



TAMPEREEN
AMMATTIKORKEAKOULU

Tuotannon töiden hienokuormitus

Tommi Jokinen

Opinnäytetyö
Toukokuu 2018
Tieto- ja viestintätekniikan koulutusohjelma
Ohjelmistotekniikka



TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintätekniikan koulutusohjelma
Ohjelmistotekniikka

JOKINEN TOMMI:
Tuotannon töiden hienokuormitus

Opinnäytetyö 27 sivua
Toukokuu 2018

Opinnäytetyössä oli tavoitteena toteuttaa Web-sovellus osaksi Collapick Company Oy:n valmista Ponniste-järjestelmää. Ponniste on yrityksille myytävä tuote, joka tehostaa tuotannon prosessointia, digitalisoimalla sen toimintaa. Opinnäytetyönä luodun Web-sovelluksen avulla loppukäyttäjä pystyy hallitsemaan visuaalisesti yrityksensä tuotannon töitä Ponniste-järjestelmän kautta.

Web-sovelluksen hallinta toiminnallisuus toteutettiin Gantt-kaavioiden avulla. Suunnittelu vaiheessa tarkasteltiin sopivia Gantt-kirjastoja vaadituilla ominaisuuksilla sekä pohdittiin sovelluksen tietomallille toimintakaaviota hienokuormitusta varten. Normaalisissa työnkuormituksessa työ nimetään sopivalle resurssille tehtäväksi. Hienokuormituksessa huomioidaan koko tuotannon kulku aikaa vasten sekä resursseille kuormitettu työmäärä. Näiden tietojen pohjalta työ saadaan hienokuormitettua parhaimmaksi katsotulle ajanjaksolle ja sopivalle resurssille, välttämällä tietyn osa-alueen ylikuormittamista.

Koska Web-sovelluksessa hallinnoitava data oli eri muodossa kuin Ponnisteen omassa tietojärjestelmässä, näiden välille kehitettiin adapteri hoitamaan datan muuntaminen sekä sovellukselle että Ponnisteelle ymmärrettävään muotoon. Adapterin vuoksi Ponnisteen ja Web-sovelluksen jatkokehitystä voidaan toteuttaa itsenäisesti, ilman vaikutuksia niiden keskinäiseen toimintaan.

ABSTRACT

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

JOKINEN TOMMI

Fine loading of production's tasks

Bachelor's thesis 27

May 2018

Thesis objective was to develop a Web application to be part of Collapick Company Ltd's Ponniste system. Ponniste is a product offered for other enterprises, to make their manufacturing process more effective via digitalization. With the implemented Web application, the end user is able to visually control the company's production work through the Ponniste system.

The Web application's management was implemented with Gantt charts. During the planning, different Gantt libraries were to look for one with the best features for the project and to plan an operating diagram on how the application's data is handled. Normally when appointing a workload to a resource, one appoints them to the resource whom can handle that work. A step up from this is observing the whole production against time as well as how large workloads each resource has at a time. With these informations the new workload can be appointed in a more efficient way, avoiding overloading any resources at a given time.

Since the form of the data in the Web application is different than in the Ponniste system, an adapter was planned between these two. The adapter would transform the data so that both components will be able to understand each other despite having the data in different forms. Further development for the Web application or Ponniste system can be done seperately without affecting each other's functionality, because of to the adapter.

Key words: Gantt, manufacturing execution system, programming, Web application

SISÄLLYS

1	JOHDANTO.....	6
2	YLEISTÄ	7
2.1	Tuotannonohjaus.....	7
2.2	Gantt-kaavio.....	8
2.2.1	Historia.....	8
2.2.2	Käyttäminen	10
3	TEKNOLOGIAT.....	12
3.1	Gantt-kirjastot	12
3.1.1	Google Chart Gantt-kaavio	12
3.1.2	JQuery Gantt-kaavio	13
3.1.3	JsGantt Improved-kaavio	13
3.1.4	TWProject Gantt-kaavio	14
3.1.5	Dojo Toolkit-kirjasto.....	15
3.1.6	Dhtmlx Gantt-kaavio.....	17
3.2	Loopback-moduuli.....	19
4	SOVELLUKSEN TOTEUTUS.....	21
4.1	Web-sovellus	21
4.2	Palvelin toteutus.....	25
5	POHDINTA.....	26
	LÄHTEET.....	27

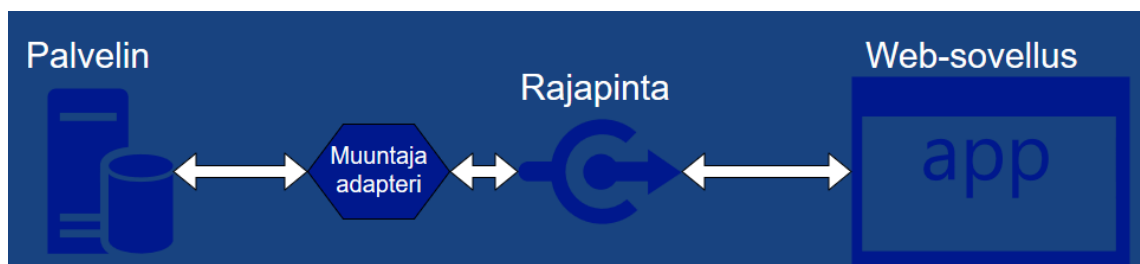
ERITYISSANASTO

EJS	Effective JavaScript on normaaliin JavaScriptiin pohjautuva tehostetumpi kieli HTML-sivustojen ohjelmointia varten.
Gantt	Gantt-kaavio on väline projektinhallinnan visualisoimiseen, josta on yksinkertaista seurata tai suunnitella projektin kulkua.
HTML5	Hypertext Markup Language 5 on nykyaikainen Web-sivustojen luonnissa ja esittämisessä käytettävä ohjelmointikieli. Se on HTML-standardin viides versio.
http	Hypertext Transfer Protocol on protokolla säännöistä, kuinka tiedostojen siirto tapahtuu maailmanlaajuisessa verkossa (WWW).
jQuery	Ketterä pieni JavaScript kirjasto HTML-sivustojen hallitsemiseen, toimintojen reagoimiseen ja yksinkertaistamaan palvelinkutsujen luomista.
JS	JS eli JavaScript on ohjelmointikieli, jota useasti käytetään Web-sivustojen teossa eri käyttötarkoituksiin.
JSON	JavaScript Object Notation on menetelmä, jolla voidaan esittää ohjelmointikielessä asiat rakenteellisessa muodossa ja se on monipuolisesti käytettävissä eri ohjelmointikielten välillä.
Node.js	Node.js on verkkosovelluksissa käytettävä alusta, joka mahdollistaa JavaScriptin käytön ilman erillisiä sovelluksia.
NoSQL	Not SQL tai not only SQL-termeillä tunnettu tietokantamalli, jossa datan muotoa ei ennalta määritetä, mahdollistaen tietojen tallettamisen sellaisenaan.
NPM	NPM on Node.js:n mukana tuleva pakkauksenhallintaväline, jolla voidaan hallita sovelluksen riippuvaisuuksia muihin komponentteihin.
REST	Representational State Transfer on alue sovelluksessa, josta pystytään kommunikoimaan sovelluksen kanssa ulkopuolelta.
SQL	Structured Query Language on ohjelmointikieli datan hallintaan relaatiotietokannoissa.
XML	Extensible Markup Language on menetelmä lähettää tai säilyttää tietoja rakenteellisessa muodossa.

1 JOHDANTO

Opinnäytetyön tavoitteena on saada toteutettua Collapick Company Oy:lle järjestelmä, jolla tuotannon töitä voidaan kuormittaa ja tarkastella visuaalisesti. Yrityksellä on tavoitteena saada kyseinen toiminnallisuus lähitulevaisuudessa järjestelmiinsä ja aihe kävi opinnäytetyöksi.

Opinnäytetyössä toteutettu sovellus integroidaan yrityksen nykyiseen Ponniste-järjestelmään. Tulevaisuuden muutoksia varten ajatuksena on toteuttaa sovelluksen ja palvelimen väliin adapteri (kuva 1). Adapteri huolehtii tuotannossa tarvittavien tietojen muuntamisen sovellukseen ymmärrettävään muotoon sekä takaisin palvelimelle. Tiedot enimmäkseen koostuvat erilaisista tuotannon resursseista, tilauksista, töistä ja näiden riippuvuuksista keskenään.



KUVA 1. Integrointi yksinkertaistettuna palvelimen ja Web-sovelluksen välillä

Web-sovellukseen käytettävissä komponenteissa on otettava huomioon nykyisen ympäristön toimintamalli. Mitä tarpeellisia ominaisuuksia nykyisessä järjestelmässä on käytössä ja kuinka saada ne käyttäjälle selkeään muotoon sovellukseen. Toiminta- ja tietomallin lisäksi on pohdittava mitä toimintoja käyttäjä voi suorittaa ja kuinka tehdä niistä käyttäjälle helposti ymmärrettäviä ratkaisuja. Alustavasti perustoiminnallisuuden kannalta vaadittavat toiminnot toteutetaan ensimmäiseen prototyyppiin. Kokonaisuudesta tehdään tarpeeksi modulaarinen ratkaisu, jotta siihen on helppoa tehdä toiminnallisuuksien lisäyksiä jatkokehityksessä.

2 YLEISTÄ

2.1 Tuotannonohjaus

Tuotannonohjaus on tapa tehostaa yrityksen tuotannonkulkua täyttääkseen vaatimusten mukaisten tuotteiden valmistuminen määritellyssä aikataulussa. Tuotannonohjaus on kilpailun kannalta hyvin tärkeää, sillä asiakkaat haluavat saada tuotteensa laadukkaana ja nopeasti toimitettuna.

Pystyäkseen toteuttamaan vaatimuksensa, täytyy yrityksellä olla suunniteltuna tuotantostrategia, jota tuotannonohjauksen on tarkoitus toteuttaa ja tehostaa. Tuotannonohjaukseen on monia tapoja, mutta pääosin se toteutetaan eri työkaluilla. Useimmiten ohjauksessa hyödynnetään eri ohjelmistoja ja tietojärjestelmiä. Käytetyimpiä ovat ERP- (Enterprise Resource Planning) eli toiminnanohjausjärjestelmät, joilla voidaan hallinnoida yrityksen liiketoimintaan vaikuttavien osa-alueiden tietoja (kuva 2).



KUVA 2. Nykyiset ERP-järjestelmät sisältävät paljon eri osa-alueisiin hyödynnettäviä työkaluja (Incarnate ICT Pvt. Ltd.)

Kaikki yritykset eivät kuitenkaan tarvitse käyttöönsä jokaista osa-aluetta ERP-järjestelmästä. Näissä tapauksissa katsotaan yritykselle tarpeellisimmat osa-alueet ja tehdään tar-

vittavat räätälöinnit toimintojen tehostamiseksi. Räätälöintiä kuitenkin rajoitetaan tiettyyn pisteeseen, sillä kaikilla ERP-järjestelmän toteuttajilla ei ole mahdollisuuksia ylläpitää jokaiselle yritykselle omaa optimoitua järjestelmää.

ERP-järjestelmän ansiosta yrityksessä pystytään selvittämään myyntiä varten, paljonko tuotannon tarvitsee tuottaa ja tuotanto osaa kertoa, paljonko työtä se vaatii tuottaakseen tarvittavan määrän. On tärkeää pystyä havainnollistamaan tuotannosta missä ja mitä tarvitaan eri työvaiheissa. Näin tietojen pohjalta voidaan aikatauluttaa tuotannonkulkua, kun tiedetään kuormitusten määrä tuotannon eri osa-alueilla. (Hemilä J., Häkkinen K. & Pötry J., 2009, 27-28)

Hienokuormituksessa on tarkoituksena saada suunniteltua työjärjestys, jolla tuotannon tavoitteet saataisiin toteutettua. Tämän mahdollistamiseksi järjestelmän täytyy välittää tietoa tuotantotasolle sekä kerätä tietoja tuotannon toteutumasta. Hienokuormituksessa on myös huomioitava reaaliajassa resurssien kapasiteetti- sekä materiaalirajoitteet, jotta vältetään turhilta seisahduksilta. Kerätyillä tiedoilla voidaan parantaa hienokuormituksen toteuttamista saaden parempia aika-arvioita tuotannon läpikulusta. Näiden vaatimusten vuoksi hienokuormitukseen perustuvat järjestelmät ovat yleisesti erillisiä sovelluksia, ja niitä käytetään ERP-järjestelmien rinnalla, mikäli yrityksellä sellainen on. (Haverila M., Uusi-Rauva E., Kouri I. & Miettinen A., 2005, 417-418; Hemilä J. ym., 2009, 27)

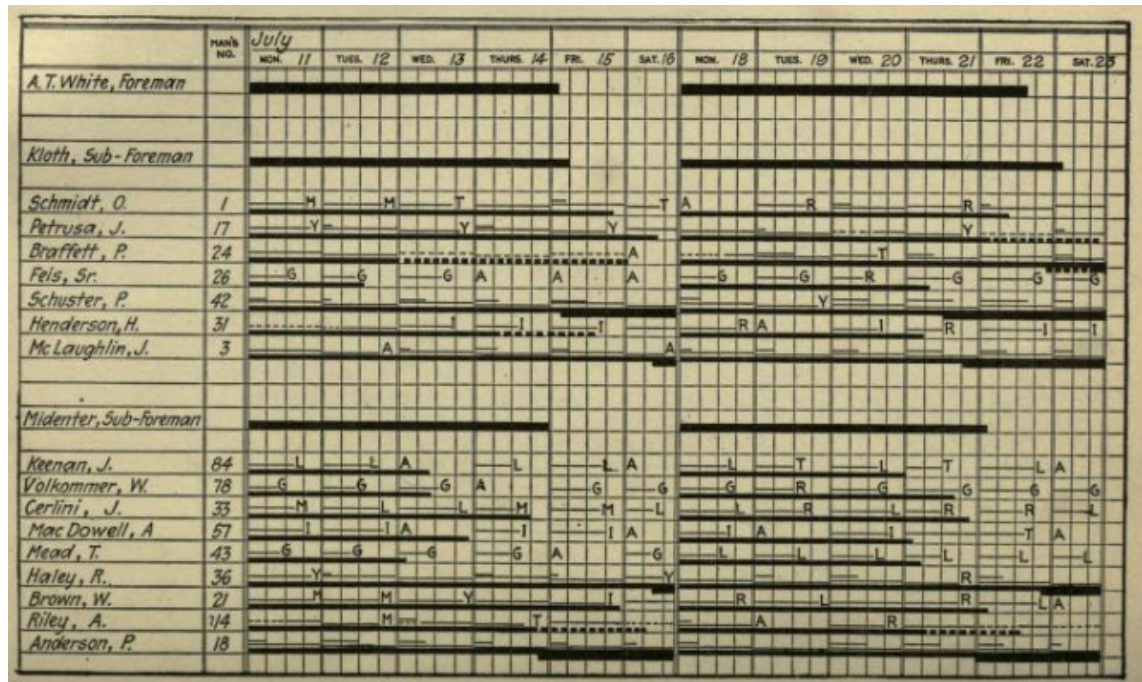
2.2 Gantt-kaavio

Gantt-kaavio on projektin hallinnassa käytettävä menetelmä, jossa voidaan suunnitella projektin töiden kulku ja seurata niiden etenemistä. Projektien lisäksi kaaviota on voitu hyödyntää työntekijöiden tai laitteiden seuraamiseen, jolloin saadaan tieto, milloin ja miten paljon kukin on työtä saanut aikaan.

2.2.1 Historia

Gantt-kaavion nimitys tulee Henry Laurence Ganttista, joka kehitti kaaviomallinsa vuosina 1910–1915. Clark Wallace kirjoitti kirjan Gantt-kaavion valmistamisesta ja käytöstä, joka julkistettiin vuonna 1923. Näihin aikoihin Gantt-kaavio piirrettiin kynillä sopivan

kokoisille papereille. Herra Gantt suositteli kansioihin sopivan kokoista paperia, sillä niihin saatiin helposti aseteltua kaksi peräkkäistä viikkoa (kuva 3) tai koko vuoden kaavio kerralla. (Wallace 1923, 8)



KUVA 3. Vanha Gantt-kaavio kuvastamassa työntekijöiden edistymissä tehtävissään kahden viikon ajalta (Wallace 1923, 38)

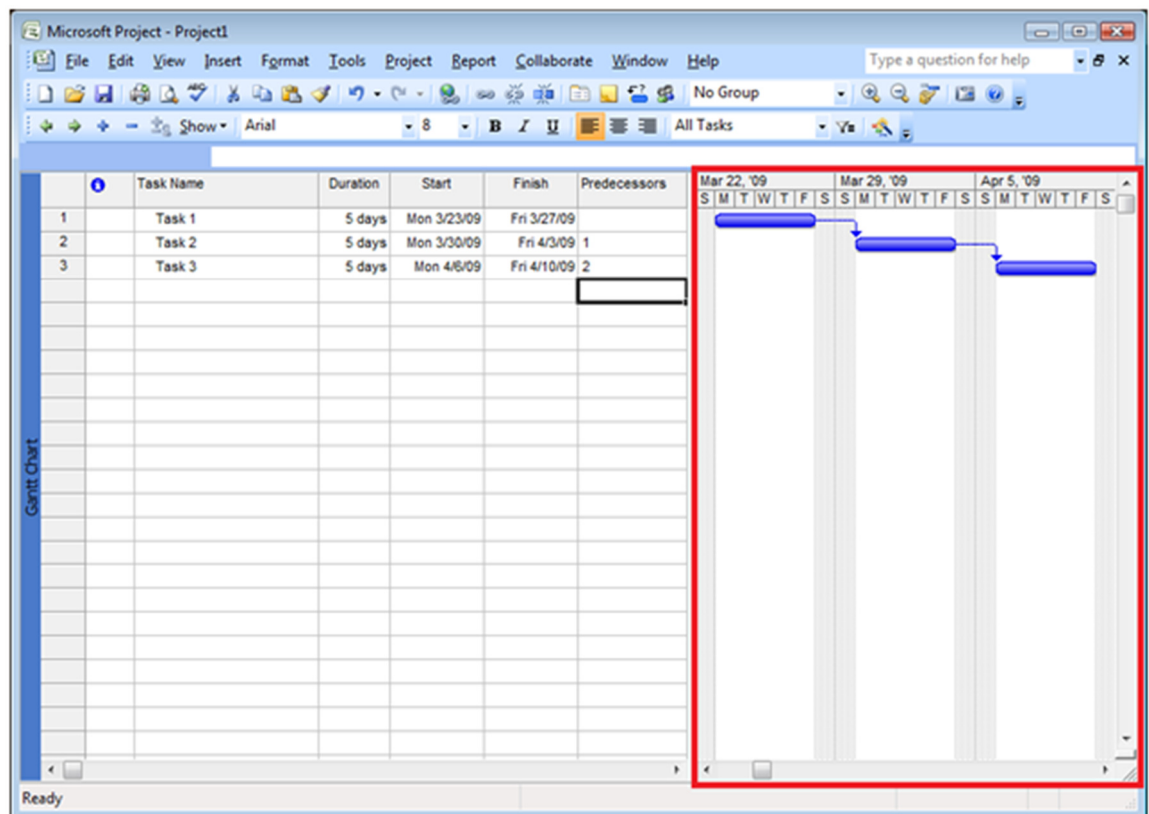
Gantt-kaavion kehittyessä modulaarisemmaksi, kaavioiden muokkaaminen kynin kävi työlääksi. Piirtämisen sijasta aloitettiin käyttämään paperinpalasia tai kuutioita kuvaamaan kaavion palkkeja. Ongelma tuli vastaan, kun kaaviota käytettiin todella suurissa tuotannoissa, missä tarkkailtavia pisteitä oli paljon ja aikataulumuutoksia saattoi aiheutua ulkopuolisista syistä. Hyvänä esimerkkinä ovat sotien aikana ollut aseteollisuus, jossa Gantt-kaaviosta saatiin todella paljon hyötyä sekä uusia näkemyksiä, kuinka kehittää kaaviota eteenpäin. (Wallace 1923, 110)

Mikrotietokoneiden tultua markkinoille, monimutkaisempien Gantt-kaavioiden valmistus mahdollistui. Internetin myötä Gantt-kaavioita pystyttiin myös tekemään Web-sovel-luksina, joka laajensi niiden käyttöä entisestään.

2.2.2 Käyttäminen

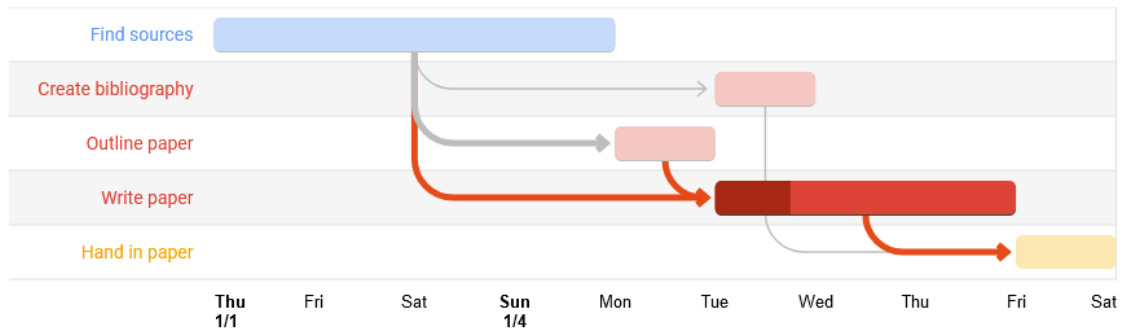
Gantt-kaavio muodostuu kahdesta eri osasta. Ensimmäisessä osassa on taulukko, johon on listattu tehtävät tai työt (kuva 4). Tehtävät ja työt voivat olla kategorisoituna esimerkiksi tilausten tai resurssien alle, jolloin tiedot voidaan esittää puurakenteen muodossa. Taulukossa voidaan esittää nimien lisäksi käyttäjälle hyödyllistä tietoa, kuten aloitus- ja lopetuspäivämäärät, kokonaiskesto ja tämänhetkinen tila.

Toisessa osassa kaaviota on asetettu listalla olevat omille riveillensä kohdistettuna aikajanaa vasten (kuva 4). Näin listatuista tiedoista tulee käyttäjälle selkeä kaavio tarkasteltavaksi.



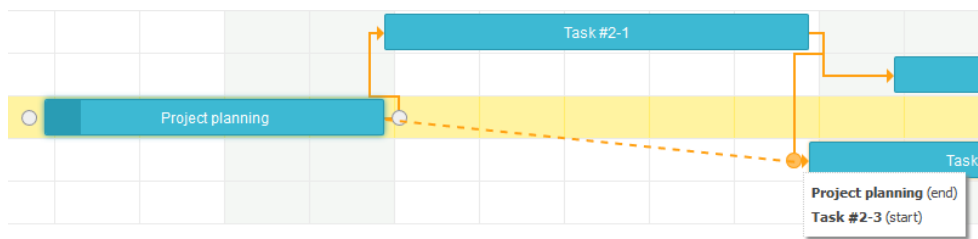
KUVA 4. Microsoft Project 2010 -projektinhallintaohjelman Gantt-kaavio (Microsoft Corporation)

Ohjelmoiduista Gantt-kaavioista löytyy lisätoimintoja antamaan hyödyllisiä lisätietoja projektinkulusta. Esimerkki tällaisesta toiminnosta on kriittisen polun havaitseminen. Kriittisellä polulla tarkoitetaan aluetta töiden riippuvaisuuksissa, joka myöhästyessään aiheuttaisi koko projektin valmistumisen myöhästymistä (kuva 5). Tällöin tuotannossa voidaan tehdä esimerkiksi varatoimenpiteitä estämään mahdollinen myöhästymisen sellaisille tehtäville, jotka sijaitsevat kriittisellä polulla.



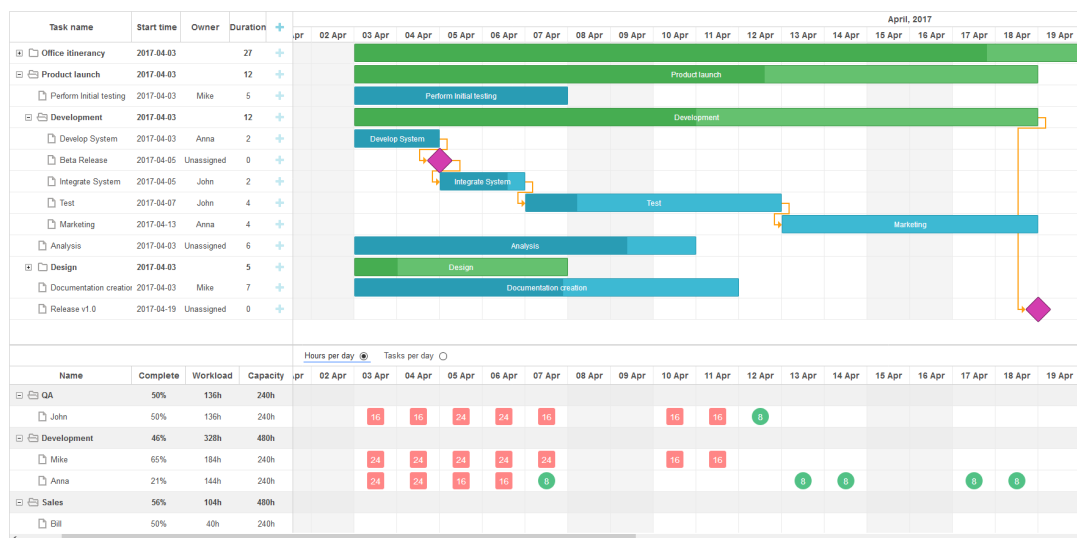
KUVA 5. Jos paperille kirjoittamisessa tapahtuu myöhästymisiä, koko tehtävän kulku tulee myöhästymään

Gantt-kaaviot helpottavat riippuvaisuuksien luontia tehtävien välille, joita yleisesti kuvataan nuolilla. Käyttäjä saa vedettyä tehtävästä toiseen nuolen, luoden kahden tehtävän välille riippuvaisuuden (kuva 6).



KUVA 6. Kahden tehtävän välille ollaan asettamassa riippuvaisuus

Jos tietoa on paljon, ne voidaan jakaa kahteen eri Gantt-kaavioon (kuva 7). Näin näkymästä saadaan entistä informoivaa ja käyttäjä pystyy katselmoimaan tietoja eri näkökulmista.



KUVA 7. Piilottamalla tarpeettomat tiedot saadaan suurestakin määrästä dataa selkeää

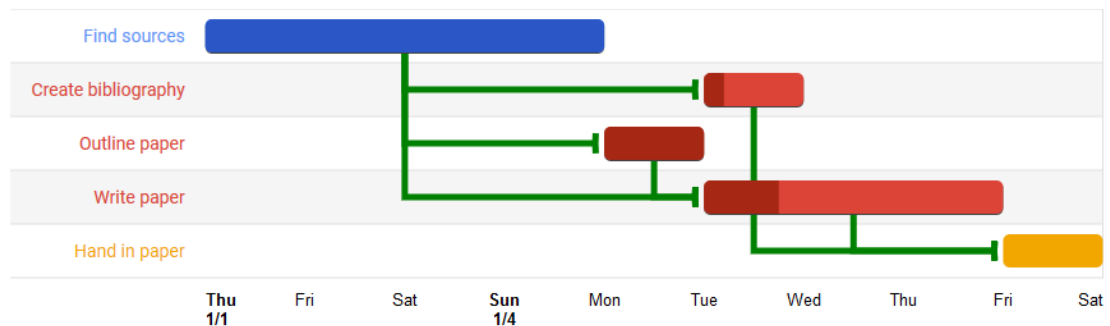
3 TEKNOLOGIAT

3.1 Gantt-kirjastot

Gantt-kaavion esittämistä varten etsittiin vertailtavaksi erilaisia kirjastoja, joita Web-sovelluksessa voitaisiin hyödyntää. Jotta Web-sovellus saataisiin integroitua vanhempiinkin Ponniste-järjestelmiin, rajattiin kirjastojen haku JavaScript pohjaisiin vaihtoehtoihin. Näin vältetään uusien ohjelmointikielien konfiguroimiselta nykyiseen järjestelmään, joka aiheuttaisi ongelmia vanhojen versioiden päivittämisessä.

3.1.1 Google Chart Gantt-kaavio

Googella on oma ilmainen Google Charts-kirjasto, joka sisältää paljon erilaisia kaavioita eri tilanteita varten. Yksi vaihtoehtoista on Gantt-kaavio (kuva 8). Kaavio on helppo toteuttaa listaten vain datan Gantt-objektiin ja kutsuen kaavion piirtometodia.

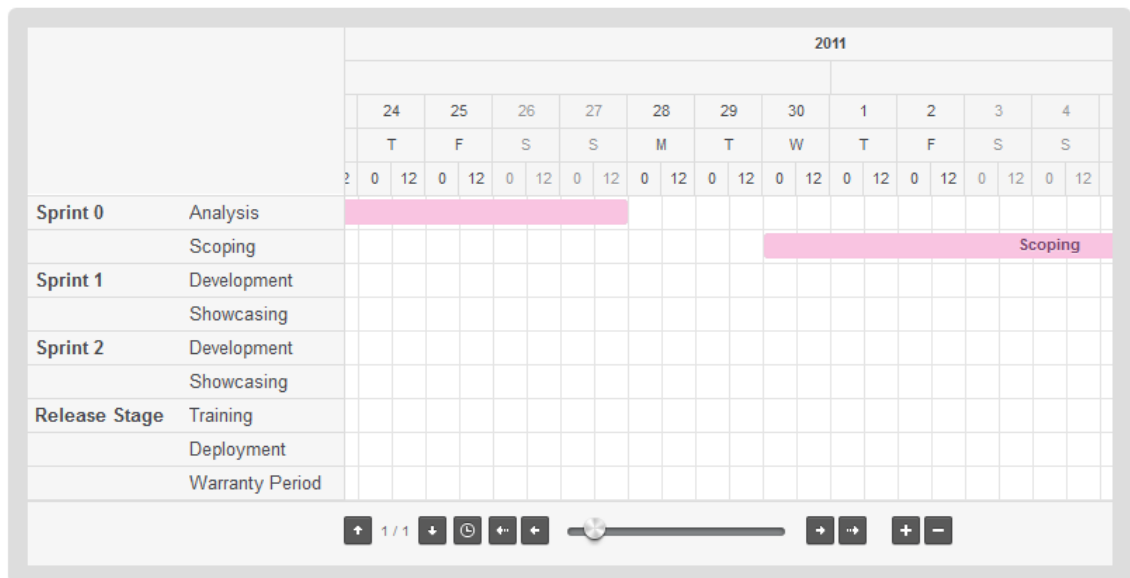


KUVA 8. Googlen oman kirjaston yksinkertainen Gantt-kaavio esimerkki

Google Chartin visuaalisesti selkeä ja yksinkertainen ulkoasu sopii hyvin staattisen datan esittämiseen. Pienet lisäominaisuudet kuten resursseihin ryhmittely ja kriittisen polun havaitseminen ovat käteviä. Google Charts ei kuitenkaan sisällä muokkaus mahdollisuutta itsessään. Se vaatii myös Internet-yhteyden, sillä kirjaston sai ainoastaan noudettua Googlen omilta palvelimilta. (Google LLC 2017)

3.1.2 JQuery Gantt-kaavio

JQuery Gantt-kaavio on ilmainen jQuery-kirjastoon pohjautuva Gantt-kaavio. Kaavio on yksinkertaisella tavalla toteutettu, sisältäen listan tehtävistä, jotka aikajanelle on asetettu. Tehtävien riippuvaisuuksia kaaviossa ei ole mahdollista asettaa. Aikajanelle asetettuja tehtäviä pystyi selaamaan hiiren rullan lisäksi alla olevan liukusäätimen avulla (kuva 9).



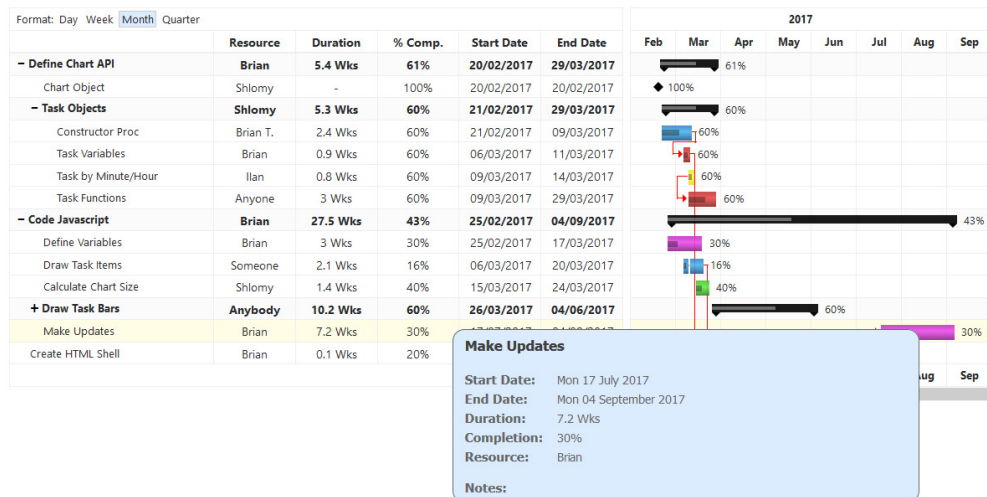
KUVA 9. Gantt-kaavio esimerkki toteutettu JQuery Gantt -kirjastolla

Kaavion zoomaus -toiminto lisää näytettävien tuntien määrää aikajanelle. Tämä aiheutti aikajanan selaamisessa ongelmia, sillä selaaminen hyppi kokonaisia päiviä, tehden tehtävien tarkastelusta hankalaa. Zoomaus-toiminto osoittautui myös hitaaksi pienilläkin datamäärillä.

Tällä Gantt-kaaviolla on kuitenkin yksinkertaisen lyhyt ja ytimekäs dokumentaatio sen vähäisten ominaisuuksien vuoksi. Kaaviota voisi suositella sellaisiin ympäristöihin, joissa jQueryä hyödynnetään sekä data halutaan esittää staattisesti.

3.1.3 JsGantt Improved-kaavio

JsGantt Improved on puhtaasti JavaScriptiin pohjautuva ilmainen kirjasto. Gantt-kaavion selkeän ulkoasun myötä voidaan esittää suuriakin määriä dataa siististi ja helposti ymmärrettävänä (kuva 10). Yksinkertaiset valinnat ajan esittämiseksi ja aikajanan helppo selaaminen teki tehtävien katselmoinnista miellyttävää.



KUVA 10. Yksinkertainen aikajanan mukauttaminen ja värien asettaminen tehtäville luotyylikkään ulkoasun

Gantt-kaavio sisälsi kattavasti perusominaisuuksia, kuten riippuvaisuudet tehtävien välillä, edistymisprosentit ja kiinnitykset resursseihin. Kaaviossa ei kuitenkaan ollut mahdollisuutta tehdä muokkauksia tehtäviin tai riippuvaisuuksiin.

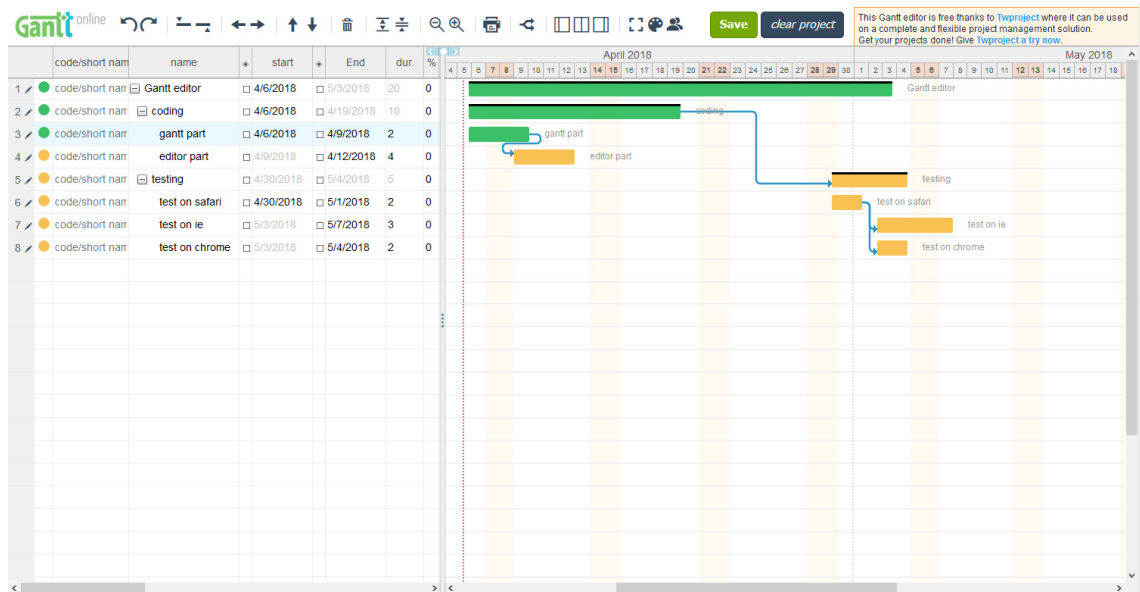
JsGantt Improved-kirjastossa oli mahdollisuus viedä ja tuoda Gantt-kaavio XML-tiedostomuodossa. XML-tiedostomuodossa Gantt-kaavion tietoja on yksinkertaista siirtää paikasta toiseen tai vain pitää tallessa. Hyvin kattava dokumentaatio kirjastoon liittyvistä metodeista ja mahdollisuus luoda eri kielille ymmärrettäväksi tekee JsGantt Improved-kirjastosta hyvän vaihtoehdon isompiin projekteihin. (Rodrigues & Cardoso, n.d.)

3.1.4 TWProject Gantt-kaavio

TWProject Gantt oli toinen jQueryyn ja Bootstrap HTML-teemaan pohjautuva ilmainen ja avoin Gantt-kirjasto. Edellisiin verrattuna kirjasto sisälsi mahdollisuuden muokata kaaviossa olevaa dataa. Kirjasto tehtiin alun perin TWProject:n sisäistä järjestelmää varten, mutta jaettiin myös muiden käytettäväksi ilmaiseksi. (Bicchierai R., 2012)

TWProject Gantt-kaaviosta löytyy paljon eri painikkeita kaavion muokkaamiseen sekä tietojen esittämiseen (kuva 11). Kaavion pystyi tallettamaan JSON-tiedostomuotoon,

josta kaavion sai takaisin ladattua esille. Kaavion ollessa JSON-muodossa, se voidaan lähettää esimerkiksi palvelimelle talletettavaksi.



KUVA 11. TWProject Gantt-kaavio sisältää paljon eri toimintoja tehtävien ja projektien esittämiseen

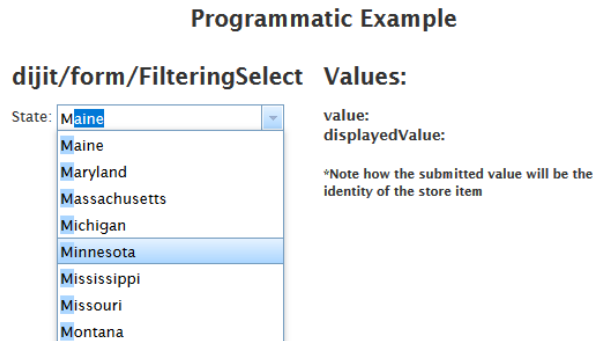
Projekteissa voi tulla konflikteja vastaan TWProject Gantt-kirjaston kanssa, sillä se hyödyntää Bootstrap-kirjastoa. Projektin muista komponenteista kannattaakin tarkistaa, muokkaavatko ne alkuperäisiä tyylitiedostoja, mikä saattaa rikkoa Gantt-kaavion toiminnallisuuksia. Muutoin TWProject Gantt-kirjastoa voi suositella, vaikkei täydellisesti kaikkia toiminnallisuuksia sisällä. Lisätoiminnallisuuksien toteuttamiseen saa tukea kehittäjältä avoimen (Bicchierai R. 2017)

3.1.5 Dojo Toolkit-kirjasto

Dojo Toolkit on JavaScriptiin pohjautuva avoimen lähdekoodin kirjasto, joka on luotu yksinkertaistamaan Web-sivustojen ja -sovellusten kehittämistä. Kirjaston kehitys oli aloitettu vuonna 2004 ja kehittäjät ympärimaailmaa ovat voineet osallistua sen Dojon kehittämiseen. (JS Foundation)

Dojo-kirjaston sisältö on lajiteltu kolmeen eri pakettiin. Itse Dojo-paketti sisältää kirjaston ytimen, joka vaaditaan muiden ominaisuuksien toimintaan. Lisäksi se sisällyttää yleisiä luokkia yleishyödyllisistä funktioista käyttäjän syötteiden testaamiseen.

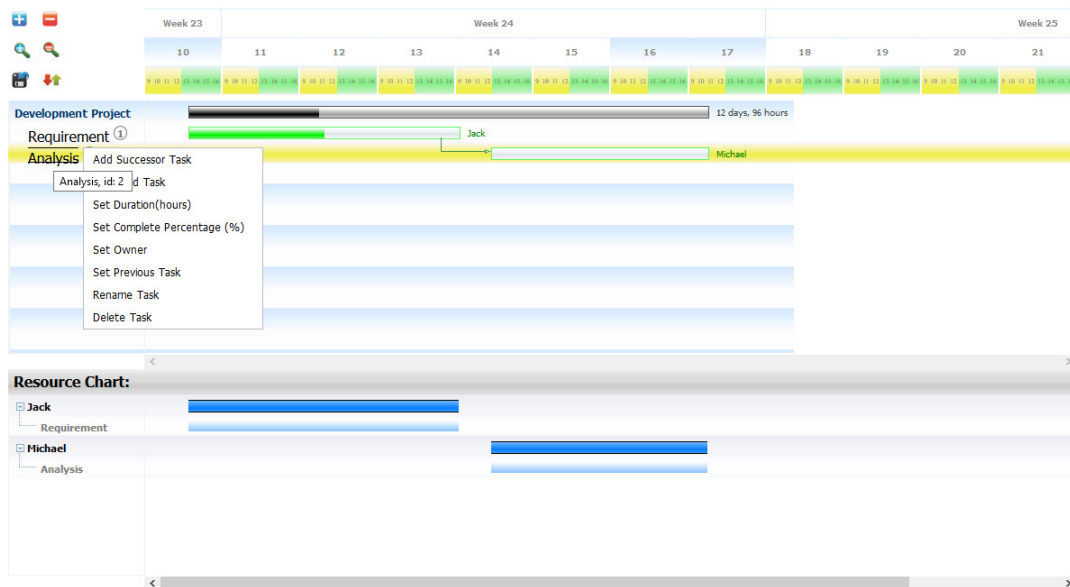
Dijit-paketti sisältää erilaisia käyttöliittymän komponentteja perusvalintalaatikoista kalentereihin asti. Jokaisella komponentilla on kattavia lisätoimintoja, kuten mahdollisuus suodattaa pudotusvalikon sisältö käyttäjän syötteen perusteella (kuva 12).



KUVA 12. Pudotusvalikko osaa suodattaa tekstin käyttäjän syötteen mukaisesti

Dojox-paketti sisältää Dojon kehitysvuosien aikana toteutettuja erikoiskomponentteja. Näihin Gantt-kaavion lisäksi kuuluu muita tiedon visualisoinnin, visuaalisten tehosteiden, tietokanta kommunikoinnin komponentteja. Jokaisen komponentin ylläpito on jokaisen kehittäjän omalla vastuulla ja niiden toimivuutta ei taata uudemmissa versioissa.

Dojox-paketin Gantt-kaavio toistaiseksi ollaan ylläpitämässä nykyisessä versiossa. Kaavioon kuuluu kattavat perustoiminallisuudet ja jopa tietojen muuttaminen lennosta. Lisäksi kaaviossa oli valmis resurssi kaavio, johon jokaisen Gantt-kaavion tehtävät jaoteltiin resurssi kohtaisesti reaaliajassa (kuva 13). Gantt-kaaviossa voi myös suurentaa ja pienentää aikajanaa sekä tallentaa JSON-tiedostoon sekä tuoda JSON-tiedostosta takaisin.



KUVA 13. Dojox Gantt-kaaviossa on mahdollista tehdä lisäyksiä ja muokkauksia lennosta

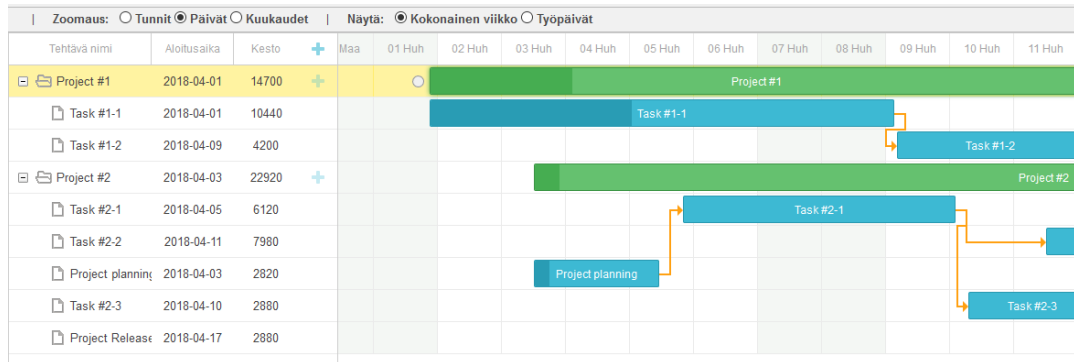
Näiden lisäksi kirjastoon kuuluu testaus paketti, joka sisältää toimintoja testien suorittamiseen Dojo-kirjastolla kehitettyihin sovelluksiin. Dojo on pätevä Web-sovelluksen kehittämiseen sopiva työkalu, joka sisältää moniin tilanteisiin sopivia komponentteja sekä tukea niiden käyttöön. Kokeellisten komponenttien käyttöä kannattaa kuitenkin harkita niiden epävarman ylläpidon vuoksi.

Dojosta ollaan kehittämässä seuraavaa merkittävää Dojo 2-versiota, josta on beetajulkaisu saatavilla. Uudempi versio on toteutettu täysin JavaScriptiin pohjautuvalla TypeScript-kielellä. Dojo 2-versio ei tule sisältämään kaikkia Dojox-paketin komponentteja, joten niiden käyttöönottoa kannattaa harkita, mikäli tahtoo pysyä ylläpidetyllä versiolla. (Kitson K. 2017)

3.1.6 Dhtmlx Gantt-kaavio

Dhtmlx Gantt-kaavio on yksi Dinamenta-yrityksen JavaScript-pohjaisista käyttöliittymä kirjastoista. Dhtmlx tuotteet ovat GPLv3-lisensioituja, joten rajatussa ilmaisikäytössä tuotettu koodi on jaettava avoimena muidenkin saataville. Muutoin Dhtmlx myy lisenssejä kaupalliseen käyttöön sekä erikseen PRO-lisenssi, jolla ostetuista kirjastoista saadaan PRO-ominaisuudet käytettäväksi.

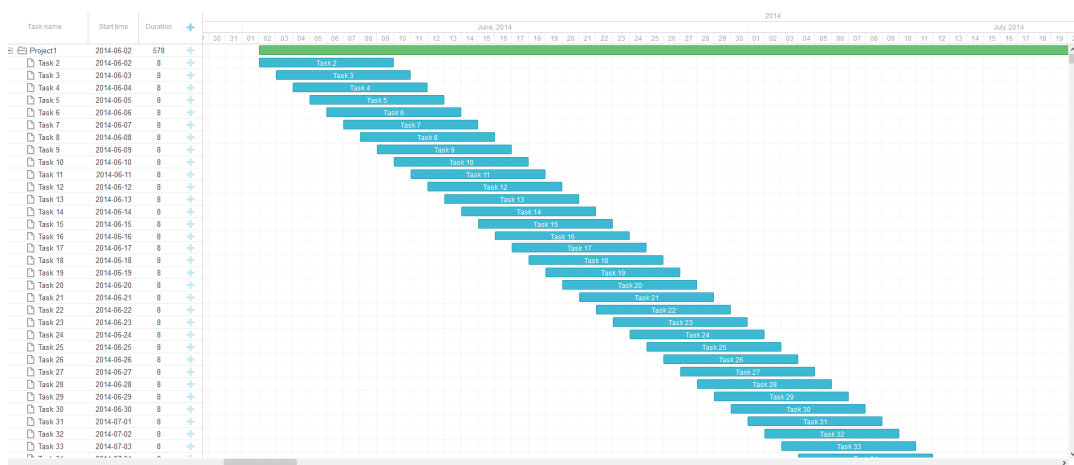
Maksullisuuden vuoksi Dhtmlx Gantt-kaaviossa on runsaasti toimintoja ja kattavan dokumentaation. Selkeällä käyttöliittymän pohjalla oli helppoa lähteä lisäämään omia toimintomalleja kaavioon sekä poistamaan tarpeettomia ominaisuuksia. Kaaviosta löytyi miltei valmiina kattava asetelma aikajanan hallintaan ja taulukko skaalautui nätisti asetusten mukaisesti (kuva 14).



KUVA 14. Dhtmlx Gantt-kaaviossa on eroteltavissa viikonloput sekä lomapäivät

Dhtmlx Gantt-kaavio sisältää muokkaamisen ajonaikana sekä adapterin, jonka kautta muunnetut tiedot saadaan helposti lähetettyä konfiguroituun rajapintaan. Tiedonsiirron muodon sai valita XML:n ja JSON:n väliltä, avaten erilaisia mahdollisuuksia toteuttaa tiedonvienti. PRO-lisenssin kanssa on mahdollista viedä kaavio kuvaksi tai PDF-dokumentiksi tulostamisen lisäksi.

Järkevän koodauksen ansioista kaavio pystyy käsittelemään monien tuhansien verran dataa kerralla, sillä se piirtää vain ne palkit, jotka sillä hetkellä käyttäjälle näkyy (kuva 15).



KUVA 15. Yli 30 000 tiedon määrä ei hidasta kaavion käyttöä

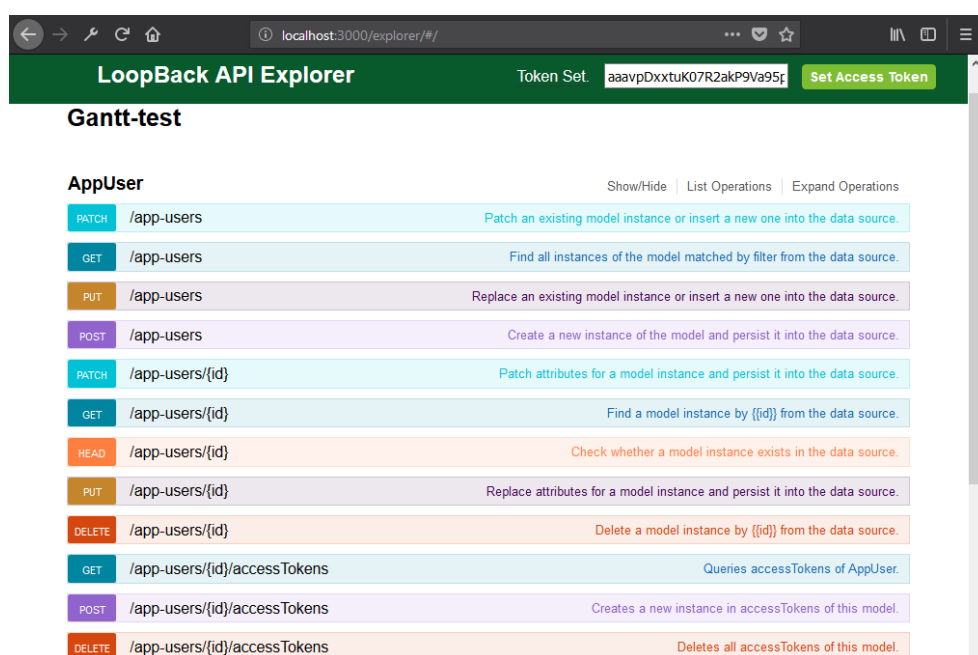
Dhtmlx Gantt-kaavio toimii niin yksinään kuin yhdessä muiden Dinamentan tuottamien komponenttien kanssa. Vaikka Dhtmlx Gantt-kaavio oli ominaisuuksiltaan hyvä, sen GPLv3-lisenssi tai maksullisuus ei ole vaihtoehto jokaiselle projektille.

3.2 Loopback-moduuli

Loopback on Node.js:n päälle asennettava moduuli, joka mahdollistaa yhtenäisen rakenteen tietokannan ja rajapinnan välille. Se on yhteensopiva monen eri tietokantatyypin kanssa SQL-tietokannoista erilaisiin NoSQL-tietokantavaihtoehtoihin.

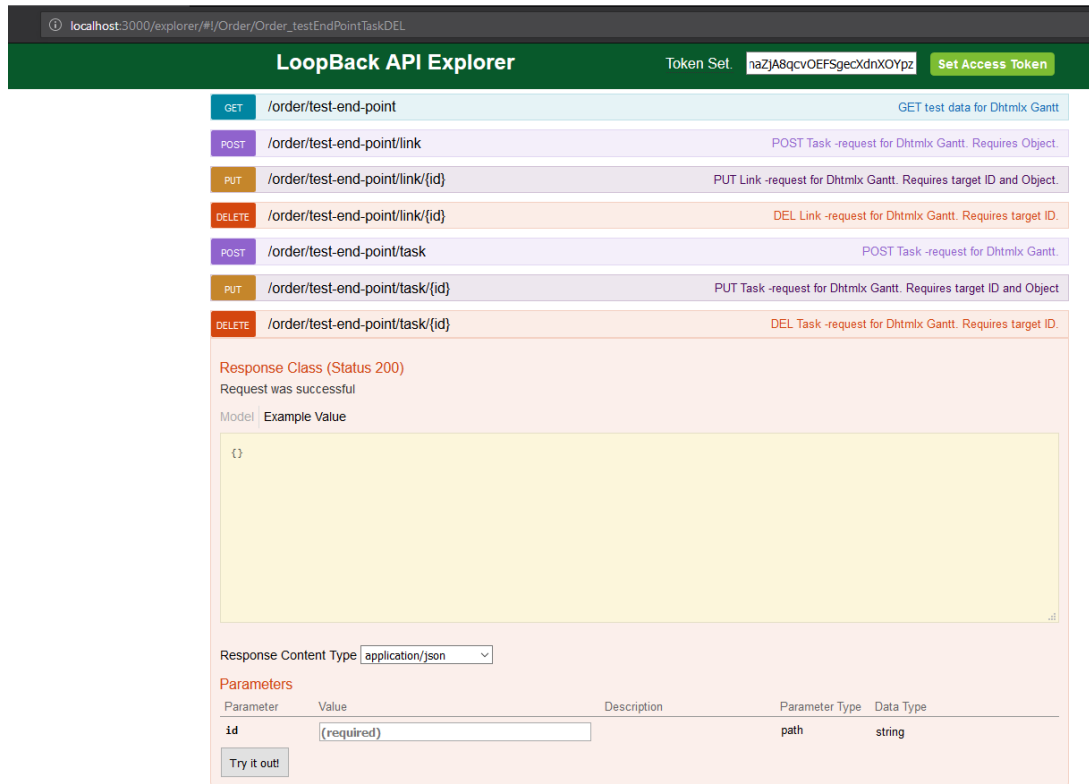
Loopback sisältää valmiiksi autentikointia varten järjestelmän sekä rajapinnan. Käyttäjän tarvitsee kirjautua kyseisen rajapinnan kautta sisään, josta käyttäjä saa avaimen. Tätä avainta tarvitaan rajapintoihin, joihin on konfiguroitu todentamisvaatimus päälle. Muutoin se palauttaa rajapintaa kutsuttaessa virheen todentamisen epäonnistumisesta.

Jotta Loopbackia voi hyödyntää tietokantojen käytössä, täytyy talletettavalle datalle määritellä mallit. Malleihin määritetään kaikki sille tarpeelliset ominaisuudet, niiden muoto tietokannassa ja mahdolliset relaatiot toisiin malleihin. Lisäksi malleihin voidaan konfiguroida lisätoiminnallisuuksia Loopbackin sisällä, kuten todentamisvaatimus. Mallien määritettyä, Loopback luo automaattisesti rajapinnat jokaiselle mallille (kuva 16).



KUVA 16. Loopbackin automaattisesti luodut rajapinnat AppUser-mallille

Perusrajapintojen lisäksi, malleille voidaan luoda JavaScriptillä omia päätteitä. Niihin voidaan määrittää itse päätteen osoite, tyyppi ja logiikka, mikä suoritetaan osoitetta kutsuttaessa (kuva 17). Koska Loopback hoitaa tiedonkulun tietokantoihin, itse luotujen mallien rajapintojen koodia ei tarvitse muokata tietokantojentyypin perusteella.



KUVA 17. Itse toteutetut testausrajapinnat Dhtmlx Gantt-kaavion kyselyille

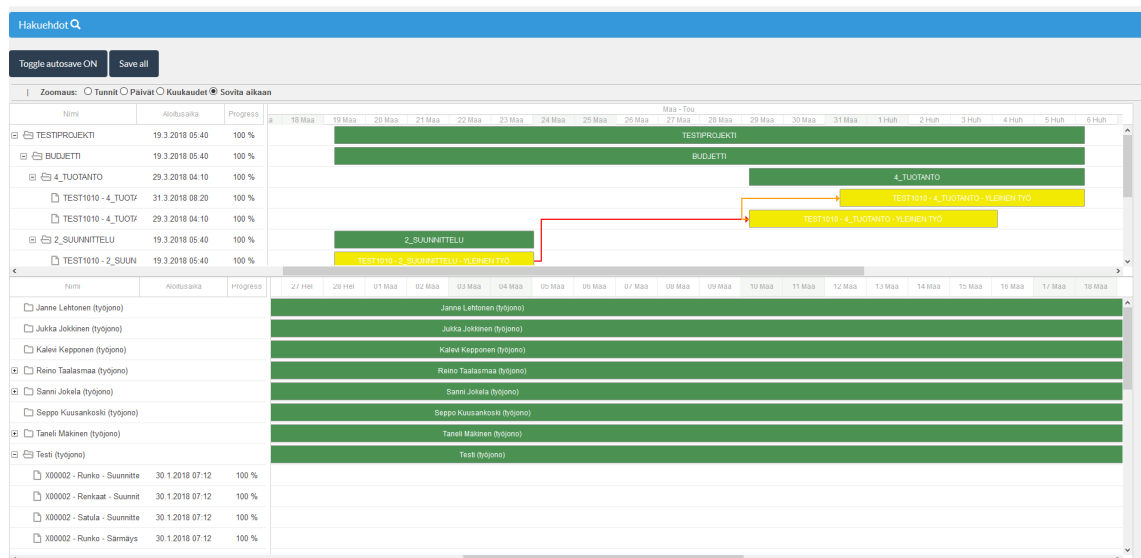
Tietomallien ja muun logiikan lisääminen jälkikäteen Loopbackiin onnistuu ilman, että edellisiä tietoja tietokannasta katoaisi. Loopback ei kuitenkaan ole heti käyttövalmis, vaan se vaatii paljon konfigurointia ennen käyttöä. Siksi se ei ole paras valinta projekteihin, joissa rajapinnan ei tarvitse olla kattava tai tietokantamalliin ei tulla tekemään suuria skaalautuvia muutoksia. Suurissa ja skaalautuvissa järjestelmissä Loopback helpottaa rajapintojen luontia ja ylläpitoa merkittävästi.

4 SOVELLUKSEN TOTEUTUS

4.1 Web-sovellus

TWProject Gantt-kirjasto oli ilmaisista vaihtoehdoista sopiva projektiin, mutta sen riippuvaisuus Bootstrap-teemaan aiheutti yhteensopivuus ongelmia muiden komponenttien kanssa. Dojo Toolkit-kirjaston ongelmaksi koitui Gantt-kaavion epävarma jatkokehitys tilanne uuden Dojo 2-version alla. Muista ilmaisista vaihtoehdoista puuttui tarpeelliseksi kaivattu muokkaus toiminnallisuus, joten jäljelle jäi Dhtmlx Gantt-kirjasto ajan ja tarpeettoman työn säästämiseksi. Lisäksi Dhtmlx Gantt-kirjastolla oli kattava dokumentaatio ja sisälsi muita toimintoja, joita pystyttiin hyödyntämään yksinkertaisten vaadittavien toimenpiteiden toimintaa.

Ensimmäisenä askeleena oli saada visuaalinen toteutus Web-sovellukselle. Näkymään haluttiin saada tilaukset ja niiden sisältö näkyviin, sekä esittää jokaisen resurssin työt. Lähtökohtana oli tehdä kaksi Gantt-kaaviota, jossa toisessa näkyy tilaus- ja toisessa resurssikohtaisesti (kuva 18).



KUVA 18. Web-sovelluksen Gantt-kaaviot on täytetty testitiedoilla

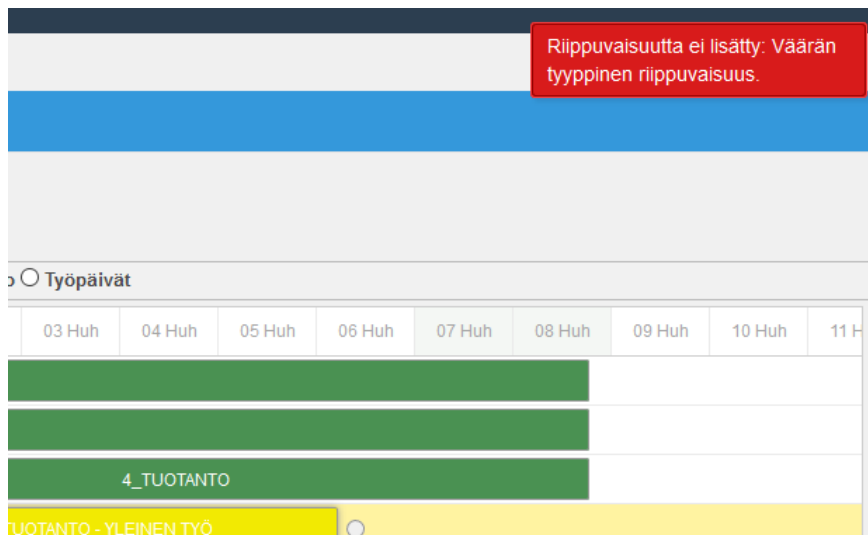
Tiedoille määriteltiin värit kuvaamaan mitä kukin palkki tarkoittaa. Tilaukset ja niiden osat sekä resurssi käyttivät samaa väritystä, kuvastamaan niiden olevan kaaviossa otsik-

kotason tietoja. Töiden väriytyksiin vaikuttaa mikä tila töillä saattoi olla. Esimerkiksi valmiilla töillä on vaaleampi vihreä väritys ja keskeneräisellä keltainen. Näin käyttäjä pystyy kuormitusta tehdessään havainnollistamaan töiden tilat.

Katselmoinnin lisäksi piti tietojen olla muokattavissa tietyn rajoittein. Tuplaklikkaamalla resurssia, tilausta tai työtä avautuu tietomalliin sopiva ikkuna, jossa käyttäjä voi muuttaa tarpeellisia tietoja. Työ ikkunassa käyttäjän on mahdollista vaihtaa kuormitettu resurssi sekä muuttamaan aloitus- ja lopetuspäivää (kuva 19). Päivämääriä käyttäjä myös pystyy muuttamaan raahaamalla työtä kaavion aikajanalla. Virheellisistä tiedoista käyttäjä saa virheilmoituksen.

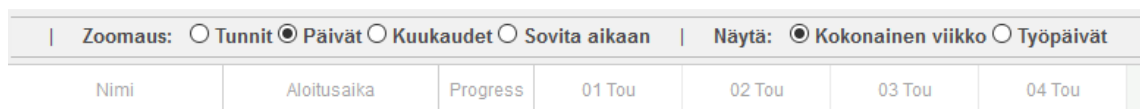
KUVA 19. Tehtävien tietoja voidaan muokata lennosta Gantt-kaaviossa

Muokkauksien lisäksi käyttäjä pystyy luomaan riippuvaisuuksia töiden välille. Riippuvaisuuksilla visualisoidaan töiden ketjutusta, minkä jälkeen tai kanssa minkäkin työn voi aloittaa. Mikäli käyttäjä yrittää asettaa vääränlaista riippuvaisuutta, käyttäjälle ilmoitetaan virheen mukainen virheviesti ja riippuvaisuutta ei luoda (kuva 20).



KUVA 20. Käyttäjä saa virheilmoituksen virheellisestä toiminnasta

Käyttäjän on mahdollista säätää aikajanaa, mikäli se on liian suurena tai pienenä. Aikajanaa on mahdollista säätää tunteihin, päiviin, kuukausien mukaan tai antaa sovelluksen automaattisesti sovittaa kaikki yhteen näkymään. Tunti ja päivä valinnoissa on mahdollista suodattaa ajat jolloin työtä ei tehdä, kuten esimerkiksi viikonloput ja yö tunnit (kuva 21).



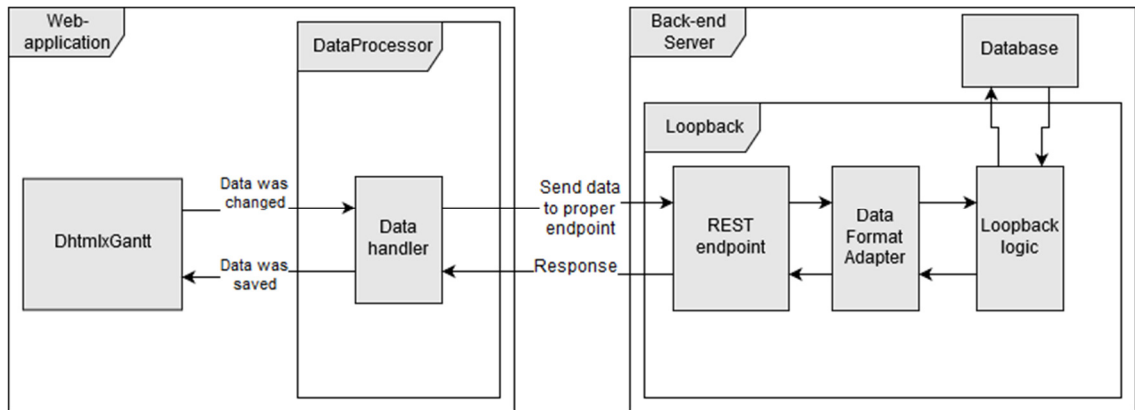
KUVA 21. Käyttäjä pystyy valitsemaan aikajanan koon

Käyttäjä voi myös rajata mitä tietoja kaavioissa näytetään. Aikarajauksella rajataan tilaukset tilauskaavioon ja työt resurssi kaavioon. Lisäksi tilauksia voidaan rajata tehtyihin ja tuotannossa oleviin tai etsiä suoraan tilaus nimenperusteella (kuva 22).

The screenshot shows a search interface for orders. At the top, there is a search bar labeled "Hakuehdot". Below the search bar, there are two input fields for dates: "Päivästä" (2018-01-01) and "Päivään" (2018-05-31). There are also two checkboxes: "Tehdyt tilaukset" and "Tuotannossa". Below these, there is a search bar labeled "Nimi" with the text "TESTIPROJ" entered. A dropdown menu is open below the search bar, showing a list of project names: "X00003 - TESTIPROJEKTI 3", "X00002 - TESTIPROJEKTI 2", "X00004 - TESTIPROJEKTI 4", "X00001 - TESTIPROJEKTI 1", and "TEST1010 - TESTIPROJEKTI".

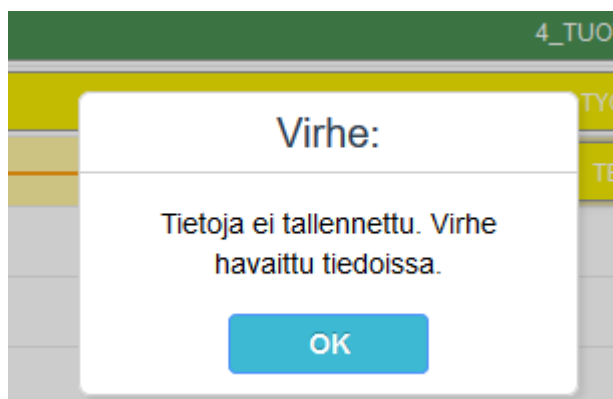
KUVA 22. Haku kriteerien lisäksi käyttäjä voi hakea tilauksen nimellä

Web-sovelluksessa tietojen tallentamisessa hyödynnettiin Gantt-kaavion dataprocessoria. Dataprocessori pystyy havaitsemaan kaaviossa tehdyt muutokset ja rakentaa niiden pohjalta pyyntö toteutettuun rajapintaan palvelimella (kuva 23).



KUVA 23. Toimintakaavio Web-sovelluksen tietomallin kulusta

Ennen tehtyjen muutosten lähettämistä, tietojen oikeellisuus varmistetaan, ettei virheellistä tietoa lähetetä tietokannalle. Varmistuksessa tarkistetaan jokaisen riippuvaisuuden ja tehtävän oikeellisuus. Esimerkiksi tehtävät joiden välillä on riippuvaisuus, riippuvaisuuden kohteen aloituspäivämäärä ei saa aikaisemmin kuin lähteen aloituspäivämäärä. Jos varmistus epäonnistuu, käyttäjälle tulee virheilmoitus ja dataprocessori jättää kutsun tekemättä rajapintaan (kuva 24).



KUVA 24. Tallennuksen aikana havaittu virhe

Lähetetty viesti prosessoidaan rajapinnalla ja palautetaan viesti takaisin sovellukselle. Riippuen rajapinnalta palautetusta viestistä, suoritetaan tarvittavat toimenpiteet, jotta tiedot ovat eheät sekä Web-sovelluksessa että tietokannassa.

4.2 Palvelin toteutus

Loopback-moduuliin rinnalle toteutettiin adapteri sekä rajapinnat Dhtmlx Gantt-kaavion kyselyitä varten. Koska kaavio käyttää rajapintakutsuissaan samaa osoitepääteettä eri tyypeillä, jouduttiin rajapinnalle tulevan tiedon muoto tarkistamaan. Tietoihin päätettiin lisätä tyyppi tieto, josta pystyttiin erottelemaan mitä tieto on ja ohjaamaan se adapterin läpi oikealle polulle Loopback-moduulissa.

Tiedon nouto kyselyistä piti tehdä tarpeeksi mukautuvia, että kaaviolle haettu data on mahdollista myös suodattaa käyttäjän syötteiden perusteella. Tietojen noudossa oli huomioitava olemassa olevan järjestelmän tietomalli. Tarpeelliset tiedot on ensin kerättävä hajautetusta tietomallista ja viedä ne adapteriin, jossa niiden muuntaminen Web-sovellukselle ymmärrettävään muotoon on toteutettu. Adapteri on osittain riippuvainen Web-sovelluksen Gantt-kaavion konfiguroinneista, joten konfigurointi tiedot asetettiin JSON-tiedostoon helpottamaan niiden muuntamista.

Tietojen muuntamisen lisäksi, adapteri rakentaa palauteviestit Web-sovellukselle ymmärrettävässä muodossa. Näin viestit ovat aina määritellyssä muodossa ja niiden prosessointi sovelluksessa on selkeämpää. Näin myös virheet pystyttiin erottelemaan selkeästi ja esittämään käyttäjälle tarpeenmukaiset virheviestit.

5 POHDINTA

Opinnäytetyön taustan tuntemattomuus antoi haastetta työn alkuun. Toimeksiantajan Ponniste-järjestelmästä sai pohjataustaa aiheelle sekä vertailupohjan mahdollisten kilpailijoiden toteutuksiin. Lisäksi yrityksellä aikaisempi työskentely auttoi aiheen käsittelemisessä.

Web-sovellukseen toteutuksen ohjelmointikieli valittiin olemassa olevan järjestelmän pohjalta, jonka kautta muiden käytettävien kirjastojen rajaukset aloitettiin. Kirjastojen valikoinnissa joutui pohtimaan kuinka paljon aikaa käyttää niiden soveltamiseen ja testaamiseen Ponniste-järjestelmää vasten. Osassa tapauksia ilmeni vakavia ongelmia. Mikäli ongelman lähde oli helposti korjattavissa, testausta voitiin jatkaa, mutta pahemmissa tapauksissa oli kannattavampaa katselmoida seuraavaa kirjastoa.

Työn haasteena oli myös saada tieto järkevänä esiintymään sovelluksessa. Adapterin rinnalle oli pohdittava logiikkaa keräämään nykyisestä tietomallista tarpeelliset tiedot kaaviota varten sekä lisäysten tai muokkausten myötä viemään ne takaisin tietomalliin. Toteutuksessa täytyi ottaa huomioon adapterin tehokkuus, sillä joissakin tapauksissa tietoa voi olla valtava määrä.

Web-sovelluksen toteutuksen aikana oli suunniteltava mitä toiminnallisuuksia sovitussa aikataulussa on mahdollista toteuttaa ja mitä niistä jätetään jatkokehityksen puolelle. Jatkokehitykseen jätettyjä toiminallisuuksia varten oli kuitenkin toteutettava vakaaversio sovelluksesta, jotta niiden toteuttaminen tulevaisuudessa olisi helpompaa.

LÄHTEET

Bicchierai R., 2012, The JavaScript Gantt odyssey Luettu 16.3.2018

<http://roberto.open-lab.com/>

Bicchierai R., 2017, New Gantt Editor Release: The Best Ever! Luettu 13.3.2018

<http://roberto.open-lab.com/>

Dinamenta, 2018, Interactive JavaScript/HTML5 Gantt Chart for Your Solutions. Luettu 14.4.2018

<https://dhtmlx.com/docs/products/dhtmlxGantt/>

Google LLC, 2017. Using Google Charts. Luettu 16.3.2018

<https://developers.google.com/chart/interactive/docs/>

Haverila M., Uusi-Rauva E., Kouri I. & Miettinen A., 2005, Teollisuustalous, 5. painos., Luettu 13.4.2018, Tampere: Tammer-Paino

Hemilä J., Häkkinen K. & Pötry J., 2009, Tuotannonohjaus ja tietojärjestelmät: kokemuksia sekä kehittämisperiaatteita - Prologi-hankkeen loppuraportti, Luettu 8.4.2018

<http://www.vtt.fi/inf/pdf/workingpapers/2009/W130.pdf>

Incarnate ICT Pvt. Ltd., 2017. The modern approach to choosing an ERP system. Luettu 4.4.2018

<http://incarnateict.com/2017/09/25/>

JS Foundation, n.d., Dojo Toolkit Reference Guide Luettu 11.4.2018

<https://dojotoolkit.org/reference-guide/>

Kitson K., 2017, Announcing Dojo 2 beta 4! Luettu 12.4.2018

<https://dojo.io/blog/2017/12/04/Announcing-Dojo-2-beta-4/>

Microsoft Corporation, 2009. Back to basics of Gantt Chart view. Julkaistu 1.4.2009, Luettu 1.4.2018

<https://blogs.msdn.microsoft.com/project/>

Rodrigues E. & Cardoso R., n.d. jsGanttImproved. Luettu 21.2.2018

<https://jsganttImproved.github.io/jsgantt-improved/>

Wallace C., 1923. The Gantt chart, a working tool of management. New York: The Ronald press company