

# TITUS-PROJEKTIN SOVELLUSRAPORTTI

*Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen  
ja Enni Stylman*

Versio 1.2.0

4.7.2018

# Sisältö

<b>Allekirjoitukset</b>	<b>1</b>
<b>Muutoshistoria</b>	<b>2</b>
<b>Projektiorganisaatio</b>	<b>3</b>
<b>Tiivistelmä</b>	<b>3</b>
<b>1 Johdanto</b>	<b>4</b>
<b>2 Käsitteet</b>	<b>5</b>
2.1 Tausta . . . . .	5
2.2 Ohjelmointi- ja merkintäkielet . . . . .	5
2.3 Työkalut ja tiedostomuodot . . . . .	6
<b>3 Sovelluksen tarpeet ja tavoitteet</b>	<b>7</b>
3.1 Tarpeet . . . . .	7
3.2 Tavoitteet ja niiden toteutuminen . . . . .	7
<b>4 Sovelluksen rakenne ja toiminta</b>	<b>9</b>
4.1 TIM-järjestelmän rakenne . . . . .	9
4.2 Taulukkoliitännäinen . . . . .	11
4.3 Taulukkoliitännäisen LaTeX-tulostus . . . . .	11
4.4 Taulukkoeditori . . . . .	11
4.5 Liitteiden leimaaminen . . . . .	12
4.6 Liitteiden yhdistäminen . . . . .	12
4.7 Pöytäkirjaotteiden ottamistoiminto . . . . .	12
4.8 Kokouskutsusta pöytäkirjaksi -muunnin . . . . .	12
4.9 Muut toiminnot . . . . .	13
<b>5 Suoritettujen testausten yhteenveto</b>	<b>14</b>
<b>6 Tavoitteiden toteutuminen</b>	<b>15</b>
6.1 Vaatimusten toteutuminen . . . . .	15
6.2 Toteutusratkaisujen muutokset projektin aikana . . . . .	15
6.3 Heikot toteutusratkaisut . . . . .	16
6.4 Jatkokehitysideat . . . . .	16
6.5 Testauksessa löydetyt jatkokehitysideat . . . . .	17
<b>7 Ylläpitäjän tiedot ja ohjeet</b>	<b>19</b>
<b>8 Yhteenveto</b>	<b>19</b>
<b>Lähteet</b>	<b>20</b>

## Allekirjoitukset

---

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö			Rami Pasanen
Tilaaajan edustaja			Vesa Lappalainen
Ohjaaja			Jukka-Pekka Santanen

---

# Muutoshistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.0.1	20.4.2018	Dokumentille hahmoteltiin pohja.	RP
0.0.2	23.4.2018	Dokumentin rakennetta tarkennettiin.	RP
0.0.3	26.4.2018	Täydennettiin lukua sovelluksen taustasta, tarpeista ja tavoitteista.	RP
0.0.4	4.5.2018	Kirjoitettiin sovelluksen rakenteesta ja tavoitteiden toteutumisesta.	RP
0.0.5	5.5.2018	Kirjoitettiin luvut heikoista toteutusratkaisuksista ja jatkokehitysideoista.	RP
0.0.6	5.5.2018	Kuvattiin pöytäkirjaotteiden ottamistoiminnon ja kokouskutsusta pöytäkirjaksi -muuntimen toimintaa.	RP
0.1.0	7.5.2018	Lisättiin taulukko koko projektin vaatimusten toteutumisesta ja poistettiin muistiinpanoja tekstin seasta.	RP
0.1.1	11.5.2018	Paranneltiin dokumenttia vastaavan ohjaajan palautteen perusteella. Lisättiin TIMin käyttöliittymän rakenteesta kertova kaavio lukuun 4.1.	RP
0.1.2	13.5.2018	Kirjoitettiin liitteiden leimaamista kuvaava luku 4.4.	VN
0.1.3	24.5.2018	Kirjoitettiin liitteiden yhdistämistä kuvaava luku 4.5.	VN
0.1.4	6.6.2018	Kirjoitettiin taulukkoliitännäisen LaTeX-tulostusta kuvaava luku 4.3.	VN
0.1.5	10.6.2018	Kirjoitettiin luvut <i>Testaus</i> ja <i>Toteutusratkaisujen muutokset projektin aikana</i> .	RP
0.1.6	10.6.2018	Kirjoitettiin tiivistelmä ja johdanto.	RP
0.2.0	13.6.2018	Täydennettiin testauksen tulokset jatkokehitysideoihin. Kirjoitettiin luku <i>Ylläpitäjän tiedot ja ohjeet</i> sekä yhteenveto.	RP
0.2.1	16.6.2018	Kirjoitettiin kuvaus preambleen toteutetuista toiminnoista.	ES
0.2.2	18.6.2018	Kuvattiin taulukkoliitännäisen ja taulukkoeditorin rakenne ja toiminta.	ML
0.2.3	20.6.2018	Parannettiin lukuja 1-4.9 vastaavan ohjaajan palautteen perusteella.	RP
0.2.4	23.6.2018	Parannettiin lukuja 5-8 vastaavan ohjaajan palautteen perusteella.	RP
1.0.0	25.6.2018	Korjattiin lähdeviitteet ja kuvattiin toimintojen vaatimat oikeudet.	RP
1.1.0	30.6.2018	Parannettiin dokumentin kirjoitusasua vastaavan ohjaajan palautteen perusteella.	RP
1.1.1	30.6.2018	Määritettiin tulostusasetukset, korjattiin taulukoiden Markdown-muotoilua.	ES
1.2.0	4.7.2018	Lisättiin lukuun 5 viittaus lukuun 6.5. Lisättiin luvun 6 johdantoon tieto projektin toimintojen ottamisesta tuotantokäyttöön.	RP

# Projektiorganisaatio

## Projektiryhmä

- Leinonen Matti, matti.o.leinonen@student.jyu.fi
- Lindholm Ronja, ronja.lindholm@gmail.com
- Naukkarinen Visa, visa.naukkarinen@gmail.com
- Pasanen Rami, rami\_p@outlook.com
- Stylman Enni, enni.stylman@gmail.com

## Tilaaajan edustajat

- Hatanpää Eija, eija.hatanpaa@jyu.fi
- Lappalainen Vesa, vesa.t.lappalainen@jyu.fi
- Markkanen Jaana, jaana.a.markkanen@jyu.fi

## Ohjaajat

- Lehtinen Mika, mika.k.lehtinen@student.jyu.fi
- Santanen Jukka-Pekka, santanen@mit.jyu.fi

## Yhteystiedot

- Sähköpostilistat
  - titus@korppi.jyu.fi
  - titus\_opetus@korppi.jyu.fi
- Sähköpostiarkistot
  - <https://korppi.jyu.fi/kotka/servlet/list-archive/titus/>
  - [https://korppi.jyu.fi/kotka/servlet/list-archive/titus\\_opetus/](https://korppi.jyu.fi/kotka/servlet/list-archive/titus_opetus/)

## Tiivistelmä

Titus-projekti kehitti kevään 2018 aikana Jyväskylän yliopiston TIM-järjestelmään toimintoja, jotka helpottavat informaatioteknologian tiedekunnan tiedekuntaneuvoston esityslistojen ja pöytäkirjojen laadintaprosessia. Projektia edeltäneen työnkulun työkaluja voidaan uusien toimintojen myötä korvata suoraan TIM-järjestelmällä. Samalla kehitettiin TIM-järjestelmään uusi joustava ja tehokas taulukkomuoto sekä taulukkoeditori. Dokumentti kuvaa projektissa toteutetun sovelluksen päämääriä, taustoja ja tavoitteita, sovellukseen kehitettyjen osien rakennetta ja toimintaa, sovelluksen tavoitteiden toteutumista sekä jatkokehitysideoita.

**Avainsanat:** Jatkokehitys, käyttöliittymä, käyttöohjeet, oppimisympäristö, projekti, sovellus, tavoitteet, testaus, TIM, vaatimukset, vaatimusmäärittely.

# 1 Johdanto

Titus-projekti kehitti Jyväskylän yliopiston informaatioteknologian tiedekunnassa kehitettyä TIM-järjestelmää paremmin soveltuvaksi tiedekuntaneuvoston kokousdokumenttien laadintaan. Tiedekuntaneuvoston kokousdokumenttien laadintaprosessissa käytettiin useita erilaisia työkaluja, mistä johtuen prosessia pystyi suoraviivaistamaan. Lisäksi dokumenttien tarkastelu oli ajoittain vaikeaa johtuen suurista määristä erillisiä liitteitä ja joidenkin tiedostomuotojen heikosta soveltumisesta esimerkiksi mobiililaitteille. Projektissa kehitettiin TIM-järjestelmää siten, että kokousdokumentit laaditaan suoraan TIM:iin.

Kokousdokumenttien laatimisen helpottamisen lisäksi projektin tavoitteena oli jatkokehittää TIM-järjestelmää erityisesti uuden taulukkomuodon ja -editorin myötä. Kokousdokumenttien laatimisen helpottamiseksi toteutetuista toiminnoista tehtiin mahdollisimman yleiskäyttöisiä, jotta niitä voisi hyödyntää myös muuten kuin kokousdokumenttien yhteydessä.

Sovellusraportti esittelee kehitetyn sovelluksen taustat ja tavoitteet, sovelluksen rakenteen ja toiminnan sekä teknisen toteutuksen. Luvussa 2 käydään läpi keskeiset dokumentissa käytetyt käsitteet. Luku 3 kuvaa sovelluksen tarpeet ja tavoitteet. Luvussa 4 kuvataan TIM-järjestelmän rakenne ja kehitettyjen toimintojen tekninen toteutus. Luku 5 kuvaa lyhyesti toimintojen testaukset ja tulokset. Luku 6 kuvaa sovellukselle asetettujen tavoitteiden toteutumisen, heikot toteutusratkaisut ja jatkokehitysideat. Luku 7 esittää ylläpitäjien tietojen ja ohjeiden sijainnin.

Dokumentin laadinnassa on otettu mallia Timantti-sovellusprojektin sovellusraportista<sup>[2]</sup>. Titus-projektin dokumenteista on hyödynnetty vaatimusmäärittelyä<sup>[3]</sup>, kehitettyjen toimintojen käyttöohjeita<sup>[4]</sup>, testausuunnitelmia ja -raportteja<sup>[5], [6], [7], [8], [9], [10], [11] ja [12]</sup> sekä jatkokehittäjien ohjeita<sup>[13], [14], [15] ja [16]</sup>. Projektin läpivienti kuvataan projektiraportissa<sup>[1]</sup>.

## 2 Käsitteet

Luvussa määritellään dokumentissa käytetyt olennaiset käsitteet.

### 2.1 Tausta

Dokumentissa käytetään seuraavia kohdealueen käsitteitä:

- **Asiakohta** tai **listakohta** sisältää kaikki yksittäiseen asialistan kohtaan sisältyvät tiedot kokouskutsussa tai pöytäkirjassa esityksineen ja liitteineen sekä pöytäkirjassa päätöksineen.
- **Asialista** tai **esityslista** on kokouksessa käsiteltävien asiakohtien listaus, josta ilmenee kokouksessa käsiteltävät asiakohdat ja niiden alustava käsittelyjärjestys kokouksen alussa. Informaatioteknologian tiedekunnan tiedekuntaneuvostossa käytetään termiä asialista.
- **Kokouskutsu** on asiakirja, jolla kokous kutsutaan koolle ja joka sisältää tiedon kokouksessa käsiteltävistä asiakohdista. Usein kokouskutsun yhteydessä välitetään asialista.
- **Liite** on erillinen dokumentti, joka olennaisilta osin liittyy tietyn asiakohdan käsitteelyyn. Liitteen koko sisältöä ei ole tiedostomuodon tai sisällön laajuuden vuoksi mielekästä kirjata auki asialistaan tai pöytäkirjaan.
- **Pöytäkirja** on kokouskutsun pohjalta laadittu virallinen asiakirja, johon on kirjattu muun muassa kokouksen osanottajat, tehdyt päätökset ja keskeiset keskustelunaiheet. Pöytäkirjan järjestys noudattaa kokouksen käsittelyjärjestystä.
- **Pöytäkirjaote** on pöytäkirjasta otettu yksittäinen asiakohta päätöksineen ja liitteineen.
- **Taulukko** on esitysmuoto ja tietorakenne. Esitysmuodossa haluttu tieto esitetään riveittäin ja sarakkeittain.
- **Taulukkomuoto** on tapa ilmaista taulukon esitysmuodon rakenne valitulla esityskielellä.
- **Tiedekuntaneuvosto** on tiedekunnan hallinnossa ylin päättävä elin.
- **TIM** on Jyväskylän yliopistossa kehitetty vuorovaikutteinen oppimisympäristö.
- **Vesileima** on tiedekuntaneuvoston kokousten pöytäkirjojen liitteisiin joka sivulle erikseen lisättävä tunniste, joka sisältää liitteen tunnistetiedot, mukaanlukien kokouksen päivämäärän, asiakohdan numeron ja liitteen numeron.

### 2.2 Ohjelmointi- ja merkintäkielet

Dokumentissa käytetään seuraavia projektille olennaisia ohjelmointi- ja merkintäkieliin liittyviä käsitteitä:

- **AngularJS** on JavaScript-pohjainen kehys yksisivuisten sovellusten kehittämiseen.
- **camelCase** viittaa tapaan nimetä esimerkiksi luokkia ja aliohjelmia lähdekoodissa siten, että nimen jokainen sana aloitetaan isolla kirjaimella.
- **CSS** on tyylitiedostokieli, jolla määritellään HTML:llä tehtyjen WWW-sivujen ulkoasu.
- **HTML** on WWW-sivujen laadintaan käytetty merkintäkieli.
- **JavaScript** on ohjelmointikieli, jota käytetään ensisijaisesti selainpuolen toiminnallisuuden lisäämiseen WWW-sivuilla.
- **jQuery** on JavaScript-kirjasto, joka yksinkertaistaa selainpuolen ohjelmointia ja erityisesti dynaamisten verkkosivujen tekemistä.
- **JSDoc** on merkintäkieli, jolla dokumentoidaan JavaScript-lähdekoodia.

- **Makro** on sääntö, jolla muutetaan syöte haluttuun muotoon.
- **Markdown** on merkintäkieli, jota TIM-dokumentit käyttävät.
- **Python** on yleiskäyttöinen ohjelmointikieli, jolla TIMin palvelinpuolen toiminnallisuus on pääosin kirjoitettu.
- **PEP8**<sup>[17]</sup> määrittelee Pythonia kirjoittaessa suositeltuja ohjeita ja sääntöjä.
- **SCSS** on CSS:ksi kääntyvä tyylitiedostokieli, jonka tarkoitus on laajentaa CSS:n ominaisuuksia.
- **TypeScript** on JavaScriptin laajennos, jonka ensisijainen tarkoitus on lisätä kieleen tuki tyyppitykselle.
- **YAML** on tiedon tallentamiseen ja siirtämiseen tarkoitettu kieli.

## 2.3 Työkalut ja tiedostomuodot

Dokumentissa käytetään seuraavia työkaluihin ja tiedostomuotoihin liittyviä käsitteitä:

- **AJAX** on joukko tekniikoita, joita hyödynnetään dynaamisia nettiselaimilla toimivia sovelluksia toteutettaessa.
- **Docker** on sovellus, jonka avulla ajetaan sovelluksia virtuaalisoidussa säilössä.
- **Flask** on Python-ohjelmointikielillä WWW-sovellusten luomista varten tehty ohjelmistokehys.
- **Kontti** tai **säilö** on Dockerin kautta suoritettava sovellus. Docker ajaa kontin sovellusta virtualisoidussa käyttöjärjestelmässä, mikä tarjoaa etuja muun muassa turvallisuuteen liittyen.
- **LaTeX** on tekstin ja matemaattisten kaavojen esittämiseen tarkoitettu ladontajärjestelmä. TIM-järjestelmä hyödyntää LaTeXia muuttaessaan dokumentteja PDF-tiedostoiksi.
- **Nginx** on palvelinohjelmisto, jolla isännöidään WWW-palvelimia.
- **PDF** on dokumenttien alustariippumattomaan esittämiseen tarkoitettu tiedostomuoto.
- **Plugin** tai **liitännäinen** on TIM-järjestelmän osa, joka lisää toiminnallisuutta dokumentteihin.
- **Preamble** (*johdanto*) on TIM-järjestelmässä pohjatiedosto, joka mahdollistaa koko kansion dokumenttien määritysten tekemisen kerralla. Preamblen sisältö lisätään kansiossa olevien dokumenttien alkuun aina, kun niitä tarkastellaan.
- **PyCharm** on erityisesti Python-kielen kirjoittamiseen käytetty ohjelmointiympäristö. Sitä käytetään myös TIMin kehittämiseen.



## 3 Sovelluksen tarpeet ja tavoitteet

Luvussa kuvataan tarpeet, joiden vuoksi sovellusta kehitettiin. Lisäksi kerrotaan tarpeista johdetut sovellukselle asetetut tavoitteet.

### 3.1 Tarpeet

Informaatioteknologian tiedekunnan tiedekuntaneuvoston kokousdokumenttien eli kokouskutsujen ja pöytäkirjojen laadintaprosessissa käytetään nykyään useita erilaisia työkaluja. Lisäksi laadittujen kokousdokumenttien tarkastelu on vaikeaa johtuen suurista määristä erillisiä liitteitä ja joidenkin tiedostomuotojen heikosta soveltumisesta esimerkiksi mobiililaitteille. Tarkemmin laadintaprosessia kuvataan projektin vaatimusmäärittelydokumentissa<sup>[3]</sup>.

Projektissa selvitettiin, että TIM-järjestelmän kehittämisen ja käyttöönoton avulla kokousdokumenttien tarkastelua voi helpottaa ja laadintaprosessia suoraviivaistaa. Useita nykyisessä prosessissa käytettyjä työkaluja on mahdollista korvata TIM-järjestelmällä siten, että dokumentit laaditaan suoraan TIM-järjestelmään. Samalla helpotetaan dokumenttien tarkastelua upottamalla erilliset liitteet suoraan kokousdokumentteihin. TIM-järjestelmän dokumentit myös skaalautuvat hyvin mobiililaitteille.

Kokousdokumenttien laatimisen ja tarkastelun helpottamisen lisäksi yksi tarve sovelluksen kehittämiseksi oli itse TIM-järjestelmän kehittäminen eteenpäin. Vaikka sovelluksen kehittäminen on tapahtunut IT-tiedekunnan tiedekuntaneuvoston tarpeiden kontekstissa, on toiminnoista pyritty tekemään mahdollisimman yleiskäyttöisiä siten, että kehitystyö hyödyttäisi myös TIMin käyttäjiä laajemmin. Erityisesti yleiskäyttöisyys koskee TIMiin kehitettyä uutta taulukkomuotoa ja taulukkoeditoria. Sovelluksen tavoitteita, toimintoja ja toteutumista kuvataan tarkemmin luvussa 3.2.

### 3.2 Tavoitteet ja niiden toteutuminen

Sovelluksen kokonaistavoitteena oli helpottaa kokousdokumenttien laadintaa ja niiden tarkastelua. Sovelluksen kehittämisen tavoitteena oli toteuttaa TIM-järjestelmään seuraavat muutokset:

- Luoda uusi taulukkomuoto, joka tukee solujen yhdistämistä ja kehittyneitä tyyliominaisuuksia.
- Luoda yksinkertainen graafinen käyttöliittymä uuden muodon mukaisten taulukoiden laatimiseen.
- Luoda toiminto PDF-liitteiden upottamiseksi TIM-dokumentteihin sekä niiden vesileimaamiseksi dokumenttiin upottamisen yhteydessä.
- Luoda toiminto, joka yhdistää dokumenttiin upotetut PDF-liitteet yhdeksi isoksi liitteeksi arkistointia varten.
- Lisätä makroja ja preambleja siten, että esityslistojen ja pöytäkirjojen laatiminen käy nopeasti valmiita pohjia hyödyntämällä.
- Helpottaa pöytäkirjojen otteiden ottamista ja tulostamista.
- Helpottaa pöytäkirjapohjan luomista kokouskutsudokumentin pohjalta.

Kaikki edellä luetellut tavoitteet toteutuivat projektissa. Tavoitteiden tarkemmat kuvaukset, vaatimukset ja prioriteetit löytyvät vaatimusmäärittelydokumentista<sup>[3]</sup>.

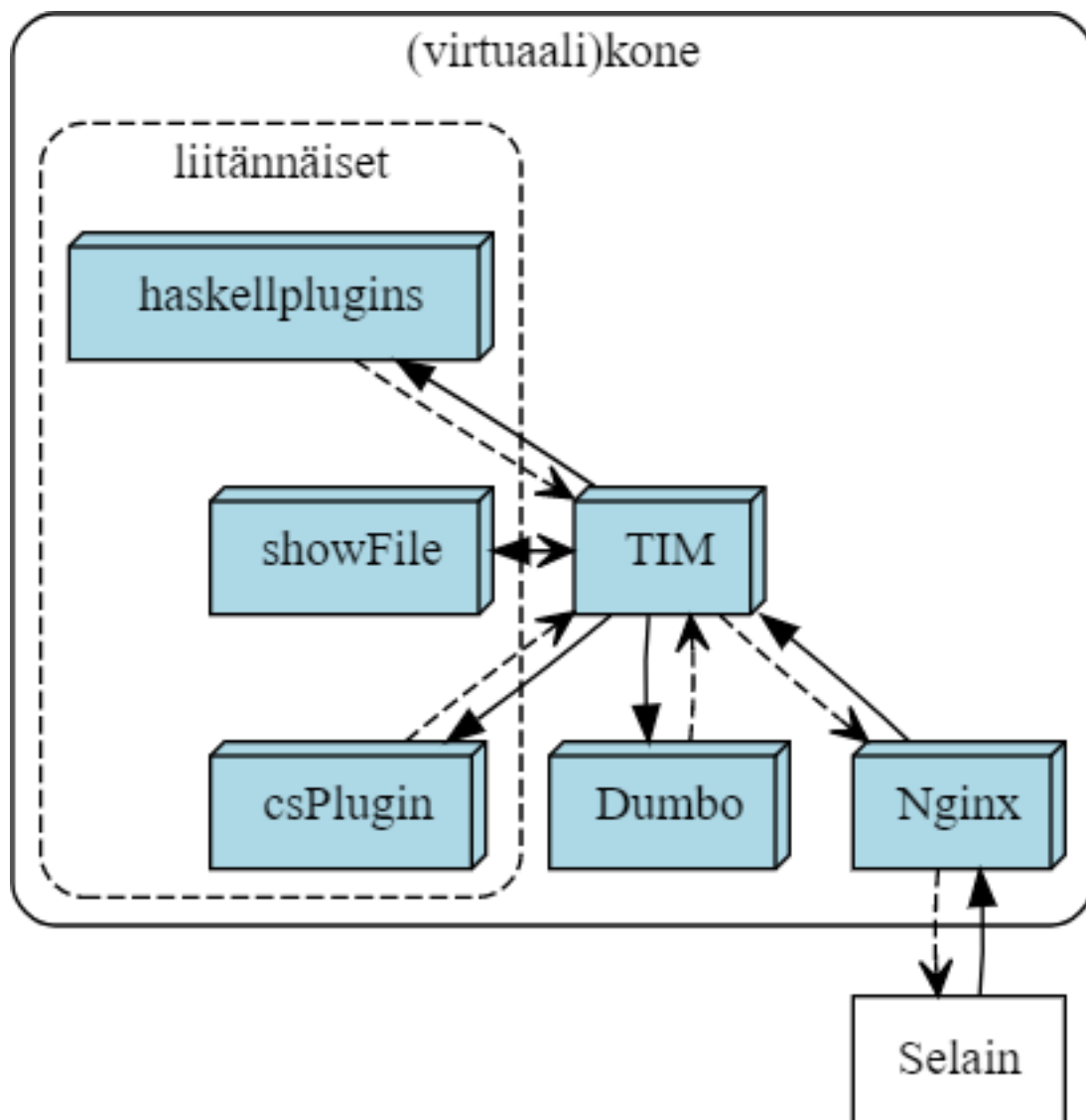
Uuden muodon mukaisten taulukoiden luominen onnistuu yksisoluisen taulukon luovan painikkeen kautta. Tämän jälkeen taulukkoon voi lisätä rivejä ja sarakkeita sekä solujen sisältöä voi muokata. Muilta osin taulukkoeditorin toiminnot sovittiin tilaajan kanssa jatkokehitykseen.

## 4 Sovelluksen rakenne ja toiminta

Luvussa kuvataan sovelluksen kokonaisrakenne. Lisäksi kuvataan yleisellä tasolla sovellukseen kehitettyjen toimintojen toimintaa. Toteutusratkaisujen yksityiskohtiin ei mennä, sillä yksityiskohdat on kuvattu jatkokehittäjien ohjeissa<sup>[13], [14], [15]</sup> ja [16] ja lähdekoodissa<sup>[18]</sup>. Käyttöliittymä on esitelty käyttöohjeissa<sup>[4]</sup>.

### 4.1 TIM-järjestelmän rakenne

TIM-järjestelmän kokonaisrakenne on esitetty kuvassa 4.1. Järjestelmä koostuu keskenään kommunikoivista Docker-konteista, jotka näkyvät kuvassa tummennettuina laatikoina.



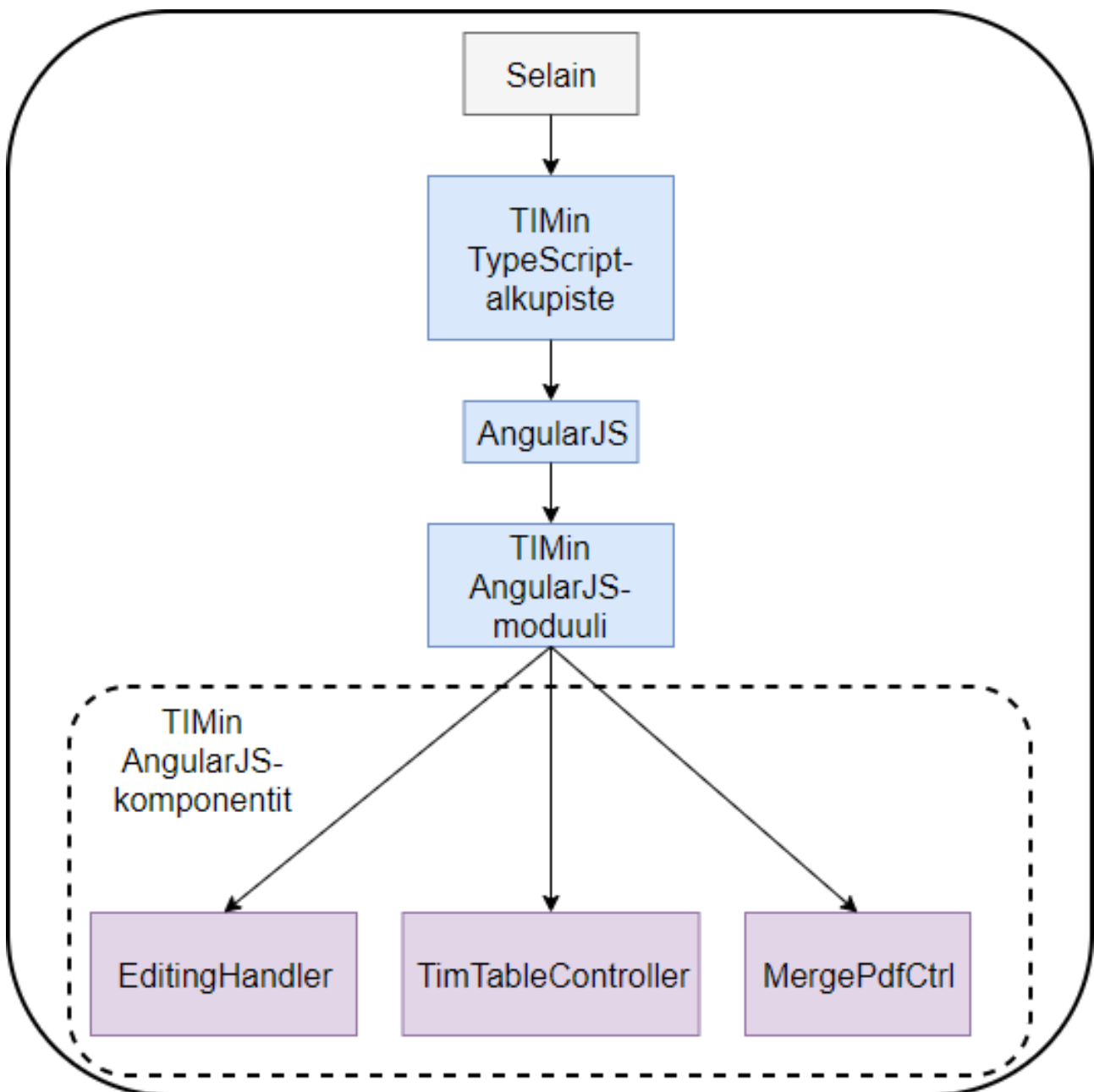
Kuva 4.1. TIM-järjestelmän kokonaisrakenne.

Nginx-palvelin vastaanottaa selaimelta tulevan pyynnön ja välittää sen TIM-kontille. Käyttäjän katsoman dokumentin HTML pyydetään tarvittaessa Dumbolta. Dokumentin hyödyntämien liitännäisten sisällön näyttämiseksi tehdään pyyntöjä liitännäiskonteille, jotka palauttavat dokumentin liitännäisiä hyödyntävän sisällön halutussa muodossa. Esimerkiksi dokumenttia katseltaessa liitännäisiä hyödyntävä sisältö palautetaan usein HTML-muodossa.

Osa liitännäisistä ajetaan oman kontin sijaan TIM-kontissa erityisesti, jos niillä on tarve päästä laajalti käsiksi TIM-järjestelmän muihin toimintoihin. Lisää tietoja TIMistä löytyy TIMin kehitysohjeista<sup>191</sup>, joista myös kuva 4.1 on lainattu.

Projektissa kehitetyt toiminnot liittyvät palvelinpuolella TIM-konttiin, ja ne toteutettiin Pythonilla Flask-sovelluskehystä hyödyntäen. Kehitetty taulukkoliitännäinen ja siihen liittyvät taulukkoeditorin toiminnot ajetaan TIM-kontissa. Liitteiden leimaaminen ja yhdistäminen, pöytäkirjaotteiden luominen sekä pöytäkirjapohjan luominen kokouskutsusta tapahtuvat myös TIM-kontissa käyttäjän selaimen pyynnöstä.

TIM-järjestelmän selainpohjaisen käyttöliittymän rakenne on esitetty kuvassa 4.2. Käyttöliittymä koostuu TypeScriptillä toteutetuista AngularJS-komponenteista. TypeScript-koodi käännetään suorittamista varten JavaScript-koodiksi.



Kuva 4.2. TIMin selainpohjaisen käyttöliittymän rakenne.

Merkittävin projektissa kehitetty AngularJS-komponentti on taulukkoeditori. Taulukkoeditoria varten myös TIMin muokkausvalikkoon lisättiin painike taulukkoeditorin avaamiselle ja

sulkemiselle. Liitteiden yhdistämistoiminnolle luotiin uutena komponenttina dialogi, jossa on painike liitteiden yhdistämiseksi ja latausanimaatio painiketta painettaessa. Pöytäkirjatoimintojen suorittamiselle ja liitteiden yhdistämisen dialogin avaamiselle lisättiin painikkeet jo valmiiksi olemassa olleeseen TIMin sivuvalikkoon.

Luvuissa 4.2-4.9 kerrotaan tarkemmin Titus-projektissa TIMiin kehitettyjen toimintojen toiminnasta. Projektissa ei tehty muutoksia TIM-järjestelmän tietokantaan.

## 4.2 Taulukkoliitännäinen

Taulukkoliitännäinen toimii omalla YAML-pohjaisella TimTable-tilukkomuodolla. Sen avulla taulukoita voidaan kirjoittaa aiempaa ilmaisuvoimaisemmin ja helpommin ylläpidettävästi. Taulukkoa voidaan editoida kokonaisuudessaan kappale-editorilla, jolloin sen rakennetta ja tyylejä voi muokata monipuolisesti. Taulukkoeditori tarjoaa rajoitetummat muokausmahdollisuudet.

Selainpuolella liitännäisen AngularJS-komponentti vastaa taulukon renderöinnistä HTML-muotoon. Palvelinpuolella on Pythonilla toteutettu reitti taulukon datan saamiseksi selaimen. Lisäksi Python-koodissa on reittejä liittyen taulukkoeditorilla tapahtuvaan taulukon muokkaamiseen.

Tarkempi kuvaus taulukkoliitännäisestä on luettavissa jatkokehittäjien ohjeesta<sup>[13]</sup>.

## 4.3 Taulukkoliitännäisen LaTeX-tulostus

LaTeX-tulostuksen moduuli saa taulukon JSON-muodossa taulukkoliitännäiseltä palvelinpuolella. Taulukosta muodostetaan oliorakenne, jossa jokainen olio tietää omat muotoilunsa ja sisältönsä sekä vastaa niiden muunnoksesta LaTeX-kielelle. Moduuli palauttaa kokonaisen LaTeX-tilukun takaisin taulukkoliitännäisen moduulille, josta se edelleen välitetään selaimelle tulostettavaksi. Tulostuksessa käytetään lisäksi tulostusmallinetta, joka sisältää taulukon, muun sivun tarvitsemat paketit ja muut LaTeX-asetukset.

Tarkempi kuvaus LaTeX-tulostuksen toiminnasta ja tulostuksen moduulista on luettavissa taulukkoliitännäisen jatkokehittäjien ohjeessa<sup>[13]</sup>.

## 4.4 Taulukkoeditori

Taulukkoeditori mahdollistaa TimTable-tilukkomuodolla kirjoitettujen taulukoiden nopean muokkaamisen. Taulukkoeditorilla voidaan muokata solujen sisältöjä sekä lisätä taulukon loppuun rivejä ja sarakkeita.

Taulukkoeditorin käyttöliittymä on toteutettu AngularJS-komponenttina. Palvelinpuolen Python-koodissa on toteutettu reitit taulukon solun sisällön saamiseksi ja tallentamiseksi sekä rivien ja sarakkeiden lisäämiseksi taulukkoon.

Tarkempi kuvaus taulukkoeditorista on luettavissa jatkokehittäjien ohjeesta<sup>[13]</sup>. Taulukkoeditorin käyttö vaatii käyttäjältä dokumenttiin *edit*-oikeuden.

## 4.5 Liitteiden leimaaminen

Liitteiden leimaaminen hyödyntää TIMin alkuperäistä tiedostojen palvelimelle tallentamista (engl. upload). Kun upload-toimintoa käytetään liitemakron kanssa, JavaScriptillä toteutettu kappale-editori saa selaimelta leimaamiseen tarvittavat tiedot kappaleen sisällöstä ja preamble-dokumenteista, sekä lähettää ne liitetiedoston kera upload-reitille.

Palvelinpuolen Pythonilla toteutettu upload-moduuli vastaanottaa datan ja kutsuu pdf-työkalumoduulia, joka suorittaa leimatiedoston luomisen ja liitteen leimaamisen käyttäen komentoriviohjelmaa. Upload-moduuli palauttaa JSON-vastauksena leimatun liitteen linkin kappale-editorille, joka edelleen välittää sen selaimen käyttäjälle.

Toiminnon käyttäminen vaatii käyttäjältä *edit*-oikeuden dokumenttiin. Tarkemmat kuvaukset leimaamisprosessista ja siihen osallistuvista moduuleista löytyvät liitetoimintojen jatkokehittäjien ohjeista<sup>[14]</sup>.

## 4.6 Liitteiden yhdistäminen

Liitteiden yhdistämisessä selain lähettää palvelimelle dokumentin sisällön kappaleiksi jaettuna. Palvelinpuolella liitemakron sisältävistä kappaleista saadaan TIM-tiedostopalvelimelle tallennettujen liitteiden polut, jotka kerätään listaan ja annetaan parametrina liitteet yhdistävälle funktiolle. Funktio tarkastaa liitetiedostojen oikeellisuuden, yhdistää ne väliaikaiskansioon käyttäen komentoriviohjelmaa ja palauttaa yhdistetyn tiedoston. Yhdistetty pdf-tiedosto välitetään selaimelle, joka palauttaa käyttäjälle linkin luotuun tiedostoon.

Toiminnon käyttäminen vaatii käyttäjältä *edit*-oikeuden dokumenttiin. Yhdistämisen toteutus ja toiminta kuvataan tarkemmin liitetoimintojen jatkokehittäjien ohjeissa<sup>[14]</sup>.

## 4.7 Pöytäkirjaotteiden ottamistoiminto

Pöytäkirjaotteiden ottamistoiminnossa selain lähettää palvelimelle pyynnön ottaa otteet pöytäkirjadokumentista. Palvelinpuolella toiminto hakee selaimen pyytämän dokumentin, luo otteen jokaisesta asiakohdasta ja palauttaa selaimelle vastauksen, jolla selain tietää siirtyä otteista luotuun koostedokumenttiin. Toiminnon käyttäminen vaatii *manage*-oikeuden pöytäkirjadokumenttiin ja sen sisältävään kansioon. Tarkemmat tiedot toiminnon toiminnasta löytyvät jatkokehittäjien ohjeesta<sup>[15]</sup>.

## 4.8 Kokouskutsusta pöytäkirjaksi -muunnin

Kokouskutsusta pöytäkirjaksi -muuntimessa selain lähettää palvelimelle pyynnön muuttaa kokouskutsudokumentti pöytäkirjapohjaksi. Palvelinpuolella muunnin hakee pyydetyn kokouskutsudokumentin, tekee siitä kopion haluttuun kansioon, merkitsee kopiodokumentin pöytäkirjaksi ja ohjaa selaimen uuteen dokumenttiin. Toiminnon käyttäminen vaatii käyttäjältä dokumenttiin *manage*-oikeuden. Tarkemmin muuntimen toimintaa ja siihen liittyviä lähdekooditiedostoja kuvataan jatkokehittäjien ohjeessa<sup>[16]</sup>.

## 4.9 Muut toiminnot

Tiedekuntaneuvoston hakemistorakenteen ylimmällä tasolla olevaan `templates`-kansioon sijoitettu `preamble`-dokumentti sisältää tiedekuntaneuvoston käyttöön laadittuja mallipohjia. Tiedostoon on sijoitettu muunmuassa asiakkoitaisten läsnäolijataulukoiden mallipohja niin vanhassa taulukkomuodossa kuin `timTable`-taulukkomuodossakin. `Preamble` lisää kappaleeditoriin `Plugins`-välilehdelle `TDK`-painikkeen ja sen alle komennot `Läsnä` ja `LäsnäTimTable`, joiden avulla voi lisätä vakiomuotoillun läsnäolijataulukkopohjan tiedekuntaneuvoston pöytäkirjojen asiakkohtiin.

## 5 Suoritettujen testausten yhteenveto

Projektissa kehitetyille ominaisuuksille suoritettiin toiminnallinen testaus ja käytettävyyss-testaus. Toiminnallinen testaus suoritettiin myös toiseen kertaan testitapauksille, jotka ensimmäisellä kerralla paljastivat puutteita, jotka projektiryhmä korjasi. Käytettävyyss-testauskertoja oli kolme. Eri kerroilla testattiin sovelluksen eri toimintoja eri koehenkilöillä.

Toiminnallisen testauksen ensimmäisellä suorituskerralla löydettiin kymmeniä puuttellisia tai virheellisiä toiminnallisuuksia. Toiminnoista löydettiin puutteellisia oikeustarkistuksia ja epäselviä ilmoituksia virhetilanteissa. Projektiryhmä korjasi näistä tärkeimmät, ja korjausten toimivuus varmistettiin toistamalla toiminnallinen testaus muokatuille toiminnoille. Loppujen puutteellisten tai virheellisten toimintojen korjaaminen sovittiin tilaajan kanssa jatkokehitykseen.

Käytettävyyss-testaus paljasti sovelluksesta useita kymmeniä käytettävyyssongelmia. Havainnot tehtiin laajasti TIM-järjestelmän eri toimintoihin liittyen, joista osa oli kehitetty ennen projektia. Projektin toiminnoista eniten huomioita liittyi taulukkoeditorin käytettävyyteen. Osa näistä oli myös toiminnallisia ongelmia. Käytettävyyss-testauksessa löydettyjen ongelmien korjaaminen sovittiin pääosin jatkokehitykseen. Projektiryhmä korjasi vain muutaman tärkeäksi ja nopeasti korjattavaksi katsotun ongelman.

Luvussa 6.5 kerrotaan testauskertojen olennaisimmat havainnot. Tarkat kuvaukset suoritetuista testauskerroista ja niiden havainnoista löytyy testausuunnitelmista ja

-raporteista<sup>[5], [6], [7], [8], [9], [10], [11] ja [12]</sup>.



## 6 Tavoitteiden toteutuminen

Luvussa käsitellään yleisellä tasolla sovellukselle asetettujen tavoitteiden ja vaatimusten toteutumista. Lisäksi pohditaan puutteellisia toteutusratkaisuja ja jatkokehitysideoita. Vaatimuksia ei käydä yksityiskohtaisesti läpi, sillä niitä sekä niiden tilaa ja prioriteetteja kuvataan vaatimusmäärittelydokumentissa<sup>[3]</sup>.

Projektissa kehitetty sovellus täyttää kaikki sovellukselle asetetut tavoitteet, jotka esitettiin luvussa 3.2. Luvussa 6.3 käsitellään heikkoja toteutusratkaisuja ja luvuissa 6.4-6.5 jatkokehitysideoita. Projektissa kehitetyt toiminnallisuudet on otettu tuotantokäyttöön.

### 6.1 Vaatimusten toteutuminen

Projektin aikana toteutettiin kaikki keskeiset toiminnot ja niiden välttämättömiksi määritellyt vaatimukset. Tärkeiksi määritellyistä vaatimuksista toteutettiin selvästi suurin osa. Mahdollisiksi ja ideoiksi määritellyistä vaatimuksista toteutettiin osa. Toteutetuista välttämättömistä ja tärkeistä vaatimuksista suurin osa myös varmistettiin toteutetuiksi testauksessa.

Eniten toteuttamattomia vaatimuksia liittyy taulukkoeditoriin, jonka vaatimuksista suurin osa rajattiin jatkokehitysideoiksi jo projektin alkuvaiheessa. Vaatimukset kokonaisuutena elivät paljon projektin aikana. Osa toteuttamattomista vaatimuksista oli määritetty tärkeiksi projektin alussa, mutta myöhemmin toiset vaatimukset korvasivat niiden toiminnallisuuden. Näiden vaatimusten prioriteettiä laskettiin ja ne jätettiin toteuttamatta tai sovittiin tilaajan kanssa jatkokehitysideoiksi.

Taulukossa 1 kuvataan koko projektin osalta, montako vaatimusta toteutettiin ja montako jätettiin toteuttamatta. Eri prioriteettien merkitysten kuvaukset ja toimintokohtainen vaatimusten toteutuminen kuvataan vaatimusmäärittelydokumentissa<sup>[3]</sup>.

Taulukko 1: Koko projektin vaatimusten toteutuminen.

Tila / Prioriteetti	Välttämätön	Tärkeä	Mahdollinen	Idea	Hylätty	Yhteensä
Löytyy TIIM:istä	13	4	2	1	0	20
Hyväksytty	0	0	0	0	0	0
Testattu	25	55	17	7	0	104
Toteutettu	7	13	9	0	0	29
Kesken	0	5	1	0	0	6
Toteuttamatta	0	12	9	29	0	50
Ei toteuteta	0	4	6	58	15	83
<b>Yhteensä</b>	<b>45</b>	<b>93</b>	<b>44</b>	<b>95</b>	<b>15</b>	<b>292</b>

### 6.2 Toteutusratkaisujen muutokset projektin aikana

Projektin aikana tehdyt merkittävimmät toteutusratkaisujen muutokset liittyvät taulukkolii-tännäiseen ja taulukkoeditoriin. Alkuperäisessä toteutuksessa taulukkolii-tännäinen muutti YAML-muotoisen taulukon HTML:ksi palvelinpuolella Python-koodin avulla. Toteutuksen ollessa kesken HTML:ksi renderöinti siirrettiin selainpuolelle, jossa TypeScriptillä toteutettu

AngularJS-komponentti renderöi YAML-taulukon HTML:ksi.

Taulukkoeditorin osalta suurin toteutusratkaisuihin liittyvä muutos oli solun sisällön syötekentän siirtäminen solutasolta taulukon tasolle. Alun perin syötekenttä oli solun sisällä elementtinä, mutta lopullisessa toteutuksessa on olemassa vain yksi syötekenttä, jota siirretään solujen välillä, kun aktiivista solua vaihdetaan.

Muiden toimintojen osalta toteutusratkaisujen muutokset olivat verrattain pieniä ja liittyivät vaatimusten muuttumiseen ja tarkentumiseen.

### 6.3 Heikot toteutusratkaisut

Pääosin projektissa tehdyt toteutusratkaisut ovat olleet toimivia ja osoittautuneet projektin aikana hyvin skaalautuviksi. Ongelmallisiin toteutusratkaisuihin pyrittiin puuttumaan jo varhain projektin aikana, ja joidenkin toimintojen koodia refaktoroiitiin useampaan kertaan ennen lopullista versiota.

Yksi puutteellinen toteutus on taulukkoliitännäisen LaTeX-tulostus. Puutteellisuus ei johdu itsessään heikosta toteutusratkaisusta, vaan rajallisesta ajasta. Taulukkoliitännäinen tukee sekä HTML:ää että LaTeXia, jotka ovat hyvin erilaisia kieliä. Aika ei riittänyt HTML-muotoa vastaavan LaTeX-tulostuksen toteuttamiseen, mikäli se olisi edes ollut teknisesti mahdollista. HTML:llä oli liitännäisessä korkeampi prioriteetti, joten taulukkomuodon käyttämä YAML-rakenne suunniteltiin ensisijaisesti HTML:n mukaan. Suurin osa taulukkoliitännäisen toiminnallisuudesta kuitenkin toimii myös LaTeX-tulostuksessa, mutta kaikkia ominaisuuksia ei saatu toimimaan, kuten solujen reunojen muotoa ja paksuutta.

Yhtenä käyttäjän näkökulmasta heikkona toteutusratkaisuna voi pitää kokousdokumentteihin liittyviä makroja ja preambleja sekä liitteiden lisäämistä. Esimerkiksi kokousten päivämäärien, kellonaikojen ja numeroiden muuttaminen vaatii preamble-tiedostojen manuaalista muokkaamista kappale-editorilla, mikä ei välttämättä luonnistu peruskäyttäjältä. Myös liitteiden lisäämisessä pitää käyttäjän olla tarkkana, että liitemakroa käyttää kappale-editorissa oikein. Näitä toimintoja varten olisi voinut kehittää käyttöliittymän, mikäli aikaa olisi ollut enemmän ja tilaajan tekninen edustaja olisi siten toivonut. Näiden toimintojen käyttöä on kuitenkin pyritty helpottamaan laatimalla niille käyttöohjeet, joiden avulla käytettävyysongelmat toimintojen kanssa voidaan välttää koulutuksen ohella. Käyttöohjeet on erityisesti suunnattu käyttäjille, joilla ei ole lainkaan aiempaa kokemusta järjestelmästä.

Joidenkin projektissa tehdyn esitutkimuksen perusteella harvoin käytettyjen yksittäisten toimintojen toteutusratkaisuihin saattaa myös piillä heikkouksia. Esimerkiksi jos pöytäkirjan liitteet halutaan yhdistää ilman leimaamista vaatimusmäärittelyn vaatimuksen 6.9 mukaisesti, tämä vaatii liitteitä sisältävien kappaleiden manuaalista muokkaamista TIMin kappale-editorilla.

### 6.4 Jatkokehitysideat

Luvussa kuvataan lyhyesti sovelluksen yleiset jatkokehitysideat sekä testauksessa löydetty ongelmat, joiden korjaaminen sovittiin tilaajan kanssa jatkokehitykseen. Jatkokehitysideoihin liittyvät vaatimukset löytyvät vaatimusmäärittelydokumentista<sup>[3]</sup>.

Taulukkoliitännäisen osalta keskeisimmät jatkokehitysideat ovat solun tyyppin määrittäminen, erilaiset editorit erilaisille solutyypeille ja laskutoiminnot Microsoftin Excel-sovelluksen tapaan.

Lisäksi liitännäinen tulisi refaktoroida toimimaan oliomaisemmin, jotta taulukko-, sarake- ja rivitasojen tyyliä saataisiin toimimaan kattavammin.

Taulukkoeditorille kirjattiin runsaasti jatkokehitysideoita. Taulukon rakenteen muokkauksen osalta jatkokehitysideoita ovat rivien ja sarakkeiden paikan vaihtaminen sekä uusien rivien ja sarakkeiden lisääminen muuallekin kuin taulukon loppuun. Kokonaisuksi uusiksi toiminnoksi jatkokehitykseen sovittiin vaatimusmäärittelyssä määritelty taulukon rakenteen editori ja taulukkolaskentaominaisuudet.

Projektissa toteutettu liitteiden käsittely soveltuu IT-tiedekunnan tiedekuntaneuvoston kokousdokumenttien laadintaan. Projektissa kehitetyt liitetoiminnot suunnattiin tosin vain tiedekuntaneuvoston kokousdokumenttien laadintaprosessia varten, eivätkä ne siten ole kovin yleiskäyttöisiä. Liitteiden käsittelyyn jäi siten runsaasti potentiaalisia jatkokehityskohteita, joiden myötä toiminnoista tulisi yleiskäyttöisempiä. Yksittäisinä jatkokehitysideoina liitteiden yhdistämisen dialogiin tulee lisätä listaus dokumentin sisältämistä yhdistettävistä liitteistä, ja liitteiden leimadatalle voisi kannattaa tehdä oma luokka `pdfTools`-moduuliin.

Pöytäkirjojen laatimisen osalta merkittävin jatkokehitysidea on asiakkohtaisten läsnäolotaulukoiden automaattinen generointi pöytäkirjan alussa sijaitsevan suuren läsnäolotaulukon pohjalta. Kokouskutsusta pöytäkirjaksi -muunnin voisi lisätä nämä asiakkohtaiset läsnäolotaulukot pöytäkirjapohjan luomisen yhteydessä. Tällä hetkellä asiakkohtaiset läsnäolotaulukot joudutaan tekemään manuaalisesti, joskin asiaa nopeuttamaan tehtiin valmis painike preambleen.

Pöytäkirjaotteiden ottamistoimintoon ei kirjattu jatkokehitysideoita.

## 6.5 Testauksessa löydetty jatkokehitysideat

Testauksessa löydettiin useita ongelmia, joiden korjaaminen sovittiin resurssien puutteen vuoksi jatkokehitykseen. Korjattavista ongelmista vain osa liittyy projektissa kehitettyihin toimintoihin. Muut ongelmat liittyvät projektin ulkopuolella kehitettyihin TIMin toimintoihin.

Toiminnallisessa testauksessa löydettyistä ongelmista seuraavien korjaaminen sovittiin jatkokehitykseen:

- Ääkköset muuttuvat heksakoodimuotoon taulukon datalohkossa, kun solun sisältöä muokataan taulukkoeditorilla.
- Liitteiden leimaaminen ei toimi, jos liitteen nimessä on alaviivoja. Väliaikaisena korjauksena leimaaminen poistaa alaviivat liitteiden nimistä, mutta tämä ei ole optimaalinen toimintatapa.
- Käyttäjä ei saa varoitusta liitteitä yhdistäessä, jos liitemakrolla on lisätty muunlaisia tiedostoja kuin PDF-liitteitä. Muunlaisia tiedostoja ei huomioida yhdistämisessä.
- *Merge attachments* -painike on näkyvässä, vaikka dokumenttiin ole muokkausosoikeuksia.
- Taulukko ei tulostu oikein LaTeX-tulostuksessa, jos `rowspan`-attribuutin arvo ylittää rivien määrän.
- LaTeX-tulostuksessa eivät toimi kaikki rivi- ja saraketyylit, jotka toimivat HTML:ssä.

Käytettävyydestestauksessa löydettyistä projektissa kehitettyjen toimintojen ongelmista seuraavien korjaaminen sovittiin jatkokehitykseen:

- Käyttäjälle ei ole selvää, että taulukkoeditori tallentaa muutokset automaattisesti.

- Solun sisällön syöttökentän sulkemiseen ei ole selvää keinoa.
- Solun sisällön kehittyneen editorin voi avata monta kertaa.
- Taulukon reunassa avattu solun sisällön syöttökenttä peittää osittain sarakkeen lisäyspainikkeen.
- Solun sisällön kehittyneessä editorissa ei ole *advanced view* oletuksena päällä.
- md:-merkintää ei lisätä solun sisältöön automaattisesti, jos käytetään markdown-merkintöjä.
- Taulukkoeditorissa lisättyä riviä ja saraketta ei voi poistaa taulukkoeditorista käsin.
- Taulukko leventyy pitkästä tekstisisällöstä siten, että solut jäävät avatun hampurilaisvalikon alle.
- TIMin yleisissä käyttöohjeissa ei ole linkkiä taulukon ohjeisiin.
- Taulukkoliitännäisen käyttöohjeissa ei ole ohjeita tietojen poistamiseen.
- Solun sisällön syöttökenttä ei ilmesty vasempaan yläkulmaan, jos solun koko vaihtuu muokattaessa.
- Kokouskutsun onnistuneesta muuntamisesta pöytäkirjaksi ei ilmoiteta käyttäjälle tarpeeksi selkeästi.

Käytettävyydestä löydetyistä projektissa kirjoitettujen ohjeiden ongelmista seuraavien korjaaminen sovittiin jatkokehitykseen:

- Taulukkoliitännäisen käyttöohjeissa ei lue tarpeeksi selvästi, missä tyylejä voi asettaa.
- Käyttöohjeissa ei mainita, että yksittäisen pöytäkirjan asiakohdan otteen saa vain ottamalla otteet kaikista asiakohdista.
- Käyttöohjeissa ei kerrota tarpeeksi tarkasti, mikä osa lisätyn liitteen kappaleen koodista tulee poistaa, jos haluaa poistaa leiman liitteestä.

Käytettävyydestä löydetyistä projektin ulkopuolella kehitettyjen TIM-järjestelmän toimintojen ongelmista seuraavien korjaaminen sovittiin jatkokehitykseen:

- Kappale-editorin esikatselua ei voi vierittää. Siten se ei toimi hyvin matalaresoluutiolla näytöillä ja suurilla kappaleilla.
- Kursiivimuotoilu rikkoutuu aloittavan merkin jälkeisestä välilyönnistä. Kursiivin asettamistoiminto ei huomioi valitun tekstin alussa olevaa välilyöntiä, vaan sallii muotoilun rikkoutumisen.
- Sivunvalikon avaaminen on vaikeaa, jos näytön resoluutio on pieni ja hampurilaisvalikko avattu.
- Kappale-editorin ikkuna on oletuksena niin pieni, että projektissa lisätty *TDK*-painike ei ole näkyvässä.

## 7 Ylläpitäjän tiedot ja ohjeet

TIM-sovellukselle on jo ennen projektia ollut olemassa kehitysohjeet <sup>[20]</sup>. Projektin aikana toteutetut TIMin lisätoiminnot eivät ole vaikuttaneet näiden ohjeiden ajantasaisuuteen. Projektissa toteutetuille toiminnoille on laadittu omat, erilliset jatkokehittäjän ohjeet <sup>[21]</sup>.

## 8 Yhteenveto

Titus-projekti kehitti Jyväskylän yliopiston informaatioteknologian tiedekunnassa kehitettyä TIM-järjestelmää paremmin soveltuvaksi tiedekuntaneuvoston kokousdokumenttien laadintaan. Tavoitteena oli korvata TIM-järjestelmällä useita projektia edeltävässä kokousdokumenttien laadinnassa käytettyjä työkaluja sekä siten yksinkertaistaa ja helpottaa dokumenttien laadintaprosessia. Lisäksi kehitettiin uusi ilmaisuvoimaisempi taulukkomuoto ja taulukkoeditori.

Kokonaisuutena projektin voi sanoa onnistuneen sovelluksen kehityksen osalta, sillä projektisuunnitelmassa sovellukselle asetetut tavoitteet täyttyivät, kaikki välttämättömät vaatimukset toteutettiin, ja tärkeistä vaatimuksista toteutettiin selvästi suurin osa. Toimintoja oli paljon erilaisia, ja vaatimukset elivät paljon projektin aikana. Joillain vaatimuksilla kesti pitkään tarkentua siten, että ne pystyttiin toteuttamaan.

Tavoitteiden toteutumisesta huolimatta sovellus ei testauksen perusteella ole vapaa käytettävyysongelmista, mutta niitä ehkäisemään on laadittu yksityiskohtaiset käyttöohjeet. Toiminnallisuustestauksessa havaituista ongelmista jäi korjausten jälkeen jäljelle lähinnä monimutkaiset virheviestit joissain LaTeX-tulostuksen ongelmatilanteissa. Testaukset suoritettiin projektissa melko myöhään, minkä vuoksi kaikkia löydettyjä ongelmia ei ehditty korjaamaan projektin aikana. Lisäksi tilaajan edustajilla oli keskenään erilaisia mielipiteitä ongelmien olemassaolosta ja korjausehdotuksista. Ongelmien korjaaminen on kuitenkin kirjattu jatkokehitykseen.

## Lähteet

- [1] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin projektiraportti, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/projektiraportti>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [2] Miika Kujala, Topi Latva-Salo, Tuomas Porvali ja Marja Similä, Timantti-sovellusprojekti: sovellusraportti, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2017/timantti-sovellusraportti>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2017.
- [3] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin vaatimusmäärittely, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/vaatimusmaarittely>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [4] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektissa kehitettyjen toimintojen käyttöohjeet, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/kayttoohjeet>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [5] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin testaussuunnitelma, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/testaussuunnitelma>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [6] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin toiminnallisuustestauksen testaussuunnitelma, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/dokumentit/testaus/toiminnallisuustestaus>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [7] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin käytettävyydestestauksen testaussuunnitelma, <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/dokumentit/testaus/kayttavyydestestaus>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [8] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin toiminnallisuustestauksen raportti, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/dokumentit/testaus/toiminnallisuustestausraportti>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [9] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin toisen toiminnallisuustestauksen raportti, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/dokumentit/testaus/toisen-toiminnallisuustestausraportti>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [10] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin läsnäolijataulukoiden käytettävyydestestauksen raportti, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/dokumentit/testaus/kayttavyydestestausraportti>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [11] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin käytettävyydestestauksen raportti - Pöytäkirjan toiminnot ja taulukkoplugin, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/dokumentit/testaus/kayttavyydestestausraportti>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [12] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin käytettävyydestestauksen raportti - taulukkomuoto ja liitteiden käsittely, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/dokumentit/testaus/kayttavyydestestausraportti>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.

- [13] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Tim-Table - Ohjeet jatkokehittäjille, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [14] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Jatkokehittäjien ohje TIM-liitetoiminnoille, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [15] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Jatkokehittäjän ohjeet: Pöytäkirjaotteiden ottamistoiminto, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/kayttoohjeet/jatkokehittajan-ohjeet/>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [16] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Jatkokehittäjän ohjeet: Kokouskutsusta pöytäkirjaksi -muunnin, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/kayttoohjeet/jatkokehittajan-ohjeet/>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [17] Guido van Rossum, Barry Warsaw and Nick Coghlan, PEP8 – Style Guide for Python, saatavilla osoitteesta <https://www.python.org/dev/peps/pep-0008/>, Python Software Foundation. Viitattu 16.3.2018.
- [18] TIMin kehittäjät, Titus-projektin TIM-haaran lähdekoodi, saatavilla osoitteesta <https://gitlab.com/Rampastring/tim>, GitLab, 2018.
- [19] TIMin kehittäjät, Johdatus TIMin kehitykseen, saatavilla osoitteesta <https://tim.jyu.fi/view/tim>, Jyväskylän yliopisto, informaatioteknologian tiedekunta. Viitattu 4.5.2018.
- [20] TIMin kehittäjät, TIMin kehitysohjeet, saatavilla osoitteesta <https://tim.jyu.fi/view/tim/TIMin-kehitysohjeet/>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.
- [21] Matti Leinonen, Ronja Lindholm, Visa Naukkarinen, Rami Pasanen ja Enni Stylman, Titus-projektin toimintojen jatkokehittäjän ohjeet, saatavilla osoitteesta <https://tim.jyu.fi/view/kurssit/tie/proj/2018/titus/kayttoohjeet/jatkokehittajan-ohjeet/>, Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2018.