Paatti-sovellusprojekti

Tapio Keränen
Toni Salminen
Jari Salokangas
Lauri Satokangas

Sovellusraportti

Julkinen
Versio 0.5.0
4.6.2012

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Hyväksyjä	**Päivämäärä**	**Allekirjoitus**	**Nimenselvennys**
Projektipäällikkö	__.__.2012		
Tilaaja	__.__.2012		
Ohjaaja	__.__.2012		

Tietoa dokumentista

Tekijät:

- Tapio Keränen (TK) t.tapio.keranen@student.jyu.fi 040-8464981
- Toni Salminen (TS) toni.a.j.salminen@student.jyu.fi 040-5378086
- Jari Salokangas (JS) jari.p.t.salokangas@student.jyu.fi 040-8652014
- Lauri Satokangas (LS) lauri.n.satokangas@student.jyu.fi 040-5735358

Dokumentin nimi: Paatti-projekti, Sovellusraportti
Sivumäärä: 68
Tiedosto: paatti_sovellusraportti.tex

Tiivistelmä: Paatti-projekti kehitti Jyväskylän yliopiston psykologian laitokselle prototyypin tietojärjestelmästä, jolla kuntoutetaan iäkkäitä, mielenterveyspotilaita ja työuupumuksesta kärsiviä sekä tuetaan kyseisiin aihealueisiin liittyvää tutkimusta. Tietojärjestelmässä on kaksi käyttöliittymää, joista tutkijan käyttöliittymällä hallitaan kuntoutettavien kuntoutusohjelmia sekä kuntoutettavan käyttöliittymällä kuntoutettava suorittaa tutkijan laatimaa kuntoutusohjelmaa. Sovellusraportissa kuvataan projektissa toteutettu tietojärjestelmän prototyyppi käyttöliittymän, tietorakenteiden, ohjelmointikäytänteiden sekä tavoitteiden toteutumisen osalta.
Avainsanat: Arkkitehtuuri, jatkokehitys, komponentit, käytänteet, käyttöliittymä, käyttöohje, rajapinnat, rakenne, sovellus, tavoitteet, testaus, tietojärjestelmä, tietokanta.

Muutoshistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.0.1	11.5.2012	Dokumentti luotu ja pohjaksi otettu projektisuunnitelman pohja.	JS
0.0.2	12.5.2012	Muokattu avainsanoja sekä lukuja ”Johdanto” ja ”Ohjelmointikäytänteet”. Kirjoitettu luvut ”Tutkijan käyttöliittymä”, ”Kuntoutettavan käyttöliittymä”, ”Sovelluksen rakenne” (osittain), ”Tavoitteiden toteutuminen” (osittain) sekä ”Yhteenveto”.	JS
0.0.3	13.5.2012	Lisätty kuvat lukuihin ”Tutkijan sovelluksen näkymäkartta”, ”Kuntoutettavan sovelluksen näkymäkartta”, ”Tietojärjestelmän asiakaspalvelinarkkitehtuuri” ja ”Sovelluksen tietokanta”. Kirjoitettu lukuja ”Sovelluksen tietokanta”, ”Tunnetut puutteet ja virheet” ja ”Jatkokehitysideat” sekä muokattu lukua ”Toteutusratkaisujen kehittyminen”.	JS
0.0.4	14.5.2012	Lisätty kuvat lukuihin ”Sovelluksen päänäkymä” ja ”Oma profiili -näkymä”. Muokattu lukuja ”Sovelluksen tietokanta”, ”Tunnetut puutteet ja virheet” sekä ”Jatkokehitysideat”.	JS
0.1.0	15.5.2012	Toimitettu versio ohjaajan tarkastettavaksi.	JS
0.1.1	17.5.2012	Muokattu avainsanoja sekä lukuja ”Johdanto”, ”Termit”, ”Tutkijan käyttöliittymä”, ”Kuntoutettavan käyttöliittymä”, ”Tietojärjestelmän rakenne” (osittain) ja ”Yhteenveto”.	JS
0.1.2	18.5.2012	Lisätty luku ”Tietojärjestelmän tietokanta”. Muokattu lukuja ”Tietojärjestelmän rakenne” (osittain), ”Ohjelmointikäytänteet” ja ”Tavoitteiden toteutuminen”.	JS
0.1.3	19.5.2012	Poistettu ylimääräisiä termejä. Muokattu lukuja ”Tapahtumien luonti ja muokkaus” sekä ”Tietojärjestelmän tietokanta”.	JS
0.1.4	20.5.2012	Muokattu lukua ”Jatkokehitysideat”.	JS

Versio	Päivämäärä	Muutokset	Tekijät
0.1.5	21.5.2012	Muokattu sovellusraporttia kokonaisvaltaisesti. Kirjoitettu lukuja ”Tietojärjestelmän rakenne ja rajapinnat” sekä ”Tietojärjestelmän komponentit”.	JS
0.1.6	22.5.2012	Lisätty kuvat lukuihin ”Sovelluksen näkymien rakenne (kuntoutettava)” sekä ”Tietojärjestelmän rakenne ja rajapinnat”.	JS
0.2.0	22.5.2012	Toimitettu versio ohjaajan tarkastettavaksi.	JS
0.2.1	26.5.2012	Muokattu sovellusraporttia kokonaisvaltaisesti. Lisätty tietokantataulujen esimerkit.	JS
0.2.2	27.5.2012	Lisätty käyttöliittymäkuvia. Muokattu luvun ”Tietojärjestelmän rakenne ja rajapinnat” kuvaa. Laskettu vaatimukset uudelleen. Lisätty lähteitä ja viittauksia niihin.	JS
0.2.3	28.5.2012	Lisätty käyttöliittymäkuvia. Jaettu alaluvut ”Tunnetut puutteet ja virheet” sekä ”Jatkokehitysideat” useammiksi alaluvuiksi.	JS
0.3.0	29.5.2012	Toimitettu versio ohjaajan tarkastettavaksi.	JS
0.3.1	30.5.2012	Lisätty käyttöohjeet ylläpitäjälle. Muokattu sovellusraporttia ohjaajan palautteen perusteella.	JS, TK, TS
0.4.0	31.5.2012	Toimitettu versio ohjaajan tarkastettavaksi.	JS
0.4.1	4.6.2012	Muokattu sovellusraporttia ohjaajan havaitsemien muutosehdotuksien pohjalta.	JS
0.5.0	4.6.2012	Toimitettu versio ohjaajan, teknisen ohjaajan ja Tero Kallioisen tarkastettavaksi.	JS

Tietoa projektista

Paatti-projekti kehitti Jyväskylän yliopiston psykologian laitokselle prototyypin tietojärjestelmästä, jolla tuetaan iäkkäiden, mielenterveyspotilaiden ja työuupumuksesta kärsivien itsenäistä elämää ja kuntoutumista, sekä tuetaan kyseisiin aihealueisiin liittyvän tutkimuksen tekemistä.

Tekijät:

- Tapio Keränen (TK) t.tapio.keranen@student.jyu.fi 040-8464981
- Toni Salminen (TS) toni.a.j.salminen@student.jyu.fi 040-5378086
- Jari Salokangas (JS) jari.p.t.salokangas@student.jyu.fi 040-8652014
- Lauri Satokangas (LS) lauri.n.satokangas@student.jyu.fi 040-5735358

Tilaaja:

- Jukka Kaartinen jukka.kaartinen@jyu.fi 0400-248148
- Päivi Lappalainen paivi.k.lappalainen@psyka.jyu.fi 040-8054192
- Raimo Lappalainen raimo.lappalainen@psyka.jyu.fi 050-4432349
- Heikki Lyytinen heikki.lyytinen@psyka.jyu.fi 050-5524892

Ohjaajat:

- Matti Lehtinen matti.k.lehtinen@jyu.fi 041-4564118
- Jukka-Pekka Santanen santanen@mit.jyu.fi 040-8053299
- Outa Valkama outa.j.valkama@gmail.com 045-3490910

Yhteystiedot:

- Sähköpostilistat: paatti@korppi.jyu.fi, paatti_opetus@korppi.jyu.fi
- Sähköpostiarkistot: https://korppi.jyu.fi/list-archive/paatti/, https://korppi.jyu.fi/list-archive/paatti_opetus/

Sisältö

1 Johdanto **1**

2 Termit **3**
2.1 Aihealueen termit 3
2.2 Vaatimusmäärittelyn termit 4
2.3 Tekniikat ja kehitysvälineet 5

3 Tutkijan käyttöliittymä **6**
3.1 Sovelluksen näkymäkartta 6
3.2 Sovelluksen näkymien rakenne 7
3.3 Sovelluksen päänäkymä 9
3.4 Käyttäjien hallinta 10
3.5 Ryhmien hallinta 13
3.6 Tutkimusten hallinta 16
3.7 Tapahtumien luonti ja muokkaus 17
3.8 Kuntoutusohjelmien luonti ja muokkaus 21
3.9 Jatkokehitykseen sovitut näkymät 23

4 Kuntoutettavan käyttöliittymä **24**
4.1 Sovelluksen näkymäkartta 24
4.2 Sovelluksen näkymien rakenne 25
4.3 Sovelluksen päänäkymä 26
4.4 Oma profiili -näkymä 27
4.5 Edistyminen-näkymä 28
4.6 Tapahtumat-näkymä 29
4.7 Tapahtumien suorittaminen 32
4.8 Harjoitteet-näkymä 35
4.9 Jatkokehitykseen sovitut näkymät 36

5 Tietojärjestelmän rakenne **38**
5.1 Tietojärjestelmän asiakas-palvelinarkkitehtuuri 38
5.2 Tietojärjestelmän rakenne ja rajapinnat 39
5.3 Ulkopuoliset komponentit 39
5.4 PaattiResearchin rakenne ja komponentit 40
5.5 Paatin rakenne ja komponentit 41

5.6 PaattiDatabaseUtilsin rakenne ja komponentit 42
5.7 Toteutusratkaisujen kehittyminen 43

6 Tietojärjestelmän tietokanta 44
6.1 Tietokannan ER-kaavio 44
6.2 USER-taulu 46
6.3 USERGROUP-taulu 46
6.4 ROLE-taulu 47
6.5 RESEARCH-taulu 47
6.6 SCHEDULE-taulu 47
6.7 EVENT-taulu 48
6.8 EVENTTIME-taulu 48
6.9 EVENTTYPE-taulu 49
6.10 TASK-taulu 49
6.11 TASKTYPE-taulu 50
6.12 CHOICE-taulu 50
6.13 TASKDATA-taulu 51
6.14 BINARYDATA-taulu 51
6.15 TEXTDATA-taulu 51
6.16 USER – USERGROUP -relaatiotaulu 52
6.17 USERGROUP – RESEARCH -relaatiotaulu 52
6.18 EVENT – USER -relaatiotaulu 52

7 Ohjelmointikäytänteet 53
7.1 Lähdekoodin muotoilu ja tiedostojen nimeäminen 53
7.2 Lähdekoodin kommentointi 53
7.3 Lähdekoodiesimerkki 54
7.4 Kehitysympäristö 54
7.5 Testauksen käytänteet 55
7.6 Testauksen tulokset 56

8 Tavoitteiden toteutuminen 57
8.1 Vaatimusten toteutuminen 57
8.2 Tutkijan käyttöliittymän tunnetut puutteet ja virheet 58
8.3 Kuntoutusohjelmat-näkymän tunnetut puutteet ja virheet 59
8.4 Kuntoutettavan käyttöliittymän tunnetut puutteet ja virheet 60
8.5 Yleiset jatkokehitysideat 60
8.6 Tutkijan käyttöliittymän jatkokehitysideat 61

8.7 Kuntoutusohjelmat-näkymän jatkokehitysideat 62
8.8 Tapahtumamuokkaimen jatkokehitysideat 62
8.9 Tietokannan jatkokehitysideat 63

9 Käyttöohjeita ylläpitäjälle 64
9.1 Sovelluksen asennus palvelimelle 64
9.2 Tietokokonaisuuden hallintanäkymän lisääminen 64
9.3 Tehtäväkomponenttien lisääminen 65

10 Yhteenveto 66

11 Lähteet 67

1 Johdanto

Iäkkäitä ihmisiä halutaan auttaa elämään tervettä ja itsenäistä elämää omassa kodissaan pidempään. Iäkkäillä on usein muistihäiriöitä, jotka estävät heitä noudattamasta hoito-ohjeita. Kuntoutettavia tulee siten muistuttaa hoito-ohjeiden noudattamisesta. Useasti varttuneempien ihmisten fyysinen aktiivisuus alenee, ja tämä johtaa toimintakyvyn heikkenemiseen, joka estää heitä elämästä itsenäisesti omassa kodissaan. Ikäihmisiä tulee motivoida liikkumaan enemmän omatoimisesti.

Työuupumuksesta kärsivät eivät useasti muista pitää taukoja elämässään ja työssään. Heitä pitää muistuttaa rentoutusharjoituksista, jotka helpottavat stressiä, auttavat toipumaan ja opettavat keinoja selviytyä uupumuksesta.

Mielenterveyspotilaiden hoito vaatii paljon hoitohenkilökunnan läsnäoloa. Läsnäolo ei kuitenkaan ole aina mahdollista, kun hätä on suurin. Potilaan tilaa tulee voida seurata tapaamisten välillä, sekä potilaan ja tutkijan tulee tarvittaessa saada yhteys toisiinsa tapaamisten välillä.

Jyväskylän yliopiston psykologian laitos haluaa joustavan yhteyden kuntoutettavan, omaisen ja tutkijan välille. Tutkijan ja kuntoutettavan välille halutaan yhteys, joka mahdollistaa rutiiniluonteisten tehtävien suorittamisen ilman tutkijan läsnäoloa ajasta ja paikasta riippumattomasti. Yhteys mahdollistaa myös nopeamman viestinnän kuntoutettavan ja tutkijan välillä sekä antaa mahdollisuuden tutkijalle ja omaiselle seurata kuntoutukseen liittyvien tapahtumien suorittamista lähes reaaliajassa.

Paatti-projekti toteutti Sovellusprojekti-kurssilla keväällä 2012 Jyväskylän yliopiston psykologian laitokselle prototyypin tietojärjestelmästä, jolla tuetaan kuntoutettavien itsenäistä elämää ja kuntoutumista sekä tutkijoiden tekemää tutkimusta. Kuntoutettaviin kuuluu muun muassa iäkkäitä, työuupumuksesta kärsiviä ja mielenterveyspotilaita. Tietojärjestelmällä voidaan tukea tutkimusta ja parantaa kuntoutettavien hyvinvointia monin eri tavoin.

Projektissa toteutettu tietojärjestelmä koostuu kahdesta erillisestä käyttöliittymästä. Tutkijan käyttöliittymä on suunniteltu tutkijoiden käytettäväksi ja sillä voi hallita käyttäjiä, ryhmiä ja tutkimuksia sekä luoda kuntoutusohjelmia ryhmille. Kuntoutettavan käyttöliittymällä voi suorittaa kuntoutusohjelmia ja tarkastella suorittamiaan kuntoutusohjelman tapahtumia. Toteutetut prototyypit ovat yhteydessä palvelimella olevaan tietokantaan.

Sovellusraportissa kuvataan vaatimusmäärittelyn [8] toiminnallisten vaatimusten toteutusratkaisut ja toteutuminen. Lisäksi raportissa kuvataan sovelluksen käyttöliittymiä, yleisiä toteutusratkaisuja, toteutuksen tietorakenteita ja yhteyksiä tiedostoihin, suoritettuja testauksia, puutteellisia toteutusratkaisuja, sekä jatkokehitysideoita. Toteutettujen sovellusten luokkia ja metodeja kuvataan tarkemmin luokkadokumentaatiossa [5]. Projektin tuloksia, käytänteitä, tehtävien työmääriä, työnjakoa, aikataulua, sekä riskejä ja muita projektin läpivientiin liittyviä asioita käsitellään projektiraportissa [6].

Luvussa 2 kuvataan aihealueeseen ja sovellukseen liittyvät termit ja niiden merkitys. Luvuissa 3 ja 4 kuvataan tutkijan ja kuntoutettavan käyttämien sovellusten käyttöliittymien näkymät ja toiminnot. Luvussa 5 kuvataan järjestelmän rakenne ja toteutusratkaisuja. Luvussa 6 kuvataan järjestelmän tietokannan taulujen ja kenttien käyttötarkoitus. Luvussa 7 kuvataan projektin ohjelmointikäytänteitä sekä testauskäytänteitä ja testauksen tuloksia. Luvussa 8 kuvataan projektin tavoitteiden toteutumista, järjestelmän tunnettuja virheitä ja puutteita sekä projektin aikana esiintulleita jatkokehitysideoita. Luvussa 9 sisältää käyttöohjeita sovelluksen ylläpitäjille ja jatkokehittäjille.

2 Termit

Luvussa kuvataan projektissa käytettäviä aihealueen termejä sekä tekniikoita ja kehitysvälineitä.

2.1 Aihealueen termit

Projektin sovellusraportin aihealueen termejä ovat seuraavat:

Hälytystehtävä	on tehtävätyyppi, joka lähettää järjestelmän kautta viestin toiselle käyttäjälle.
Kuntoutettava	on tietojärjestelmän käyttäjä, joka suorittaa kuntoutusohjelmia.
Kuntoutusohjelma	on tutkijan ryhmälle tai käyttäjälle laatima kokonaisuus aikataulutettuja tapahtumia.
Käyttäjärooli	on käyttäjän käyttöoikeuksia tietojärjestelmässä kuvaava rooli. Niitä ovat mm. kuntoutettava ja tutkija.
Mobiililaite	on älypuhelin tai sormitietokone.
Ohje	on tehtävään liittyvä kuvaus teksti-, ääni-, kuva- tai videomuodossa.
Omainen	on käyttäjä, jolla on oikeus seurata kuntoutettavan edistymistä ja aktiivisuutta.
Ryhmä	on käyttäjistä koostuva kokonaisuus, jolle mm. voidaan laatia kuntoutusohjelma.
Suorite	on tehtävän suorittamisesta tietokantaan tallennettu data.
Tapahtuma	on tutkijan laatima tehtävien kokonaisuus, joka liitetään osaksi kuntoutusohjelmaa.
Tehtävä	on tapahtuman sisällä oleva yksittäinen tehtävä, joka voi sisältää mm. tekstiä, ääntä, kuvan tai videon.

Tutkija on käyttäjä, joka laatii kuntoutusohjelmia kuntoutettaville ja seuraa niiden toteutumisia.

Tutkijan käyttöliittymä on käyttöliittymä, jolla tutkija hallitsee tapahtumia ja kuntoutusohjelmia, hallitsee käyttäjien ja ryhmien tietoja sekä muodostaa suoritteista erilaisia raportteja.

Tutkimus on samaan tutkimusaiheeseen tai -kohteeseen liittyvien ryhmien kokonaisuus.

2.2 Vaatimusmäärittelyn termit

Vaatimusmäärittelyssä esiintyviä termejä ovat seuraavat:

Etenemisvaihtoehto on monivalintatehtävän yksi valinta, jonka valitsemisen mukaan siirrytään seuraavaksi määritettyyn tehtävään.

Hälytyskomponentti on tehtävä, joka lähettää viestin käyttäjän asetuksiin määritetyille henkilöille.

Käyttäjäprofiili on järjestelmässä olevan käyttäjän profiili, joka sisältää yksilöintitietoja ja käyttöoikeuksia.

Mediasisältö koostuu tietojärjestelmässä olevista multimediatiedostoista, joita voidaan sisällyttää tehtäviin.

Muistutus huomauttaa käyttäjää ajankohtaisista tapahtumista ääni-, värinä- ja/tai tekstimuotoisella viestillä.

Sisältökomponentti on tehtävä, joka esittää käyttäjälle informaatiota, kuten tekstiä, ääntä, valokuvan tai videon.

Syötekomponentti on tehtävä, joka tallentaa järjestelmään käyttäjän antamaa informaatiota, kuten tekstiä, ääntä, valokuvan tai videon.

Tapahtumamuokkain on tutkijan käyttöliittymässä oleva kuntoutusohjelmissa käytettävien tapahtumien laatimiseen käytettävä työkalu.

Tapahtumatyyppi	on tapahtumalle asetettu tyyppi, jolla tapahtuman sisältö voidaan tunnistaa. Niitä voivat olla mm. liikunta- tai rentoutustapahtumat.
Valintakomponentti	on tehtävä, johon käyttäjä vastaa valitsemalla vaihtoehdon.

2.3 Tekniikat ja kehitysvälineet

Dokumentissa esiintyviä tekniikoita ja kehitysvälineitä ovat seuraavat:

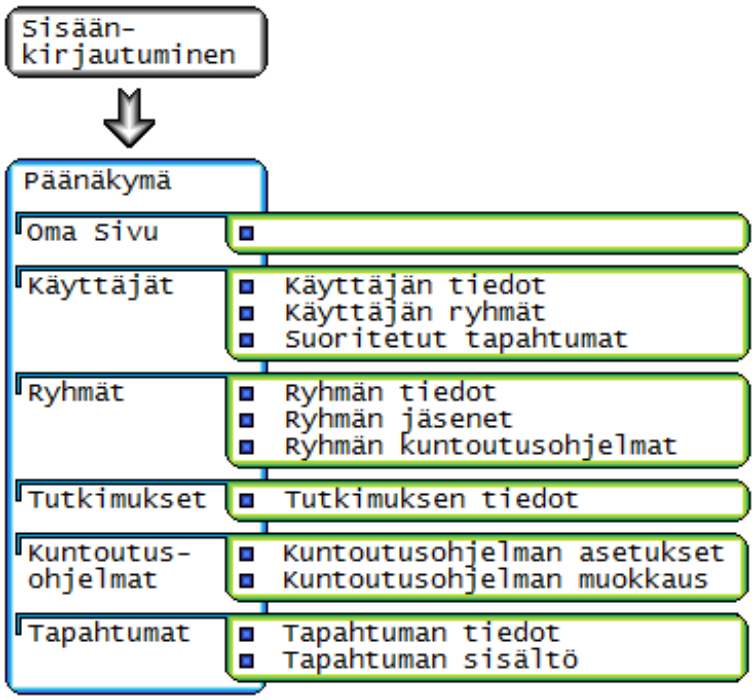
CSS	on WWW-dokumenteille kehitetty tyylikieli.
Git	on hajautettu versiohallintajärjestelmä.
HTML5	on uusi versio WWW-sivujen tekemiseen käytetystä HTML-kuvauskielestä.
JavaDoc	on ohjelma, jonka avulla Javan lähdekoodista voidaan generoida luokkadokumentaatio.
JavaScript	on WWW-ympäristössä käytettävä komentosarjakieli, jolla voidaan mm. lisätä WWW-sivuille dynaamista toiminnallisuutta.
NetBeans	on integroitu ohjelmointiympäristö mm. Java- ja JavaScript-ohjelmointikielille.
Vaadin	on Java-kielisten WWW-sovellusten kehitysympäristö.
YouSource	on Git-versiohallintaohjelmistoa tukeva lähdekoodien julkistusjärjestelmä, jota käytetään WWW-käyttöliittymällä.

3 Tutkijan käyttöliittymä

Toteutettu tutkijan käyttöliittymä ei sisällä kaikkia tarvittavia toiminnallisuuksia. Paatti-projektissa toteutettuihin toiminnallisuuksiin kuuluu sovelluksen sisäänkirjautuminen, käyttäjien luonti ja muokkaus, ryhmien luonti ja muokkaus, tutkimusten luonti ja muokkaus, tapahtumien luonti ja muokkaus sekä kuntoutusohjelmien luonti ja muokkaus. Projektin jälkeisessä jatkokehityksessä toteutetaan multimediatiedostojen lataaminen ja esikatselu, suoritedatasta luotavien raporttien luonti, viestien lähettäminen ja vastaanottaminen sekä ohjenäkymä. Luvussa esitellään pääosin tutkijan käytettäväksi toteutetun sovelluksen käyttöliittymän eri näkymät sekä niihin liittyvät toiminnot ja niiden väliset suhteet.

3.1 Sovelluksen näkymäkartta

Kuvassa 3.1 havainnollistetaan näkymäkartan avulla tutkijan käyttöliittymän välilehtinä esitettäviä näkymiä ja niiden toimintakokonaisuuksia alivälilehtinä. Luvuissa 3.3 – 3.9 kuvataan kunkin näkymän toiminnallisuus ja yhteydet muihin näkymiin.



Kuva 3.1: Tutkijan käyttöliittymän näkymäkartta.

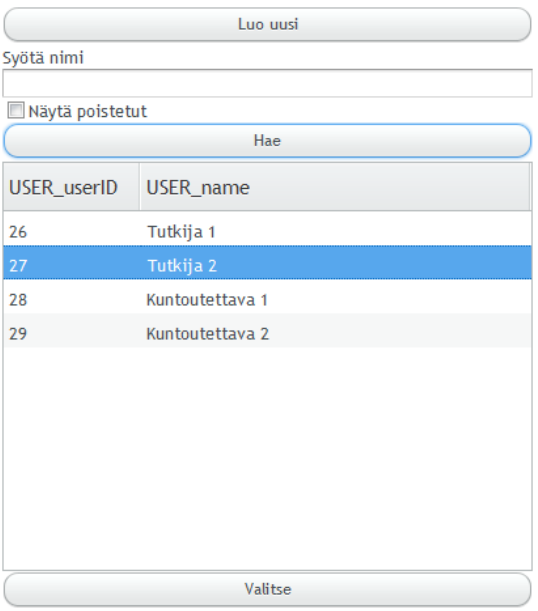
Tutkijan käyttöliittymän sovellukseen kirjaudutaan sisäänkirjautumisnäkymästä. Sisäänkirjautuva käyttäjä kirjoittaa käyttäjätunnuksensa ja salasanansa kirjautuessaan sisään. Uloskirjautuminen sovelluksesta tapahtuu *Oma sivu* -näkymästä.

3.2 Sovelluksen näkymien rakenne

Sovelluksen toiminnallisuus on sijoitettu **välilehtinä** esitettäviin näkymiin. Välilehtiratkaisu mahdollistaa sulavan siirtymisen näkymästä toiseen yläreunan välilehtikuvakkeiden kautta. Muista navigointimahdollisuuksista näkymien välillä on mainintoja luvuissa 3.4 – 3.8.

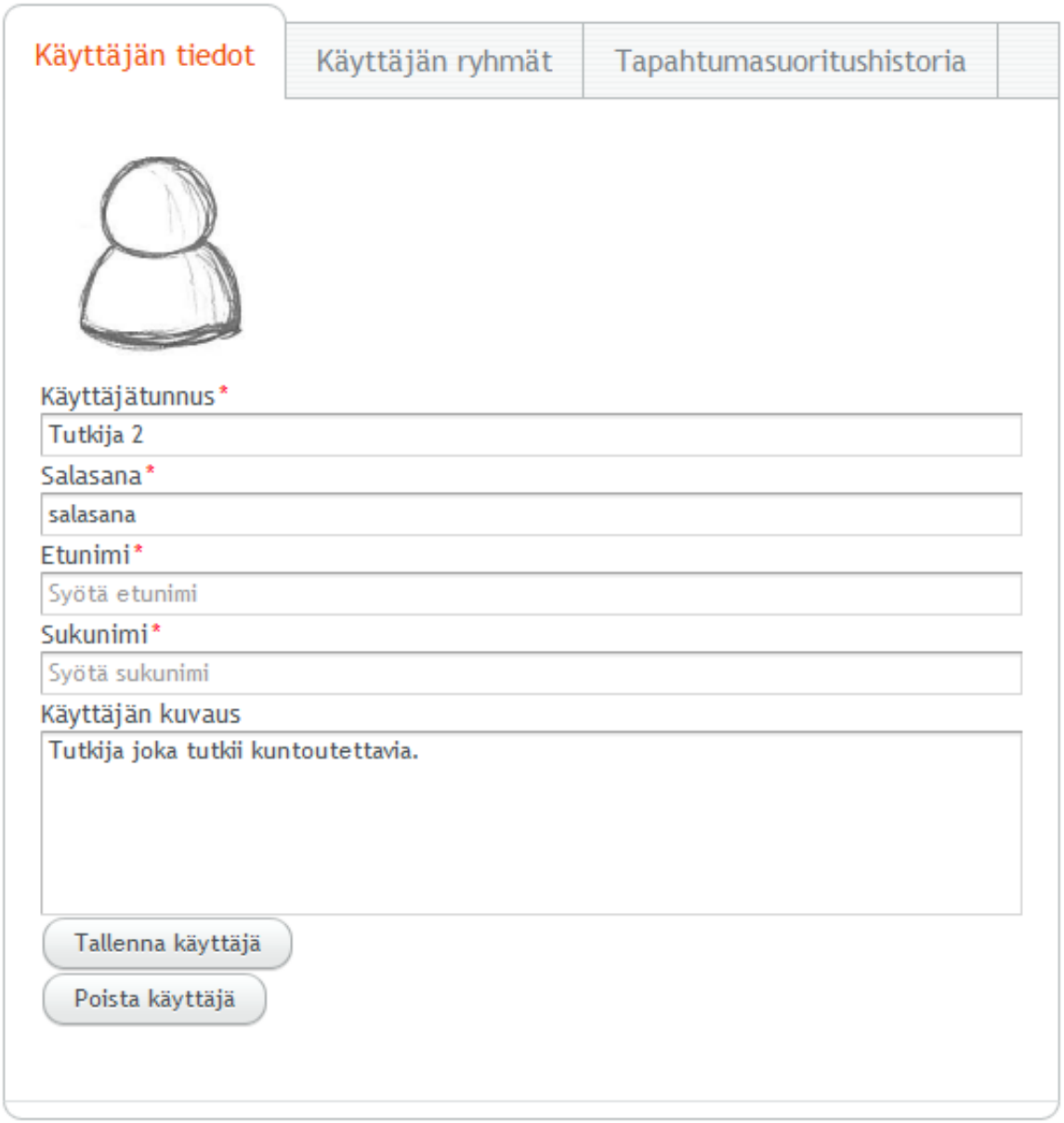
Kuva 3.2: Tutkijan käyttöliittymän välilehtivalikko.

Näkymien rakenne on ulkoasultaan ja sijoittelultaan samanlainen. Vasemmalla puolella näkymää on **hakukomponentti** (katso kuva 3.3), jolla käyttäjä voi hakea tiettyyn näkymään liittyviä tietoja. Esimerkiksi käyttäjien hallintanäkymässä kyseisellä hakukomponentilla voi hakea tietokannasta järjestelmässä olevia käyttäjiä. Hakukomponentin avulla luodaan myös uusia kohteita painamalla *Luo uusi* -painiketta.



Kuva 3.3: Tutkijan käyttöliittymän hakukomponentti.

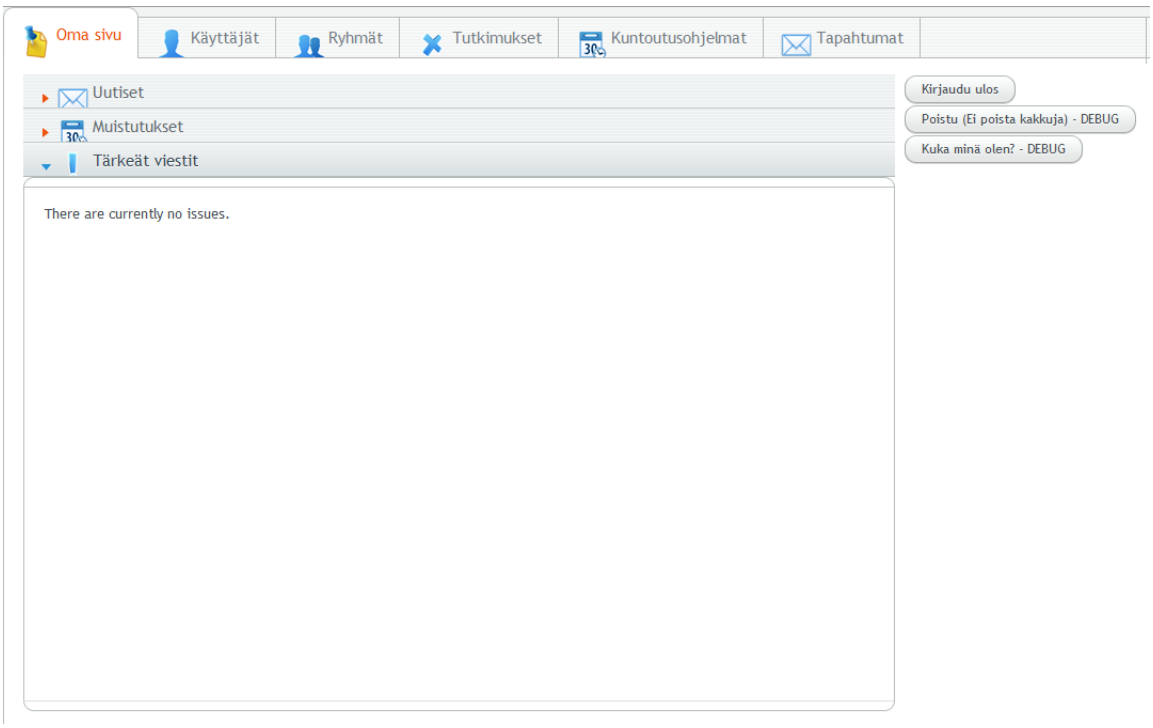
Näkymien oikealla puolella olevilla **alivälilehdillä** näytetään hakutuloslistasta valittuun tietokokonaisuuteen liittyviä tietoja (katso kuva 3.4). Niillä voidaan myös muokata kyseiseen tietokokonaisuuteen liittyviä tietoja.



Kuva 3.4: Tutkijan käyttöliittymän haetun kohteen alivälilehtivalikko.

3.3 Sovelluksen päänäkymä

Sovelluksen päänäkymänä toimii kuvan 3.5 *Oma sivu* -näkymä, joka näyttää tutkijalle ensiarvoista tietoa kuntoutettavista. Näkymä myös huomauttaa tutkijalle, jos hänellä on jokin asia jäänyt kesken aiemmilla istunnoilla. Kyseinen näkymä avautuu aina ensimmäisenä, kun sovellukseen kirjaudutaan sisään. Tämän näkymän kautta kirjaudutaan myös ulos järjestelmästä.



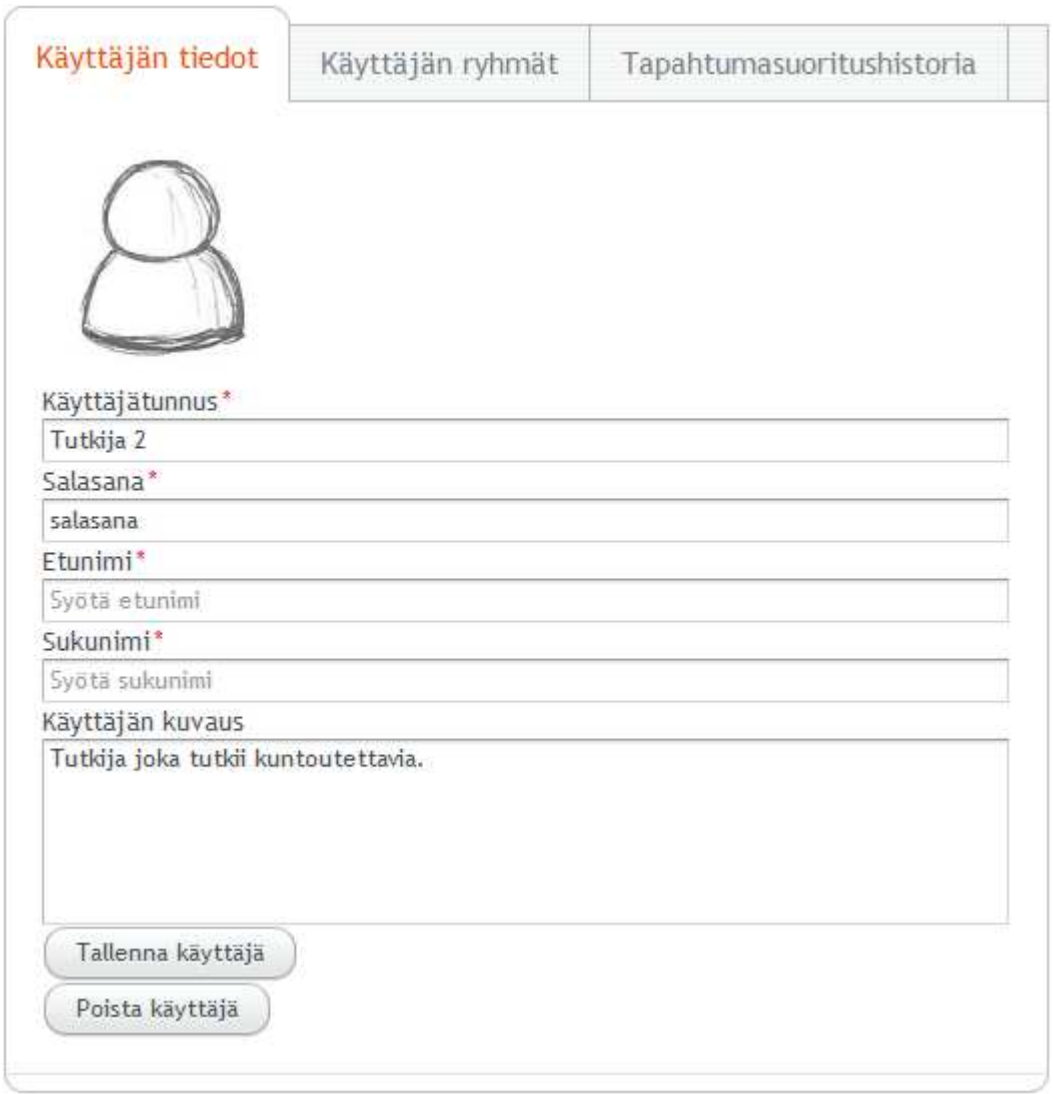
Kuva 3.5: Tutkijan käyttöliittymän päänäkymä.

Sovelluksen päänäkymän rakenne eroaa ulkoasultaan muista näkymistä, sillä päänäkymässä ei luoda tai muokata järjestelmän tietokannan tietoja, vaan päänäkymä toimii ensisijaisena informaationäkymänä sovelluksen käyttäjälle.

Näkymään liittyvät toiminnallisuudet sovittiin tilaajan edustajien kanssa projektin jälkeiseen jatkokehitykseen. Uloskirjautumisen sijainnin muuttamisen jatkokehitysideaa kuvataan luvussa 8.6.

3.4 Käyttäjien hallinta

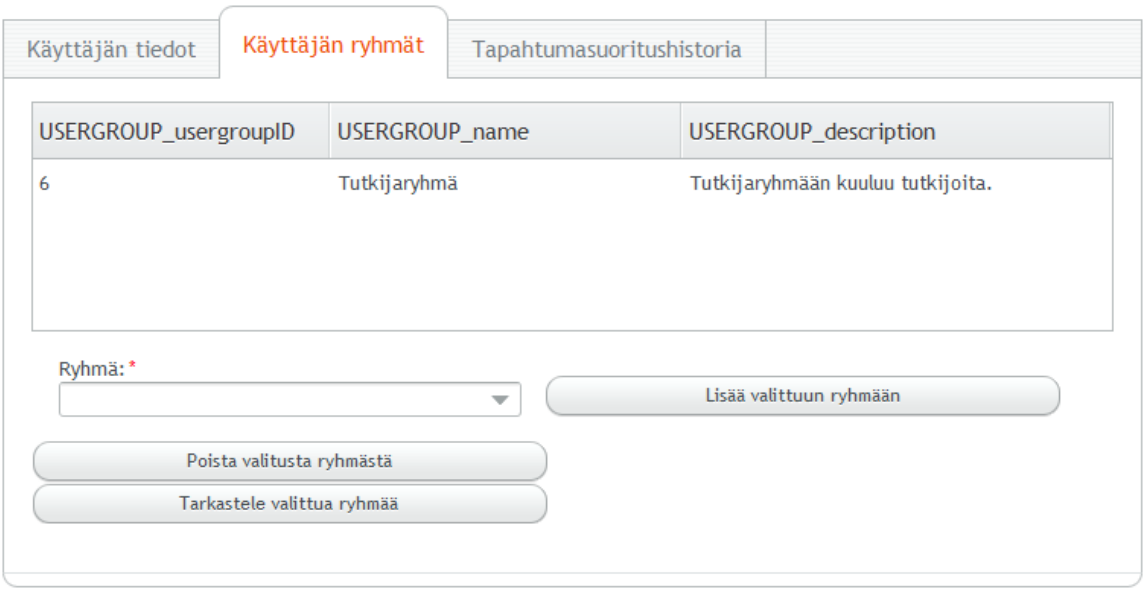
Käyttäjien hallintanäkymällä tutkija voi luoda uusia käyttäjiä järjestelmään, hakea olemassa olevia käyttäjiä ja muokata heidän tietojaan, hallita käyttäjien ryhmiä sekä tarkastella käyttäjän suorittamien tapahtumien suoritedataa. Näkymän päätoimintakokonaisuudet ovat uuden käyttäjän luonti ja haetun käyttäjän tietojen muokkaaminen. Tietojen muokkaamisen osatoimintakokonaisuudet kuvassa 3.6 on jaettu alivälilehdille, jotka ovat käyttäjän perustietojen muokkaaminen, käyttäjän ryhmien jäsenyyden tarkasteleminen, lisääminen ja poistaminen sekä tapahtumien suoritushistorian tarkastelu ja tallentaminen xls-tiedostoksi.



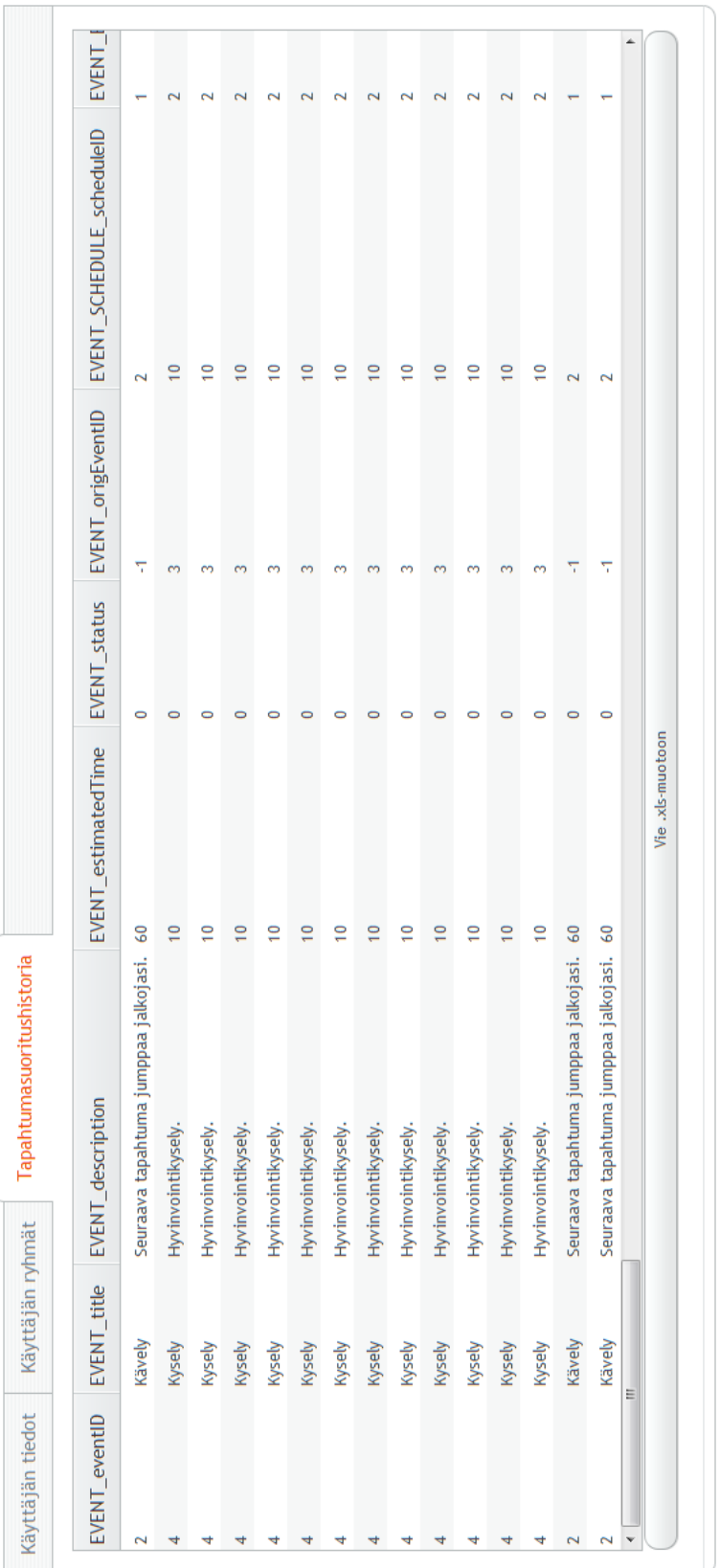
Kuva 3.6: Tutkijan käyttöliittymän *Käyttäjän tiedot* -näkymä.

Luotaessa uutta käyttäjää tutkija painaa hakukomponentin (katso kuva 3.3) *Luo uusi* -painiketta, jolloin kuvan 3.6 alivälilehti aktivoituu muokattavaksi. Tämän jälkeen tutkija syöttää pakollisiin syötekenttiin tarvittavat käyttäjän perustiedot, jonka jälkeen hän voi tallentaa käyttäjän järjestelmän tietokantaan. Tutkija voi tallentamisen jälkeen jatkaa juuri tallennetun käyttäjän muokkausta tai valita toisen käyttäjän muokattavaksi.

Käyttäjän tietoja muokattaessa tutkija voi muokata käyttäjän perustietoja kuvan 3.6 *Käyttäjän tiedot* -alivälilehdellä. Muokattaviin käyttäjän perustietoihin kuuluvat etunimi, sukunimi, käyttäjätunnus ja käyttäjän kuvaus. Välilehdellä voidaan myös tallentaa tai poistaa kyseinen muokattavaksi valittu käyttäjä painikkeilla *Tallenna käyttäjä* ja *Poista käyttäjä*. Kuvan 3.7 *Käyttäjän ryhmät* -välilehdellä tutkija voi nähdä ryhmät, joissa käyttäjä on jäsenenä ja voi halutessaan lisätä käyttäjän uuteen ryhmään tai poistaa käyttäjän valitusta ryhmästä. Näkymästä voi myös päästä tarkastelemaan valitun käyttäjän ryhmän tietoja *Tarkastele valittua ryhmää* -painikkeen avulla. Kuvan 3.8 *Tapahtumasuoritushistoria*-välilehdellä tutkija voi tarkastella käyttäjän suorittamista tapahtumista kerättyä suoritedataa ja siirtää kyseisen datan xls-tiedostomuodossa Excel-taulukkolaskentaohjelmaan painamalla näkymän painiketta *Vie .xls-muotoon*.



Kuva 3.7: *Käyttäjät*-näkymän *Käyttäjän ryhmät* -alivälilehti.

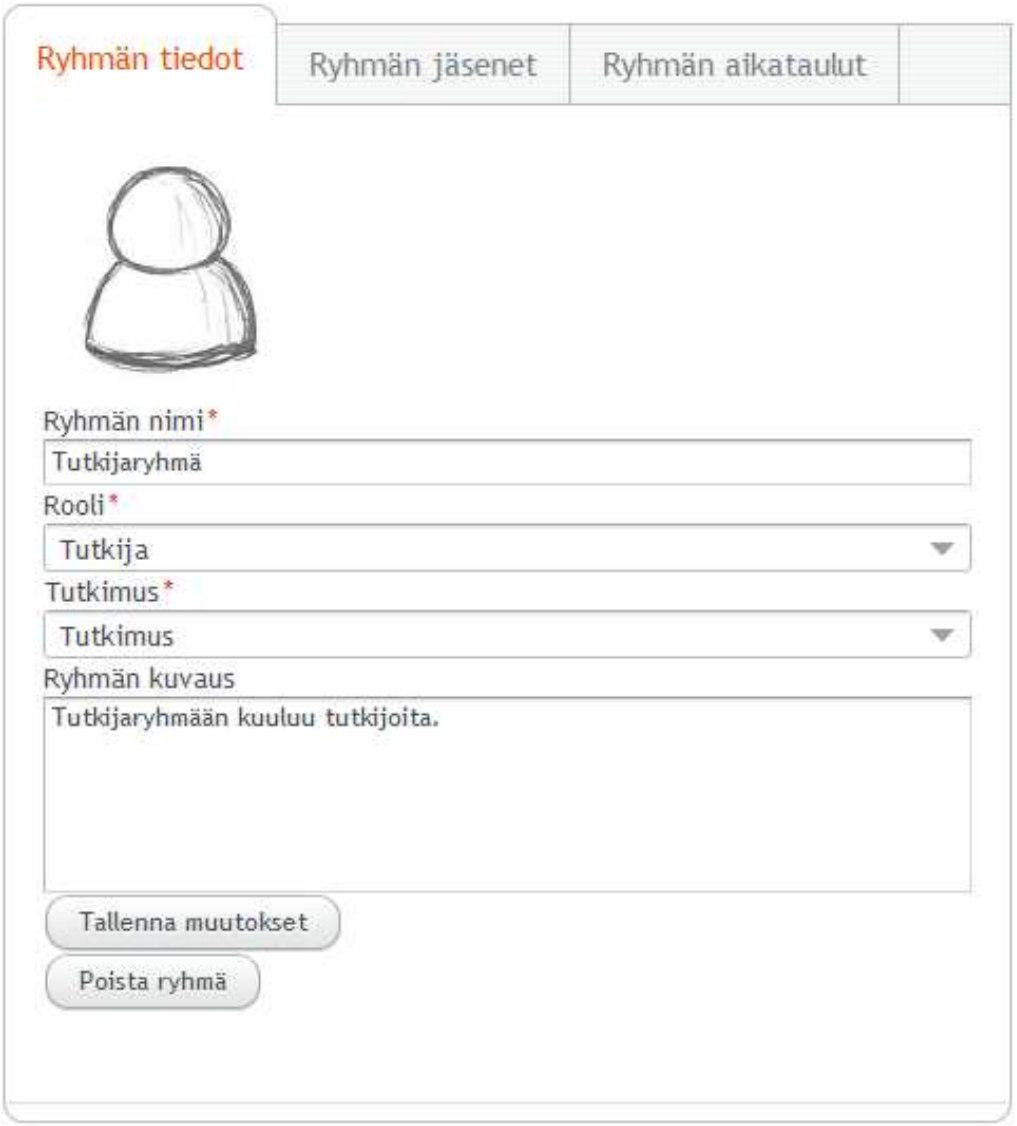


EVENT_eventID	EVENT_title	EVENT_description	EVENT_estimatedTime	EVENT_status	EVENT_origEventID	EVENT_SCHEDULE_scheduleID	EVENT_
2	Kävely	Seuraava tapahtuma jumppaa jalkojasi.	60	0	-1	2	1
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
4	Kysely	Hyvinvointikysely.	10	0	3	10	2
2	Kävely	Seuraava tapahtuma jumppaa jalkojasi.	60	0	-1	2	1
2	Kävely	Seuraava tapahtuma jumppaa jalkojasi.	60	0	-1	2	1

Kuva 3.8: *Käyttäjät*-näkymän *Tapahtumasuoritushistoria*-alivälilehti.

3.5 Ryhmien hallinta

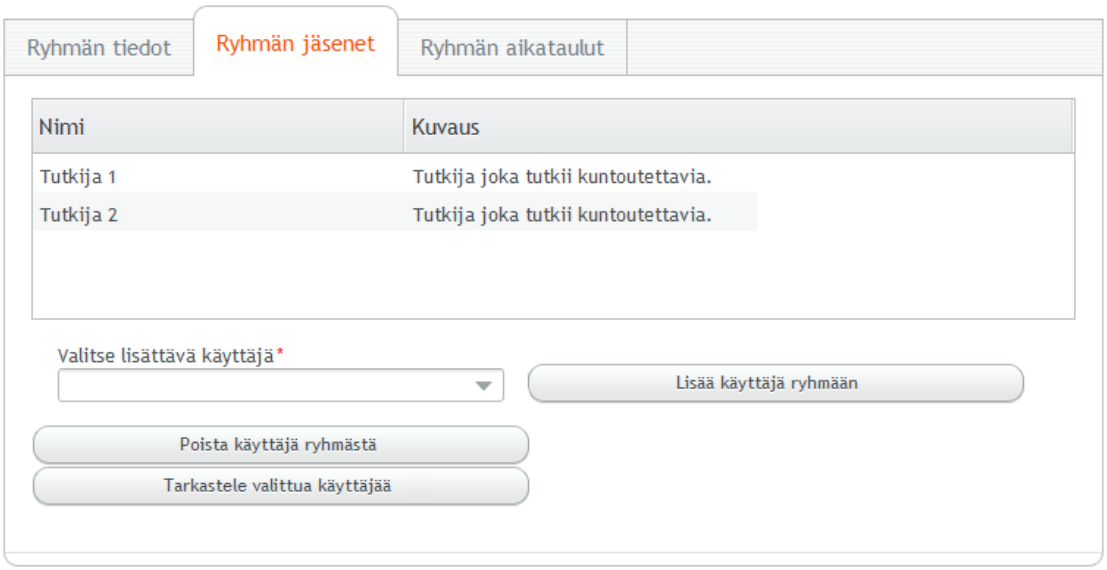
Ryhmien hallintanäkymässä tutkija voi luoda uusia ryhmiä järjestelmään, hakea olemassa olevia ryhmiä ja muokata niiden tietoja, hallita ryhmään kuuluvia käyttäjiä sekä hallita ryhmien kuntoutusohjelmia. Näkymän päätoimintakokonaisuudet kuvassa 3.9 ovat uuden ryhmän luonti ja haetun ryhmän tietojen muokkaaminen. Alivälilehdille jaetut tietojen muokkaamisen osatoimintakokonaisuudet ovat ryhmän perustietojen muokkaaminen, ryhmän käyttäjien jäsenyyksien tarkasteleminen, lisääminen ja poistaminen sekä ryhmän kuntoutusohjelmien tarkasteleminen, lisääminen ja poistaminen.



Kuva 3.9: Tutkijan käyttöliittymän *Ryhmän tiedot* -näkymä.

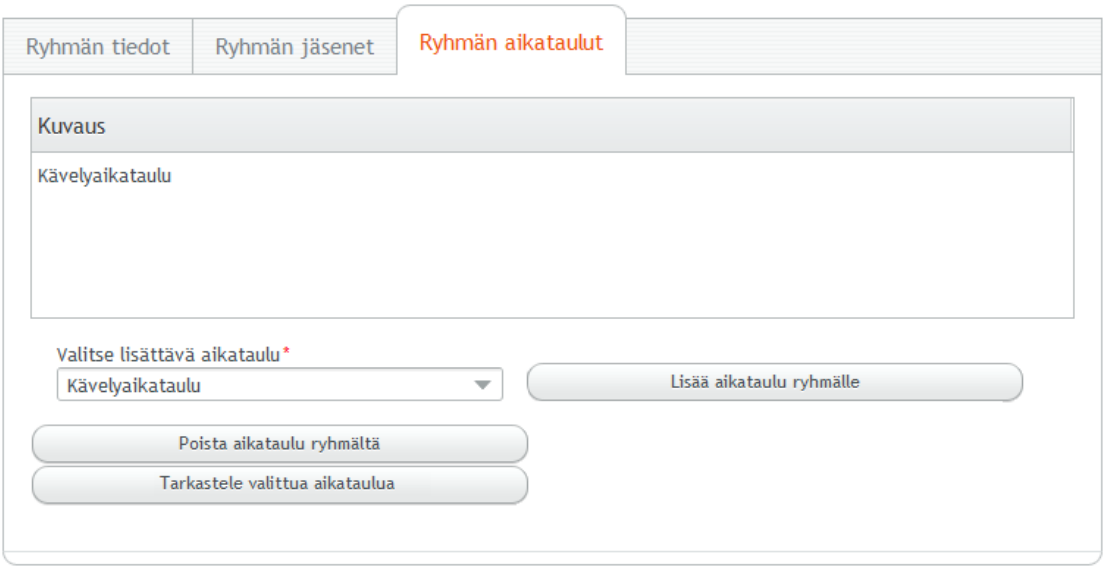
Luotaessa uutta ryhmää tutkija painaa hakukomponentin (katso kuva 3.3) *Luo uusi* -painiketta, jolloin kuvan 3.9 alivälilehti aktivoituu muokattavaksi. Tämän jälkeen tutkija syöttää pakollisiin syötekenttiin tarvittavat ryhmän perustiedot, jonka jälkeen hän voi tallentaa ryhmän järjestelmän tietokantaan. Tutkija voi tallentamisen jälkeen jatkaa juuri tallennetun ryhmän muokkausta tai valita toisen ryhmän muokattavaksi.

Ryhmän tietoja muokattaessa tutkija voi muokata valitun ryhmän perustietoja kuva 3.9 *Ryhmän tiedot* -välilehdellä. Muokattaviin ryhmän perustietoihin kuuluvat nimi, ryhmän rooli, ryhmään liitetty tutkimus ja ryhmän kuvaus. Välilehdellä voidaan myös tallentaa tai poistaa kyseinen muokattavaksi valittu ryhmä painikkeilla *Tallenna muutokset* ja *Poista ryhmä*. Kuvan 3.10 *Ryhmän jäsenet* -välilehdellä tutkija voi nähdä ryhmään kuuluvat käyttäjät ja voi halutessaan joko lisätä ryhmään uuden käyttäjän tai poistaa ryhmästä valitun käyttäjän. Välilehdeltä voi myös päästä suoraan tarkastelemaan valitun ryhmän jäsenenä olevan käyttäjän tietoja *Tarkastele valittua ryhmää* -painikkeen avulla.



Kuva 3.10: *Ryhmät*-näkymä *Ryhmän jäsenet* -alivälilehti.

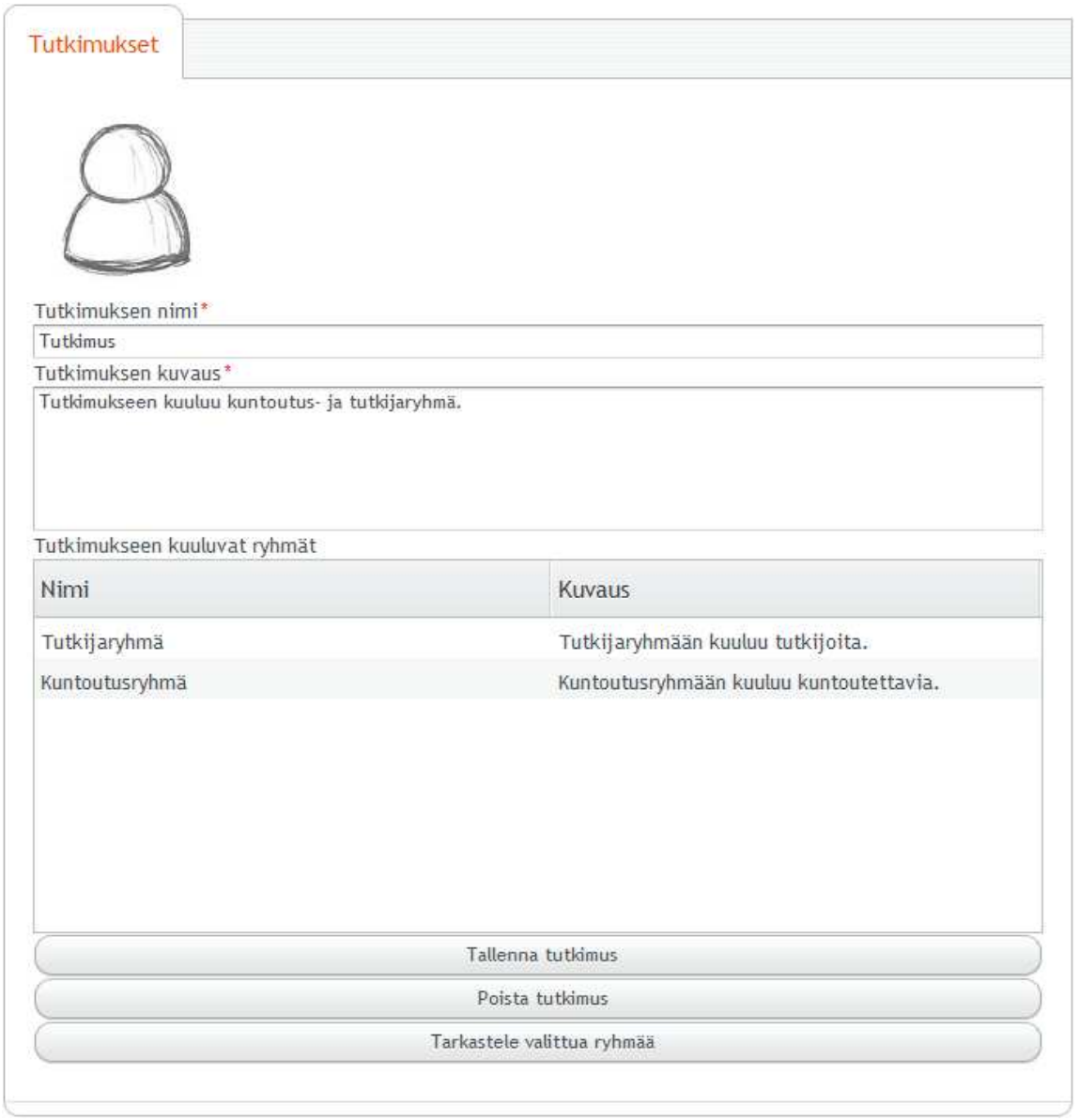
Kuvan 3.11 *Ryhmän aikataulut* -välilehdellä tutkija voi tarkastella ryhmän sisältämiä kuntoutusohjelmia, lisätä uusia kuntoutusohjelmia ryhmälle sekä poistaa ryhmältä kuntoutusohjelmia. Näkymästä voi myös siirtyä tarkastelemaan ryhmällä olevia kuntoutusohjelman tietoja *Tarkastele valittua aikataulua* -painikkeen avulla.



Kuva 3.11: *Ryhmät*-näkymän *Ryhmän aikataulut* -alivälilehti.

3.6 Tutkimusten hallinta

Tutkimusten hallintanäkymässä kuvassa 3.12 tutkija voi luoda uusia tutkimuksia järjestelmään sekä hakea ja muokata olemassa olevia tutkimuksia. Näkymän päätoimintakokonaisuudet ovat uuden tutkimuksen luonti sekä haetun tutkimuksen tietojen muokkaaminen.



Kuva 3.12: Tutkijan käyttöliittymän *Tutkimukset*-näkymä.

Luotaessa uutta tutkimusta tutkija painaa hakukomponentin (katso kuva 3.3) *Luo uusi* -painiketta, jolloin kuvan 3.12 alivälilehti aktivoituu muokattavaksi. Tämän jälkeen tutkija syöttää pakollisiin syötekenttiin tarvittavat tutkimuksen perustiedot, jonka jälkeen hän voi tallentaa tutkimuksen järjestelmän tietokantaan. Tutkija voi tallentamisen jälkeen jatkaa juuri tallennetun tutkimuksen muokkausta tai valita toisen tutkimuksen muokattavaksi.

Tutkimuksen tietoja muokattaessa tutkija voi muokata valitun tutkimuksen perustietoja *Tutkimukset* -välilehdellä. Muokattaviin tutkimuksen perustietoihin kuuluvat nimi ja tutkimuksen kuvaus. Välilehdellä voidaan myös tallentaa tai poistaa kyseinen muokattavaksi valittu tutkimus painikkeilla *Tallenna tutkimus* ja *Poista tutkimus*. Välilehdellä näytetään myös tutkimukseen kuuluvat ryhmät, joita voi siirtyä tarkastelemaan *Tarkastele valittua ryhmää* -painikkeen avulla.

3.7 Tapahtumien luonti ja muokkaus

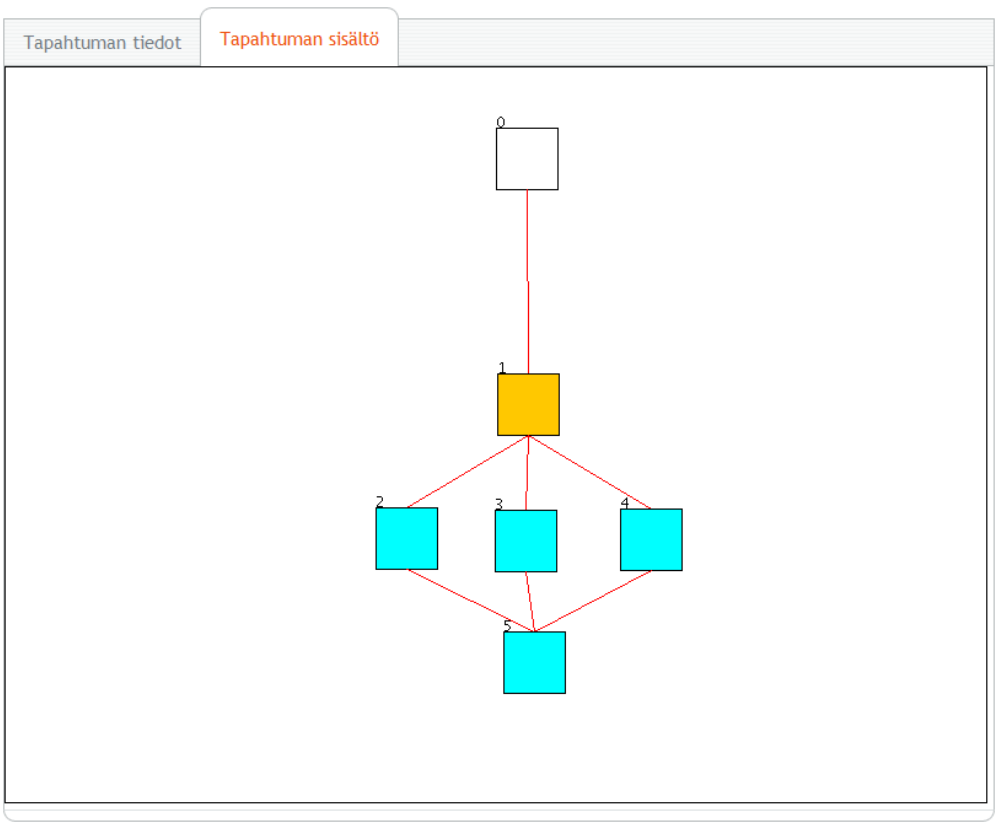
Tapahtumien luonti- ja muokkausnäkymässä kuvassa 3.13 tutkija voi luoda uusia tapahtumia järjestelmään sekä hakea ja muokata olemassa olevia tapahtumia. Näkymän päätoimintakokonaisuudet ovat uuden tapahtuman luonti sekä haetun tapahtuman muokkaaminen. Tapahtuman muokkaamisen osatoimintakokonaisuudet ovat tapahtuman perustietojen muokkaaminen sekä tapahtuman rakenteen ja sisällön muokkaaminen.

Luotaessa uutta tapahtumaa tutkija painaa hakukomponentin (katso kuva 3.3) *Luo uusi* -painiketta, jolloin kuvan 3.13 alivälilehti aktivoituu muokattavaksi. Tämän jälkeen tutkija syöttää pakollisiin syötekenttiin tarvittavat tapahtuman perustiedot, jonka jälkeen hän voi muokata tapahtuman rakennetta ja sisältöä tapahtumamuokkaimella, joka sisältää kuvissa 3.14 – 3.17 esiteltävät *Tapahtuman sisältö* -välilehden toiminnallisuudet. Tutkija voi tallentaa tapahtuman järjestelmän tietokantaan painikkeella *Tallenna tapahtuma*. Tutkija voi tallentamisen jälkeen jatkaa juuri tallennetun tapahtuman muokkausta tai valita toisen tapahtuman muokattavaksi.

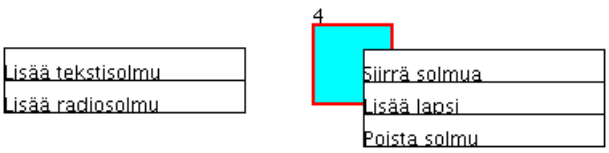
Tapahtuman tietoja muokattaessa tutkija voi muokata valitun tapahtuman perustietoja kuvan 3.13 *Tapahtuman tiedot* -välilehdellä. Muokattaviin tapahtuman perustietoihin kuuluvat otsikko, kuvaus ja tyyppi. Välilehdellä voidaan myös tallentaa tai poistaa kyseinen muokattavaksi valittu tapahtuma painikkeilla *Tallenna tapahtuma* ja *Poista tapahtuma*. Kuvissa 3.14 – 3.17 esiteltävällä *Tapahtuman sisältö* -välilehdellä tutkija voi muokata tapahtuman rakennetta ja sisältöä tapahtumamuokkaimella.

Kuva 3.13: Tutkijan käyttöliittymän *Tapahtuman tiedot* -näkymä.

Tapahtumamuokkaimen piirtoalueessa (katso kuva 3.14) näytetään tapahtuman rakenne. Uusia tehtäväkomponentteja lisätään piirtoalustaan painalla hiiren oikeaa painiketta piirtoalustan tyhjässä tilassa. Tämän jälkeen tutkija valitsee kuvan 3.15 avautuvasta hiirivalikosta lisättäväksi halutun tehtäväkomponenttityypin. Kun tutkija painaa hiiren oikeaa painiketta tehtäväkomponentin päällä (katso kuva 3.15), voi tutkija siirtää tai poistaa valitun tehtäväkomponentin tai luoda etenemispolun kahden komponentin välille. Etenemispolku kertoo seuraavaksi näytettävän tehtävän, kun halutaan siirtyä tapahtumaa suoritettaessa tehtävästä toiseen. Tehtäväkomponentin sisältöä voidaan muokata tapahtumamuokkaimen oikean puolen tietoalueessa (katso kuvat 3.16 ja 3.17), jossa näytetään piirtoalueesta hiirellä valittuun tehtäväkomponenttiin liittyviä tietoja.



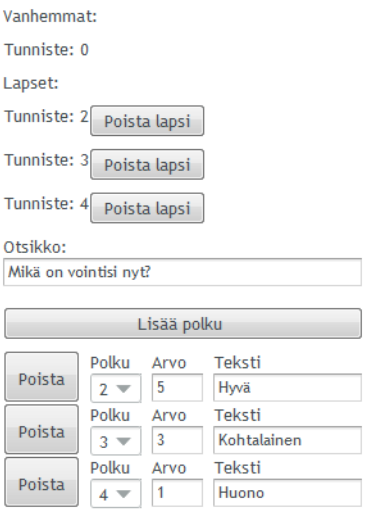
Kuva 3.14: *Tapahtumat*-näkymän *Tapahtuman sisältö*-alivälilehti.



Kuva 3.15: Tapahtumamuokkaimen hiirivalikot.

Lisättäviä tehtäväkomponenttityyppejä ovat tekstikomponentti, monivalintakomponentti, liukusäädinkomponentti, sensorikomponentti, mediakomponentti ja hälytyskomponentti, joista projektissa ehdittiin toteuttaa kuvien 3.16 ja 3.17 teksti- ja monivalintakomponentti. Tekstikomponentin muokattaviin ominaisuuksiin kuuluvat otsikko ja tekstisisältö. Monivalintakomponentin muokattaviin ominaisuuksiin kuuluvat otsikko sekä etenemisvaihtoehtojen lisääminen, etenemisvaihtoehdon seuraavan tapahtuman valitseminen, etenemisvaihtoehdon arvon valitseminen sekä etenemisvaihtoehdon tekstin lisääminen. Etenemisvaihtoehdon arvoa käytetään kuntoutettavan edistymisen seurannan laskemiseen ja analysointiin.

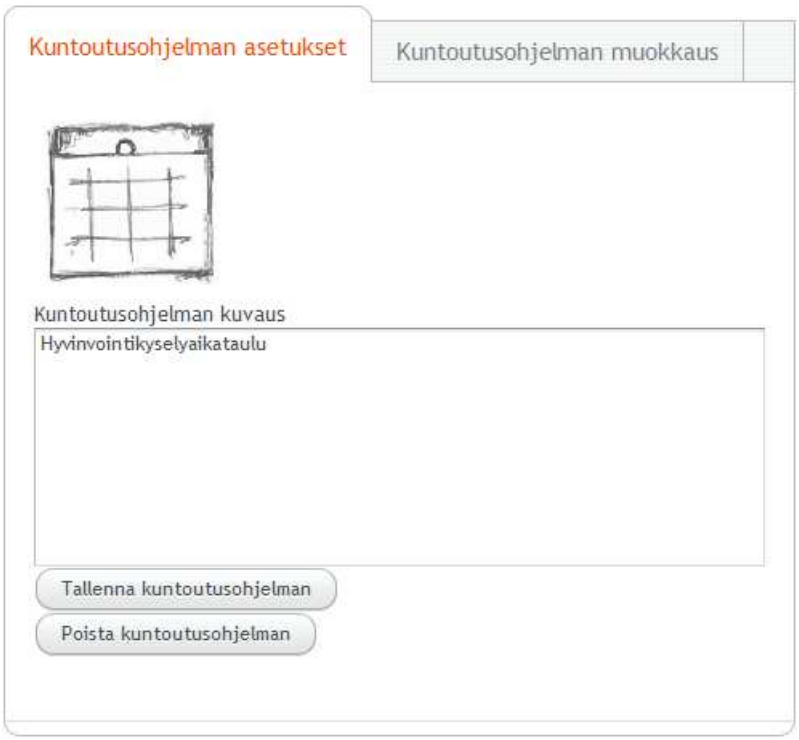
Kuva 3.16: Tekstikomponentin asetusvalikko.



Kuva 3.17: Monivalintakomponentin asetusvalikko.

3.8 Kuntoutusohjelmien luonti ja muokkaus

Kuntoutusohjelmien luonti- ja muokkausnäkymässä kuvassa 3.18 tutkija voi luoda uusia kuntoutusohjelmia järjestelmään sekä hakea ja muokata olemassa olevia kuntoutusohjelmia. Näkymän päätoimintakokonaisuudet ovat uuden kuntoutusohjelman luonti sekä haetun kuntoutusohjelman tietojen muokkaaminen. Tietojen muokkaamisen osatoimintakokonaisuudet ovat kuntoutusohjelman perustietojen muokkaaminen sekä kuntoutusohjelmassa olevien tapahtumien tarkasteleminen, lisääminen, poistaminen ja aikatauluttaminen.

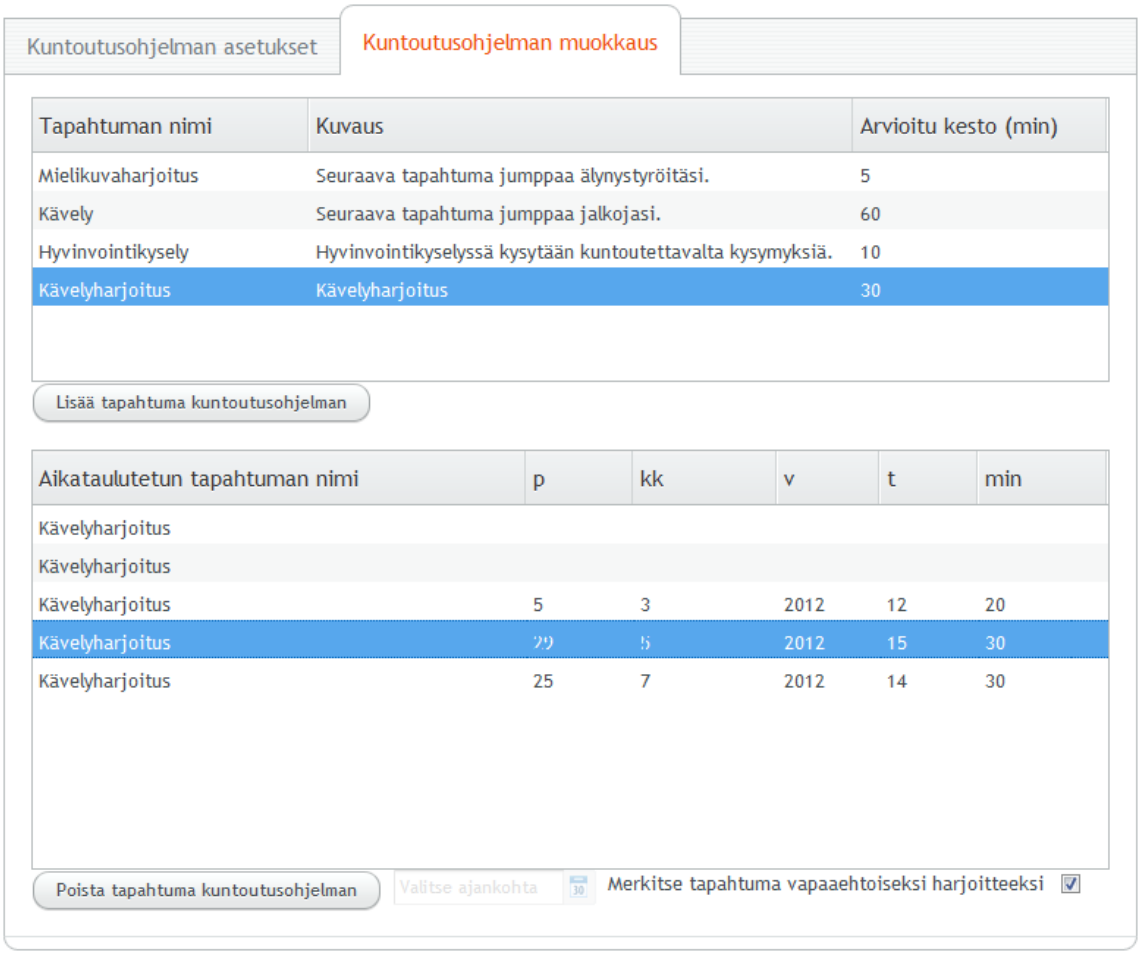


Kuva 3.18: Kuntoutusohjelman tietojen muokkaus.

Luotaessa uutta kuntoutusohjelmaa tutkija painaa hakukomponentin (katso kuva 3.3) *Luo uusi* -painiketta, jolloin kuvan 3.18 alivälilehti aktivoituu muokattavaksi. Tämän jälkeen tutkija syöttää pakollisiin syötekenttiin tarvittavat kuntoutusohjelman perustiedot, jonka jälkeen hänen täytyy lisätä tapahtumia kuntoutusohjelmaan kuvan 3.19 *Kuntoutusohjelman muokkaus* -välilehdellä. Siinä tutkija voi lisätä järjestelmässä olevia tapahtumia kuntoutusohjelmaan, poistaa kuntoutusohjelmasta tapahtumia sekä aikatauluttaa kuntoutusohjelmaan lisäämiään tapahtumia. Tutkija voi tallentamisen jälkeen jatkaa juuri tallennetun kuntoutusohjelman muokkausta tai valita toisen kuntoutusohjelman muokattavaksi.

Kuntoutusohjelman tietoja muokattaessa tutkija voi muokata valitun kuntoutusohjelman perustietoja kuvan 3.18 *Kuntoutusohjelman asetukset* -välilehdellä. Muokattaviin kuntoutusohjelman perustietoihin kuuluu kuntoutusohjelman kuvaus. Välilehdellä voidaan myös tallentaa tai poistaa kyseinen muokattavaksi valittu kuntoutusohjelma. Kuvan 3.19 *Kuntoutusohjelman muokkaus* -välilehdellä tutkija voi tarkastella kuntoutusohjelman sisältämiä tapahtumia ja niiden ajoitusta, lisätä uusia tapahtumia kuntoutusohjelmaan ja aikatauluttaa ne sekä poistaa tapahtumia kuntoutusohjelmasta.

Tutkija voi myös **luoda vapaaehtoisia kuntoutusohjelmia** *Kuntoutusohjelman muokkaus* -välilehdellä. Tällöin kuntoutusohjelmassa oleva tapahtuma asetetaan vapaaehtoiseksi *Merkitse tapahtuma vapaaehtoiseksi harjoitteeksi* -asetuspainikkeella.



Kuva 3.19: Kuntoutusohjelman tapahtumien muokkaus.

3.9 Jatkokehitykseen sovitut näkymät

Tilaajan kanssa jatkokehitykseen sovitut näkymät ovat mediatiedostojen lataamis- ja tarkastelunäkymä, raporttinäkymä sekä ohjenäkymä.

Mediatiedostojen lataamis- ja tarkastelunäkymässä tutkija voi lisätä mediatiedostoja järjestelmään ja esikatsella järjestelmässä olevia mediatiedostoja. Mediatiedostoilla tarkoitetaan kuva-, ääni- tai videotiedostoja. Mediatiedostoja voi järjestelmään lisäämisen jälkeen käyttää mediakomponentissa (katso luku 3.7) näytettävänä sisältönä.

Raporttinäkymässä tutkija voi tarkastella järjestelmään tallentunutta suoritedataa tekstinä ja/tai graafisessa muodossa sekä luoda erilaisia raportteja eri tiedostoformaatteihin, esimerkiksi pdf-muotoon.

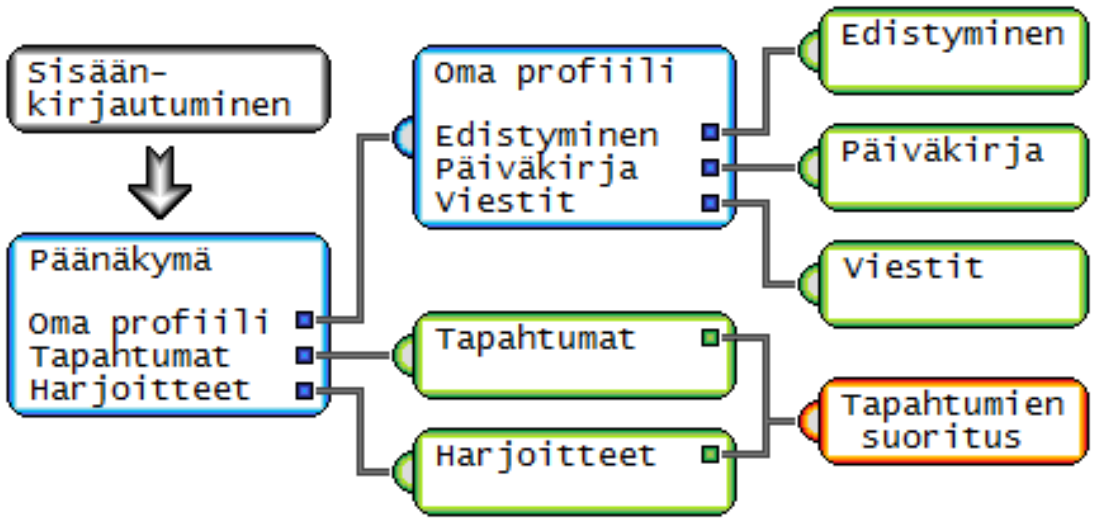
Ohjenäkymässä tutkija voi lukea sovelluksen käyttöohjeita. Mahdollisesti kyseinen ohjenäkymä saattaa olla järkevä avautua omaan selainikkunaan, eikä sen välttämättä tarvitsisi sijaita päävälilehtenä.

4 Kuntoutettavan käyttöliittymä

Toteutettu kuntoutettavan käyttöliittymä ei sisällä kaikkia tarvittavia toiminnallisuuksia. Toteutettuihin toiminnallisuuksiin kuuluvat sovelluksen sisäänkirjautuminen, edistymisen seuraaminen sekä ajoitettujen ja vapaaehtoisten tapahtumien suorittaminen. Paatti-projektin jälkeisessä jatkokehityksessä toteutetaan päiväkirjaviestien luonti ja lukeminen sekä ryhmäkohtaisten viestien lukeminen. Luvussa esitellään pääosin kuntoutettavan käytettäväksi toteutetun sovelluksen käyttöliittymän eri näkymät sekä niihin liittyvät toiminnot ja niiden väliset suhteet.

4.1 Sovelluksen näkymäkartta

Kuvassa 4.1 havainnollistetaan kuntoutettavan käyttöliittymän näkymien väliset suhteet näkymäkarttana. Luvuissa 4.3 – 4.8 kuvataan kunkin näkymän toiminnallisuus ja yhteydet muihin näkymiin.



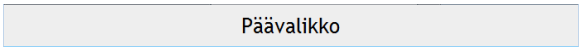
Kuva 4.1: Kuntoutettavan käyttöliittymän näkymäkartta.

Kuntoutettavan käyttöliittymän sovellukseen kirjaudutaan sisäänkirjautumisnäkymästä. Sisäänkirjautuva käyttäjä kirjoittaa käyttäjätunnuksensa ja salasanansa kirjautuessaan sisään. Uloskirjautuminen sovelluksesta tapahtuu painamalla päänäkymän vasemman alareunan painiketta (katso kuva 4.4).

4.2 Sovelluksen näkymien rakenne

Kuntoutettavan käyttöliittymän näkymien rakenne ja ulkoasu noudattaa samaa tyyliä koko sovelluksessa. Jokainen näkymä on jaettu kolmeen osaan. Ylimpänä on **otsikkotieto** (katso kuva 4.2), jossa näytetään valikoissa valikon nimi ja tapahtumaa suoritettaessa tapahtuman nimi. Näkymämallin keskellä olevassa **sisältöalueessa** (katso kuva 4.3) näytetään valikoissa navigointipainikkeet, alavalikoissa valikon sisältö ja tapahtumaa suoritettaessa tapahtuman tehtävän sisältö.

Näkymämallissa alimpana on varattu **alue toimintopainikkeille** (kuva 4.4). Vasen toimintopainike toimii aina takaisin-, peruutus- tai kieltopainikkeena, kun keskimmäinen ja oikea painike toimivat eri tavalla eri näkymissä. Esimerkiksi tapahtumaa suoritettaessa keskimmäinen ja oikea painike toimivat tapahtuman tehtävien etenemis- ja paluupainikkeina. Oikeaa painiketta käytetään myös hyväksymispainikkeena tietyissä tilanteissa, kuten esimerkiksi tapahtuman keskeyttämisessä.



Kuva 4.2: Käyttöliittymän otsikkoalue.

Kuva 4.3: Käyttöliittymän sisältöalue.

Kuva 4.4: Käyttöliittymän navigointipainikealue.

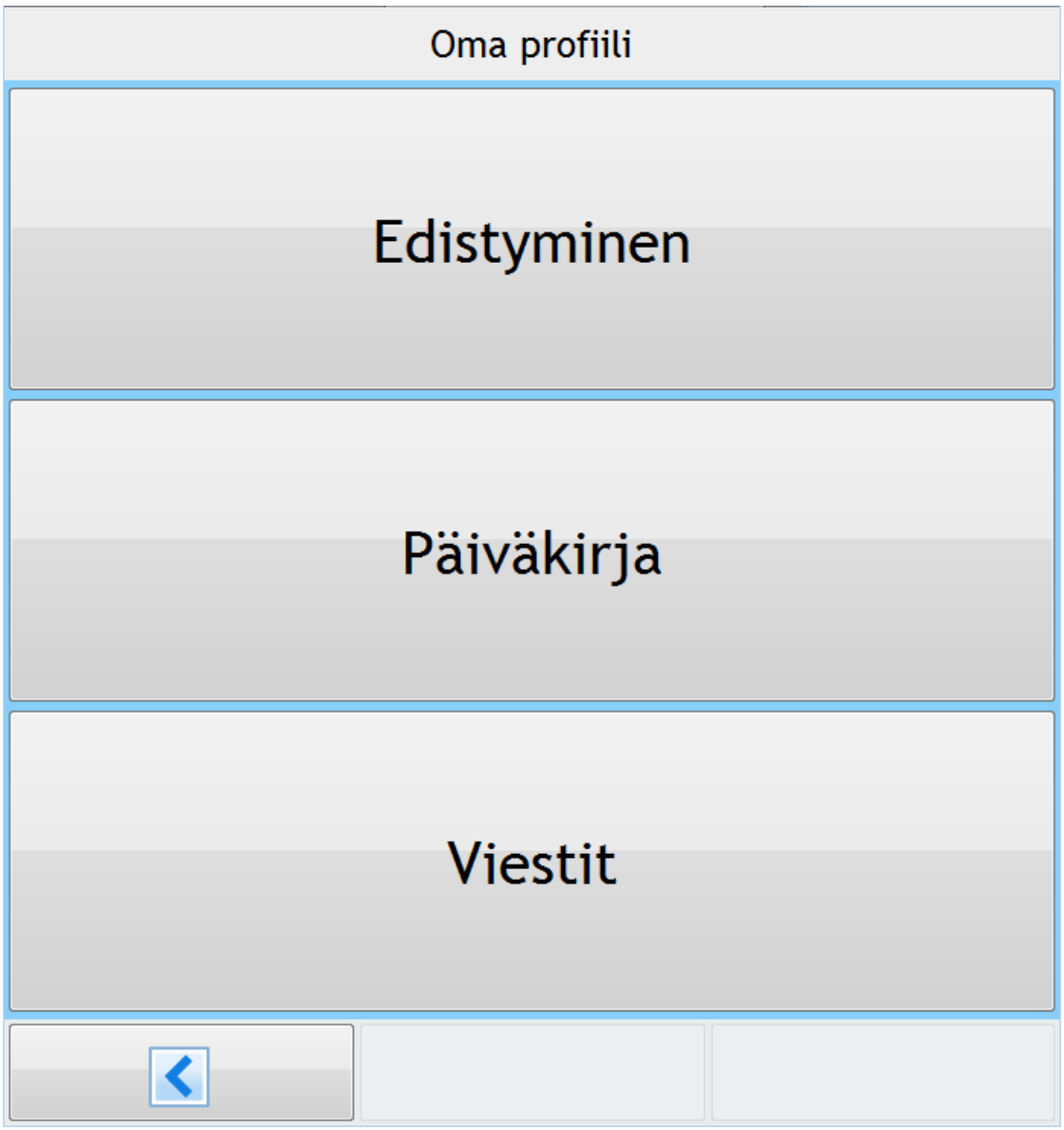
4.3 Sovelluksen päänäkymä

Sovelluksen päänäkymästä (katso kuva 4.5) pääsee näkymiin *Oma profiili*, *Tapahtumat* ja *Harjoitteet* (katso luvut 4.4, 4.6 ja 4.8) vastaavilla painikkeilla. Navigointipainikkeiden vasemmalla painikkeella kuntoutettava voi kirjautua ulos sovelluksesta ja keskipainikkeella kuntoutettavalle avautuu näkymä, jossa näytetään sovellukseen liittyviä tietoja.

Kuva 4.5: Sovelluksen päävalikko.

4.4 Oma profiili -näkymä

Kuvan 4.6 *Oma profiili* -näkymän kautta pääsee näkymiin *Edistyminen*, *Päiväkirja* sekä *Viestit* (katso luvut 4.5 ja 4.9) vastaavilla painikkeilla. Navigointipainikkeiden vasemmalla painikkeella kuntoutettava voi palata takaisin luvun 4.3 päänäkymään. Muilla alareunan painikkeilla ei näkymässä ole toiminnallisuutta.



Kuva 4.6: *Oma profiili* -alavalikko.

4.5 Edistyminen-näkymä

Kuntoutettava pääsee kuvan 4.7 *Edistyminen*-näkymään luvun 4.4 näkymän kautta. Näkymässä kuntoutettava voi tarkastella suorittamiaan tapahtumia. Navigointipainikkeiden vasemmalla painikkeella kuntoutettava voi palata takaisin luvun 4.4 näkymään. Muilla alareunan painikkeilla ei näkymässä ole toiminnallisuutta.

Kuva 4.7: *Edistyminen*-alavalikko.

4.6 Tapahtumat-näkymä

Kuntoutettava pääsee kuvan 4.8 *Tapahtumat*-näkymään luvun 4.3 päänäkymän kautta. Näkymässä näytetään kuntoutettavalle suoritettavien tapahtumien lista. Kuntoutettava voi valita tapahtumalistasta suoritettavaksi haluamansa tapahtuman, jos se on ajoituksen osalta suoritettavissa sillä hetkellä. Tapahtuman suorittamiseen siirrytään painamalla lista-alkion oikealla puolella olevaa *Aloita*-painiketta.

Kuva 4.8: *Tapahtumat*-näkymä.

Kuntoutettava voi alareunan vasemmalla painikkeella palata takaisin päävalikkoon. Muilla alareunan painikkeilla voidaan valita näkymään näytettäväksi tulevia tai rästiin jääneitä tapahtumia (katso kuvat 4.9 ja 4.10). Kyseiset painikkeet painautuvat pohjaan, kun niitä painaa ensimmäisen kerran. Painamalla painettua painiketta toisen kerran näkymä näyttää listan kaikista suorittamattomista tapahtumista.

Kuva 4.9: Tulevat tapahtumat.

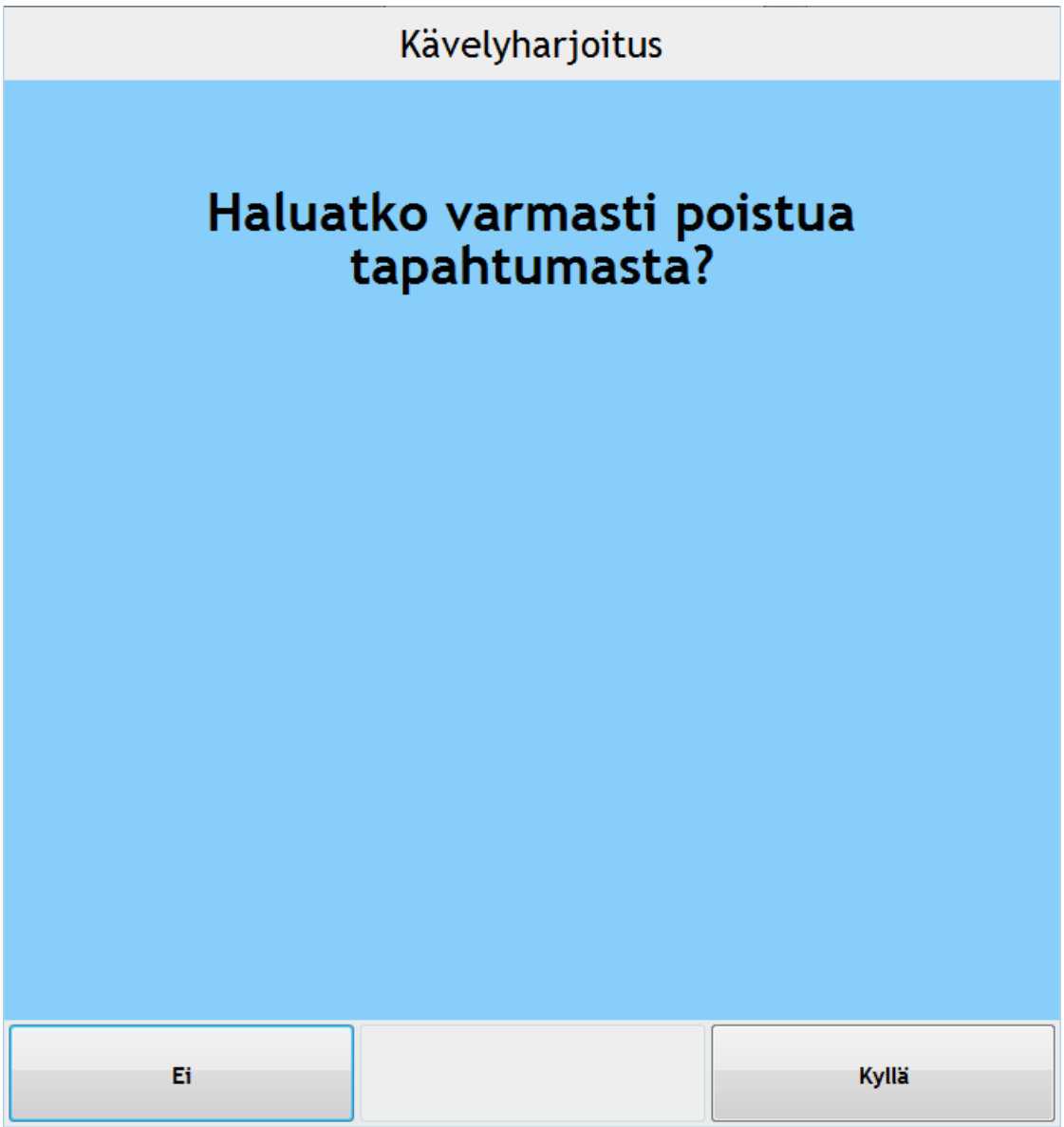
Kuva 4.10: Rästissä olevat tapahtumat.

4.7 Tapahtumien suorittaminen

Tapahtuman suorittaminen aloitetaan luvuissa 4.6 ja 4.8 esiteltyjen *Tapahtumat*- tai *Harjoitteet*-näkymien kautta. Tapahtuma sisältää yhden tai useamman tehtävän (katso kuva 4.11), jonka kuntoutettavan toivotaan suorittavan. Jotkin tehtävätyyppien näytettävistä sisällöistä kaipaavat kuntoutettavan kosketussyötettä. Esimerkiksi monivalintatehtävässä kuntoutettavan on valittava vaihtoehdoista sopivin, ennen kuin hän voi edetä tapahtumassa eteenpäin.

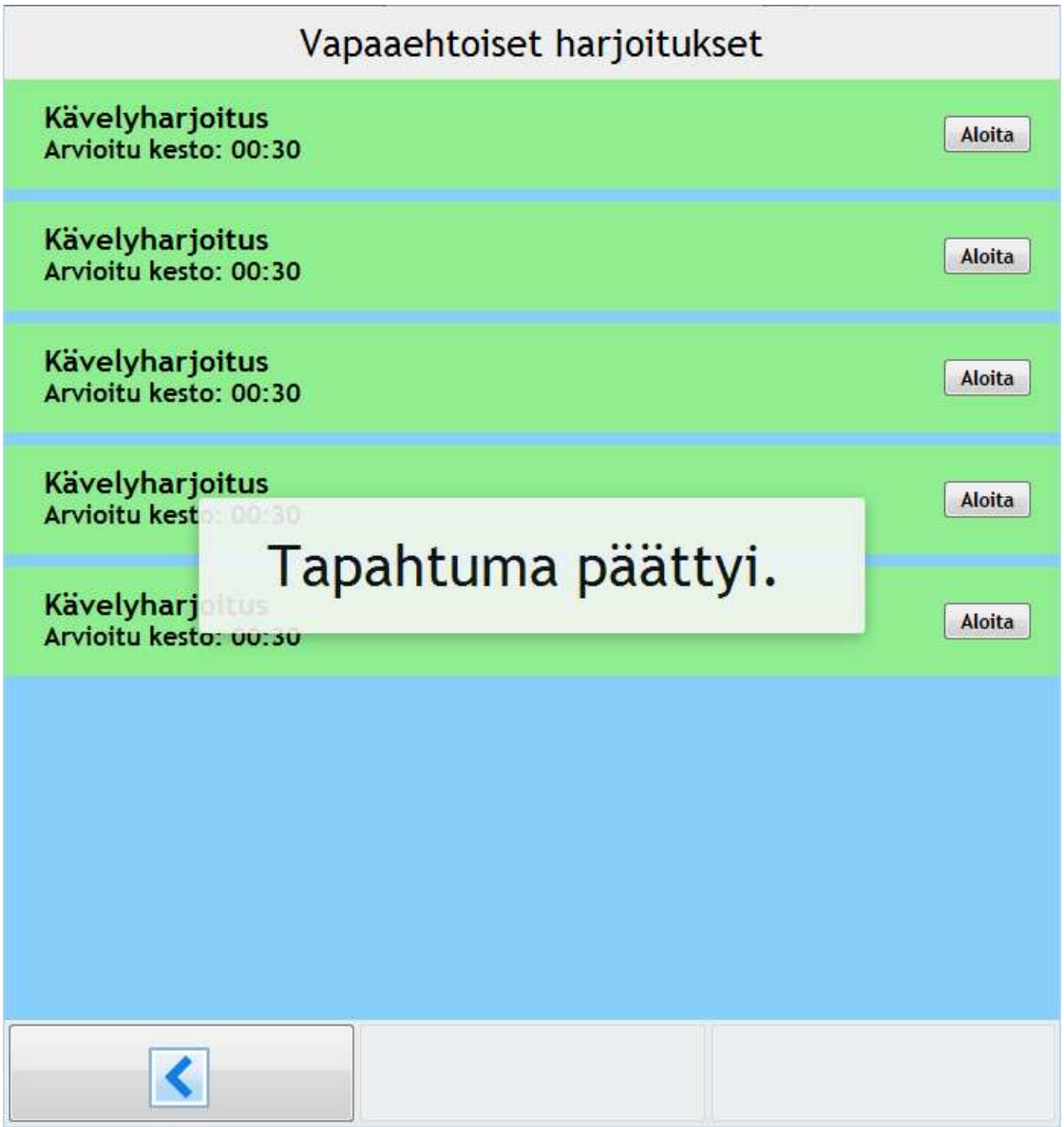
Kuva 4.11: Tapahtuman suorittaminen.

Tapahtuman tehtäviä pääsee eteen- ja taaksepäin alareunan keskimmäisellä ja oikealla painikkeella. Alareunan vasemmalla painikkeella kuntoutettava voi peruuttaa tapahtuman suorittamisen. Tällöin kuntoutettavalta varmistetaan kuvan 4.12 mukaisesti, että hän haluaa varmasti poistua tapahtumasta.



Kuva 4.12: Tapahtumasta poistumisen varmistus.

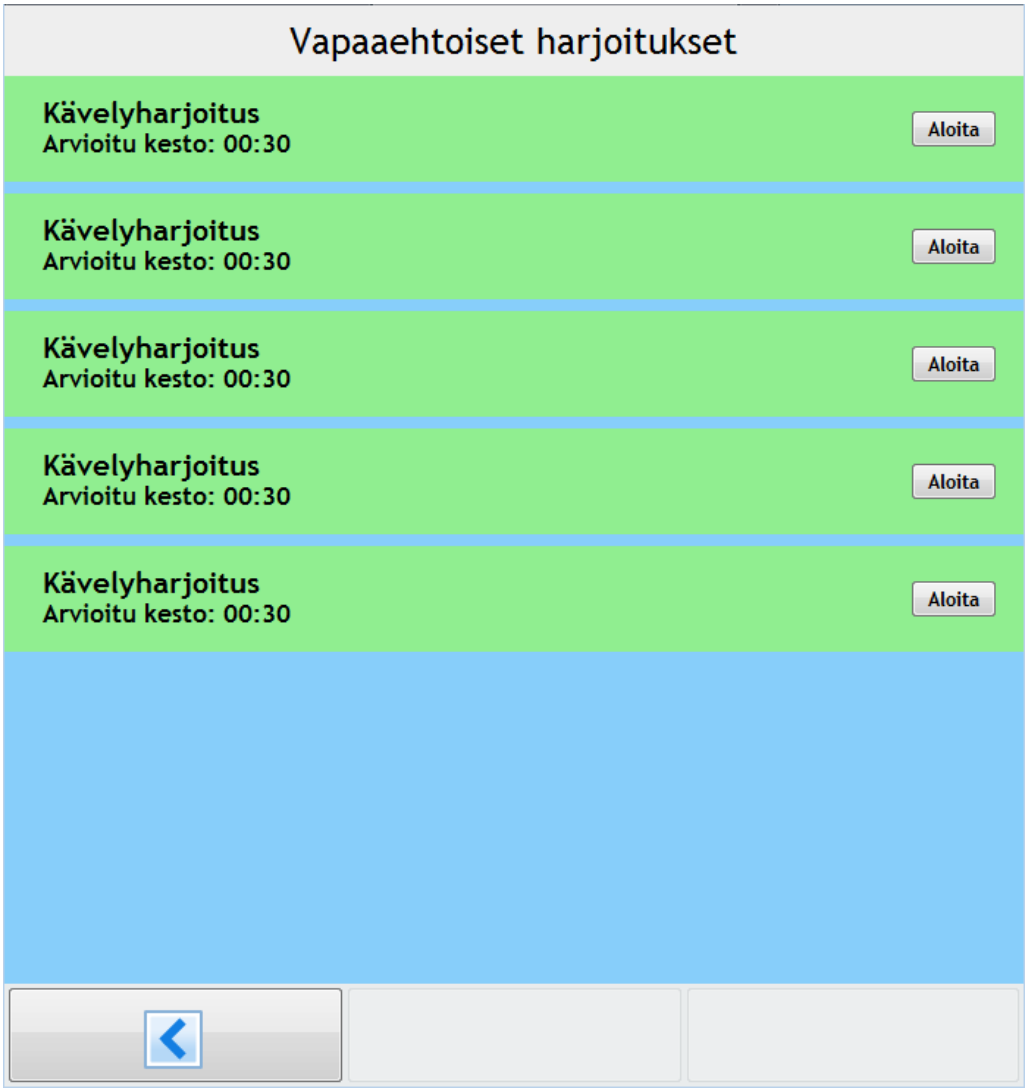
Kun kuntoutettava on suorittanut tapahtuman, palaa sovellus takaisin *Tapahtumat*- tai *Harjoitteet*-näkymään. Tapahtuman päättymisestä kuntoutettavalle näytetään ilmoitus kuvan 4.13 mukaisesti.



Kuva 4.13: Tapahtuman päättyminen.

4.8 Harjoitteet-näkymä

Kuntoutettava pääsee *Harjoitteet*-näkymään luvussa 4.3 esitetyn päänäkymän kautta. *Harjoitteet*- ja *Tapahtumat*-näkymät ovat ulkoasultaan ja toiminnallisuuksiltaan lähes samanlaisia. Kuvan 4.14 *Harjoitteet*-näkymässä näkyy kuntoutettavalle suunnattuja vapaaehtoisia tapahtumia, joita hän voi suorittaa useamman kerran tai niitä ei ole välttämätöntä suorittaa ollenkaan.

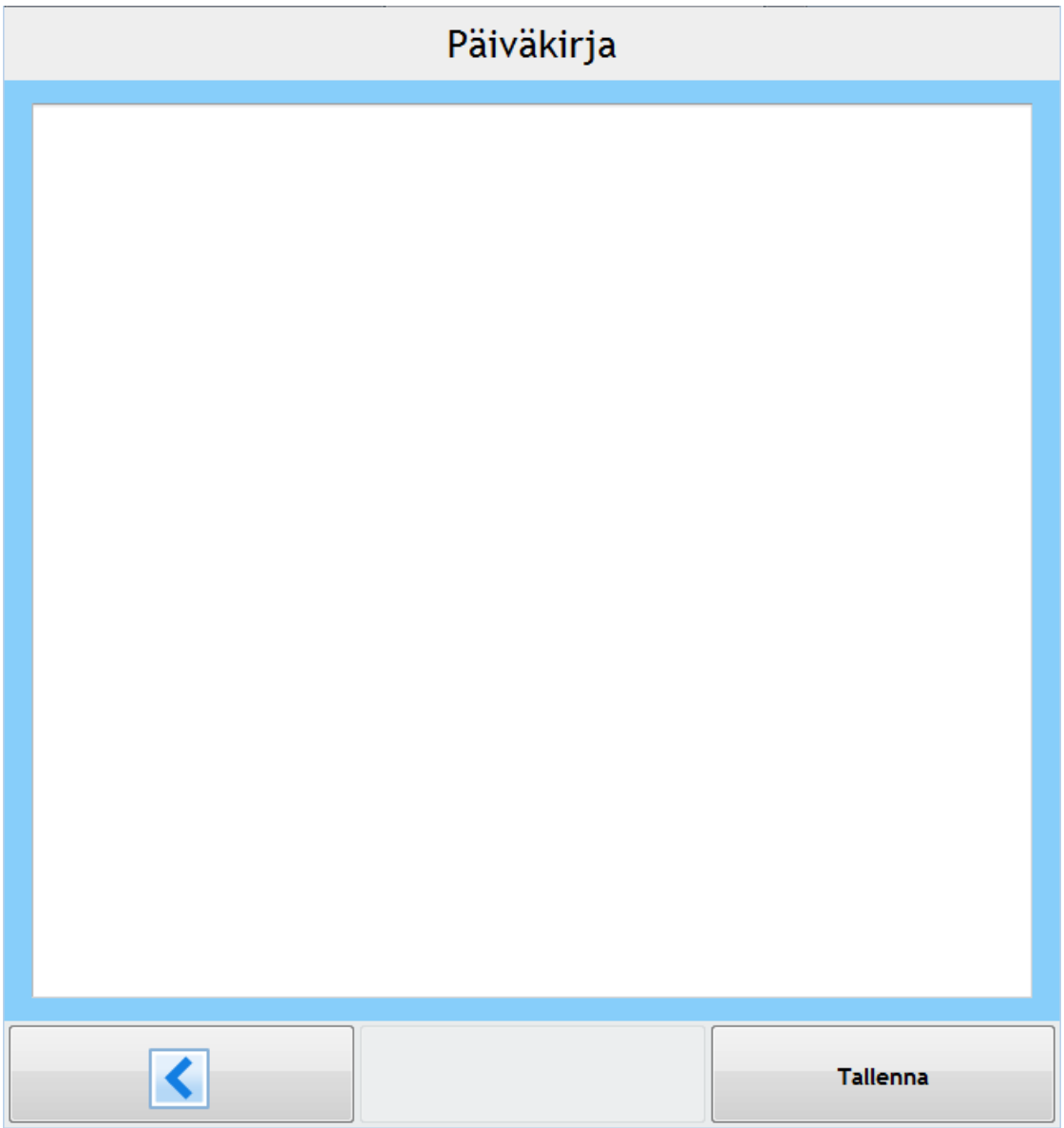


Kuva 4.14: *Harjoitteet*-näkymä.

Harjoitteet-näkymän vasemman alareunan painikkeella kuntoutettava voi palata takaisin luvun 4.3 päänäkymään. Tapahtuman suorittamiseen siirrytään samalla tavalla kuin luvussa 4.6 kuvatussa *Tapahtumat*-näkymässä.

4.9 Jatkokehitykseen sovitut näkymät

Tilaajan kanssa jatkokehitykseen sovitut näkymät ovat *Päiväkirja* ja *Viestit*. Kuvan 4.15 *Päiväkirja*-näkymässä kuntoutettava voi tallentaa päiväkirjamerkintöjä joko kuva-, ääni- tai tekstimuodossa. Kukin päiväkirjan merkintä tallentuu järjestelmän tietokantaan.



Kuva 4.15: *Päiväkirja*-näkymä.

Kuvan 4.16 *Viestit*-näkymässä kuntoutettava voi tarkastella tutkijoiden lähettämiä ryhmäkohtaisia viestejä. *Viestit*-näkymän olennaisin käyttötarkoitus on toimia palautekanavana tutkijan ja kuntoutettavan välillä.

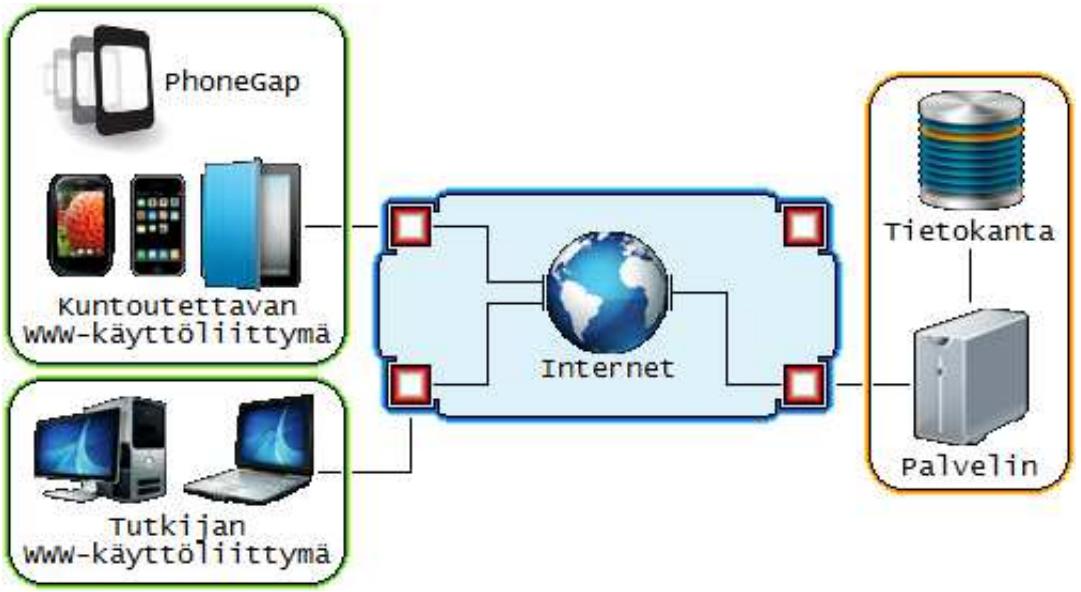
Kuva 4.16: *Viestit*-näkymä.

5 Tietojärjestelmän rakenne

Sovellus koostuu kahdesta WWW-selainsovelluksesta, jotka ovat yhteydessä palvelimella olevaan projektissa suunniteltuun tietokantaan. Sovellus käyttää Vaadin-sovelluskehitysympäristön käyttöliittymäkirjastoja ja komponentteja. Prototyypin käyttöliittymistä kuntoutettavan mobiilisovellus noudattaa enemmän MVC-mallia, kun taas tutkijapuolen sovellus hyödyntää sitä vain osittain. Luvussa kuvataan sovelluksen rakenne ja sen toteutusratkaisut yleisellä tasolla.

5.1 Tietojärjestelmän asiakas-palvelinarkkitehtuuri

Kuvassa 5.1 esitetään tietojärjestelmän yleinen rakenne sekä ulkopuoliset ohjelmistot, joita järjestelmä käyttää. Palvelimella olevaan MySQL-tietokannanhallintajärjestelmään tallennetaan tietojärjestelmän tiedot. Tomcat-palvelin tarjoaa suoritusalustan tutkijan ja kuntoutettavan käyttöliittymille, joita käytetään WWW-selainten kautta mobiililaitteilla ja tietokoneilla.

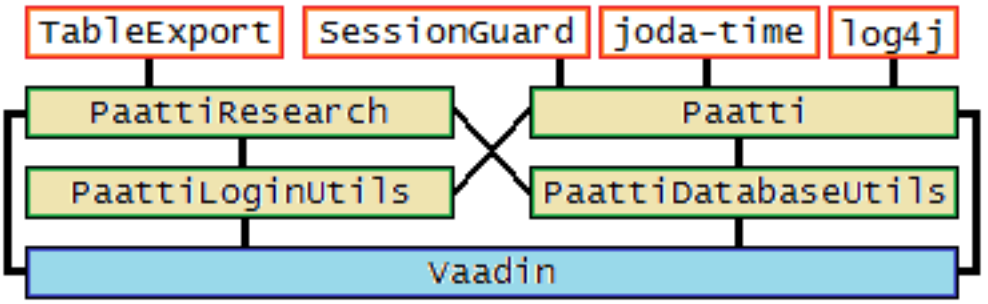


Kuva 5.1: Tietojärjestelmän asiakas-palvelinarkkitehtuuri.

Kuntoutettavan käyttöliittymä on suunniteltu käytettäväksi mobiililaitteella. Kuntoutettavan käyttöliittymä on suunniteltu käyttämään PhoneGap-laajennusta käsittelemään mobiililaitteiden erilaisia antureita, kuten kiihtyvyysanturia, kameraa ja värinätoimintoa.

5.2 Tietojärjestelmän rakenne ja rajapinnat

Kuvassa 5.2 kuvataan projektissa toteutetun tietojärjestelmän arkkitehtuuri kerrosmallina. Tietojärjestelmä hyödyntää Vaadin-sovelluskehitysalustan tarjoamaa rajapintaa. Luvuissa 5.3 – 5.7 kuvataan tarkemmin valittuja rakenteita sekä lyhyesti niissä olevien komponenttien käyttötarkoitus. Kaikki projektissa toteutetut komponentit, luokat ja metodit kuvataan tarkemmin luokkadokumentaatiossa [5].



Kuva 5.2: Tietojärjestelmän rakenne.

PaattiLoginUtils-pakettia ei käsitellä luvuissa 5.3 – 5.7 tarkemmin sen yksinkertaisen rakenteen takia. Se on kokoelma luokkia, joiden avulla sisäänkirjaudutaan sovelluksiin.

5.3 Ulkopuoliset komponentit

Tietojärjestelmä hyödyntää seuraavia ulkopuolisia komponentteja:

joda-time on Java-kirjasto, jota käytetään tapahtumien aikojen esittämisessä ja laskemisessa.

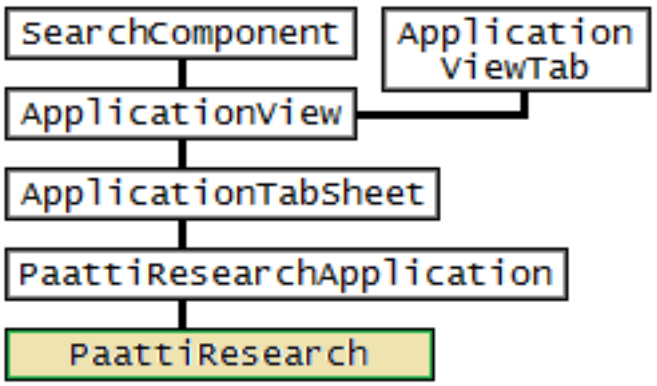
log4j on Java-kirjasto lokien kirjoittamiseen.

SessionGuard on Vaadinin lisäosa, joka pitää palvelimen ja sovelluksen yhteyden päällä, jotta käyttäjän ”istunto” ei vanhene.

TableExport on Vaadinin lisäosa, jota käytetään suoritedatan tallentamiseen Excel-taulukkomuodossa.

5.4 PaattiResearchin rakenne ja komponentit

Kuvassa 5.3 kuvataan PaattiResearch-paketin arkkitehtuuri kerrosmallina. Paketti sisältää kokoelman luokkia, joiden avulla käytetään tutkijan käyttöliittymää. Luvussa kuvataan tiivistetysti kuvassa 5.3 olevien komponenttien käyttötarkoitus.



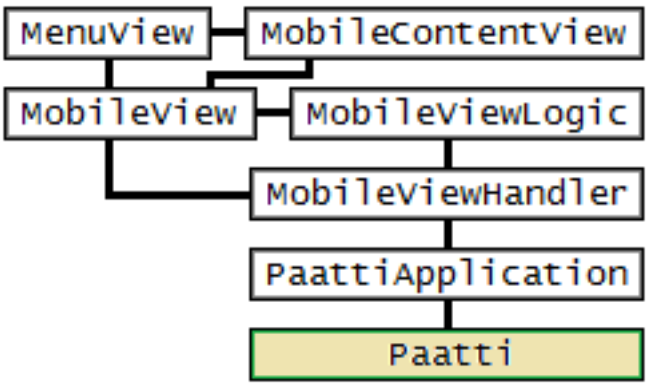
Kuva 5.3: PaattiResearchin rakenne.

PaattiResearch koostuu seuraavista komponenteista:

PaattiResearchApplication	on tutkijan käyttöliittymän ohjelma.
ApplicationTabSheet	sisältää kaikki tutkijan käyttöliittymän sovelluksen näkymien välilehdet.
ApplicationView	on abstraktiluokka, josta on peritty tutkijan käyttöliittymän sovelluksen muut näkymät.
ApplicationViewTab	on abstraktiluokka, josta on peritty tutkijan käyttöliittymän sovelluksen näkymissä olevat välilehdet.
SearchComponent	on tutkijan käyttöliittymän näkymissä oleva hakukomponentti.

5.5 Paatin rakenne ja komponentit

Kuvassa 5.4 kuvataan Paatti-paketin arkkitehtuuri kerrosmallina. Paketti sisältää kokoelman luokkia, joiden avulla käytetään kuntoutettavan käyttöliittymää. Luvussa kuvataan tiivistetysti kuvassa 5.4 olevien komponenttien käyttötarkoitus.



Kuva 5.4: Paatin rakenne.

Paatti koostuu seuraavista komponenteista:

PaattiApplication on kuntoutettavan käyttöliittymän ohjelma.

MobileViewHandler on kuntoutettavan käyttöliittymän näkymähallitsija, joka hoitaa yhteyden kaikkien näkymien ja tietokannan välillä.

MobileView on käyttöliittymän runko, johon asetetaan näkymän logiikka ja sisältö.

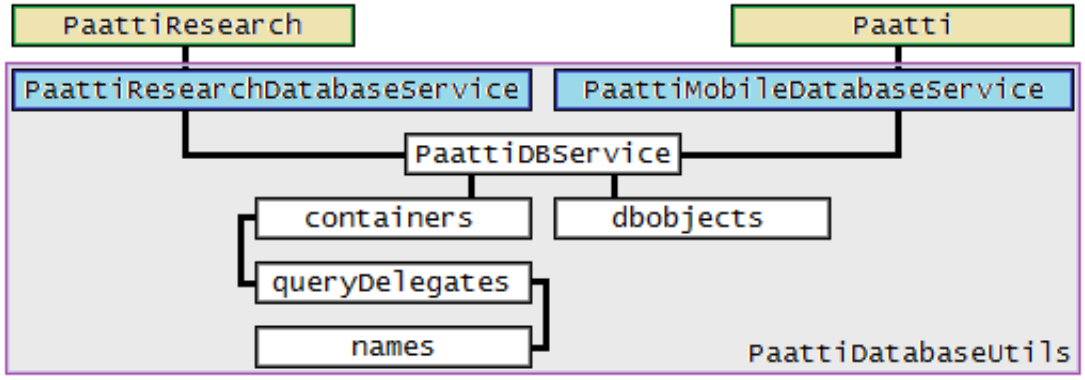
MobileViewLogic on luokka, josta on peritty kuntoutettavan käyttöliittymän näkymien logiikat.

MenuView on luokka, josta on peritty kuntoutettavan käyttöliittymän sovelluksen valikkonäkymät.

MobileContentView on luokka, josta on peritty kuntoutettavan käyttöliittymän sovelluksen näkymät.

5.6 PaattiDatabaseUtilsin rakenne ja komponentit

Kuvassa 5.5 kuvataan PaattiDatabaseUtils-paketin arkkitehtuuri kerrosmallina. Se sisältää kokoelman luokkia, joiden avulla käytetään tietojärjestelmän tietokantaa. Luvussa kuvataan tiivistetysti kuvassa 5.5 olevien komponenttien käyttötarkoitus.



Kuva 5.5: PaattiDatabaseUtilsin rakenne.

PaattiDatabaseUtils koostuu seuraavista komponenteista:

PaattiResearchDatabaseService on tutkijan käyttöliittymän käyttämä rajapinta olennaisten tietokantametodien käyttöön. Sillä estetään myös kaikkien tietokantametodien käyttö.

PaattiMobileDatabaseService on kuntoutettavan käyttöliittymän käyttämä rajapinta olennaisten tietokantametodien käyttöön. Sillä estetään myös kaikkien tietokantametodien käyttö.

PaattiDBService on tietojärjestelmän tietokannan toteutus.

dbobjects on käytön aikana olevien käyttäjän sekä hänen ryhmiensä ja rooliensa väliaikainen tallennuspaikka.

containers on luokka, joka on peritty SQLContainerista ja jota käytetään tapahtumien tehtävien valintojen tallentamiseen.

queryDelegates on luokka, joka avustaa tiedon käsittelyssä, laskemisessa ja tallentamisessa. Se toteuttaa FreeformStatementDelegate-rajapintaa.

names on luokka, joka sisältää tietokannan taulujen nimet.

5.7 Toteutusratkaisujen kehittyminen

Tietojärjestelmän toteutus aloitettiin maaliskuun alkupuolella kuntoutettavan käyttöliittymästä. Tietokannan suunnittelu ja tutkijan käyttöliittymän toteutus aloitettiin maaliskuun loppupuolella. Virtuaalipalvelimen ryhmä sai käyttöönsä huhtikuun alkupuolella testausta varten.

Kuntoutettavan käyttöliittymän toteutusratkaisujen suurin muutos tapahtui näkymien näyttämisessä. Kuntoutettavan käyttöliittymä latasi sovellusta käytettäessä seuraavan näkymän uudelleen aina, kun näkymästä siirryttiin toiseen. Myöhemmin kuntoutettavan käyttöliittymän näkymien latausta muutettiin niin, että kaikki sovelluksen näkymät luodaan vain kerran silloin, kun niihin mennään sovelluksessa ensimmäisen kerran. Sovellus tallentaa luodut näkymät ja lataa näytettäviä näkymiä uudelleen sisäisestä tietorakenteesta.

Tutkijan käyttöliittymän toteutusratkaisujen suurin muutos tapahtui projektin viimeistelyvaiheessa toukokuun alkupuolella. Sovelluksen rakennetta muutettiin noudattamaan enemmän MVC-arkkitehtuuria. Sovellusta myös refaktoroitiin hyödyntämään enemmän perintää ja rajapintoja koodin siistimisen ja sovelluksen suorituskyvyn parantamisen vuoksi.

6 Tietojärjestelmän tietokanta

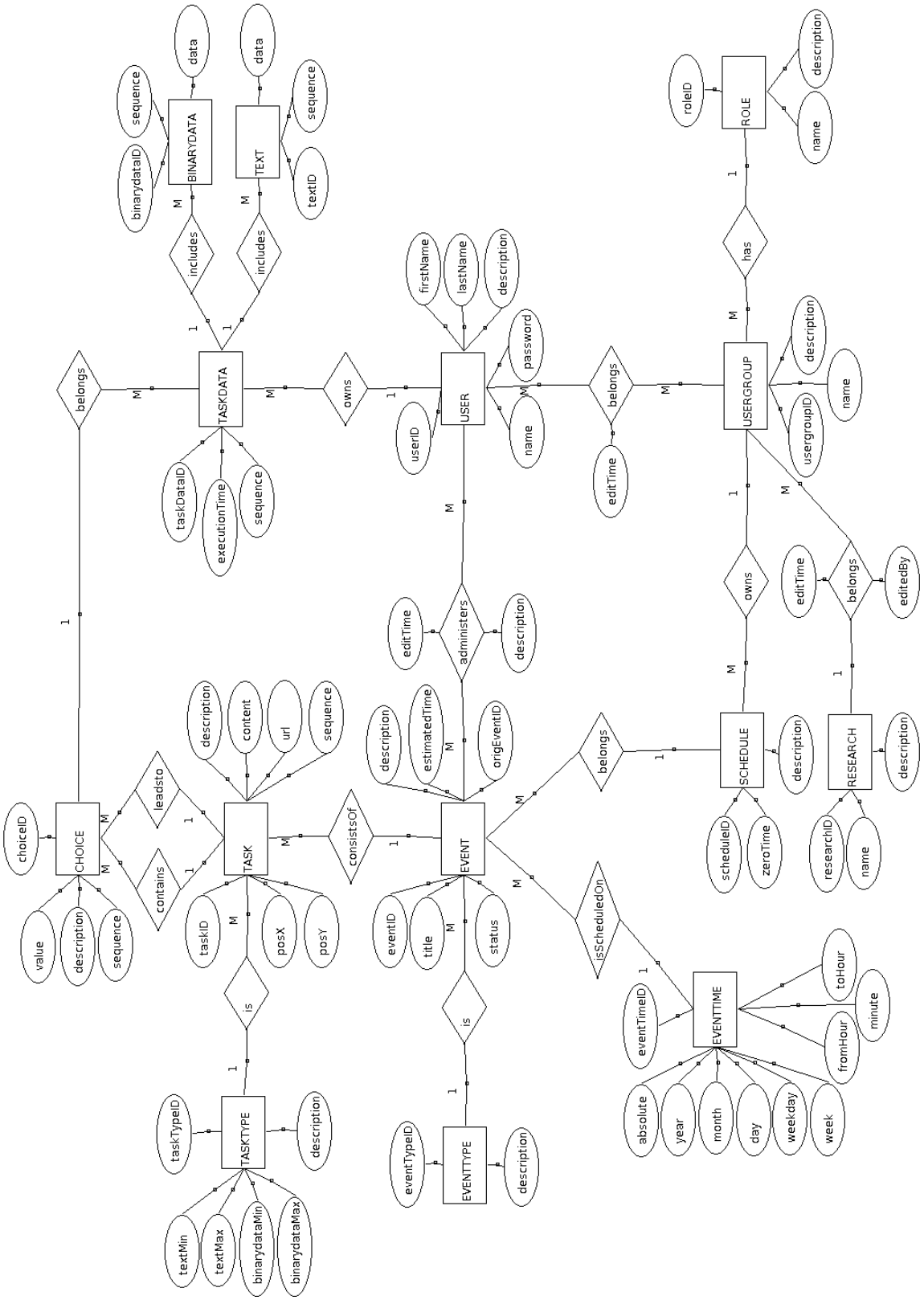
Projektryhmä suunnitteli tietokannan järjestelmälle. Tietokantaan tallennetaan sekä tutkijan luomia käyttäjiä, ryhmiä, tutkimuksia, kuntoutusohjelmia ja tapahtumia että kuntoutettavan tapahtumien suorittamisesta kerättyä suoritedataa.

6.1 Tietokannan ER-kaavio

Kuvan 6.1 ER-kaaviossa on esitetty tietokannan taulut ja niiden kentät sekä suhteet toisiin tietokannan tauluihin. Luvuissa 6.2 – 6.18 kuvataan tietokannan taulujen ja kenttien merkitys tarkemmin sekä kuvataan esimerkkitilanteita taulujen käyttöön liittyen.

Kuvan 6.1 tauluihin ei ole piirretty `rowStatus`-kenttää, sillä se on jokaisella taululla kenttänä. `RowStatus`-kenttään tallennetaan taulukossa olevan sarakkeen tila (kuten `ACTIVE` tai `DELETED`). Kyseiseen kenttään tallennetaan tekstimuotoista dataa (`string`).

Tietokannan tauluista `BINARYDATA` ja `TEXTDATA` eivät ole tällä hetkellä käytössä.



Kuva 6.1: Järjestelmän tietokanta ER-kaaviona.

6.2 USER-taulu

USER-tauluun tallennetaan järjestelmän käyttäjät. Taulu sisältää seuraavat kentät:

userID on käyttäjän yksilöivä numero tietokannassa (int).

firstName on käyttäjän etunimi (string).

lastName on käyttäjän sukunimi (string).

name on käyttäjän käyttäjätunnus (string).

password on käyttäjän salasana (string).

description on lyhyt kuvaus käyttäjästä (string).

USER-taulu on sidottu relaatiolla TASKDATA-, EVENT- ja USERGROUP-tauluihin. Yksi käyttäjä voi kuulua useampaan ryhmään, luoda ja muokata useita tapahtumia sekä omistaa tapahtumista kerättyä suoritedataa.

6.3 USERGROUP-taulu

USERGROUP-tauluun tallennetaan järjestelmän käyttäjäryhmät. Taulu sisältää seuraavat kentät:

usergroupID on käyttäjäryhmän yksilöivä numero tietokannassa (int).

name on ryhmän nimi (string).

description on lyhyt kuvaus ryhmästä (string).

USERGROUP-taulu on sidottu ROLE-tauluun relaatiolla, joka määrää ryhmän roolin. Rooli voi olla esimerkiksi tutkija tai kuntoutettava. RESEARCH-relaatio BELONGS-tauluun ilmaisee, kuka ja milloin on lisännyt tai poistanut ryhmän tutkimuksesta.

6.4 ROLE-taulu

ROLE-tauluun tallennetaan järjestelmän ryhmien roolityypit. Taulu sisältää seuraavat kentät:

roleID on roolin yksilöivä numero tietokannassa (int).

name on roolin nimi (string).

description on lyhyt kuvaus roolista (string).

6.5 RESEARCH-taulu

RESEARCH-tauluun tallennetaan järjestelmän tutkimukset. Taulu sisältää seuraavat kentät:

researchID on tutkimuksen yksilöivä numero tietokannassa (int).

name on tutkimuksen nimi (string).

description on lyhyt kuvaus tutkimuksesta (string).

6.6 SCHEDULE-taulu

SCHEDULE-tauluun tallennetaan järjestelmän kuntoutusohjelmat. Taulu sisältää seuraavat kentät:

scheduleID on kuntoutusohjelman yksilöivä numero tietokannassa (int).

zeroTime on kuntoutusohjelman alkamisajankohta (Date).

description on lyhyt kuvaus kuntoutusohjelmasta (string).

SCHEDULE-taulu on sidottu relaatiolla EVENT- ja USERGROUP-tauluihin. Yksi kuntoutusohjelma voi sisältää useamman tapahtuman ja olla useammalla ryhmällä.

6.7 EVENT-taulu

EVENT-tauluun tallennetaan järjestelmän tapahtumat. Taulu sisältää seuraavat kentät:

eventID on tapahtuman yksilöivä numero tietokannassa (int).

title on tapahtuman otsikko (string).

status on tapahtuman tila (int).

origEventID on tapahtuman alkuperäisyyden ilmaisemiseen käytettävä tunnistenumero (int).

estimatedTime on tapahtuman arvioitu kesto (int).

description on lyhyt kuvaus tapahtumasta (string).

EVENT-taulu on sidottu relaatiolla tauluihin EVENT, SCHEDULE, EVENTTIME, EVENTTYPE ja TASK. Tapahtuma voi kuulua vain yhteen kuntoutusohjelmaan. Tapahtumalle asetetaan vain yksi ennalta määrätyistä tyypeistä. Relaatio EVENTTIME-tauluun kertoo tapahtuman ajastuksen kuntoutusohjelmassa.

6.8 EVENTTIME-taulu

EVENTTIME-tauluun tallennetaan aikataulutettujen tapahtumien ajoitus kuntoutusohjelmassa. Taulu sisältää seuraavat kentät:

eventTimeID on ajan yksilöivä numero tietokannassa (int).

absolute ilmaisee ajan absoluuttisuuden tai suhteellisuuden (int).

year ilmaisee suoritushetken vuoden (int).

month ilmaisee suoritushetken kuukauden (int).

day ilmaisee suoritushetken päivän (int).

weekday ilmaisee suoritushetken viikonpäivän (string).

week	ilmaisee suoritushetken viikon (int).
fromHour	ilmaisee suoritushetken aloitustunnin (int).
toHour	ilmaisee suoritushetken lopetustunnin (int).
minute	ilmaisee suoritushetken minuuttia (int).

Kentät fromHour ja toHour ilmaisevat ajanjaksoa, jolloin tapahtuma on tehtävissä. Jos kuntoutettava ei suorita tapahtumaa asetetulla ajankohdalla, tapahtuma siirtyy rästitapahtumaksi.

6.9 EVENTTYPE-taulu

EVENTTYPE-tauluun tallennetaan tapahtumatyypit (kuten kysely ja liikunta). Taulu sisältää seuraavat kentät:

eventTypeID	on tapahtumatyypin yksilöivä numero tietokannassa (int).
description	on tapahtumatyypin kuvaus (string).

6.10 TASK-taulu

TASK-tauluun tallennetaan tapahtumiin liittyvät tehtävät. Taulu sisältää seuraavat kentät:

taskID	on tehtävän yksilöivä numero tietokannassa (int).
content	on tehtävän otsikko (string).
url	on tehtävän URL-osoite (string).
sequence	on tehtävän järjestys tapahtumassa (int).
description	on lyhyt kuvaus tehtävästä (string).
posX	on tapahtuman x-koordinaatti piirtoalueella (int).
posY	on tapahtuman y-koordinaatti piirtoalueella (int).

TASK-taulu on sidottu relaatiolla EVENT-, TASKTYPE- ja CHOICE-tauluihin. Yhdelle tehtävälle voidaan asettaa vain yksi tehtävätyyppi. Yksi tehtävä voi kuulua vain yhteen tapahtumaan.

6.11 TASKTYPE-taulu

TASKTYPE-tauluun tallennetaan tehtävätyypit (kuten teksti ja monivalinta). Taulu sisältää seuraavat kentät:

taskTypeID on tehtävätyypin yksilöivä numero tietokannassa (int).

textMin on tehtävätyypin tekstimuotoisen datan minimimäärä (int).

textMax on tehtävätyypin tekstimuotoisen datan maksimimäärä (int).

binaryDataMin on tehtävätyypin binäärimuotoisen datan minimimäärä (int).

binaryDataMax on tehtävätyypin binäärimuotoisen datan maksimimäärä (int).

description on lyhyt kuvaus tehtävätyypistä (string).

6.12 CHOICE-taulu

CHOICE-tauluun tallennetaan tehtävien sisällä olevat valinnat. Taulu sisältää seuraavat kentät:

choiceID on valinnan yksilöivä numero tietokannassa (int).

value on valinnan arvo (int).

sequence on valinnan järjestys tehtävässä (int).

description on valinnassa näytettävä teksti (string).

CHOICE-taulu on sidottu relaatiolla TASK- ja TASKDATA-tauluihin. Valinta kuuluu vain yhteen tehtävään ja vastaavasti johtaa vain yhteen tehtävään. Tehtävään voi kuulua yksi tai useampi valinta, ja yhteen tehtävään voi johtaa useampi valinta.

6.13 TASKDATA-taulu

TASKDATA-tauluun tallennetaan tapahtumien suoritedataan liittyvät tiedot. Taulu sisältää seuraavat kentät:

taskdataID on suoritedatan yksilöivä numero tietokannassa (int).

executionTime on suoritedatan suoritusajankohta (timestamp).

sequence on suoritedatan järjestys tehtävässä (int).

TASKDATA-taulu on sidottu relaatiolla tauluihin CHOICE, BINARYDATA, TEXTDATA ja USER. Käyttäjällä voi olla useampi saman tehtävän suorittamisesta kerätty suoritedata. Suoritedata tallentuu aina, kun tapahtumassa siirrytään uuteen tehtävään. Suoritedata voi liittyä vain yhteen valintaan.

6.14 BINARYDATA-taulu

BINARYDATA-tauluun tallennetaan binäärimuotoinen suoritedata (kuten kuva ja ääni). Taulu sisältää seuraavat kentät:

binarydataID on binäärimuotoista dataa sisältävän suoritedatan yksilöivä numero tietokannassa (int).

sequence on datan järjestys suoritedatassa (int).

data on binaarimuotoinen suoritedata (blob).

6.15 TEXTDATA-taulu

TEXTDATA-tauluun tallennetaan tekstimuotoinen suoritedata. Taulu sisältää seuraavat kentät:

textID on tekstimuotoista dataa sisältävän suoritedatan yksilöivä numero tietokannassa (int).

sequence on datan järjestys suoritedatassa (int).

data on tekstimuotoinen suoritedata (text).

6.16 USER – USERGROUP -relaatiotaulu

`USER – USERGROUP` -relaatiotauluun tallennetaan käyttäjäryhmän käyttäjien muokkausajankohta. Taulu sisältää seuraavan kentän:

`editTime` on muokkaamisen ajankohta (`timestamp`).

6.17 USERGROUP – RESEARCH -relaatiotaulu

`USERGROUP – RESEARCH` -relaatiotauluun tallennetaan tutkimuksen ryhmien muokkausajankohta ja kuvaus. Taulu sisältää seuraavan kentän:

`editTime` on muokkaamisen ajankohta (`timestamp`).

`editedBy` kertoo muokkauksen tekijän (`string`).

6.18 EVENT – USER -relaatiotaulu

`EVENT – USER` -relaatiotauluun tallennetaan tapahtuman muokkausajankohta ja kuvaus. Taulu sisältää seuraavan kentän:

`editTime` on muokkaamisen ajankohta (`timestamp`).

`description` on muokkaajan kirjaama muokkaussyy (`string`).

7 Ohjelmointikäytänteet

Luvussa kuvataan projektin ohjelmointikäytänteet. Projektissa noudatettiin suunnitteluvaiheessa määriteltyjä ohjelmointikäytänteitä.

7.1 Lähdekoodin muotoilu ja tiedostojen nimeäminen

Lähdekoodi ohjelmoitiin projektisuunnitelmassa [7] sovittujen käytänteiden mukaisesti käyttäen mallina Javan yleisiä käytänteitä [9]. Tiedostojen ja lähdekoodin nimeäminen toteutui suunnitelman mukaisesti, sekä verkkosivulle ja CD:lle sijoitetut dokumentit vastasivat määriteltyjä käytänteitä.

Luokkien nimissä ensimmäinen kirjain kirjoitettiin pienellä ja sanat kirjoitettiin yhteen isoin alkukirjaimin. Funktioiden ja muuttujien nimissä ensimmäinen kirjain kirjoitettiin pienellä ja sanat kirjoitettiin yhteen isoin alkukirjaimin. Lähdekooditiedostojen hakemistorakenne oli valmiiksi määritelty Vaadin-sovelluskehitysympäristössä, eikä rakennetta muutettu. Muiden tiedostojen ja hakemistojen nimet kirjoitettiin pienillä kirjaimilla ja englanniksi.

7.2 Lähdekoodin kommentointi

Lähdekoodit kommentointiin englanniksi. Kommentit aloitettiin yksittäisten rivien tai rivin loppujen tapauksissa merkillä `//`. Useamman rivin tapauksessa kommentti aloitettiin merkillä `/*` ja päätettiin merkillä `*/`. Luokkien ja funktioiden kommentit aloitettiin `/**` ja päätettiin `*/`. Luokkadokumentointi muodostettiin JavaDocin avulla, jonka kommentointiohjeita noudatettiin lähdekoodissa.

7.3 Lähdekoodiesimerkki

Sovelluksessa noudatettua lähdekoodin sisennystä ja kommentointia havainnollistaa seuraava esimerkki:

```
/**
* Layout for all the mobile views has slots for header,
* footer and the content.
*
* @author Lauri Satokangas, lauri.n.satokangas@student.jyu.fi
* @date 13.5.2012
*/
public abstract class MobileView extends VerticalLayout {

    private Panel pageContent;
    private final HashMap<Object, Component> footerComponents;

    // This is the left button in the footer layout.
    public Object FOOTER_BUTTON_LEFT = "leftfooterbutton";

    /**
     * Get the component in the footer.
     * @return footerComponent
     */
    public HashMap<Object, Component> getFooterComponents() {
        return footerComponents;
    }

...
```

7.4 Kehitysympäristö

Tietojärjestelmän sovellusten kehitystyökaluna käytettiin NetBeans IDE:n versiota 7.1. Lisäosina käytettiin Maven-automatisointityökalua lähdekoodin kääntämiseen sekä Vaadin-sovelluskehitysalustan käyttöliittymäkirjastoja ja komponentteja sovelluksen toteutukseen hyödyntäen lähteitä [1] ja [14].

7.5 Testauksen käytänteet

Sovellusten toteutusvaiheessa jokainen ryhmän jäsen suoritti käytännön yksikkötestausta heti koodiin tekemänsä muutoksen jälkeen omatoimisesti. Ajan puutteen vuoksi tietojärjestelmälle ei laadittu järjestelmätestaussuunnitelmaa, eikä ryhmä ehtinyt myöskään laatia yksikkötestejä sovelluksille. Muutama testitiedosto ohjelmoitiin tietokantaa käsitteleville tiedostoille. Integraatiotestausta ryhmä suoritti projektin viimeistelyvaiheessa jonkin verran.

Ryhmälle jaettiin ennen toteutuksen aloittamista erilaisia mobiililaitteita ja sormitietokoneita. Niiden avulla ryhmä testasi kuntoutettavan sovellusta tietyn väliajoin ja sai näin arvokasta tietoa liittyen sovelluksen toimivuuteen eri mobiililaitteissa.

Ryhmälle järjestettiin maaliskuun puolessavälissä käytettävyyspäivä Meeri Mäntylän toimesta. Käytettävyyspäivän aikana ryhmä sai käyttöliittymien käytettävyyteen liittyvää tietoa ja palautetta projektissa toteutettavista käyttöliittymistä kirjaten ne muistioksi [4].

Ryhmän jäsenet harjoittivat myös omatoimista käyttäjätestausta havaitakseen sovellusten toiminnallisuuksiin liittyviä puutteita ja bugeja. Ryhmä testasi sovellusten toiminnallisuuksien toimivuutta tasaisin väliajoin, varsinkin ennen projektin loppuesitystä.

Sovellusta ryhmän ohella testasi pari muuta henkilöä projektiorganisaatiosta projektin aikana. Laajimman palautteen sovellusten toimivuudesta projektiryhmä sai vastaavalta ohjaajalta Jukka-Pekka Santaselta sähköposteina [10], [11], [12] ja [13]. Sovelluksen testaukseen osallistui tilaajan edustajista projektin jatkokehityksestä kiinnostunut Tero Kallioinen lähettäen sähköpostit [2] ja [3].

Projektin tilaajan muiden edustajien kiireellisen aikataulun vuoksi heiltä saatiin vain vähän palautetta liittyen sovellusten käyttöliittymien toimivuuteen. Heidän ainoat käyttötestaukset tapahtuivat projektin toiseksi viimeisessä kokouksessa. Projektin tilaajilta saatiin kuitenkin testidataa testiaineiston luomiselle, ja sitä hyödynnettiin projektissa jonkin verran.

Projektiryhmä ei ajan puutteen vuoksi ehtinyt testaamaan sovelluksia niiden loppukäyttäjillä. Tämä asia on otettava huomioon heti jatkokehityksen alkuvaiheessa.

7.6 Testauksen tulokset

Ryhmän jäsenten suorittamien sovellusten testausten tulokset osoittivat, että toteutetut sovellukset toimivat asetettujen pakollisten ja tärkeiden toiminnallisten vaatimusten mukaisesti. Usein toistuvia virheitä ei löydetty testausten aikana. Muita sovellusten tunnettuja puutteita ja virheitä kuvataan tarkemmin luvuissa 8.2 – 8.4.

8 Tavoitteiden toteutuminen

Projektissa saatiin toteutettua kaikki 47 pakollista tai tärkeää vaatimusta ja 85:stä mahdollisesta tai jatkokehitykseen kirjatusta toiminnallisesta vaatimuksesta 28. Luvussa kuvataan projektin vaatimusten toteutuminen, tunnetut virheet, puutteelliset toteutusratkaisut sekä jatkokehitysideat.

Tilaajan kanssa sovittiin osia toteutuksesta jo aikaisessa vaiheessa jatkokehitykseen projektin rajallisen ajan vuoksi. Jatkokehitykseen sovittiin tutkijan käyttöliittymästä mediatiedostojen hallintanäkymä, suoritedatan raporttinäkymä, viestien lähetys- ja lukunäkymä sekä ohjenäkymä. Kuntoutettavan käyttöliittymästä jatkokehitykseen sovittiin kiihtyvyysanturidatan lukeminen, päiväkirjamerkintöjen luku- ja tallennusnäkymä sekä viestien lähetys- ja lukunäkymä.

8.1 Vaatimusten toteutuminen

Luvussa kuvataan pääpiirteittäin sovellukselle asetettujen vaatimusten toteutuminen. Vaatimusmäärittely [8] käsittelee tarkemmin vaatimuksia ja niiden toteutumista.

Tutkijan käyttöliittymän 72:sta vaatimuksesta toteutettiin kaikki 27 pakollisiksi tai tärkeiksi määriteltyä vaatimusta. Näkymiä *Oma sivu*, *Media*, *Raportti* ja *Ohje* ei toteutettu projektissa, ja ne sovittiin tilaajan kanssa jatkokehitykseen.

Kuntoutettavan käyttöliittymän 42:sta vaatimuksesta toteutettiin kaikki 12 pakollisiksi tai tärkeiksi määriteltyä vaatimusta. Vaatimukset yksinkertaisten kuvakkeiden lisäämisestä näkymiin vieviin painikkeisiin ja tapahtuman tyyppien ilmaisuun sekä näkymiin *Päiväkirja*, *Viestit* ja *Ohje* liittyvät vaatimukset sovittiin jatkokehitykseen.

Tapahtuman suorittamisen kahdeksasta vaatimuksesta toteutettiin kaikki kolme pakollisiksi tai tärkeiksi määriteltyä vaatimusta. Kuntoutettava voi suorittaa tapahtumia ja keskeyttää suoritettavan tapahtuman.

Sisäänkirjautumiseen liittyvistä mahdollisista vaatimuksista toteutettiin kaksi projektissa. Sisäänkirjautuminen tallentaa kirjautumistiedot keksiin, jolloin esimerkiksi kuntoutettavan tarvitsee kirjautua sovellukseen vain yhden kerran, ellei hän poista keksitietoja oman mobiililaitteen selaimesta. Kirjautumistietojen salaamiseen liittyvä vaatimus sovittiin jatkokehitykseen.

Tapahtumien tehtäväkomponentteihin liittyvistä 17:sta vaatimuksesta toteutettiin kaikki seitsemän pakollisiksi tai tärkeiksi määriteltyä vaatimusta. Media-, sensori-, syöte-, liukusäädin- ja hälytystehtäväkomponentteja ei toteutettu projektissa, joten ne sovittiin jatkokehitykseen.

Suoritedatan tallentamisesta eri tiedostomuodoissa projektiryhmä toteutti vain xls-muotoiseen dataan liittyvän vaatimuksen. Muissa tiedostomuodoissa vietyä dataa ei ehditty projektin aikana edes harkitsemaan.

Kiihtyvyysanturidatan ja muiden sensoridatojen keräämiseen asetettuja vaatimuksia ei ehditty toteuttaa projektin aikana. Kiireisen aikataulun takia projektissa ei toteutettu kuntoutettavan käyttöliittymää hyödyntämään PhoneGap-rajapintaa kiihtyvyysanturidatan keräämiseen. Kyseiset vaatimukset sovittiin tilaajan kanssa jatkokehitykseen.

Viesteihin liittyviä vaatimuksia ei ehditty toteuttaa projektissa. Ryhmäkohtaisilla viesteillä tutkija voisi tiedottaa haluamiaan ryhmiä joistain tarpeellisista asioista. Viestien tarkoitus on luoda eri käyttäjäkuntien välille vuorovaikutus- ja palautekanavia.

Projektissa toteutettuja vaatimuksia ei kirjattu testatuiksi tai hyväksytyiksi, koska suunnitelmallista ja hallittua järjestelmätestausta ei ehditty suorittaa projektissa.

8.2 Tutkijan käyttöliittymän tunnetut puutteet ja virheet

Järjestelmän käyttäjien roolit määräytyvät käyttäjän ryhmien jäsenyyksien mukaan. Tämä tarkoittaa sitä, että yksi käyttäjä voi olla eri rooleissa eri ryhmissä. Järjestelmän rooleja on ajateltu käytettävän sovellusten käyttöoikeuksien jakamiseen. Esimerkiksi tutkijalla voisi olla täydet käyttöoikeudet tutkijan sovelluksessa, sekä omaisella tai opiskelijalla voisi olla rajoitetut tutkijan oikeudet sovelluksen käyttöön. Käyttöoikeuksien määräämisessä voi syntyä ongelmia silloin, kun yksi käyttäjä on jäsenenä useassa ryhmässä eri rooleilla. Ongelman voisi ratkaista asettamalla käyttäjäroolit `USER`- ja `USERGROUP`-taulujen välisen `BELONGS`-taulun kautta.

Näkymien hakukomponentin *Hae*-painikkeella pystyy myös valitsemaan valitun kohteen tiedot tarkasteltavaksi *Valitse*-painikkeen tapaan. Hakukomponentin *Hae*-painikkeella olisi vain tarkoitus hakea hakukenttään syötetyn tekstin perusteella tietoa tietokannasta.

Tutkija pystyy tällä hetkellä hakemaan käyttäjiä ainoastaan käyttäjän käyttäjätunnuksen mukaan. Tutkijan olisi halutessaan myös voitava hakea tietokannasta käyttäjiä etunimen tai sukunimen mukaan.

Tutkijalta ei kysytä varmennusta hänen tallentaessa muutoksia tai poistaessa valittua kohdetta. Tällä voisi estää tahattomat poistot ja muutosten tallennukset. Vaihtoehtoisesti tutkija voisi asetuksistaan ottaa pois päältä varmennuksen kysymisen, jos jatkuva varmentaminen haittaa käyttöliittymän käytettävyyttä.

Käyttäjien salasanoihin liittyy pari puutteellista ratkaisua. **Käyttäjien salasanoja ei tallennetta salatussa muodossa tietokantaan**, ja tutkijan käyttöliittymässä **tutkija näkee tarkasteltavan käyttäjän salasanan *Käyttäjän tiedot* -alivälilehdellä**. Kyseiset puutteelliset toteutusratkaisut tulisi ottaa käsittelyyn projektin jatkokehityksen alkuvaiheessa.

Tapahtumamuokkaimen alueen piirto tapahtuu palvelimella, joten palvelin lähettää aina uudelleenpiirretyn kuvan asiakaspäätteelle. Tapahtumanmuokkaimen alueen piirto voisi jatkokehityksessä toteuttaa tapahtumaan asiakaspäässä, jos se on mahdollista. Ongelmia voi syntyä tapahtumamuokkaimen käytettävyyteen liittyen silloin, jos tapahtumamuokkainta muutetaan niin, että piirtoalueesta lähetettävien kuvien koko kasvaa suureksi.

8.3 Kuntoutusohjelmat-näkymän tunnetut puutteet ja virheet

Kuntoutusohjelmien aikatauluttamisnäkymässä **tutkijan ei ole mahdollista valita kaikkia tietokannan mahdollistavia ajoitukseen liittyviä ominaisuuksia**. Tällaisia ovat mm. absoluuttisen ajan asettaminen ja aikaväli, jolloin tapahtuman voi suorittaa.

Kuntoutusohjelmien **aikataulutukseen käytettävä ajanasetuskomponentti ilmoittaa virheestä**, jos tutkija syöttää syöteriville virheellisen aikatiedon. Ongelmaa ei tule, jos tutkija asettaa ajan vieressä olevan painikkeen painalluksen kautta avautuvasta päivämäärän ja ajan asetusikkunasta.

Kuntoutusohjelman ajastetut tapahtumat eivät ota huomioon siirtoja kesä- tai talviaikaan. Näin ollen ajastetut tapahtumat tulevat juuri siihen aikaan, kun ne on kuntoutusohjelmaan asetettu. Ongelmia syntyy, jos kuntoutusohjelma on ajoitettu kesä- ja talviajalle tai sitä suoritetaan sekä kesä- että talviaikana.

Uuden kuntoutusohjelman tallentaminen luo tapahtumista uudet kopiot tietokantaan suoritedatan tunnistamista varten. Näin käy myös, jos tutkija muokkaa ja tallentaa kussakin välissä jo olemassa olevan kuntoutusohjelman. Jos esimerkiksi tutkija luo kymmenen tapahtumaa sisältävän kuntoutusohjelman, yhdellä tapahtumalla on kymmenen monivalintatehtävää ja jokaisella niistä on kymmenen valintaa, tulee uuden kuntoutusohjelman tallennuksessa luotua kokonaisuudessaan 1001 uutta taulualkiota tietokantaan. Kyseinen kopioiminen voi aiheuttaa ongelmia jo pelkästään yhden käyttäjän toimesta.

Kuntoutusohjelmaa tallentaessa sovellus ilmoittaa virheestä, jos kuntoutusohjelmaan ei ole lisätty tapahtumia.

8.4 Kuntoutettavan käyttöliittymän tunnetut puutteet ja virheet

Kuntoutettavan sovelluksen tapahtuman suorittamisessa on pieni bugi. Jos tapahtumassa on monivalintatehtävä, voi kuntoutettava vahingossa pystyä poistumaan tapahtumasta kesken sen suorituksen. Kyseinen ilmiö tapahtuu, jos kuntoutettava valitsee monivalinnasta vaihtoehdon, palaa takaisin kyseiseen monivalintatehtävään ja painaa oikean alareunan painiketta ilman, että on valinnut valintavaihtoehtoa.

Kiihtyvyysanturidatan keräämiseen käytettävälle PhoneGap-rajapinnalle on mahdollisesti tulossa tuki Vaadin-sovelluskehitysalustaan tulevaisuudessa. Jos kyseinen laajennus tulee saataville lähitulevaisuudessa, niin jatkokehittäjien kannattaa odottaa kyseistä laajennusta ennen kiihtyvyysanturiin liittyvien toiminnallisten vaatimusten toteuttamista.

8.5 Yleiset jatkokehitysideat

Lokin kirjauksen voisi asettaa tapahtumaan sovelluksia käytettäessä. Loki helpottaisi järjestelmän virheiden havaitsemista ja kyseiset tiedostot voi helposti arkistoida muualle.

Projektin vaatimusmäärittelyssä [8] useista **vaatimuksista puuttuu maininta milloin poistaminen on mahdollista.** Kyseiset vaatimukset tulee tarkentaa jatkokehityksessä.

8.6 Tutkijan käyttöliittymän jatkokehitysideat

Tutkijan käyttöliittymän yläreunaan päävälilehden yläpuolelle tulisi asettaa kapea alue, jossa näkyisi **sovelluksen nimi** ja jossa olisi oikeassa yläreunassa **uloskirjautumispainike**. Tällöin sovelluksesta uloskirjautuminen voisi tapahtua käyttöliittymän kaikkien välilehtien kautta.

***Oma sivu* -näkymä tulisi vaihtaa *Viestit*-näkymäksi**. Siinä tutkija voisi tarkastella kuntoutettavilta tulleita viestejä ja hälytyksiä sekä lähettää ryhmille ryhmäkohtaisia viestejä. Kyseisen näkymän tulisi myös olla päänäkymä, kun tutkija kirjautuu sisään tutkijan käyttöliittymään.

Alivälilehtien sisälle sijoitetut ***Tallenna*- ja *Poista*-painikkeet voisivat sijaita alivälilehtien ulkopuolella** alhaalla, jolloin tallentamisen ja poistamisen voisi tehdä riippumatta valittuna olevasta alivälilehdestä.

Tutkijan tulisi pystyä **asettamaan käyttäjien tiedoista yhteyshenkilöitä käyttäjälle**. Yhteyshenkilöt määritellään, jos kuntoutettava suorittaessaan tapahtumia laukaisee hälytyskomponentin kriittisellä tavalla, jolloin yhteyshenkilöille lähetetään viesti kuntoutettavaan liittyen.

Käyttäjän syöttämiä **syötteitä syötekenttiin ei tarkasteta millään lailla**. Syötteiden tarkistuksella voitaisiin eliminoida `DROP TABLE *` -tyyppisten syötteiden meno tietokantaan.

Kaikkien toimintojen järjestystä käyttömäärien ja yleisyyden mukaan ei mietitty tarpeeksi projektin aikana. Esimerkiksi hakukomponentin toiminnot voisi järjestää yleisimmästä harvemmin käytettyihin painikkeisiin ylhäältä alas.

Tutkijan käyttöliittymä kaipaa **vihjetekstejä** useisiin paikkoihin. Sovelluksen käyttötestauksen palautteessa ilmeni, että tapahtumamuokkaimen käyttämisen oppiminen ei välttämättä ole aloittelijalle kaikista helpoin tehtävä.

Toteutettujen käyttöliittymien **ulkoasuihin** ei ehditty käyttämään tarpeeksi aikaa projektin aikana. Käyttöliittymien teema-asetuksia voi muokata CSS-tyylitiedostojen kautta. Käyttöliittymien graafista ulkoasua kannattaa suunnitella paremmin asiaan perehtyneiden.

8.7 Kuntoutusohjelmat-näkymän jatkokehitysideat

Tutkijan tulisi olla mahdollista laatia **jaksottain tapahtuvia kuntoutusohjelmia**. Tämä tarkoittaisi sitä, että tutkijan olisi mahdollisuus määrittää aikaväli, jonka mukaisesti kuntoutusohjelman tapahtumia toistetaan. Esimerkiksi tutkija voisi asettaa kuntoutusohjelman tapahtumat suoritettavaksi kerran viikossa yhden kuukauden ajan.

Tutkijan tulisi olla mahdollista laatia **kuntoutusohjelmia, jotka sisältävät linkitettyjä tapahtumia**. Tämä tarkoittaisi sitä, että tutkijan olisi mahdollisuus linkittää kaksi tapahtumaa yhteen. Kun kuntoutettava suorittaa kyseisen tapahtuman loppuun, siirtyy hän tekemään suoraan toista tapahtumaa.

8.8 Tapahtumamuokkaimen jatkokehitysideat

Tutkijoilla voisi olla **tapahtumien näkemiseen ja muokkaamiseen liittyviä erityisoikeuksia**. Esimerkiksi tapahtuman näkeminen hakulistalla ja muokkaus voisi olla rajattu vain sen luoneelle ja muokanneille sekä tapahtumaan sidottujen ryhmien tutkijoille. Tietokannan taulujen `rowStatus`-kentän toimintaa voisi laajentaa kattamaan kyseisen ominaisuuden käyttöliittymässä.

Tapahtuman muokkauksesta syntyvää **muutoshistoriaa** tulisi pystyä tarkastelemaan tapahtumaa muokattaessa. Muutoshistorian voisi sijoittaa omaan alivälilehteen, josta tutkija voisi tarkastella tapahtumaan tehtyjä muutoksia.

Tapahtumamuokkaimeen liittyy seuraavia parannusehdotuksia:

- Yhden tehtävän siirtäminen piirtoalueella tulisi olla mahdollista raahamalla tapahtumaa hiiren painike pohjassa.
- Tehtävien etenemissuunnan tehtävien välillä tulisi esittää joko nuolella tai vastaavalla havainnollistavalla visualisoinnilla.
- Piirtoalueella näytettävät tehtävätyypit tulisi visuaalisesti olla erotettavissa muun kuin värin avulla.
- Tapahtumamuokkaimessa tulisi näyttää tapahtumassa olevat tehtävät listana käyttäjälle.
- Tutkijalla tulisi lisäksi myös olla mahdollisuus tehdä uusia tapahtumatyyppejä tapahtumille.

8.9 Tietokannan jatkokehitysideat

Tietojärjestelmälle voisi asettaa oman **testitietokannan** tietokannan testaukselle ja testaustiedostoille. Näin tietokantaa voisi testata eri kannalla kuin mitä tuotantoversion tietojärjestelmä käyttäisi.

Tiedon salauksen tietokantaan voisi asettaa tapahtumaan salasanojen ohella joillekin muillekin kentille. Tällaisia kohteita voisivat olla käyttäjien etunimi, sukunimi ja käyttäjätunnus. Tällä hetkellä tietokantaan tallentuu vain salaamatonta dataa.

9 Käyttöohjeita ylläpitäjälle

Luvussa kuvataan lyhyesti, kuinka järjestelmän voi asettaa toimimaan palvelimella, kuinka tutkijan käyttöliittymään pystyy lisäämään uuden näkymän ja kuinka tapahtumamuokkaimissa käytettävien tehtäväkomponenttien lisäys tapahtuu.

9.1 Sovelluksen asennus palvelimelle

Sovellus vaatii MySQL-tietokannanhallintajärjestelmän ja TomCat-palvelimen. Sovellusta varten täytyy ensin **luoda** MySQL-tietokannanhallintajärjestelmään oma **tietokanta** ja **luoda käyttäjä**, jolla on oikeudet käyttää kyseistä kantaa. Tietokannan rakenteen voi muodostaa käyttäen tiedostoa `dokumentit/tietokanta/paatti_database_1.0.0.sql`. Muodostettu tietokanta sisältää myös yhden käyttäjän, jonka käyttäjätunnus on `admin` ja jonka oletussalasana on `admin`. Tutkijan käyttöliittymä määritetään käyttämään kyseistä tietokantaa tiedostossa `PaattiResearchApplication.java` ja kuntoutettavan käyttöliittymä tiedostossa `PaattiApplication.java`. Tietokannan hallinnassa voi apuna käyttää myPhpAdmin-apuohjelmaa.

Sovellukset voidaan lähettää **TomCat-palvelimelle** käyttäen Cargo-laajennusta. Cargo-laajennuksen asetukset täytyy konfiguroida käytettävälle palvelimelle sopiviksi pakettien `pom.xml`-tiedostoista. Itse palvelimelle lähettäminen tapahtuu valitsemalla asetuksista komento "`custom->cargoUndeployAndDeploy`". Sovellukset voi tietenkin siirtää palvelimelle myös käyttäen TomCat-palvelimen omaa hallintaohjelmaa.

9.2 Tietokokonaisuuden hallintanäkymän lisääminen

Hallintanäkymän lisääminen aloitetaan **luomalla** `views`-kansioon **uusi paketti** näkymän `java`-tiedostoille. Kansio nimetään sitä vastaavan näkymän mukaan. Paketin sisälle luodaan `java`-tiedostot näkymälle ja näkymän välilehdille.

Näkymän `java`-tiedosto tulee **periä luokasta** `ApplicationView.java`. Näkymien välilehtien tulee käyttää `ApplicationViewTab.java`-luokan rajapintaa. Luodut välilehdet lisätään näkymään `ApplicationView.java`-tiedoston kons-

truktorissa. Tiedostojen muun rakenteen suunnitteluun ja toteutukseen kannattaa ottaa mallia jonkun toteutetun näkymän lähdekoodista.

Näkymä lisätään sovellukseen `ApplicationTabSheet.java`-luokan konstruktorissa. Sinne lisätään näkymän luontirivi ja asetetaan näkymälle täysi koko. Tämän jälkeen näkymä lisätään sovelluksen välilehdeksi. Mallia tähän voi katsoa edellä mainitun tiedoston konstruktorista.

9.3 Tehtäväkomponenttien lisääminen

Uusien tehtäväkomponenttien lisääminen tapahtumamuokkaimeen edellyttää tiedostojen `NodeFactory.java`, `ContextMenu.java` ja `SettingsView.java` muokkaamista.

`NodeFactory`-luokassa oleva `enum`-muuttuja `NODE_TYPE` listaa kaikki tehtäväkomponenttityypit, jonka tyyppisiä solmuolioita tehtaalla on mahdollista luoda. Jokaista `NODE_TYPE`-arvoa vastaa `NodeFactory`-luokan sisällä sijaitseva sisäluokka, joka perii halutun poistumuspolkujen lukumäärän mukaan joko `SinglePathNode`- tai `MultiPathNode`-luokan. Uutta tehtäväkomponenttiluokkaa luotaessa kannattaa katsoa mallia olemassaolevien sisäluokkien lähdekoodeista.

Tehtävien lisääminen tapahtumapuuhun tehdään **`ContextMenu`-luokan** tarjoaman valikkotoiminnallisuuden kautta. Luokan alkupuolella alustettava `nullMenuItems`-lista sisältää tiedot niistä tehtävätyypeistä, joita puuhun on mahdollista lisätä. Tehtaaseen lisättyä uutta tehtäväkomponenttiluokkaa vastaava `NODE_TYPE`-arvo on lisättävä myös tähän listaan. Operaatiota varten kannattaa katsoa mallia luokan lähdekoodista.

`SettingsView`-luokan toiminnallisuuteen kuuluu valittua tehtäväsolmua vastaavien käyttöliittymäelementtien lisääminen asetusnäkymään. Luokan `setViewContent`-metodi saa parametrina tapahtumapuussa valitun tehtävän, jonka luokkatyyppiä `instanceOf`-operaatiolla tarkastelemalla muodostetaan tehtävätyyppiä vastaavat käyttöliittymäelementit kyseisen tehtävän sisällön muokkaamista varten. Uuden tehtäväkomponentin lisääminen pitää huomioida kyseisen metodin toteutuksessa lisäämällä vastaava vertailuoperaatio kyseessä olevalle komponentille.

10 Yhteenveto

Paatti-projekti toteutti Sovellusprojekti-kurssilla keväällä 2012 Jyväskylän yliopiston psykologian laitokselle prototyypin tietojärjestelmästä, jolla tuetaan iäkkäiden, työuupumuksesta kärsivien ja mielenterveyspotilaiden itsenäistä elämää ja kuntoutumista sekä tuetaan tutkijoiden tekemää tutkimusta. Projektissa toteutettiin prototyypit tutkijan ja kuntoutettavan sovelluksista, joissa käsiteltävä tieto tallennetaan tietokantaan. Toteutettu järjestelmäkokonaisuus toimii asetettujen päätavoitteiden mukaisesti ja antaa siten hyvän pohjan tuotantoversion kehittämiselle.

Tietojärjestelmässä tutkija luo tutkimuksiinsa liittyen kuntoutettavista ryhmiä, joille määritetään omat kuntoutusohjelmat. Kuntoutusohjelmat sisältävät aikataulutettuja tapahtumia, jotka sisältävät määritetyssä järjestyksessä suoritettavia tehtäviä. Tehtävät voivat olla kysymyksen tai ohjeen muodossa ja voivat sisältää tekstiä tai monivalinnan.

Kuntoutettavia voi siirtää ryhmästä toiseen edistymisen mukaan tutkijan toimesta. Kuntoutettava voi olla monessa ryhmässä, jolloin hän suorittaa useampaa kuntoutusohjelmaa samanaikaisesti. Tutkija pystyy seuraamaan kuntoutettavan tai ryhmän edistymistä omasta käyttöliittymästään. Tietojärjestelmän tietokantaan keräämää tietoa saa myös siirrettyä toisiin järjestelmiin xls-tiedostomuodossa.

Kuntoutettava näkee ja kuittaa suoritettavat kuntoutusohjelman tapahtumat mobiililaitteessa olevasta WWW-sovelluksesta. Tapahtumat voivat olla esimerkiksi liikuntaharjoitteita, rentoutusharjoitteita tai lääkkeiden ottamisen muistutuksia.

Paatti-projektissa toteutettu tietojärjestelmän prototyyppi ei sisällä kaikkia vaadittuja toimintakokonaisuuksia. Tilaajien kanssa jatkokehitykseen sovittiin tutkijan käyttöliittymästä mediatiedostojen lataus ja esikatselunäkymä, suoritedatan graafiseen analysointiin käytettävä raporttinäkymä, viestien lukemiseen ja lähettämiseen tarkoitettu viestinäkymä sekä ohjenäkymä. Kuntoutettavan käyttöliittymästä tilaajien kanssa jatkokehitykseen sovittiin päiväkirjamerkintöjen tallentamiseen ja lukemiseen tarkoitettu päiväkirjanäkymä sekä viestien lukuun tarkoitettu viestinäkymä.

Projektin rajallinen aikataulu ei mahdollistanut kaikkien tilaajien vaatimusten toteuttamista ja siksi toteutettu järjestelmäprototyyppi kaipaa jatkokehitystä. Toteutettu järjestelmäprototyyppi sovittiin tilaajien kanssa jatkokehitykseen, jossa kehitetään projektin aikana toteuttamatta jääneet toiminnallisuudet sekä korjataan sovelluksissa esiintyvät puutteet ja heikot toteutusratkaisut.

11 Lähteet

[1] Frankel Nicolas, ”Learning Vaadin”, Packt Publishing, 2011

[2] Kallioinen Tero, ”Havaintoja tapahtumista”, sähköpostipalaute, Jyväskylän yliopisto, psykologian laitos, lähetetty 8.5.2012.

[3] Kallioinen Tero, ”Pari havaintoa ja ideaa”, sähköpostipalaute, Jyväskylän yliopisto, psykologian laitos, lähetetty 16.5.2012.

[4] Keränen Tapio, Salminen Toni, Salokangas Jari ja Satokangas Lauri, ”Paatti-projekti, Käytettävyyspäivän muistio”, Jyväskylän yliopisto, tietotekniikan laitos, 2012.

[5] Keränen Tapio, Salminen Toni, Salokangas Jari ja Satokangas Lauri, ”Paatti-projekti, Luokkadokumentaatio”, Jyväskylän yliopisto, tietotekniikan laitos, 2012.

[6] Keränen Tapio, Salminen Toni, Salokangas Jari ja Satokangas Lauri, ”Paatti-projekti, Projektiraportti”, Jyväskylän yliopisto, tietotekniikan laitos, 2012.

[7] Keränen Tapio, Salminen Toni, Salokangas Jari ja Satokangas Lauri, ”Paatti-projekti, Projektisuunnitelma”, Jyväskylän yliopisto, tietotekniikan laitos, 2012.

[8] Keränen Tapio, Salminen Toni, Salokangas Jari ja Satokangas Lauri, ”Paatti-projekti, Vaatimusmäärittely”, Jyväskylän yliopisto, tietotekniikan laitos, 2012.

[9] Oracle, ”Code Conventions for the Java Programming Language”, saatavilla PDF-muodossa <URL: `http://www.oracle.com/technetwork/java/codeconventions-150003.pdf`>, viitattu 25.3.2012.

[10] Santanen Jukka-Pekka, ”Huomioita kuntoutettavan käyttoliittymästä”, sähköpostipalaute, Jyväskylän yliopisto, tietotekniikan laitos, lähetetty 7.5.2012.

[11] Santanen Jukka-Pekka, ”Huomioita tutkijan käyttoliittymästä”, sähköpostipalaute, Jyväskylän yliopisto, tietotekniikan laitos, lähetetty 7.5.2012.

[12] Santanen Jukka-Pekka, ”Huomioita tapahtumien hallintanäkymästä”, sähköpostipalaute, Jyväskylän yliopisto, tietotekniikan laitos, lähetetty 15.5.2012.

[13] Santanen Jukka-Pekka, ”Huomioita tapahtumien hallintanäkymästä jatkuvat”, sähköpostipalaute, Jyväskylän yliopisto, tietotekniikan laitos, lähetetty 15.5.2012.

[14] Vaadin Ltd, ”Book of Vaadin - 4th Edition”, 2012