

Opinnäytetyö (AMK)

Tieto- ja viestintätekniikka

2023

Antton Heinonen

Aikataulutus rakennusalan projektinhallintasovelluksessa

– projektien tehokkuutta parantavan ominaisuuden
suunnittelu ja kehittäminen



Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintätekniikka

2023 | 27 sivua

Antton Heinonen

Aikataulutus rakennusalan projektinhallintasovelluksessa

– projektien tehokkuutta parantavan ominaisuuden suunnittelu ja kehittäminen

Tämä opinnäytetyö kuvaa aikatauluhallinnan ominaisuuden suunnittelua ja kehittämistä rakennusalan projektinhallintasovellukselle. Ominaisuus mahdollistaa projektitehtävien visuaalisen esittämisen Gantt-kaaviossa ja kyvyn hallita näitä tehtäviä samasta näkymästä.

Sovellusosio rakennettiin Vue.js-ohjelmistokehysympäristöön käyttäen hyväksi avoimen lähdekoodin grafiikkakirjastoa. Koska ominaisuus tuli olemassa olevaan sovellukseen, tuli se integroida olemassa oleviin järjestelmiin ja palvelimiin. Projektin tavoitteena oli luoda helposti laajennettava kehys tuleville toiminnoille ja työkaluille.

Asiasanat:

Projektinhallintasovellus, JavaScript, Vue

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Information and Communications Technology

2023 | 27 pages

Antton Heinonen

Scheduling in a construction project management application

– Designing and developing a feature that improves the efficiency of projects

This thesis describes the design and development of a time management feature for a project management application used in the construction industry. The feature allows for a visual representation of project tasks in a Gantt chart and the ability to manage these tasks from the same view.

The application was built using the Vue.js framework and an open-source graphics library. Since the feature was added to an existing application, it had to be integrated with the existing systems and back end. The project aimed to create an easily expandable framework for future functions and tools.

Keywords:

Project management software, JavaScript, Vue

Sisältö

Käytetyt lyhenteet tai sanasto	5
1 Johdanto	6
2 Projektissa käytetyt teknologiat	7
2.1 Sovellusrunko	7
2.2 Vue.js-ohjelmistokehys	8
3 Projektin suunnittelu ja toteutus	10
4 Yhteenveto	11
Lähteet	12

Liitteet

- Liite 1. Projektisuunnitelma
- Liite 2. Vaatimusmäärittely
- Liite 3. Loppuraportti

Kuvat

Kuva 1. Esimerkki Vuen mallisyntaksista	9
---	---

Kuviot

Kuvio 1. Rekordin elinkaari (Rekord 2023)	8
---	---

Käytetyt lyhenteet

DOM	Dokumenttimalli (engl. Document Object Model)
ORM	Objektisuhdekartoitus (engl. Object-Relational Mapping)
PHP	Ohjelmointikieli (engl. PHP: Hypertext Preprocessor)
REST	Ohjelmointirajapinnan arkkitehtuurityyli (engl. Representational State Transfer)
SVG	Vektorikuvien kuvauskieli (engl. Scalable Vector Graphics)

1 Johdanto

Tämä opinnäytetyö käsittelee projektinhallintasovelluksen ajanhallintaominaisuuden suunnittelua ja kehittämistä. Tämän ominaisuuden tärkeimmät elementit ovat sovelluksen projektien työtehtävien visuaalinen esitys Gantt-kaaviossa sekä näiden työtehtävien hallinointi samasta näkymästä.

Projekti toteutettiin osana Punchzee-projektinhallintasovellusta. Punchzee on rakennusalaan erikoistunut sovellus, jossa on kattavat työkalut muun muassa työn seurantaan, pohjapiirrosten hallinointiin, työntekijöiden ja managerien välisen kommunikointiin ja työn dokumentointiin. Sovellus toimii kaikilla yleisimmillä alustoilla eli selaimissa sekä Android- ja iOS-alustoilla. (Punchzee 2023.)

Sovelluksen aikasempi versio työtehtävien aikajanatoiminnosta ei vastannut asiakkaiden tarpeita ja oli vaikeasti laajennettavissa, joten tässä opinnäytetyössä käsiteltävä projekti aloitettiin alusta. Tavoitteena oli rakentaa visuaalisesti selkeä esitys projektien sisäisistä työtehtävistä aikajanalla, sekä tarjota helposti jatkokehitettävissä oleva runko tuleville toiminnoille ja työkaluille.

Koska projektin puitteissa kehitetty osio tuli osaksi isompaa kokonaisuutta, ei käytettyjä teknologioita valittu itse, vaan ne määräytyivät sovelluksen aikasempien valintojen mukaan. Pääasiallisena ohjelmointikielenä toimi natiivi JavaScript sekä samaiseen kieleen perustuva Vue.js-ohjelmistokehys.

Opinnäytetyön alussa esitellään käytetyt teknologiat. Sen jälkeen kerrotaan projektin suunnittelusta ja toteutuksesta lyhyesti. Näiden asioiden tarkempi käsittely on projektin dokumentoinnissa opinnäytetyön liitteissä.

2 Projektissa käytetyt teknologiat

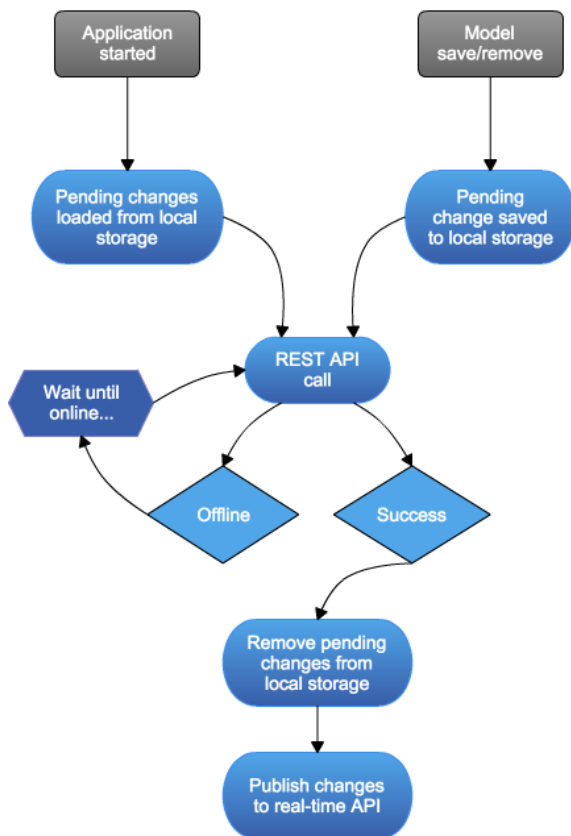
2.1 Sovellusrunko

Punchzee-sovelluksen yksi myyntivalteista on sen laaja toimintaympäristö, sillä se toimii niin verkkoselaimissa kuin Android- ja iOS-sovelluksina. Näiden eri alustojen koodikanta on yksi ja sama. Tämä on saavutettu käyttämällä avoimen lähdekoodin Apache Cordova -mobiilisovelluskehystä. Se yhdistää verkkoselainteknologian mobiilialustojen natiiviominaisuuksien kanssa ja näin mahdollistaa niin sanottujen hybridisovellusten kehittämisen käyttäen JavaScriptiä. (The Apache Software Foundation 2023.)

Jotta Punchzeen kehittäminen olisi mielekästä monessa eri ympäristössä, sovellus on rakennettu niin sanotun konttitekologian varaan Docker-sovelluksella. Docker paketoii sovelluksen eri osat standardisoituihin ajoympäristöihin, minkä ansiosta niin kehitys-, testaus- kuin tuotantoympäristöt pysyvät samoina. Tämä kokonaisuudessaan nopeuttaa sovelluksen käyttöönottoa tuotannossa sekä varmistaa kehitysympäristöstä huolimatta yhdenmukaisen ajoympäristön sovellukselle. Näin ei pääse syntymään tilanteita, joissa samassa projektissa työskentelevillä sovellus käyttäytyisi eri tavoin. (Docker Inc. 2023.)

Sovelluksen tietokantana on avoimen lähdekoodin relaatiopohjainen MySQL. Se tunnetaan erityisesti sen nopeudesta, luotettavuudesta ja skaalautuvuudesta, ja se onkin pitkään ollut eniten käytössä oleva tietokantaohjelmisto. (Oracle 2023.) Sovelluksen keskusteluyhteys kyseisen tietokannan kanssa on rakennettu PHP-pohjaisen Laravel-ohjelmistokehyksen avulla.

Tietokannan kanssa kommunikoinnissa on aina viivettä, ja aina tietokanta ei välttämättä ole edes saatavilla esimerkiksi kun käyttäjä ei ole verkossa. Tästä huolimatta sovelluksen on toimittava saumattomasti. Tähän ongelmaan on monia työkaluja, joista tässä projektissa on käytössä Rekord. Rekord on REST ORM JavaScript-kirjasto, joka toimii RESTful-rajapinnan kanssa, yksinkertaistaen eri toimenpiteiden suorittamista tietokantaan (Kuvio 1). (Rekord 2023.)



Kuvio 1. Rekordin elinkaari (Rekord 2023)

2.2 Vue.js-ohjelmistokehys

Sovelluksen käyttöliittymä on rakennettu Vue.js-ohjelmistokehyksellä. Se oli vuonna 2022 kolmanneksi käytetyin JavaScript-pohjainen front end -kehys (Stack Overflow 2022). Vue tarjoaa työkalut käyttöliittymän rakentamiseen. Se on keskittynyt deklaratiiiseen renderöintiin ja komponentteihin. Näiden avulla dataa voidaan

renderöidä suoraan DOMiin käyttäen yksinkertaista mallisyntaksia (Kuva 1).

```
1  <template>
2    <div>
3      <h1>{{ message }}</h1>
4    </div>
5  </template>
6
7  <script>
8    export default {
9      data() {
10        return {
11          message: 'Hello World!'
12        },
13      };
14    }
15  </script>
```

Kuva 1. Esimerkki Vuen mallisyntaksista

Deklaratiivisen renderöinnin avulla Vue linkittää datan ja DOMin, jolloin käyttöliittymästä tulee reaktiivinen. Vue myös mahdollistaa HTML-attribuuttien, eli direktiivien, käyttämisen. Vuessa on monia sisäänrakennettuja direktiivejä, jotka laajentavat HTML:ää, esimerkiksi v-if -direktiivi, joka ehdollisesti renderöi DOMiin dataa. Käyttäjä voi myös itse luoda omia direktiivejä.

Vuen komponenttirakenne pilkkoo käyttöliittymän itsenäisiin ja uudelleenkäytettävissä oleviin osiin. Tämän avulla duplikaattikoodin määrä vähenee ja koodikanta selkeytyy. Nämä komponentit vastaanottavat reaktiivisesti dataa ja renderöityvät itsenäisesti.

Käyttöliittymän rakentamisen lisäksi Vuen oheiskirjastot tarjoavat monimutkaisempia ominaisuuksia, kuten reitityksen ja globaalin tilanhallinnan (engl. state management). (Vue 2023.)

3 Projektin suunnittelu ja toteutus

Projektin suunnittelu aloitettiin tekemällä projektisuunnitelmadokumentti (Liite 1). Projektisuunnitelma pitää sisällään kuvauksen projektiorganisaatiosta, tavoitellusta aikataulusta, teknisestä toteutussuunnitelmasta sekä projektinhallinnasta.

Projektisuunnitelman lisäksi aluksi tehtiin vaatimusmäärittelydokumentti (Liite 2). Siinä on määritelty projektille raamit, joiden sisällä työskennellään. Käyttökokemuksen vaatimusmäärittelyn pääasiallinen tehtävä on määrittää projektin lopputuotteelta halutut toiminnot sekä niiden käytettävyyteen, saavutettavuuteen ja suorituskykyyn liittyvät seikat. Näiden määrittelyjen avulla projektin loppuunsaattaminen helpottuu, kun jo projektin alussa tiedetään haluttu taso, jota tavoitella. Näin vältetään usein tapahtuvalta tehtävälistan kasvamiselta työn edetessä.

Projektin kehitysvaihe alkoi grafiikkakirjaston valinnalla. Valinnan muodostaminen tapahtui nopeasti sopivan kirjaston löydyttyä. Kyseinen kirjasto tarjosi kevyen alustan, jonka päälle oli helppo kehittää projektin sovellusosiota. Ensi vaiheissa aikajananäkymän käyttöliittymää rakennettiin sovellusrunkoon sopivaksi. Tämän jälkeen komponentti muokattiin käyttämään ja muokkaamaan sovelluksen omaa dataa. Erinäisten toimintojen rakentaminen suoritettiin viimeiseksi. Toteutuksesta kerrotaan lisää projektin loppuraportissa (Liite 3).

4 Yhteenveto

Tämän opinnäytetyön puiteissa suunniteltiin ja toteutettiin sovellusosio rakennusalan projektinhallintasovellukseen visualisoimaan rakennusprojektien eri osia aikajanalla. Opinnäytetyön aikana työprosessista tuotettiin dokumentteja, jotka ovat luettavissa opinnäytetyön liitteissä.

Sovellusosio toteutettiin käyttäen moderneja verkkokehitysteknologioita. Käyttöliittymän toteutuksessa pääpaino oli JavaScript-pohjaisilla ratkaisuilla. Sovellusosion suunnittelussa ja toteutuksessa oli huomioitava sovelluksen kattavat käyttöympäristöt sisältäen niin mobiililaitteet kuin verkkoselaimet.

Projekti eteni halutulla tavalla projektisuunnitelman ja vaatimusmäärittelyn mukaisesti. Sovellusosion alustava versio tuli valmiiksi aikataulussa. Osio suunniteltiin ja toteutettiin helposti jatkokehitettäväksi, joten sen kehitys jatkuu tulevaisuudessa.

Lähteet

The Apache Software Foundation 2023. Apache Cordova -dokumentaatio. Viitattu 24.4.2023. <https://cordova.apache.org/docs/en/11.x/guide/overview/index.html>

Docker Inc. 2023. Docker-dokumentaatio. Viitattu 25.4.2023
<https://docs.docker.com/get-started/overview/>

Oracle 2023. My-SQL -dokumentaatio. Viitattu 25.4.2023
<https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>

Punchzee 2023. Kotisivu. Viitattu 24.4.2023. <https://www.punchzee.com/>

Rekord 2023. Rekord-dokumentaatio. Viitattu 25.4.2023.
<https://github.com/Rekord/rekord>

Stack Overflow 2022. Kehittäjäkysely. Viitattu 23.4.2023.
<https://survey.stackoverflow.co/2022/#section-most-popular-technologies-web-frameworks-and-technologies>

Vue 2023. Vue-dokumentaatio. Viitattu 25.5.2023
<https://v2.vuejs.org/v2/guide/index.html>

Projektisuunnitelma

Antton Heinonen

Gantt-kaaviosovellus tahtituotannon tueksi

Projektisuunnitelma

1 Johdanto

1.1 Johdanto ja projektin tavoite

Tahtituotanto on projektinhallinnallinen tuotantomalli, jossa työtehtävä jaetaan ajanjaksollisesti yhtä pitkiin paketteihin, jotka seuraavat toisiaan ilman keskeytyksiä. Tämän mallin hyödyntäminen rakennusalan projektienhallinnassa on vielä kohtuu uusi ilmiö, joten sen käyttöönoton helpottamista varten tehtyjä työkaluja ei toistaiseksi ole markkinoilla monta.

Tämä projekti toteutetaan osana suurempaa projektikokonaisuutta. Kyseessä on selaimissa ja mobiililaitteissa toimiva rakennusalan projektinhallintatyökalu Punchzee. Tämä projektisuunnitelma käsittelee Punchzeehen toteutettavaa osiota, joka perustuu tahtituotannon tueksi suunniteltuun gantt-kaavioon ja oheisiin työkaluihin.

1.2 Projektin laajuus ja lopputulos

Projektin tavoitteena on suunnitella ja toteuttaa asiakkaalle sovellusosio, jossa on toiminnallinen gantt-kaavio projektinhallintatyökalu Punchzeehen. Punchzee on selainpohjainen sovellus, joka toimii sekä mobiililaitteilla että verkkoselaimissa. Kaavion ja sen työkalujen tulee toimia Punchzeen ekosysteemissä ja hyödyntää jo olemassa olevia rakenteita ja tietokantoja.

1.3 Projektin rajoitukset

Sovellusosiota ei tarvitse testata selaimissa, jotka eivät tue ECMAScript 6:sta.

2 Organisaatio

Projektiryhmä koostuu yhdestä kehittäjästä sekä asiakasyrityksen vastuuhenkilöstä. Vastuuhenkilön tehtävänä on valvoa työn edistymistä, seurata aikatauluja ja tarvittaessa tarjota teknistä tukea, jotta projekti pysyy aikataulussa ja tavoitteet saavutetaan. Sovelluskehittäjä puolestaan keskittyy sovelluksen tekniseen toteutukseen, eli suunnittelee ja kehittää sovelluksen toiminnallisuutta sekä testaa ja korjaa mahdollisia virheitä.

3 Toteutussuunnitelma

3.1 Aikataulu

Projektin aloitus: tammikuu 2023

Alustava aikataulu: Viikot 1 ja 2 ovat suunnitteluvaihe. Tähän sisältyy ominaisuuksien suunnittelu, alustava ulkoasu sekä käytettävien teknologioiden vertailu. Viikot 3-13 ovat toteutusvaihetta, jonka tavoitteena on valmistaa toimiva beta-versio sovellusosioista. Lopullinen julkaisu on toisella vuosikvartaalilla.

3.2 Ohjelmisto

Projekti toteutetaan Punchzee-alustan puitteissa, joten ohjelmisto- ja teknologiavalinnat ovat pitkälti ennalta määrättyjä. Sovellusosio kirjoitetaan JavaScriptillä hyödyntäen Vue.js -ohjelmistokehystä (versio 2.6). Koko sovellus toimii Cordova mobiilisovellusten kehityskehyksessä (versio 11.0). Tämä mahdollistaa sovelluksen toimivuuden niin verkkoselaimissa kuin iOS- sekä Android-alustoilla. Taustajärjestelmä on kirjoitettu PHP-pohjaisella Laravel-kehyksellä (versio 7.0). Toteutettavan sovellusosion on toimittava yhdessä kyseisen taustajärjestelmän kanssa. Koko sovellus vielä paketoidaan Docker-sovelluksella.

Gantt-kaavion piirtämiseen käytetään SVG-kuvatiedostoja ja niiden rakentamiseen voidaan mahdollisesti käyttää ulkopuolista kirjastoa. Näiden vertailu ja valinta sisältyy suunnitteluvaiheeseen.

3.3 Tuotteen luovutus

Projektin lopputuotos luovutetaan asiakkaalle antamalla lähdekoodi asiakkaan GitLab-versionhallintajärjestelmään.

4 Projektinhallinta

4.1 Kommunikointi ja palaverit

Projekti hyödyntää ketterien menetelmien periaatteita, sillä pääasiallinen kommunikointikanava projektin osapuolien välillä ovat viikkopalaverit. Näissä käydään menneen viikon aikana työstetyt asiat ja suunnittellaan seuraavan viikon työtehtäviä. Näin asiakas myös pystyy antamaan jatkuvaa palautetta projektin kulusta ja ohjaamaan työskentelyä haluttuun suuntaan.

Viikkopalaverien lisäksi jatkuva yhteydenpito on mahdollista sähköpostien lisäksi myös projektin Microsoft Teams -viestintäalustalla.

4.2 Dokumentaatio ja versionhallinta

Projektiin kuuluvat dokumentit tallentaa asiakkaan Teams-alustalle. Viikottaisia työtehtäviä pidetään kirjalla VivifyScrum-palvelussa. Myös asiakkaalla on pääsy Vivifyyn, jolloin työnkulun seuranta on helppoa.

Koodin versionhallinta on asiakkaan GitLab-versionhallintajärjestelmässä. Kuluneen viikon aikana kirjoitettu koodi lähetetään GitLabiin ennen viikkopalaveria.

4.3 Laatuvaatimukset

Laadunvarmistuksen vuoksi ennen muuhun koodikantaan liittämistä projektin manageri vertaisarvioi kaiken tuotetun koodin. Sekä koodin että sovelluksen käyttökokemuksen tulee noudattaa vaatimusmäärittelyssä (Liite 2) linjattuja ohjeita.

4.4 Riskianalyysi

Projektin päällimmäiset riskit liittyvät käytettäviin uusiin teknologioihin ja niiden tuomiin haasteisiin. Sillä yhden hengen toteutusryhmän luonteesta johtuen ongelmien kohtaaminen on väistämätöntä, ja näiden haasteiden ratkominen tavanomaista hitaampaa. Näihin uusien teknologioiden tuomiin riskeihin on kuitenkin varauduttu hyvällä tukiverkolla. Tarvittaessa kokeneet kehittäjät pystyvät auttamaan ja näin

ehkäisemään projektin pitkittymisen. Uusien teknologioiden opettelun lisäksi ne tuovat mahdollisia haasteita niiden yhdistämisessä olemassa olevaan sovellusrunkoon. Esimerkiksi SVG-grafiikkakirjastoa valittaessa on huomioitava vaatimusmäärittelyssä annetut ehdot ja toimintaympäristöt tarkasti. Kun kehitys tapahtuu pääasiallisesti selainympäristössä, on mobiililaitteiden käyttömukavuus muistettava ja testattava säännöllisesti.

Vaatimusmäärittely

Antton Heinonen

Gantt-kaaviosovellus tahtituotannon tueksi

Vaatimusmäärittely

1 Johdanto

Projektin aiheena on suunnitella ja kehittää Punchzee-projektinhallintasovellukseen gantt-kaaviosovellus ja siihen liittyvät työkalut. Punchzee on rakennusalaan erikoistunut projektinhallinta-alusta, joka on toteutettu verkkoselainteknologioita käyttäen ja toimii kaikissa yleisimmissä selaimissa sekä sovelluksena Android- ja iOS-laitteilla. Sovellusta käyttävät sekä erinäiset projektimanagerit että kentällä työskentelevät työntekijät.

Sovellusosion pääasiallinen tarkoitus on näyttää projektiin kuuluvat työvaiheet ja -tehtävät visuaalisessa muodossa aikajanalla. Tätä aikajanaa käyttäen projektin johtavat henkilöt pystyvät muun muassa hallitsemaan projektikokonaisuutta, muokkaamaan työtehtäviä ja seuraamaan työn edistymistä.

2 Käyttökokemuksen vaatimukset

2.1 Toiminnot

Valmiin sovelluksen näkyvin ja tärkein osa on gantt-aikajana. Sen tulee näyttää projektin eri osat ja niiden ajalliset pituudet visuaalisesti. Näiden työtehtävien etenemisen seuraaminen tapahtuu myös suoraan aikajanalta. Projektin tarkastelun lisäksi näitä osia tulee myös pystyä hallitsemaan suoraan aikajanalta. Vaatimuksena on että työtehtäviä pystytään liikuttamaan ja järjestelemään uudestaan sekä niiden perustietoja, kuten nimeä, ajankohtaa tai tehtävälle määritelty väriä voidaan muokata gantt-osion kautta.

2.2 Käytettävyys

Sovelluksen tulee olla intuitiivinen ja helppokäyttöinen käyttöliittymältään, jotta sen käyttö onnistuu helposti sekä rakennustyömailla että toimistoissa kaikilta henkilöiltä. Myös käyttölaitteet vaihtelevat ja sovelluksen tuleekin tukea tietokoneen verkkoselaimia sekä tabletin että puhelimen sovelluksia.

2.3 Saavutettavuus

Sovellusosion suunnittelussa huomioidaan saavutettavuus kohtuullisesti. Täysin lukijalle suunniteltua sivua ei aikajanasta ole mahdollista tehdä, joten tyydytään tavanomaista lievempiin saavutettavuusratkaisuihin. Värit suunnitellaan helposti nähtävästi ja värisokeille sopiviksi.

2.4 Suorituskyky

Nykyaikasessa verkkokehityksessä sivun suorituskyky ja vasteaika ovat tärkeitä, jotta potentiaalinen käyttäjä ei välittömästi poistu sivulta. Näin ollen sovelluksen tulee toimia nopeasti ja suorittaa kommunikointi tietokannan kanssa ilman käyttäjälle aiheutuvaa viivettä jos mahdollista.

3 Tekniset vaatimukset

3.1 Toteutusteknologia

Sovellusosio toteutetaan noudattaen muussa sovelluksessa käytössä olevia teknologioita niin pitkälti kuin mahdollista. Pääasiallisena ohjelmointikielenä käytetään siis Vue.js-JavaScript-ohjelmistokehystä. Sovellus keskustelee tietokannan kanssa PHP-pohjaisen Laravel-ohjelmistokehityksen avulla. SVG-grafiikkoja varten voidaan projektin aikana valita käyttötarkoitukseen sopiva Javascript-kirjasto.

3.2 Käyttöympäristö

Sovelluksen tulee toimia yleisimmillä moderneilla verkkoselaimilla, jotka tukevat ECMAScript 6:sta.

Selain	Chrome	Firefox	Edge	Safari	Opera
Vanhin tuettu versio	51	52	14	10	38

Näiden selainten lisäksi sovelluksen pitää toimia kännykkäalustoilla sekä selaimissa että natiivisovelluksena. Kännykkäsovelluksen toteutus pohjautuu Apache Cordova-kehikseen, jonka avulla tämän projektin puitteissa mobiililaitteiden tukemisen varmistamiseksi pääasiallinen huomio kiinnittyy vain eri näyttöjen resoluutioiden tukemiseen.

Antton Heinonen

Gantt-kaaviosovellus tahtituotannon tueksi

Loppuraportti

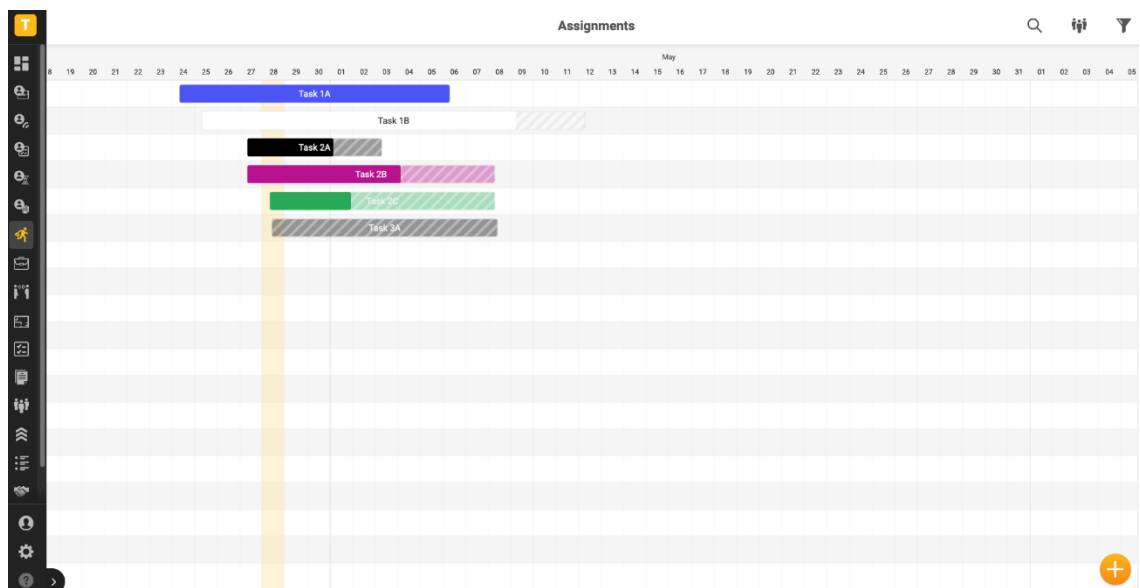
1 Johdanto

1.1 Johdanto ja projektin laajuus

Projektin tarkoituksena oli suunnitella ja valmistaa rakennusalan projektinhallintaan tarkoitettuun Punchzee-sovellukseen Gantt-kaavioon perustuva sovellusosio. Gantt-kaavio visualisoi aikajanalla projektin eri osia ja työvaihetta. Visualisoimisen lisäksi osioon kuuluvat näiden projektien hallintaan liittyvät työkalut.

1.2 Tavoite ja lopputulos

Projektin aikana suunniteltiin ja toteutettiin projektisuunnitelman ja vaatimusmäärittelyn mukainen sovellusosio (Kuva 1).



Kuva 1. Kuvakaappaus aikajananäkymästä

Aikajananäkymän piirtämiseen käytettiin hyväksi avoimen lähdekoodin Frappe Gantt - JavaScript-kirjastoa. Tämä kirjasto tarjosi hyvät työkalut SVG-vektorien piirtämiseen, sillä se on toiminnoiltaan hyvin kevyt ja on keskittynyt vain aikajanan visualisointiin. Tämä on toivottua, sillä tällainen mahdollisimman yksipiirteinen kirjasto tarjoaa parhaimman rungon jatkokehittää omaa tuotetta sen varaan. Ensi töikseen tästä grafiikkakirjastosta kloonattiin oma haara, jossa tähän projektiin kuuluva kehitys pitkälti tapahtui. Tällaisen uuden haaran luominen on paras käytäntömenetelmä, kun halutaan

itse jatkokehittää valmista kirjastoa omaan käyttöön. Tällöin pystyy itse tekemään haluamiansa muutoksia vaikuttamatta päähaaraan. Kun kyseessä on avoimen lähdekoodin kirjasto, on hyvien tapojen mukaista vielä mahdollisesti liittää tiettyjä ominaisuuksia alkuperäiseen haaraan, jos sen omistaja kokee tehdyt muutokset toivottaviksi. Tämän projektin puitteissa ei tällaista takaisinliittämistä ei vielä tehty, vaikka osa muutoksista olisi luonteeltaan voinutkin sopia siihen. Tämä asia jää tulevaisuuden harkintaan.

Vaatusmäärittelyssä kirjatut toiminnot saatiin toteutettua onnistuneesti. Aikajana on liitetty muun sovellukseen runkoon ja näyttää työtehtävät ja niiden progression halutulla tavalla. Työtehtävien ajankohtaa pystyy muokkaamaan siirtämällä työtehtävän palkkia, ja niiden pituutta voi muokata pidentämällä tai lyhentämällä vastaavaa palkkia raahamalla. Työtehtäviä voi myös järjestää uudelleen raahamalla palkkeja riviltä toiselle. Tämän lisäksi mobiilisovellukset huomioitiin tekemällä aikajanasta ja hallintatyökaluista kosketuksella ohjailtavat sekä erilaisiin näytön resoluutioihin oikein skaalautuvat (Kuva 2).

```
653 ✓ $.on(this.$svg, 'mousedown', '.grid-row, .today-highlight', (e) => {  
654     is_dragging = true;  
655     x_on_start = e.offsetX;  
656 ✓     if (this.$svg.parentElement) {  
657         this.$svg.parentElement.style.cursor = 'move';  
658     }  
659 });  
660 ✓ $.on(this.$svg, 'mousemove', '.grid-row, .today-highlight', (e) => {  
661 ✓     if (!is_dragging) {  
662         return;  
663     }  
664     const dx = e.offsetX - x_on_start;  
665     const parent_element = this.$svg.parentElement;  
666     if (!parent_element) return;  
667     parent_element.style.cursor = 'move';  
668     parent_element.scrollLeft -= dx * 1.5;  
669     x_on_start = e.offsetX;  
670 });  
671  
672 ✓ document.addEventListener('mouseup', (e) => {  
673 ✓     if (this.$svg.parentElement) {  
674         this.$svg.parentElement.style.cursor = 'default';  
675     }  
676     is_dragging = false;  
677     x_on_start = 0;  
678 });  
679 }  
680
```

Kuva 2. Esimerkki aikajanahan raahausominaisuuden lisäämisestä

2 Onnistumiset ja riskit

Projektisuunnitelmassa oli määritelty alustavan version valmistumiseksi ensimmäinen vuosineljännes. Tähän tavoitteeseen päästiin, vaikka aivan kaikkia toimintoja ei tähän ajanjaksoon vielä saatu valmiiksi. Tavoitteeseen pääsyssä avainasemassa olivat viikottaiset palaverit projektin managerin kanssa. Näissä palavereissa saatu tuki ja ohjaus auttoivat pitämään kehityksessä oikean suunnan sekä ratkomaan viikon aikana tulleita haasteita.

Projektin suunnitteluvaiheessa havaittiin kaksi riskitekijää, joista ensimmäinen oli uusien teknologioiden opetteluun tuoma epävarmuus. Tämä realisoitui useamman kerran projektin aikana, kun tekniset seikat viivyttivät kehitystyötä.

Projektisuunnitelmassa varauduttiin tähän hyvällä kehittäjien tukiverkolla, jonka avulla näitä ongelmia saatiin selvitettyä.

Toinen riskitekijä oli moninaisten eri laitteiden tukemisen tuomat haasteet. Kun kehitys tapahtuu pitkälti verkkoselaimessa tietokoneella, on mobiililaitteiden tukeminen muistettava kehityksen kaikissa vaiheissa. Tämä riski pysyi hallinnassa juurikin koska alusta lähtien sovellusosiota testattiin selaimen lisäksi mobiilinäkymässä.

Mobiilinäkymän testauksen avulla havaittiin myös haluttuja ominaisuuksia, kuten aikajananäkymän raahamisen perinteisten vierityspalkkien käyttämisen sijaan. Nämä ominaisuudet sitten lisättiin sovellukseen.

3 Tulevaisuuden parannukset

Projekti eteni pääpiirteittäin hyvin, ja aikataulu piti kohtuullisesti paikkansa.

Luonnollisesti parannettavaa tulevaisuutta varten kuitenkin jäi vielä.

Projektin edetessä tuli ilmi kuinka tärkeässä osassa suunnitteluvaihe on projektin etenemisen kannalta. Vaikka projektisuunnitelma sekä vaatimusmäärittely olivat tehty, olisivat niiden tekemiseen voinnut käyttää enemmän aikaa ja huolta. Niiden täsmällisyyden puute jätti tilaa tulkinnanvaraisuudelle. Esimerkiksi toiminnalliset vaatimusmäärittelyt olisivat voineet olla tarkemmat. Nyt ne antoivat liian isot raamit, joita käytännössä piti täsmentää koko projektin ajan viikottaisissa palavereissa. Myös projektisuunnitelmassa määriteltyn aikatauluun olisi voinnut kiinnittää enemmän huomiota: siinä ei ollut huomioitu uusien teknologioiden opiskeluun kuluva aikaa. Myös toteutusvaiheen eri osien aikataulutus olisi tuonnut selkeyttä työnteon tahdittamiseen.

Teknisistä seikoista parannusta olisi kaivannut ulkoisen grafiikkakirjaston käyttö ja sen suunnittelu. Tässä projektissa käytetystä avoimen lähdekoodin kirjastosta tehtiin oma haara, jossa kirjastoa jatkokehitettiin. Kaikkien toimintojen osalta ei kuitenkaan aina ollut selvää tulisiko kehitys tapahtua nimeenomaan tämän kirjaston haarassa, vaiko itse projektin omassa kehitysympäristössä. Tämän projektin puitteissa näiden kahden menetelmän välillä ei ole suuria eroja, mutta jatkokehityksen kannalta yhtenäinen toimintasuunnitelma tämän seikan osalta on tärkeää.