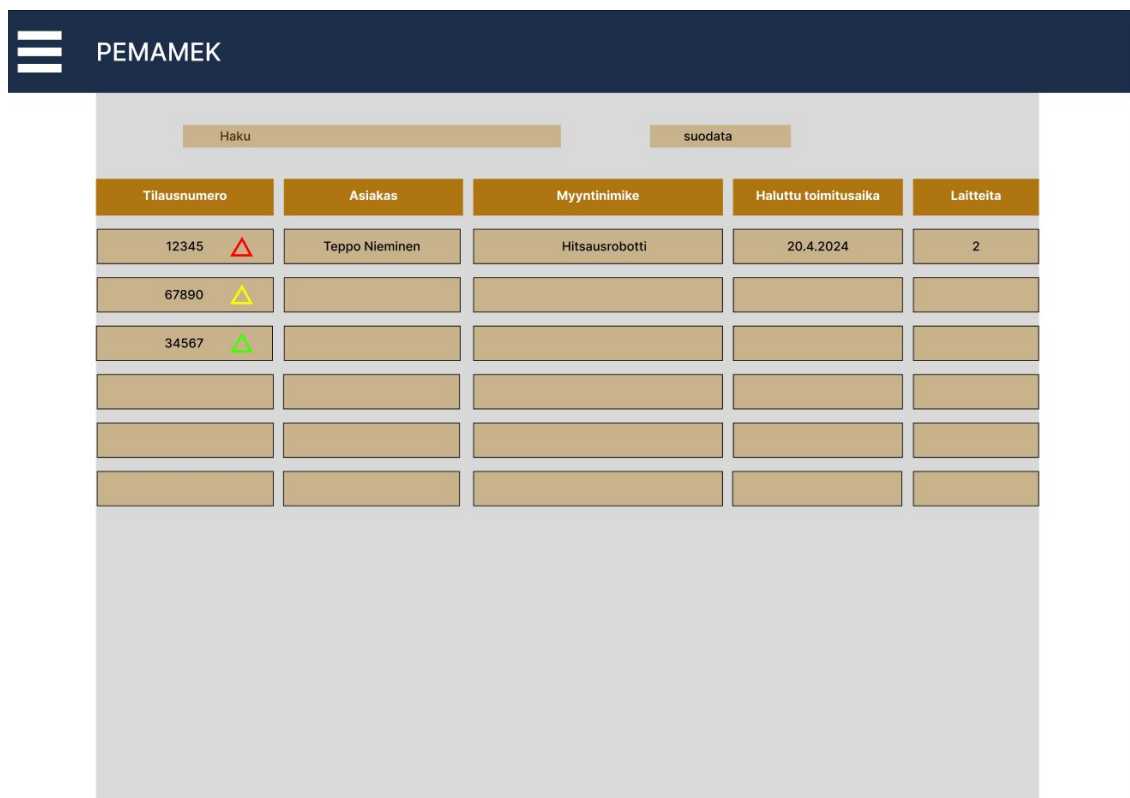
Projekti käynnistyi helmikuussa aloituspalaverilla, jossa työryhmän kanssa käytiin läpi suunnitteluvaiheelle ominaiset asianhaarat. Näitä ovat muun muassa toteutukseen liittyvät tehtävät ja tarvittavat resurssit (Artto, Martinsuo & Kujala, 49). Projektille ei määritelty yksityiskohtaista tehtäväälistaa tavoitteiden saavuttamiseksi, mikä antoi enemmän vapautta ja vastuuta tekijöille työn etenemiseen liittyen. Havainnollistavana materiaalina ja paremman käsityksen saamiseksi tekijöille näytettiin esimerkkikuva sovelluksen mahdollisesta ulkonäöstä, jonka avulla toiminnalliset tavoitteet ja asiakkaan toivomukset selitettiin auki.

Pääasiallisena resurssina projektille oli kaksi tekijää, jotka työskentelivät sen parissa kuuden kuukauden ajan lähes täysipäiväisesti. Tekijöiden avuksi laadittiin kattava lista, joka sisälsi yrityksen IT-puolen työntekijöiden yhteystiedot ja erityiset osaamisalueet. Osaamisalue oli joko yleiseen ohjelmointiin tarkoitettu apu tai asiantuntijuus yhteen yrityksen sisäisistä järjestelmistä kuten toiminnannohjausjärjestelmästä. Yhteydenotot tapahtuivat yleensä joko sähköpostitse tai etäpalaverin muodossa.

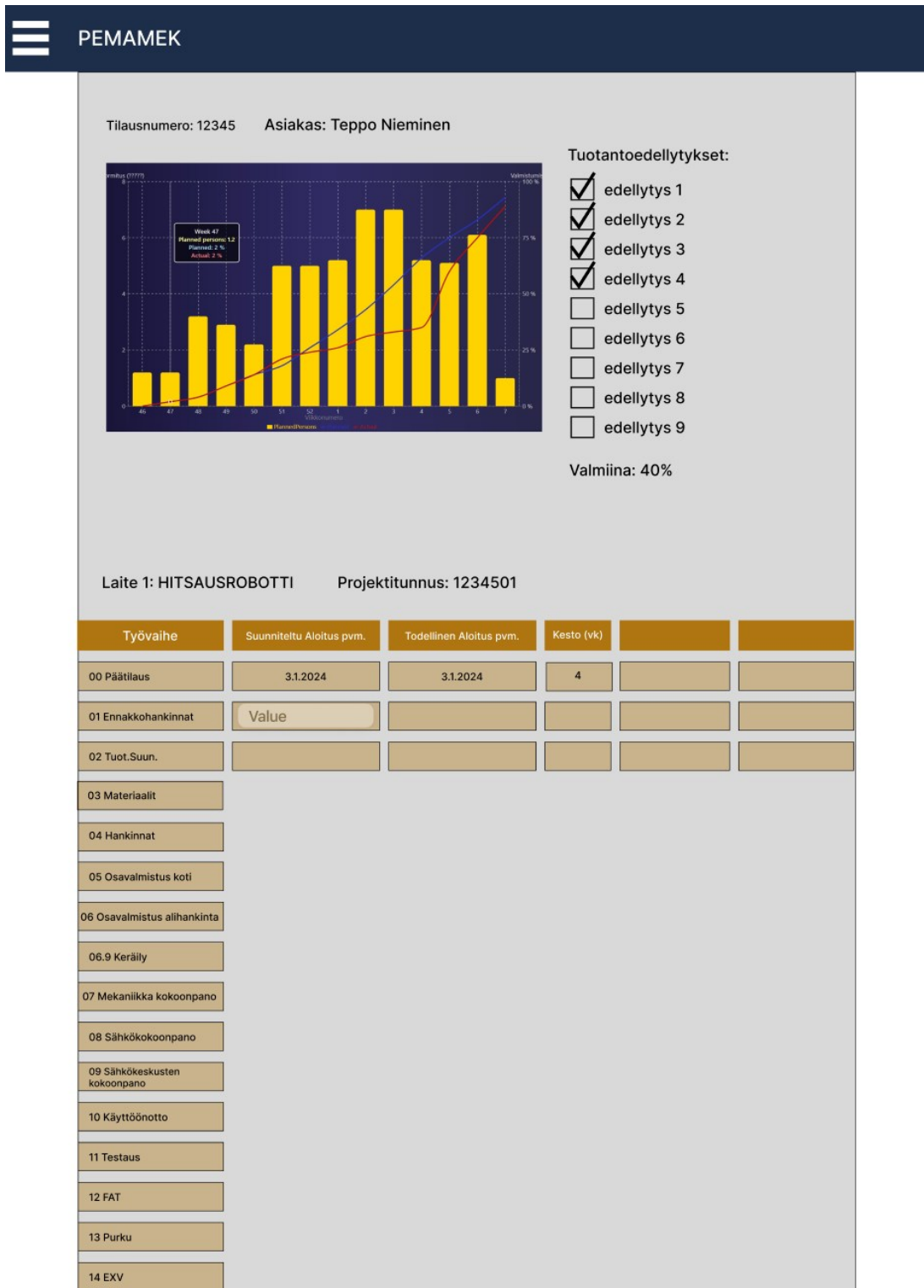
Projekti alkoi yrityksen nykyiseen tuotantosuunnitelmaan ja edistymäseurantaan tutustumisella. Alussa oli tärkeää ymmärtää niiden toiminnot mahdollisimman hyvin, jotta voitiin kehittää niistä uusi versio. Tuotantosuunnitelma-Excelin tutkiminen ja ymmärtäminen vei jonkin verran aikaa sen laajuuden sekä monimutkaisuuden vuoksi. Tuotantosuunnitelma on myös kytköksissä Pemamekillä käytössä olevaan Powered-toiminnanohjausjärjestelmään, jonka sisällä on lähtökohteisesti kaikki tärkein ja viimeisin tieto tilauksista ja projekteista. Oli alusta asti

selvää, että Poweredista haluttaisiin hakea tietoja myös uudessa verkkosovellusversiossa ja ehkä jopa laajemminkin, kuin aiemmin. Ennen Poweredin toiminnan ymmärtämistä oli vaikea sanoa, onnistuisiko integraatio koodin kanssa ollenkaan. Siksi yrityksen ohjelmistoihin, kuten Powerediin ja erityisesti sen raporttigeneraattoriin, tutustuminen oli hyvin tärkeää. Raporttigeneraattorilla pystytään hakemaan kaikki halutut tiedot eri rajausmenetelmin projekteista ja saamaan ne ulos Excel-tilukossa. Kun tämä selvisi ja raporttigeneraattoria testattiin muutaman kerran, todettiin, että Powered-integraatio onnistuu hyvin myös uuden tuotantosuunnitelman kanssa.

Koska tuotantosuunnitelman ja edistymäseurannan, Poweredin sekä muiden ohjelmistojen ymmärtäminen vei aikaa ja oli vielä hieman epäselvää, mitä sovelluksessa tulisi olla, tehtiin välissä käyttöliittymäsuunnittelua. Suunnittelun tavoitteena oli selkeyttää projektin tavoitteita ja visualisoida, mitä kaikkea sovellukseen haluttiin ja miltä se voisi näyttää. Suunnitelma tehtiin Figmalla. Figma on sovellus, joka on kehitetty käyttöliittymäsuunnittelua ja prototyyppien tekoa varten. Tehty prototyyppi on kuvissa 1 ja 2. Prototyyppiä esiteltiin myös seurantapalaverissa, jossa siitä saatiin palautetta ja varmistettiin, että tavoitteet ymmärrettiin oikein.



KUVA 1. Figmalla tehty havainnekuva sovelluksen etusivusta.



KUVA 2. Figmalla tehty havainnekuva sovelluksen laitekohtaisesta näkymästä.

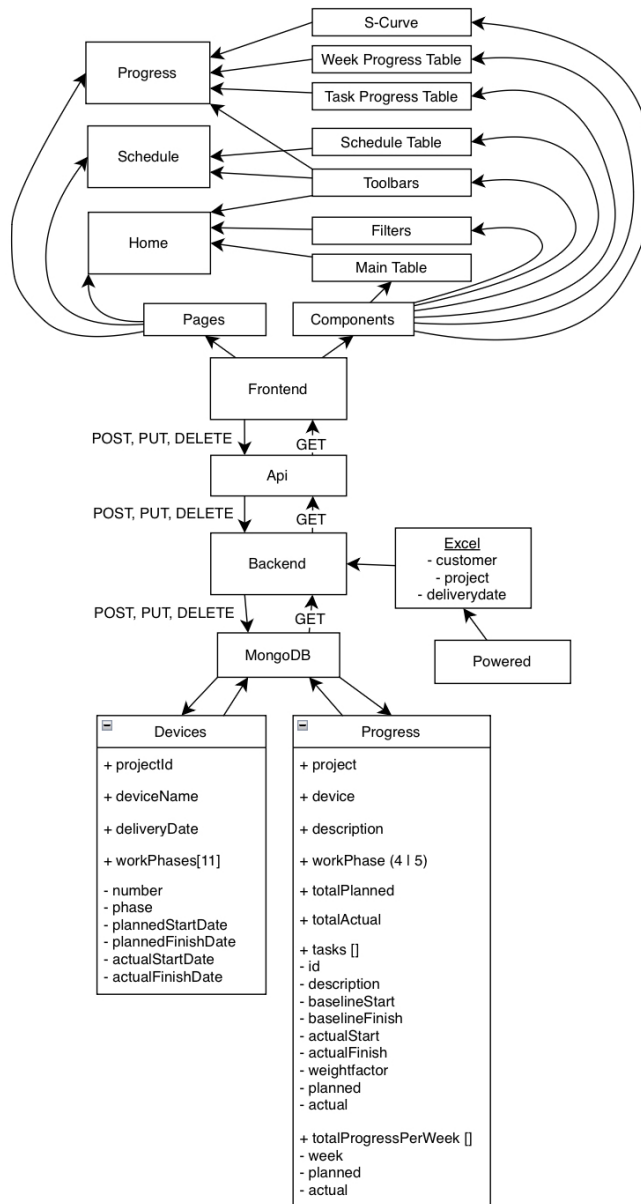
### 3.2 Toteutus

Yleisesti projektin toteutusvaiheessa tarkennetaan projektitiimin keskinäiset vastualueet ja tehtävien ja niihin liittyvien töiden sisältö (Arto ym. 49). Luodun ohjelmistoprojektin tapauksessa tämä tarkoittaa kahden tekijän välistä työnjakoa ja

tehtävien tarkempaa määrittelemistä. Varsinaisessa ohjelmoinnissa vastualueet jakautuivat suurelta osin backendin eli palvelinpuolen ja frontendin eli selainpuolen kehitykseen, mutta järjestelystä joustettiin tilanteen mukaan ja vuoroin tekijät keskittyivät yhdessä samaan tehtävään. Esimerkiksi palvelinpuolen kehitystyön ohella tapahtuva selvitystyö järjestelmien yhteensopivuudesta oli molempien tekijöiden vastuulla.

Kun käytettävät teknologiat saatiin päätettyä yrityksen kanssa, aloitettiin itse koodaaminen. Ensimmäisenä lähdettiin liikkeelle selainpuolesta, jota tehtiin prototyypin pohjalta. Koska datan saanti Poweredista ei vielä ollut selvää, käytettiin selainpuolen kehityksessä esimerkkidataa. Käyttöliittymää pystyttiin kehittämään melko pitkälle ilman palvelinpuolta. Pelkällä selainpuolella tehtiin muun muassa etusivu, jossa oli tilauksia listattuna, sekä sivu edistymäseurannalle, jossa oli S-käyrä. Myös yläpalkit ja hakutoiminto luotiin jo tässä vaiheessa. Kuvio 1 havainnollistaa sovelluksen teknistä arkkitehtuuria. Kuviosta nähdään, että selainpuoli on jaettu sivuihin ja komponentteihin, ja nämä sivut koostuvat tietyistä komponenteista. Komponentit ovat lähinnä erilaiset taulukot, yläpalkit, suodatukset ja S-käyrä.

Palvelinpuolen kehitys alkoi heti, kun Poweredin raporttigeneraattorin käyttö selkiytyi ja saatiin päätettyä, mitä tietoja sieltä halutaan hakea. Kuviosta 1 nähdään, mitä tietoja Poweredista tuotiin Excelin kautta palvelinpuolen käyttöön, jolloin sitä dataa pystyttiin käyttämään myös selainpuolella esimerkkidatan sijaan. Nyt saatiin sovelluksen etusivulle, kuviossa Home, näkymään oikeita asiakkaita ja tilauksia. Koska jokaisella projektilla tuli olla työvaiheet omassa näkymässä, luotiin ne palvelinpuolella ja lisättiin ne myös lokaaliin tietokantaan, jotta niitä pääsisi muokkaamaan ja tarkastelemaan. Kuviosta näkee, mitä tietojäseniä jokaisella projektilla tietokannan kokoelmassa Devices on. Nyt voitiin myös luoda työvaihenäkymä, kuviossa Schedule, jokaiselle projektille. Alkuperäisen edistymäseuranta-Excelin toiminnallisuudet ja kaavat luotiin palvelinpuolelle viimeisimpänä, jonka jälkeen selainpuolelle edistymänäkymään, kuviossa Progress, saatiin taulukot sekä toimiva S-käyrä, joiden tiedot olivat muokattavissa ja tallentuivat myös tietokantaan. Kuviosta tietokannan kokoelmassa Progress näkyy edistymään liittyvät tietojäsenet, joista edistymänäkymän taulukoiden sekä S-käyrän tiedotkin koostuvat. Selainpuolen eri näkymistä kerrotaan lisää luvussa 5.



KUVIO 1. Havainnollistava kuvio teknisestä arkkitehtuurista.

Kuviossa 1 nuolet havainnollistavat hieman, miten ja mihin suuntiin tieto sovelluksessa liikkuu. Sovelluksessa on käytössä kaikki neljä perustoimintoa eli CRUD-toiminnot. Lyhenne CRUD tulee sanoista create, read, update ja delete. Kun sovelluksessa käytetään näitä toimintoja, se tarkoittaa, että tietoja voidaan luoda, lukea, päivittää sekä poistaa. Kuviossa nämä näkyvät HTTP-metodeina POST, GET, PUT ja DELETE. Tieto sovelluksessa liikkuu siis siten, että käyttäjä tekee selainpuolella pyynnön, joka on joko datan haku, lisäys, päivitys tai poisto, jolloin se lähetetään sopivana metodina API:n kautta palvelinpuolelle. API on

tässä tapauksessa rajapinta, joka auttaa selain- ja palvelinpuolen välisessä kommunikoinnissa sekä datanvaihdossa. Palvelinpuolella tehdään kommunikointi tietokannan kanssa, josta tieto haetaan tai jonne se lähetetään, ja tehdään mahdolliset muokkaukset tai toiminnot, ennen kuin lähetetään selainpuolen pyyntöön vastaus.

Koska kyseessä oli konseptitodistus, jonka yhteydessä syntyy pienimuotoinen ohjelmistoprojekti, erilliselle versionhallinnan pystytykselle ei nähty yrityksen puolesta tarpeeksi perusteita. Versionhallinnan mahdollisuutta päätettiin katsoa yhdessä uudelleen mikäli konseptitodistuksen tulokset todettaisiin kannattaviksi ja tuotantosuunnitelma siirtyisi jatkokehitykseen. Ohjelmistoprojektin kokoluokka kasvoi odotettua laajemmaksi, jolloin versionhallinnasta olisi ollut hyötyä alusta alkaen. Kaikki ajankohtaiset ohjelmistokoodit olisivat löytyneet yhdestä sijainnista, jolloin tiedon hakeminen ja säilöminen olisi avustanut kooditiedostojen yhdistämisessä ja ohjelmistoprojektin aikaisempien versioiden palauttamisessa.

Työskentely projektin parissa oli melko vapaamuotoista. Koska versionhallintaa ei ollut saatavilla, tekijät turvautuivat tiiviiseen yhteistyöhön ja vilkkaseen kommunikointiin varmistuakseen tiedon yhtenäisyydestä ja projektin tilasta. Tarpeen vaatiessa tekijät pitivät keskenään kokouksia puhuakseen käsillä olevista työtehtävistä ja niiden toteutumisesta. Ohjelmointikoodia jaettiin tekijöiden kesken yrityksen Teams-alustaa käyttäen. Lyhyemmät ohjelmointikoodit, kuten funktio- tai komponenttikoodit jaettiin tekstitiedostona ja isommat muutokset zip-kansioina alustan keskustelutoimintoa hyödyntäen.

Seurantapalavereja pidettiin aina, kun sellainen koettiin tarpeelliseksi. Yleensä seurantalaverien tarkoituksena oli esitellä sovellusta ja siihen tehtyjä uusia toimintoja sekä keskustella jatkosta ja seuraavista työvaiheista. Oikeaa backlogia eli ns. prioriteettilistaa ei käytetty, joten työvaiheet ja tavoitteet kirjattiin ylös palaverimuistiinpanoihin ja jaettiin myöhemmin työntekijöiden kesken. Kehitys oli täysin irrallaan yrityksen operatiivisesta toiminnasta, sillä kyseessä oli konseptitodistus, eikä siksi esimerkiksi testaus ollut niin suuressa roolissa. Sovellusta kehitettiin pääosin tuotantojohtajan kanssa, eikä työryhmässä ollut juuri muita tekijöitä mukana. Yhdessä seurantalaverissa, jossa sen hetkistä sovellusta esiteltiin,

oli mukana työntekijöitä myös tuotannon puolelta. Heiltä saatiin palautetta ja mielipiteitä sovelluksen toimintoihin sekä käyttöliittymään liittyen. Työryhmä todennäköisesti kasvaa jatkokehityksessä ja mukaan tulee myös varsinainen testaaminen.

## 4 TYÖKALUT JA TEKNOLOGIAT

Sovelluksen kehityksessä käytettiin monia moderneja työkaluja ja teknologioita, jotka mahdollistivat joustavan ja tehokkaan ohjelmoinnin. Näiden teknologioiden valinta oli keskeinen osa projektin toteutusta, sillä ne tarjosivat tarvittavat ominaisuudet sekä selain- että palvelinpuolen kehittämiseen. Teknologiaopinolla oli suuri merkitys ohjelmiston suorituskyvyn, skaalautuvuuden ja käyttäjäkokemuksen kannalta. Tämän luvun aikana tarkastellaan käytettyjen työkalujen ja teknologioiden keskeisiä ominaisuuksia sekä niiden roolia sovelluksen toiminnallisuuden toteuttamisessa.

### 4.1 Vue

Vue on vuonna 2014 julkaistu avoimen lähdekoodin JavaScript-viitekehys käyttöliittymien kehittämiseen. Se on saanut alkunsa pyrkimyksestä yhdistää tunnetuista viitekehysistä parhaimmat ominaisuudet yhteen kehykseen painottaen samalla yksinkertaisuutta ja helppokäyttöisyyttä. Vuelle tunnusomaisina ja sen suosiota selittävinä piirteinä voidaan pitää aloittelijaystävällistä käyttökokemusta, perusteellista dokumentaatiota ja joustavuutta hallita niin pieniä kuin suuria ohjelmistoprojekteja. (Monterail, 2024.)

### 4.2 Vuetify

Vuen tueksi on kehitetty ilmainen komponenttikirjasto Vuetify, joka tarjoaa kattavan valikoiman yleisimmin käytettyjä UI-komponenttiobjekteja. Tällaisia komponentteja ovat muun muassa napit, alasvetovalikot ja tekstinsyöttökentät. Valmiskomponenttiratkaisut vauhdittavat sovelluskehitystyötä, kun kehittäjän ei ole tarpeen rakentaa komponentteja tyhjästä ja niiden logiikkaan voi vaikuttaa sisäänrakennettujen funktioiden kautta. (Vuetify, 2024).

Vuetifyn UI-komponentit ovat oletukseltaan jo tyyliteltyjä, mutta niiden ulkoasua voidaan muokata pidemmälle attribuuttien, luokkien ja CSS-koodin kautta. CSS eli Cascading Sheets Style on yleisessä käytössä oleva ohjelmointikieli, jonka

avulla voidaan muun muassa muokata elementtien värejä, kokoa, reunoja ja sijaintia (MDN Web Docs, n.d.).

Vuetifyn tarjoamia UI-komponentteja hyödynnetään ohjelmistoprojektin selainpuolella käytännöllisesti katsoen aina kun mahdollista. Monimutkaista logiikkaa noudattavat komponentit on rakennettu yhdistämällä kahta tai useampaa valmis-komponenttia ja koodaamalla niiden välille yhteinen toiminnallisuus.

### 4.3 vue-chartjs

vue-chartjs-kirjasto on Vuen viitekehukseen sopeutettu versio suositusta Chartjs-kirjastosta. vue-chartjs on datan havainnollistamiseen ja erilaisten diagrammien piirtämiseen tarkoitettu avoimen lähdekoodin kirjasto ja valittiin työkaluksi sen tarjoaman kätevyyden ja teknologisen yhteensopivuuden vuoksi.

Kirjastoa hyödynnettiin projektissa S-edistymäkäyrän ohjelmointiin, jotta Excel-kaavio alkuperäisestä tuotantosuunnitelmasta voitiin rakentaa osaksi verkkosovellusta. Kirjasto soveltui tehtävään erityisen hyvin, koska sen avulla voi toteuttaa usean erityyppisten kuvaajien piirtämisen samaan ikkunaan. Projektin edessä usean kuvaajan mallista kuitenkin luovuttiin kuormituksen jäädessä prioriteettilistan ulkopuolelle, mutta mahdollista jatkokehitystä ajatellen kirjaston tiedetään taipuvan vaativampiinkin kuvaajanäkymiin.

### 4.4 TypeScript

TypeScript on JavaScript-ohjelmointikielestä juonnettu kieli, jolle ominaisinta on staattinen tyyppitys ja sen käänösominaisuus. JavaScriptissä erityyppiset muuttajat voivat erehdyttävästi käyttäytyä samalla tavalla ohjelmassa ja sallia ohjelman ajon, mikä virheiden lopulta ilmetessä vaikeuttaa niiden alkuperän löytämistä. TypeScriptin staattinen tyyppitys ratkaisee ongelman raportoimalla ohjelmoijalle tyyppivirheistä ajon ulkopuolella ja tyyppitystä korostava ohjelmakoodi helpottaa työskentelyä. Ajettava koodi säilyy JavaScriptinä, jonka kääntäjä muuttaa TypeScript-syötteestä. (TypeScript, 2024.)

Nykyohjelmistoprojektien kompleksisuuden ja kokoluokan kasvaessa TypeScript kehitettiin ohjelmistokehittäjien avuksi tunnistamaan tyyppivirheitä ajon ulkopuolella ja helpottamaan isompien koodikokonaisuuksien ohjelmointia. Koska ajettava koodi on JavaScriptiä, TypeScript on yhteensopiva teknologioiden kanssa, jotka ovat yleisesti tunnetuimpia ja edistyneempiä. Nämä kaksi ominaisuutta selittävät kielen käyttösuosiota. (TypeScript, 2024.)

## 4.5 Node.js

Sovelluksen palvelinpuoli on luotu käyttäen Node.js-ajoympäristöä ja Express-sovelluskehystä. Node.js on avoimen lähdekoodin JavaScript-ajoympäristö, ja se on myös alustariippumaton. Sitä käytetään usein luotaessa palvelinpuolen verkkosovelluksia, mikä tarkoittaa, että suurin osa sovelluksen logiikasta, tiedonkäsittelystä, laskennoista ja tietokantakyselyistä suoritetaan palvelimella käyttäjän selaimen sijaan. Node.js on hyvin suosittu ajoympäristö muun muassa sen hyvän suorituskyvyn ansiosta. Se on myös asynkroninen, eli se pystyy käsittelemään useita pyyntöjä samanaikaisesti pysähtymättä odottamaan yksittäisen tehtävän, kuten tietokantakyselyn tai API-kutsun, valmistumista. Tämän ansiosta Node.js hyödyntää tehokkaasti resurssinsa ja soveltuu erityisen hyvin reaaliaikaisten ja paljon dataa käsittelevien verkkosovellusten rakentamiseen. (Node.js n.d.)

## 4.6 Express.js

Express.js on Node.js:n päälle rakennettu nopea ja minimalistinen sovelluskehys, joka tarjoaa kehittäjille valmiit työkalut ja rakenteet verkkosovellusten ja API:en rakentamiseen. Se laajentaa Node.js:n perustoimintoja, kuten HTTP-pyyntöjen ja -vastausten käsittelyä, ja tuo mukanaan monia lisäominaisuuksia. Näitä ovat muun muassa reititys, jonka avulla sovellukseen voidaan määritellä selkeät reitit eri toiminnoille, middleware-hallinta, jolla voidaan käsitellä pyyntöjä ja vastauksia eri vaiheissa, sekä tehokas virheenkäsittelymekanismi. (Express.js n.d.)

Express.js:n suosio perustuu sen yksinkertaisuuteen, joustavuuteen ja käyttäjäystävällisyyteen, mikä tekee siitä erityisen sopivan niin pienille projekteille kuin laajoille, skaalautuville verkkosovelluksille. Vaikka sekä Express.js että Node.js

käyttävät pääasiassa JavaScriptiä, ne tukevat myös TypeScriptiä, mikä lisää kehitysprojektien tyyppiturvallisuutta ja ylläpidettävyyttä. Express.js:n ja Node.js:n yhdistelmä tarjoaa tehokkaan ja modernin pohjan palvelinpuolen kehitykselle.

#### **4.7 MongoDB**

MongoDB on dokumenttipohjainen tietokanta, joka käyttää joustavaa skeemaa tietojen tallentamiseen. Tämä tarkoittaa, että tietueiden rakenne voi vaihdella: eri tietueilla voi olla erilaiset kentät tai rakenteet, toisin kuin perinteisissä relaatiotietokannoissa, joissa kaikilla tietueilla on oltava täsmälleen sama rakenne. Tiedot tallennetaan BSON-muodossa (binäärinen JSON) ja voidaan noutaa helposti luettavassa JSON-muodossa. MongoDB-dokumentit voivat sisältää myös sisäkkäisiä dokumentteja, mikä lisää tietokannan joustavuutta. Se on suunniteltu skaalautuvaksi ja soveltuu hyvin suurten tietomäärien käsittelyyn. MongoDB on erityisen suosittu kehittäjien keskuudessa sen monipuolisuuden ja monenlaisiin soveluksiin soveltuvuuden ansiosta. (MongoDB n.d.)

## 5 SOVELLUKSEN KUVAUS

Ohjelmisto toteutettiin yksisivuisena verkkosovelluksena (Single Page Application, SPA), joka tarjoaa selkeän ja dynaamisen käyttöliittymän tuotantosuunnitelmaa varten. Sovellus koostuu kolmesta keskeisestä näkymästä, jotka on suunniteltu tukemaan eri tuotannonhallinnan työvaiheita. Käyttäjälle tarjotaan selkeät työkalut tilausten selaamiseen, aikataulutukseen ja edistymän seuraamiseen. Yhtenäinen yläpalkki toimii navigoinnin pääelementtinä, ja näkymät on optimoitu ensisijaisesti tietokoneen työpöydälle, vaikka responsiivisuus mahdollistaa sovelluksen käytön myös erikokoisilla näytöillä. Sovelluksen suunnittelussa hyödynnettiin Vuetify-komponenttikirjastoa, mikä nopeutti kehitystä ja varmisti yhtenäisen ulkoasun.

### 5.1 Näkymät

Sovelluksen näkymät on jaettu kolmeen pääosioon: etusivu tilausten yleiskuvaa ja selaamista varten, työvaihenäkymä yksittäisten laitteiden aikataulujen tarkempaa hallintaa varten ja edistymäseurantasivu kokoonpanovaiheiden seurantaa varten. Jokainen näkymä on suunniteltu selkeäksi ja käyttäjäystävälliseksi, mikä mahdollistaa tehokkaan työskentelyn. Kaikissa näkymissä hyödynnetään Vuetifyn komponentteja, kuten responsiivista taulukkokomponenttia, joka skaalautuu käyttäjän näytön tai selainikkunan koosta riippuen.

#### 5.1.1 Etusivu

Sovellus avautuu etusivulle, jossa näkyvät keskeneräisten tilausten tilausnumerot ja asiakkaiden nimet Vuetify-taulukossa (KUVA 3). Tilauksen voi laajentaa klikkaamalla tilausnumeroa tai asiakkaan nimeä, jolloin esiin tulevat tilaukseen kuuluvien laitteiden nimet, halutut toimitusajat ja projektinumerot (KUVA 4). Näin näkymä pysyy selkeänä ja eri tilaukset on helppo erottaa toisistaan.

| Order | Customer       |   |
|-------|----------------|---|
| 12345 | Reijo Saarinen | ⚠ |
| 67899 | Miska Olamo    | ⚠ |

KUVA 3. Päätaulukon tiivistetty näkymä.

Koska etusivu kerää kaikki käynnissä olevat tilaukset, tiedon suodattaminen on oleellinen osa halutun tilauksen löytämiseksi. Etusivun yläpalkissa on haku-kenttä, josta voi hakea tiettyä tilausnumeroa tai asiakasta. Taulukon esittämien tietojen määrää pystyy hallitsemaan paginoinnin avulla. Yläpalkista löytyy myös suodatinpainike, jolla voi suodattaa näkyviin 82-päätteiset projektit eli asennukset, joita asiakas halusi eritoten korostaa. Painikkeella voi myös suodattaa esiin kaikki sellaiset laitteet, joiden toimitusaika on seuraavan viikon sisällä sekä tuotantosuunnitelmaan tulleet uudet tilaukset.

| Order        | Customer       |                                     |                    |
|--------------|----------------|-------------------------------------|--------------------|
| 12345        | Reijo Saarinen | ⚠                                   |                    |
| Deliverydate | Project id     | Project/Device                      |                    |
| 24.9.2024    | 1234501        | HITSAUSTORNI                        |                    |
| 24.9.2024    | 1234502        | KOKOONPANOASEMA                     |                    |
| 24.9.2024    | 1234503        | RULLASTOSARJA                       |                    |
| 24.9.2024    | 1234504        | KÄSITTELYPÖYTÄ                      | ⚠ Missing progress |
| 24.9.2024    | 1234505        | HITSAUSROBOTTIPORTAALI              | ⚠ Missing progress |
| 30.10.2024   | 1234506        | LEVYN JATKOHITSAUSASEMA             | ⚠ Missing progress |
| 67899        | Miska Olamo    | ⚠                                   |                    |
| Deliverydate | Project id     | Project/Device                      |                    |
| 27.12.2023   | 6789901        | LEIKKAUSLAITE                       | ⚠ Missing progress |
| 27.12.2023   | 6789902        | ALAIMUPOLTTOLEIKKAUSPÖYTÄ           | ⚠ Missing progress |
| 10.2.2024    | 6789903        | PROFIILIN ASENNUS- JA HITSAUSPORTAA | ⚠ Missing progress |
| 10.3.2024    | 6789904        | PROFIILIN KASETTIASEMA              | ⚠ Missing progress |
| 17.4.2024    | 6789905        | PROFIILINHITSAUSPORTAALI            | ⚠ Missing progress |
| 17.3.2024    | 6789906        | JÄYKISTEEN ASENNUSPORTAALI          | ⚠ Missing progress |
| 17.3.2025    | 6789907        | HITSAUSROBOTTIPORTAALI              | ⚠ Missing progress |

KUVA 4. Päätaulukko avatussa näkymässä.

## 5.1.2 Työvaiheet ja aikataulu

Etusivulla voi klikata myös yksittäisen laitteen nimeä tai projektinumeroa, jolloin avautuu työvaihenäkymä (KUVA 5). Tässä näkymässä on taulukko, jossa on listattuna kaikki 11 työvaihetta. Taulukko on aluksi tyhjä, mutta siihen voi täyttää suunnitellut ja todelliset aloitus- ja lopetuspäivämäärät sekä työvaiheen edistysprosentin. Edistysprosentin taustaväri muuttuu punaisesta (0-49 %) oranssiksi (50-99 %) ja vihreäksi (100 %), kun työvaihe on valmis. Työvaihenäkymän yläpalkissa on tilausnumeron ja asiakkaan lisäksi valitun laitteen nimi ja projektinnumero sekä "laittekaruselli", jonka avulla voi siirtyä saman tilauksen muiden laitteiden työvaihenäkymiin.

| Workphase number | Workphase                 | Remark | Planned start | Actual start | Planned finish | Actual finish | Progress % |
|------------------|---------------------------|--------|---------------|--------------|----------------|---------------|------------|
| 1                | Ennakkohankinnat          |        | 08.07.2024    | 08.07.2024   | 11.07.2024     | 11.07.2024    | 100 %      |
| 2                | Tuotantosunnittelu        |        | 08.07.2024    | 08.07.2024   | 09.07.2024     | 10.07.2024    | 100 %      |
| 3                | Keräily                   |        | 11.07.2024    | 11.07.2024   | 15.07.2024     | 15.07.2024    | 100 %      |
| 4                | Mekaniikkakokoonpano      |        | 29.07.2024    | 29.07.2024   | 23.08.2024     |               | 48 %       |
| 5                | Sähkökokoonpano           |        | 29.07.2024    | 29.07.2024   | 23.08.2024     |               | 60 %       |
| 6                | Sähkökeskusten kokoonpano |        | 26.08.2024    |              | 06.09.2024     |               | 0 %        |
| 7                | Käyttöönotto              |        | 09.09.2024    |              | 11.09.2024     |               | 0 %        |
| 8                | Testaus                   |        | 12.09.2024    |              | 16.09.2024     |               | 0 %        |
| 9                | FAT                       |        | 17.09.2024    |              | 17.09.2024     |               | 0 %        |
| 10               | Purku                     |        | 19.09.2024    |              | 20.09.2024     |               | 0 %        |
| 11               | EXW                       |        | 23.09.2024    |              | 24.09.2024     |               | 0 %        |

KUVA 5. Aikataulunäkymä.

## 5.1.3 Edistymäseuranta

Työvaihetaulukossa vaiheet 4 (mekaniikkakokoonpano) ja 5 (sähkökokoonpano) ovat tarkemmin seurattavia, joten niitä klikkaamalla pääsee edistymäseurantaan. Edistymäseurannassa seurataan vaiheiden alikokoonpanojen edistystä. Sivulla voi lisätä alikokoonpanoja taulukkoon manuaalisesti. Lisätyt alikokoonpanot näkyvät molemmissa kokoonpanovaiheissa, sillä alikokoonpanot ovat aina samat laitteen molemmilla kokoonpanovaiheilla 4 ja 5. Edistymäsivulla on taulukko, jo-

hön alikokoonpanojen ID ja kuvaus ilmestyvät käyttäjän syötettyä tiedot manuaalisesti lomakkeeseen. Sekä taulukko että lomake näkyvät kuvassa 6. Taulukkoon voi myös kirjata suunnitellut ja todelliset aloitus- ja lopetuspäivät, painoarvot sekä todelliset edistysprosentit. Taulukossa on myös sarake suunnitelluille edistysprosentteille, jotka ohjelma laskee kaavalla jokaiselle alikokoonpanolle, kun suunnitellut päivämäärät sekä painoarvot on kirjattu. Alimmainen rivi taulukossa on ”total”-rivi, joka näyttää aikaisimman aloituspäivän, viimeisimmän lopetuspäivän, painoarvot yhteensä sekä koko edistymän suunnitellun ja todellisen edistysprosentin. Sivun alaosassa on s-edistymäkäyrä, josta kerrotaan lisää kappaleessa 5.2.

← Back to schedule
1234501

Progress for HITSAUSTORNI - Mekaniikkakokoonpano

**Activities:**

| ID            | Description     | Baseline Start    | Baseline Finish   | Weight Factor % | Planned %  | Actual %   | Actual Start | Actual Finish |
|---------------|-----------------|-------------------|-------------------|-----------------|------------|------------|--------------|---------------|
| 1234501A      | Jalusta         | 29.07.2024        | 01.08.2024        | 10%             | 100%       | 100%       | 29.07.2024   | 01.08.2024    |
| 1234501B      | Pystypuomi      | 29.07.2024        | 02.08.2024        | 5%              | 100%       | 100%       | 29.07.2024   | 02.08.2024    |
| 1234501C      | Vaakapuomi      | 30.07.2024        | 05.08.2024        | 10%             | 100%       | 100%       | 30.07.2024   | 06.08.2024    |
| 1234501W      | Hitsauspää SAW  | 12.08.2024        | 16.08.2024        | 5%              | 75%        | 70%        | 12.08.2024   |               |
| 1234501W1     | Hitsauspää MIG  | 12.08.2024        | 16.08.2024        | 20%             | 75%        | 75%        | 12.08.2024   |               |
| 1234501L      | Loppukokoonpano | 14.08.2024        | 21.08.2024        | 35%             | 14%        | 10%        | 14.08.2024   |               |
| 1234501P      | Lähetys         | 22.08.2024        | 23.08.2024        | 15%             | 0%         | 0%         |              |               |
| <b>TOTAL:</b> |                 | <b>29.07.2024</b> | <b>23.08.2024</b> | <b>100%</b>     | <b>49%</b> | <b>48%</b> |              |               |

**Add New Activity**

**Activity ID:**

**Description:**

Add Task

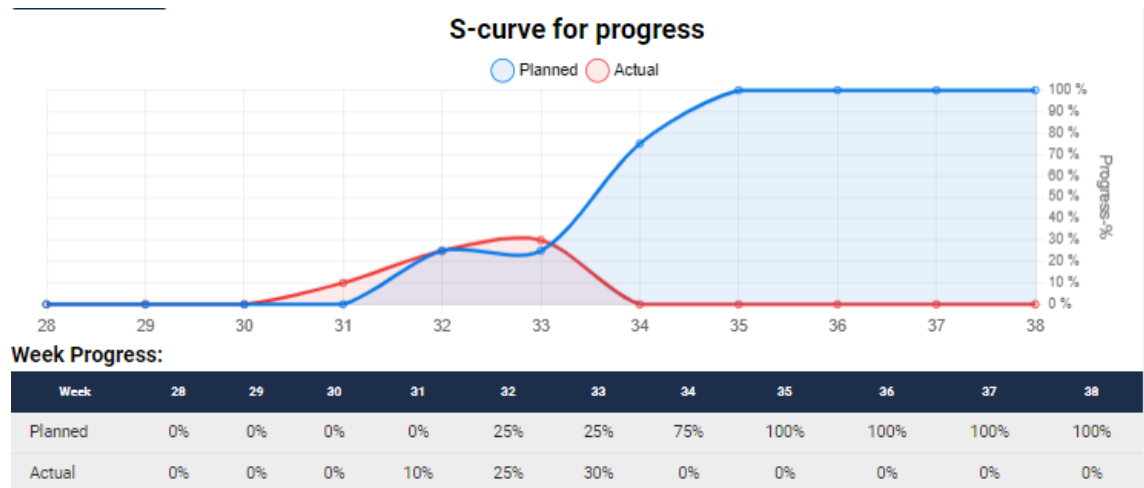
Hide

KUVA 6. Alikokoonpanotaulukko ja lisäyslomake.

## 5.2 S-edistymäkäyrä

S-edistymäkäyrän tehtävänä on antaa kokonaiskuva projektin kulusta ja havainnollistaa työn toteutumista suunnitellussa aikataulussa (KUVA 7). Suunniteltu ja

todellinen edistymä piirtyvät viivadiagrammiin, jonka x-akseli kuvaa aikaa viikoissa ja y-akseli prosentuaalista edistymää asteikolla 0-100 (0 ei aloitettu, 100 valmis). S-edistymäkäyrä on apuväline, josta on nopeasti nähtävissä suunnitellun ja todellisen edistymän suhde.



KUVA 7. S-edistymäkäyräkomponentti testidatalla.

Suunnitellun edistymän muodostaa datajoukko, joka saadaan alikokoonpanotaulukon arvoista ja laskennallisesta kaavasta. Kaavaan syötetään argumentteina sekä alikokoonpanojen aloitus- ja päättymispäivämäärät että weight factor (painotusarvo) ja tuloksena saadaan viikkonäkymään soveltuva datajoukko. Kaavan logiikka on peräisin alkuperäisestä edistymäseurannasta ja on muunnettu Excelin funktioista TypeScript-ohjelmointikielelle.

X-akselin viikkonumerot määräytyvät ensimmäisen alikokoonpanon aloituspäivämäärän ja viimeisen alikokoonpanon päättymispäivämäärän mukaan, mutta aikahaarukkaa laajennetaan lisäämällä kolme viikkoa sen alkuun ja loppuun. Tämä mahdollistaa edistymän seuraamisen tapauksissa, joissa työ aloitetaan suunniteltua aikaisemmin tai venyy asetetun valmistumispäivän yli. Todellisen edistymän käyttäjä syöttää taulukkoon käsin oikean viikkonumeron kohdalle. Kun työ on saatettu päätökseen ja todellinen edistymä on viikon kohdalla saanut arvon 100, ohjelmisto päivittää loput arvot automaattisesti.

## 6 POHDINTA

Projektin edetessä nousi esiin useita tekijöitä, jotka vaikuttivat ohjelmiston kehitykseen ja sen loppukäyttöön. Erityisesti datan hakeminen, yrityksen toimintamallien ymmärtäminen ja uusien teknologioiden omaksuminen osoittautuivat alkuvaiheessa haasteiksi. Monet näistä tekijöistä vaikuttavat myös sovelluksen jatkokehitykseen, ja niiden käsittely on keskeistä tulevien parannusten ja toiminnallisuuksien suunnittelussa. Lisäksi on pohdittu mahdollisuuksia ja ideoita, jotka voisivat parantaa sovelluksen toimivuutta, visuaalisuutta ja käytettävyyttä tulevaisuudessa.

### 6.1 Haasteet

Projektin alussa suurimpana haasteena koettiin yrityksen nykyisten toimintamallien ymmärtäminen sekä selkeiden tavoitteiden asettaminen. Yrityksen omien ohjelmistojen käytön oppimiseen sekä niiden koodiin integrointiin meni myös jonkin verran aikaa. Projektin edistystä hidastivat datan ulos saanti Poweredista sekä sen määrä. Jotta pystyttiin kehittämään mahdollisimman lopullista koodia, olisi ollut hyvä tietää heti alussa, mitä kaikkea dataa Poweredista haluttiin ohjelmaan tuoda ja missä datamuodossa sen ohjelmaan tulisi. Se, että mitä dataa haluttiin käyttää oli kuitenkin alussa hieman epäselvää sekä muuttui muutamaan otteeseen projektin aikana.

Yrityksellä on monia projekteja meneillään samaan aikaan eli datan määrä on suuri. Jatkokehitykseen jätettiin pohdinta siitä, miten kaikki keskeneräiset projektit saataisiin ohjelmaan järkevästi ilman, että tietokanta paisuisi liikaa ja kuinka ne rajattaisiin Poweredista. Yhtenä haasteena koettiin myös yrityksen toimintatapojen muuttuminen ja niiden kehittäminen, sillä samalla meidän tehtävät ja tavoitteet muuttuivat ikään kuin lennosta. Itse ohjelmointia hidastivat meille täysin uudet teknologiat sekä versionhallinnan puuttuminen.

## 6.2 Jatkokehitysideat

Konseptitodistuksen kehittämisen aikana oli paljon puhetta myös jatkokehitysideoista. Pääkohtina jatkokehitykselle pidettiin Powered-toiminnallisuuden lisäämistä, visuaalisuutta sekä kuormitustyökalua. Kun nämä pääkohdat saataisiin kuntoon, voisi nykyisen taulukkopohjaisen tuotantosuunnitelman korvata uudella verkkosovellusversiolla.

Vaikka Poweredista haetaan jo tilaukset ja niiden laitteet sovellukseen, on jatkokehityksessä toiveena saada sieltä myös automaattisesti muun muassa päivämäärät alikokoonpanoille, joilla korvattaisiin manuaalisesti syötetyt versiot. Alikokoonpanojen toiminnallisuutta haluttaisiin kehittää myös niin, että edistymäsivulla voisi manuaalisen syöttämisen sijaan valita alikokoonpanot alasvetovalikosta, tai ohjelma voisi tunnistaa laitteen ja lisätä siihen sopivat alikokoonpanot automaattisesti. Yrityksellä ei kuitenkaan ole vielä yleistettyjä alikokoonpanoja tietyille laitteille, joten toistaiseksi käytössä on manuaalinen syöttö.

Visuaalisuutta haluttaisiin parantaa esimerkiksi erilaisilla kuvaajilla, varoituksilla ja viesteillä. Varoituksia toivottaisiin lisää ainakin työvaiheisiin ja aikatauluun liittyen, jotta olisi mahdollisimman helppoa nähdä, jos jokin tilaus on aikataulusta jäljessä. Toiveena oli myös saada lisää suodatusmahdollisuuksia työvaiheisiin, jotta voisi esimerkiksi nähdä kaikkien laitteiden lähetykset omassa taulukossa.

Kuormitustyökalu on tärkeä osa tuotantosuunnitelmaa ja se tulisi saada toimivaksi osaksi sovellusta, jotta nykyisen version voisi korvata. Kuormituksella tarkoitetaan vaadittavia työtunteja, jotka tarvitaan projektin tekemiseen. Kuormitustyökalulla voisi siis sijoittaa projektin alikokoonpanoja työntekijöille, joiden perusteella työkalu pystyy laskemaan tunnit, jotka projektiin on kuormitettu. Näin nähdään, onko jokaiseen projektiin kuormitettu tarpeeksi, liikaa tai liian vähän työtunteja.

Näiden jatkokehitysideoiden lisäksi sovellukseen tarvittaisiin eritasoisia käyttäjiä. Admin-käyttäjät pystyisivät muokkaamaan päivämäärätietoja ja muuttamaan tau-

lukoiden arvoja. Katsojakäyttäjät pääsisivät sovellukseen tarkastelemaan ja hakemaan tietoa ja selaamaan eri näkymien välillä pystymättä kuitenkaan tekemään muutoksia tuotantosuunnitelmaan.

## LÄHTEET

Artto, K., Martinsuo, M. & Kujala, J. 2008. Projekttiliiketoiminta - projektinhallinnan oppikirja. Pdf-dokumentti. Viitattu 23.10.2024.

<https://www.aalto.fi/sites/g/files/flghsv161/files/2020-08/Projekttiliiketoiminta.pdf>

Express.js. n.d. Express. Verkkosivu. Viitattu 16.11.2024.

<https://expressjs.com/>

MDN Web Docs. n.d. CSS: Cascading Sheets Style. Verkkosivu. Viitattu 15.10.2024.

<https://developer.mozilla.org/en-US/docs/Web/CSS>

MongoDB. n.d. Why use MongoDB and when to use it? Verkkosivu. Viitattu 30.7.2024.

<https://www.mongodb.com/resources/products/fundamentals/why-use-mongodb>

Monterail. 2024. Vue vs. React in 2025: Which Framework To Choose For Your Development? Verkkosivu. Viitattu 15.10.2024.

<https://www.monterail.com/blog/vue-vs-react#a:Introduction---Why-Vue.js-vs.-React?>

Node.js. n.d. Introduction to Node.js. Verkkosivu. Viitattu 16.11.2024

<https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>

TechTarget. Maaliskuu 2023. Proof of Concept. Verkkosivu. Viitattu 25.7.2024.

<https://www.techtarget.com/searchcio/definition/proof-of-concept-POC>

TypeScript. Lokakuu 2024. TypeScript for the New Programmer. Verkkosivu. Viitattu 15.10.2024.

<https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>

TypeScript. Lokakuu 2024. The TypeScript Handbook. Verkkosivu. Viitattu 15.10.2024.

<https://www.typescriptlang.org/docs/handbook/intro.html>

Virolainen, H. & Yli-Knuutila, V. n.d. PoC ja MVP: Lyhyt oppimäärä. Verkkosivu. Viitattu 14.10.2024.

<https://www.itewiki.fi/p/poc-ja-mvp-lyhyt-oppimaara>

Vue.js. 2024. Introduction. Verkkosivu. Viitattu 15.10.2024.

<https://vuejs.org/guide/introduction.html>

Vuetify. 2024. Why Vuetify? Verkkosivu. Viitattu 15.10.2024.

<https://vuetifyjs.com/en/introduction/why-vuetify/#what-is-vuetify3f>

Vue with TypeScript. 2024. Verkkosivu. Viitattu 15.10.2024.

<https://vuejs.org/guide/typescript/overview>