

# **Moveatis-sovellusprojekti**

**Jarmo Juujärvi**

**Sami Kallio**

**Kai Korhonen**

**Juha Moisio**

**Ilari Paananen**



## **Sovellusraportti**

**Julkinen**

**Versio 1.0.0**

**17.8.2016**

**Jyväskylän yliopisto**

**Tietotekniikan laitos**

<b>Hyväksyjä</b>	<b>Päivämäärä</b>	<b>Allekirjoitus</b>	<b>Nimenselvennys</b>
Projektipäällikkö	__.__.20__		
Tilaaja	__.__.20__		
Ohjaaja	__.__.20__		

## Tietoja dokumentista

### Tekijät:

Jarmo Juujärvi	jarmojuujarvi@gmail.com	050-3651136
Sami Kallio	sami.m.j.kallio@student.jyu.fi	046-6418125
Kai Korhonen	kaivkorhonen@gmail.com	045-6362355
Juha Moisio	juha.pa.moisio@student.jyu.fi	040-7314776
Ilari Paananen	ilari.k.paananen@student.jyu.fi	044-9746743

**Dokumentin nimi:** Moveatis-projekti, Sovellusraportti

**Sivumäärä:** 49

**Tiedosto:** Moveatis\_Sovellusraportti.doc

**Tiivistelmä:** Moveatis-sovellusprojektiryhmä kehitti Moveatis-sovelluksen, joka korvaa käytössä olevan Lotas Observer -ohjelman Jyväskylän yliopiston liikuntakasvatuksen laitoksella. Sovellukseen on mahdollista määrittellä toimenpiteitä kategorioina, joiden esiintymistä opetustapahtumassa halutaan tarkkailla. Tapahtuman tarkkailun päätyttyä sovelluksessa on nähtävissä ja tallennettavissa yhteenveto kategorioiden kestoista ja määristä sekä kuvaaja kategorioiden ajoittumisesta. Sovellusraportissa kuvataan projektissa kehitetyn tietojärjestelmän käyttöliittymää ja kokonaisrakennetta, tavoitteiden toteutumista, tietokannan rakennetta sekä jatkokehitysideoita.

**Avainsanat:** jatkokehitys, kokonaisrakenne, komponentit, käyttöliittymä, luokkakaavio, sovellus, tavoitteet, tietojärjestelmä, tietokanta, toteutusratkaisut, ylläpito.

## Muutoshistoria

<b>Versio</b>	<b>Päivä</b>	<b>Muutokset</b>	<b>Tekijä</b>
0.1.0	2.5.2016	Sovellusraportin kirjoittaminen aloitettiin projektiraportin pohjalta.	Kai Korhonen
0.2.0	20.5.2016	Lisättiin sovelluksen toimintaa kuvaava luku.	Jarmo Juujärvi
0.3.0	3.6.2016	Sovelluksen toiminnan kuvausta täydennettiin.	Jarmo Juujärvi
0.4.0	4.7.2016	Lisättiin sovelluksen sivujen teknistä toteutusta kuvaava luku.	Jarmo Juujärvi
0.4.1	3.8.2016	Muotoiluja korjattiin ja jatkokehitysoasiota täydennettiin.	Jarmo Juujärvi
0.4.2	7.8.2016	Lisättiin jatkokehitystä ja ylläpitoa kuvaava luku.	Sami Kallio
0.5.0	8.8.2016	Täydennettiin tavoitteiden toteutumista kuvaavaa lukua.	Jarmo Juujärvi
0.6.0	13.8.2016	Kieli- ja ulkoasua korjattiin. Korjattiin termistön kaavio.	Jarmo Juujärvi
0.7.0	15.8.2016	Täydennettiin lukua 7.	Sami Kallio
1.0.0	17.8.2016	Kieli- ja ulkoasua korjattiin.	Jarmo Juujärvi

## Tietoja projektista

Moveatis-projekti kehitti Lotas Observer -ohjelman korvaavan Moveatis-tietojärjestelmän Jyväskylän yliopiston liikuntakasvatuksen laitokselle. Sovellukseen on mahdollista määrittellä kategorioina opetustapahtumassa seurattavia toimenpiteitä. Tapahtuman tarkkailun päätyttyä sovelluksessa on nähtävissä yhteenveto kategorioiden yhteenlasketuista kestoista ja määristä sekä kuvaaja kategorioiden ajoittumisesta.

### Tekijät:

Jarmo Juujärvi	jarmojuujarvi@gmail.com	050-3651136
Sami Kallio	sami.m.j.kallio@student.jyu.fi	046-6418125
Kai Korhonen	kai.v.korhonen@gmail.com	045-6362355
Juha Moisio	juha.pa.moisio@student.jyu.fi	040-7314776
Ilari Paananen	ilari.k.paananen@student.jyu.fi	044-9746743

### Tilaajan edustajat:

Pilvikki Heikinaro-Johansson	pilvikki.heikinaro-johansson@jyu.fi	040-7243352
Mirja Hirvensalo	mirja.hirvensalo@jyu.fi	040-8053944
Nelli Lyyra	nelli.lyyra@jyu.fi	040-8427624
Sanna Palomäki	sanna.h.palomaki@jyu.fi	040-8053967

### Ohjaajat:

Mika Lehtinen	mika.k.lehtinen@student.jyu.fi	040-4160776
Jukka-Pekka Santanen	santanen@mit.jyu.fi	050-5504666

**Yhteystiedot:**

- Sähköpostilistat     `moveatis@korppi.fi,`  
                              `moveatis_opetus@korppi.fi`
- Sähköpostiarkistot   `https://korppi.jyu.fi/kotka/servlet/`  
                              `list-archive/moveatis,`  
                              `https://korppi.jyu.fi/kotka/servlet/`  
                              `list-archive/moveatis_opetus`
- WWW-sivusto         `http://sovellusprojektit.it.jyu.fi/`  
                              `moveatis/`

## Sisällys

<b>1</b>	<b>Johdanto.....</b>	<b>1</b>
<b>2</b>	<b>Termit.....</b>	<b>2</b>
2.1	Kohdealueen termejä .....	2
2.2	Sovelluksen käyttäjäroolit .....	3
2.3	Käyttöliittymän termejä .....	4
2.4	Kaavio termeistä.....	6
<b>3</b>	<b>Sovelluksen käyttöliittymä ja käyttö .....</b>	<b>7</b>
3.1	Sovelluksen käyttöliittymän rakenne ja näkymäkartta.....	7
3.2	Sovelluksen etusivu.....	9
3.3	Komentovalikon toiminnot .....	9
3.4	Kirjautumisvalinnat ja niiden vaikutus käyttöön .....	11
3.5	Tapahtumien, kategoriaryhmien ja kategorioiden hallintasivu .....	12
3.6	Observoinnissa käytettävien kategorioiden valinta .....	17
3.7	Observoinnin suorittaminen.....	20
3.8	Observoinnin yhteenvetosivu .....	22
3.9	Observoinnin kirjausten ja yhteenvedon tallennus .....	24
<b>4</b>	<b>Sovelluksen toteutusratkaisut .....</b>	<b>26</b>
4.1	Kehitystyökalut.....	26
4.2	Sivujen toiminnallisuus .....	27
4.3	Toimintalogiikka .....	28
4.4	Sovelluspalvelin .....	29

<b>5</b>	<b>Sovelluksen tietokannan rakenne .....</b>	<b>30</b>
5.1	Käyttäjät ja roolit .....	30
5.2	Kategoriat, tapahtumat ja observoinnit .....	32
5.3	Taulujen käyttötarkoitukset.....	35
<b>6</b>	<b>Sovelluksen tavoitteiden toteutuminen .....</b>	<b>37</b>
6.1	Toteutetut ominaisuudet .....	37
6.2	Jatkokehitykseen sovitut ominaisuudet.....	39
<b>7</b>	<b>Ohjeita sovelluksen ylläpitoon ja jatkokehitykseen.....</b>	<b>41</b>
7.1	Sovelluksen asennus- ja käyttöönotto-ohjeet .....	41
7.2	Ohjeita jatkokehittäjälle .....	44
7.3	Ohjeita ylläpitäjälle .....	45
<b>8</b>	<b>Yhteenveto .....</b>	<b>47</b>
	<b>Lähteet.....</b>	<b>48</b>



# 1 Johdanto

Jyväskylän yliopiston liikuntakasvatuksen laitoksella järjestetään vuosittain kaikille liikunnanopettajaopiskelijoille yhteinen liikuntapedagogiikan peruskurssi nimeltä *Tutkiva opettaja*. Tärkeä osa kurssia ja liikunnanopettajaopintoja on opetustapahtuman tarkkailu ja opiskelijan kehittyminen opetustilanteessa. Moveatis-projekti kehitti opetustapahtuman tarkkailuun WWW-sovelluksen, joka perustuu kerta- ja kestorekisteröintiin. Sovellukseen on mahdollista määritellä opetustapahtumassa seurattavia toimenpiteitä kategorioina. Tapahtuman tarkkailun päätyttyä sovelluksessa on nähtävissä ja tallennettavissa yhteenveto kategorioiden kestoista ja määristä sekä kuvaaja kategorioiden ajoittumisesta. Sovellusta tullaan käyttämään aluksi pääosin edellä mainitulla kurssilla, mutta jatkossa sitä pyritään hyödyntämään muiden kurssien lisäksi myös pro gradu -tutkielmissa ja tutkimuskäytössä.

Sovellusraportti kuvaa toteutetun sovelluksen toteutusratkaisuja ja käyttöliittymää, asetettujen tavoitteiden toteutumista sekä olennaisimpia puutteita, virheitä ja jatkokehitysideoita. Sovellusraportin laatimisessa on hyödynnetty Kepler-projektin sovellusraporttia [1] sekä Moveatis-projektin projektiraporttia [2], luokkadokumentaatioita [3] ja [4], vaatimusmäärittelyä [5], tietokantakuvausta [6], käytettävyytestausdokumenteja [7] ja [8] sekä järjestelmätestausraportteja [9], [10] ja [11].

Sovellusraportti muodostuu kahdeksasta luvusta. Luvussa 2 esitellään sovelluksessa ja dokumenteissa käytettäviä termejä. Luvussa 3 kuvataan sovelluksen käyttöliittymää ja käyttöä. Luvussa 4 esitellään sovelluksen kokonaisrakennetta ja toteutusratkaisuja. Luku 5 kuvaa tietokannan rakennetta. Luvussa 6 verrataan sovelluksen vastaavuutta vaatimusmäärittelyyn ja sovellussuunnitelmiin sekä kuvataan jatkokehitykseen sovitut ominaisuudet. Luvussa 7 on ohjeita ylläpitäjille ja jatkokehittäjille.

## 2 Termit

Luvussa kuvataan Moveatis-projektiin liittyviä termejä, jotka on jaoteltu erillisiin alalukuihin kohdealueen, käyttäjäroolien ja käyttöliittymän mukaan.

### 2.1 Kohdealueen termejä

<b>Tapahtumaryhmä</b> (event group)	on observeitavia vuorovaikutustilanteita yhdistävä kokonaisuus. Esimerkiksi kurssin ”Tutkiva opettaja” jokainen opetustapahtuma on sovelluksessa yksi <i>tapahtuma</i> . <i>Tapahtumaryhmälle</i> määritetään mm. <i>kategoriaryhmät</i> ja <i>ryhmäavain</i> .
<b>Tapahtuma</b> (event)	on yksi sovellukseen luotava vuorovaikutustilanne. <i>Tapahtuma</i> kuuluu yhteen <i>tapahtumaryhmään</i> ja on <i>observoinnin</i> kohteena oleva tilanne. <i>Ryhmäavain</i> ja <i>kategoriaryhmät</i> haetaan <i>tapahtumaryhmästä</i> , johon <i>tapahtuma</i> kuuluu.
<b>Ryhmäavain</b> (group key)	on <i>tapahtumaryhmän</i> tunniste, jonka syöttämällä sovellus hakee siihen liittyvät <i>kategoriaryhmät</i> käytettäväksi <i>observoinnissa</i> .
<b>Observointi</b> (observation)	on yksittäisen <i>tapahtuman</i> tarkkailukerta, jonka aikana tehdyistä <i>kirjauksista</i> muodostetaan <i>observointidata</i> .
<b>Observointidata</b> (observation data)	on yhdestä <i>observoinnista kirjauksen</i> avulla kerätty tieto. <i>Observointidata</i> voidaan lähettää sähköpostiin, tallentaa laitteelle ja tallentaa sovelluksen tietokantaan.
<b>Observointitulokset</b> (observation results)	on <i>observointidatasta</i> muodostettava yhteenveto, joka esitetään yhteenvetosivulla.

<b>Kirjaus (recording)</b>	on yksi <i>observoinnin</i> aikana tehty merkintä. <i>Kirjauksella</i> on ajankohta ja <i>kategoria</i> , sekä siihen voi liittää ääni- tai tekstikommentteja.
<b>Kategoriaryhmä (category set)</b>	on <i>kategorioista</i> koostuva ryhmä. <i>Kategoriaryhmiä</i> voi ottaa käyttöön <i>tapahtumaryhmän avaimella</i> tai <i>valikosta</i> .
<b>Kategoria (category)</b>	on yksittäisen <i>kirjauksen</i> tyyppi, kuten tehtävänselitys, ohjaus tai palautteenanto.
<b>Systemaattiset observointimenetelmät (methods of systematic observation)</b>	ovat keinoja tarkastella ja havainnoida <i>tapahtumaa</i> ennalta asetettujen <i>kategorioiden</i> mukaan.
<b>Keston rekisteröinti (duration recording)</b>	on <i>systemaattisen observoinnin</i> menetelmä, jolla mitataan <i>kategorian</i> kirjausten kestoa tapahtumassa kirjaamalla <i>kategorian</i> alkamisen ja päättymisen ajankohdat.
<b>Kertarekisteröinti (count recording)</b>	on <i>systemaattisen observoinnin</i> menetelmä, jolla kirjataan <i>tapahtuman</i> aikana ilmenevät ennalta määritellyt tilanteet ja ajankohdat. <i>Kertarekisteröinnillä</i> mitataan tiettyyn <i>kategoriaan</i> kuuluvan toiminnan määriä <i>tapahtuman</i> aikana.

## 2.2 Sovelluksen käyttäjät

<b>Julkinen käyttäjä (public user)</b>	käyttää sovellusta kirjautumatta JYU-tunnuksilla. Hän voi käyttää sovellusta antamalla <i>kategoriat</i> itse tai hyödyntämällä <i>ryhmäavaimella</i> saatavia <i>kategoriaryhmiä</i> . Hän voi tallentaa observoinnin tulokset käyttämäänsä laitteeseen.
--	---

<b>Kirjautunut käyttäjä (logged-in user)</b>	on kirjautunut sovellukseen JYU-tunnuksilla. Hän voi julkisen käyttäjän toimintojen lisäksi luoda ja hallita omistamiaan <i>tapahtumia</i> , sekä lähettää observoinnin tulokset sähköpostiin tai tallentaa ne sovelluksen tietokantaan.
<b>Pääkäyttäjä (superuser)</b>	on kirjautunut sovellukseen JYU-tunnuksilla, joille on määritetty sovelluksen <i>pääkäyttäjän</i> oikeudet. <i>Pääkäyttäjille</i> näytetään edellä mainittujen <i>kirjautuneen käyttäjän</i> toimintojen lisäksi pääkäyttäjän toiminnot, kuten pääkäyttäjäoikeuksien lisääminen ja poistaminen sekä kaikkien tapahtumaryhmien ja tapahtumien tarkastelu.
<b>Tapahtuman omistaja (event owner)</b>	on <i>käyttäjä</i> , joka on luonut kyseisen <i>tapahtuman</i> tai jolle on annettu omistajan oikeudet <i>pääkäyttäjän</i> toimesta. <i>Tapahtumaa</i> voi <i>tapahtuman omistajien</i> lisäksi hallita <i>pääkäyttäjät</i> .

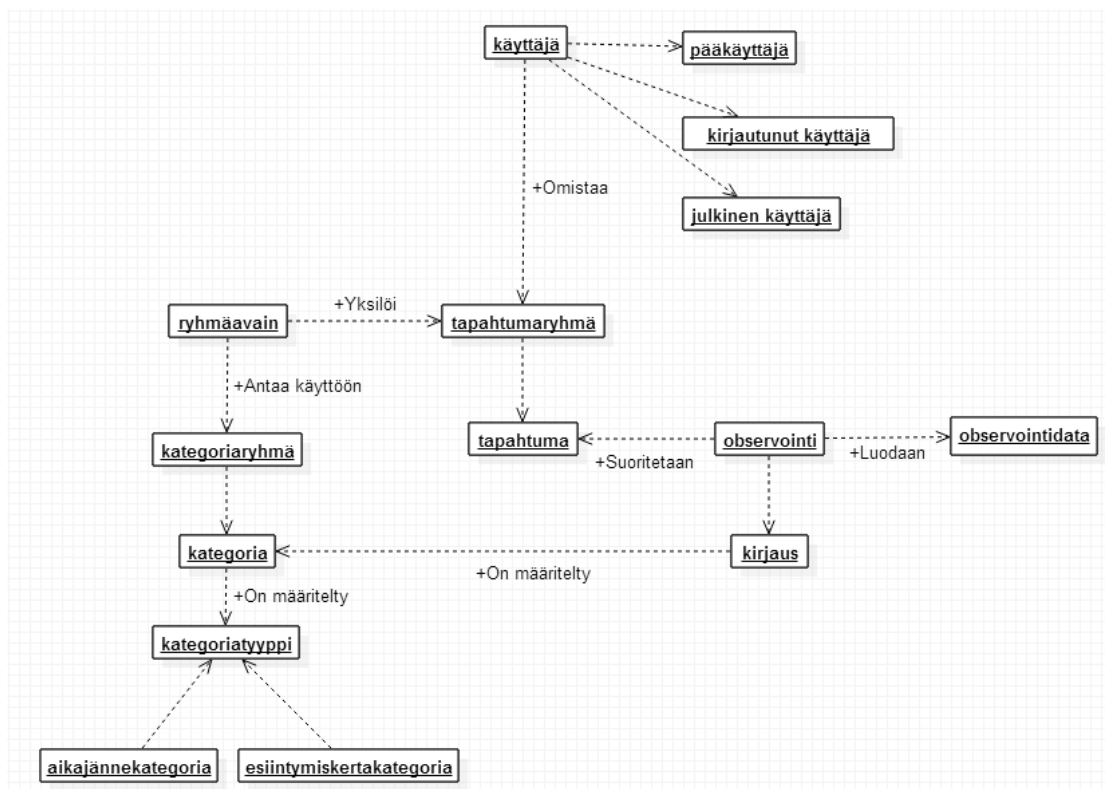
### 2.3 Käyttöliittymän termejä

<b>Etusivu (front page)</b>	on sovelluksen aloitussivu, jossa voi kirjautua sisään, syöttää <i>ryhmäavaimen</i> tai siirtyä käyttämään sovellusta kirjautumatta.
<b>Kategoriaryhmien valintasivu</b>	on sivu, jolla käyttäjä voi ottaa hallita observoinnissa käytettäviä <i>kategoriaryhmiä</i> ja <i>kategorioita</i> .
<b>Observointisivu (recording sheet)</b>	on sovelluksen sivu, jossa suoritetaan itse <i>observointi</i> . Sivun tärkein sisältö ovat painikkeet eri <i>kategorioiden kirjauksille</i> ja <i>observointikello</i> .
<b>Yhteenvetosivu (summary)</b>	on sovelluksen sivu, jossa näytetään <i>observointitulokset</i> <i>observoinnin</i> aikana tehdyistä <i>kirjauksista</i> . Sivulla esitetään <i>kategorioitten kirjausten</i> kestojen ja määrien yhteenvedo sekä <i>kirjausten</i> ajoitusta kuvaava janakaavio.
<b>Hallintasivu (control page)</b>	on sivu, jolla <i>kirjautunut käyttäjä</i> voi muun muassa luoda ja hallita omia <i>tapahtumiaan</i> , sekä selata tallentamia <i>observointidatoja</i> .

<b>Valikko (menu)</b>	on sovelluksen oikeassa ylä laidassa sijaitseva pudotusvalikko, joka sisältää sovelluksen asetuksia.
<b>Kategoriatyyppi (category type)</b>	on <i>kategorialle</i> määritelty tyyppi kuvaten, mitaako <i>kategoria</i> aikaa vai esiintymiskertoja.
<b>Aikajännekatgoria (duration category)</b>	on <i>kategoriatyyppi</i> , joka mittaa kategorian kirjausten aikaa. Kohdealueen vastaava käsite on <i>keston rekisteröinti</i> .
<b>Esiintymiskertakategoria (count category)</b>	on <i>kategoriatyyppi</i> , joka mittaa kategorian kirjausten esiintymiskertoja. Kohdealueen vastaava käsite on <i>kertarekisteröinti</i> .
<b>Raportti (report)</b>	on tekstiraportti suoritetusta observoinnista sisältäen tiedot <i>observoinnin</i> aikana tehdyistä <i>kirjauksista</i> .
<b>Janakaavio</b>	on <i>observoinnin kirjauksista</i> muodostettu visuaalinen kuvaaja.
<b>Käynnistyspainike (start)</b>	on painike, joka käynnistää <i>observointisivulla</i> sijaitsevan <i>observointikellon</i> .
<b>Taukopainike (pause)</b>	on painike, joka pysäyttää <i>observointisivulla observointikellon</i> . Tämän jälkeen observointia voidaan vielä jatkaa.
<b>Lopetuspainike (stop)</b>	lopettaa <i>observointisivulla observoinnin</i> .
<b>Observointikello (observation clock)</b>	on kello, joka käynnistetään <i>observointi</i> aloitettaessa. Kellon voi pysäyttää ja käynnistää uudelleen <i>observoinnin</i> aikana.
<b>Kommentti (comment)</b>	on <i>observointiin</i> lisättävä ääni- tai tekstikommentti.

## 2.4 Kaavio termeistä

Kuvassa 1 esitetään projektin olennaisimmat käsitteet ja niiden yhteydet kaaviona. Kaaviota luetaan nuolen suuntaisesti. Esimerkiksi käyttäjän ja tapahtumaryhmän välinen yhteys ”omistaa” tarkoittaa, että ”käyttäjä omistaa tapahtumaryhmän”. Yhteydet ilman kuvausta tarkoittavat koostumista. Esimerkiksi käyttäjät koostuvat pääkäyttäjistä, kirjautuneista käyttäjistä ja julkisista käyttäjistä.



Kuva 1. Kohdealueen käsitteiden yhteydet.

### 3 Sovelluksen käyttöliittymä ja käyttö

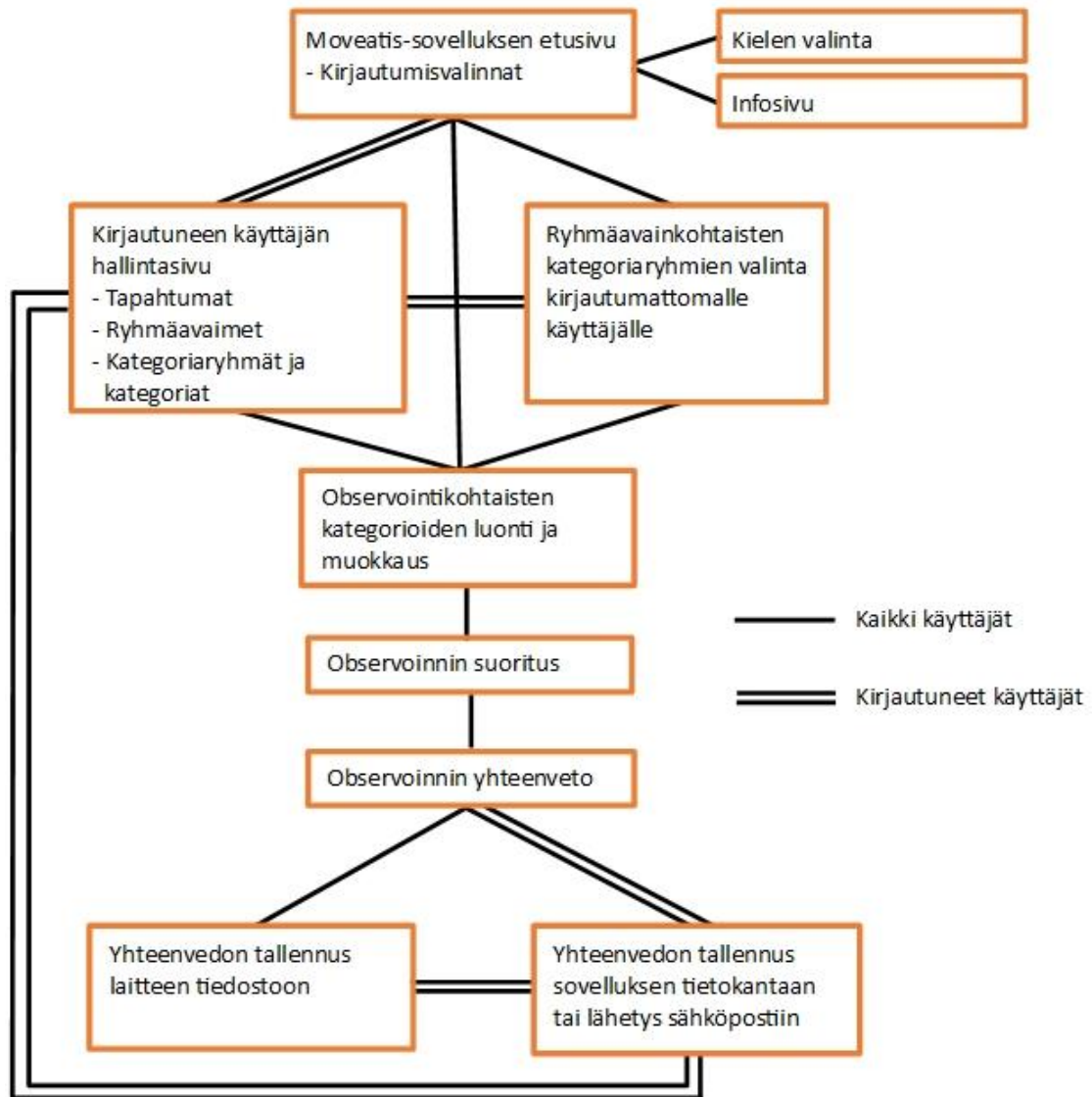
Luvussa esitellään sovelluksen käyttöliittymä ja sovelluksen käyttöä. Sovellus on toteutettu sekä suomeksi että englanniksi sisällöltään identtisenä. Raportissa käytetään esimerkkinä suomenkielistä kieliversiota.

#### 3.1 Sovelluksen käyttöliittymän rakenne ja näkymäkartta

Sovelluksen käyttöliittymän sivut ovat kolmiosaisia (katso kuva 3). Ylinä olevasta yläotsakkeesta löytyvät valikkopainikkeet. Alinna olevassa alaotsakkeessa sijaitsee tekijänoikeuslauseke, joka toimii myös linkkinä infisivulle. Varsinainen sovellussisältö on keskellä.

Työpöytänäkymässä etusivun navigointipainikkeet ovat vasemmalla ja teksti oikealla. Mobiilinäkymässä navigointipainikkeet ovat ylhäällä ja teksti alhaalla. Sovelluksen toiminnallisuudet ovat oleellisesti samat päätelaitteesta riippumatta. Raportissa käsitellään työpöytänäkymää.

Kuvassa 2 on kuvattu Moveatis-sovelluksen näkymien väliset suhteet. JYU-tunnuksilla kirjautuneen käyttäjän sekä kirjautumattoman käyttäjän käytettävissä olevat näkymät poikkeavat toisistaan, ja ne on esitetty kuvassa erilaisilla viivoilla. Osasta sivuja avautuu dialogeja, joissa on mahdollista muokata tietoja. Kuvaajan selkeyttämiseksi nämä on katsottu osaksi ko. sivua.



Kuva 2. Moveatis-sovelluksen näkymien riippuvuussuhteet.



### 3.2 Sovelluksen etusivu

Etusivu koostuu kirjautumisvalintapainikkeista sekä tervetulo- ja pikaohjetekstistä kuvan 3 mukaisesti. Kirjautumispainikkeilla valitaan sovelluksen käyttötapa, joita ovat ryhmäavain, kirjautuminen Jyväskylän yliopiston käyttäjätunnuksilla ja julkinen käyttö. Sovelluksen käyttötavat on kuvattu tarkemmin luvussa 3.4.



Kuva 3. Etusivu.

### 3.3 Komentovalikon toiminnot

Yläotsakkeen muodostavat värillinen vaakapalkki, sovelluksen nimi ja komentovalikko (katso kuva 4). Sovelluksen nimi vasemmassa laidassa on myös linkki etusivulle.

Oikeaan laitaan sijoitetun komentovalikon sisältö muuttuu riippuen siitä, onko käyttäjä kirjautunut palveluun JYU-tunnuksilla vai ei.



Kuva 4. Yläpalkki ja komentovalikko kirjautuneen käyttäjän tapauksessa.

Kirjautuneen käyttäjän komentovalikon kautta (katso kuva 5) käyttäjä voi siirtyä hallintasivulle tai kirjautua ulos sovelluksesta. Jos käyttäjä ei ole kirjautunut sovellukseen, kirjautuneen käyttäjän valikko ei ole näkyvissä. Kirjautuneen käyttäjän nimi näkyy myös osana komentovalikkoa.



Kuva 5. Kirjautuneen käyttäjän valikko.

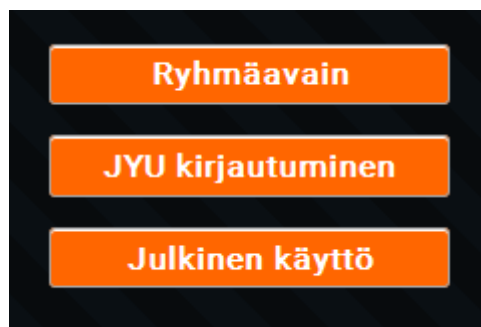
Asetusvalikosta (katso kuva 6) käyttäjä voi siirtyä käyttäjän oppaaseen (info) tai etusivulle. Myös kielen vaihto suomen ja englannin välillä on sijoitettu asetusvalikkoon. Kielivalinnasta on näkyvissä vain se kieli, johon sovelluksessa on kulloinkin mahdollista vaihtaa.



Kuva 6. Asetusvalikko.

### 3.4 Kirjautumisvalinnat ja niiden vaikutus käyttöön

Etusivun (katso kuva 3) kirjautumisvalikossa on kuvan 7 mukaisesti valittavissa Moveatis-sovellukselle kolme eri käyttötapaa: ryhmäavain, JYU-kirjautuminen ja julkinen käyttö.



Kuva 7. Kirjautumisvalikko.

**Ryhmäavaimella kirjautuessaan** käyttäjä saa käyttöönsä tapahtuman luojaan ennalta määrittelemät kategoriaryhmät ja kategoriat. Käyttäjä voi valita näistä kategorioista haluamansa ja myös tarvittaessa lisätä omia kategorioita. Ryhmäavaimella kirjautuva

käyttäjä voi tallentaa observointitulokset paikallisesti omalle päätelaitteelleen yhteenvetosivulla.

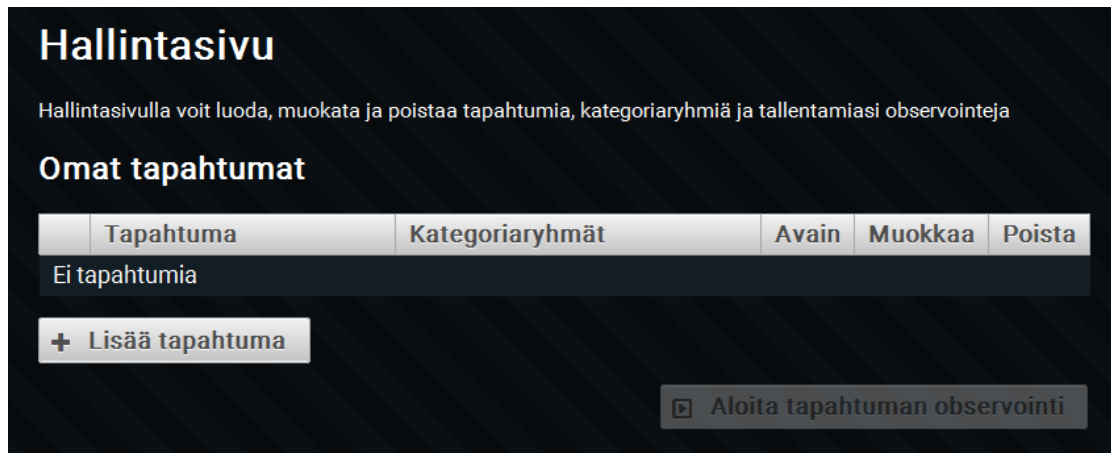
**JYU-kirjautuminen** tapahtuu Jyväskylän yliopiston käyttäjätunnuksilla antaen sovelluksen käyttäjälle oikeuden luoda tapahtumia, kategoriaryhmiä, kategorioita ja ryhmäavaimia. Kirjautunut käyttäjä voi paikallisen tallennuksen lisäksi lähettää observoinnin tulokset syöttämäänsä sähköpostiosoitteeseen tai tallentaa ne sovelluksen tietokantaan. Tietokantaan tallennetut observointitulokset ovat myös myöhemmin katseltavissa. Käyttäjä myös näkee kaikki luomallaan ryhmäavaimella suoritettut observoinnit riippumatta siitä, kuka ne on suorittanut.

**Julkinen käyttö** on kenen tahansa valittavissa oleva käyttömuoto. Julkisessa käytössä ei ole oletuskategorioita, vaan käyttäjän on luotava omat kategoriaryhmät ja kategoriat jokaista sovelluksen käyttökertaa varten. Julkinen käyttäjä voi tallentaa suorittamansa observoinnin tiedot ainoastaan paikallisesti omalle päätelaitteelleen.

### 3.5 Tapahtumien, kategoriaryhmien ja kategorioiden hallintasivu

Kirjaututtuaan sovellukseen JYU-tunnuksilla käyttäjä ohjataan automaattisesti kuvissa 8 ja 9 esitetyille hallintasisivulle. Hallintasisivulla käyttäjä voi luoda, muokata ja poistaa tapahtumia, kategoriaryhmiä ja kategorioita.

Tapahtuman lisääminen tapahtuu *Lisää tapahtuma* -painikkeella. Käyttäjän tulee antaa tapahtumalle yksilöllinen nimi ja vahvistaa toiminto *Lisää*-painikkeella. Jotta tapahtumaan voisi tehdä observointeja, on luotava myös vähintään yksi kategoriaryhmä, jossa on vähintään yksi tarkkailtava kategoria.



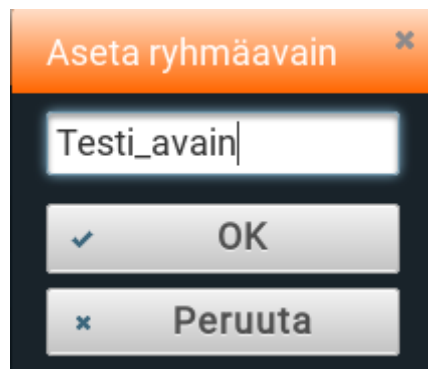
Kuva 8. Hallintasivu ilman olemassa olevia tapahtumia.



Kuva 9. Hallintasivu, jossa on listattuna luotuja tapahtumia.

**Tapahtumataulukossa** (katso kuva 9) kategoriaryhmän nimen edessä olevaa nuolta painamalla käyttäjä saa näkyviin tapahtumalle jo suoritettut observoinnit, mikäli ne on tallennettu sovelluksen tietokantaan. Mikäli hallintasivulla on jo olemassa tapahtumia kuvan 9 mukaisesti, voi käyttäjä valita jonkin olemassa olevista tapahtumista ja halutessaan lisätä sille uusia kategoriaryhmiä taulukon lisäyispainikkeista. Tapahtuman ollessa aktiivisena, voi käyttäjä siirtyä uuteen observointiin *Aloita tapahtuman observointi* -painiketta painamalla.

Rekisteröitynyt käyttäjä voi hallintasivulla myös asettaa tapahtumalle **ryhmäavaimen** harmaasta avainkuvakkeesta (katso kuva 9) avautuvasta dialogista (katso kuva 10). Jos tapahtumalle on jo asetettu ryhmäavain, tapahtuman avainkuvake näkyy keltaisena (katso kuva 9). Ryhmäavainta käyttämällä voi kirjautumatonkin käyttäjä saada käyttöönsä tapahtumaan ennalta määritellyt kategoriat. Kirjautuneen käyttäjän suorittamat observoinnit ovat nähtävissä myös ryhmäavaimen luojaan hallintanäkymässä, jos observoija käyttää ryhmäavainta ja tallentaa observointidatan sovelluksen tietokantaan.




Kuva 10. Ryhmäavaimen syöttöikkuna.

Mikäli kirjautunut käyttäjä siirtyy observointiin suoraan etusivulta ryhmäavaimella tai julkisena käyttäjänä haluten silti tallentaa observointitiedot, tulevat ne näkyviin hallintasivulle omien tapahtumien alle otsikolla *Muut suoritettut observoinnit* kuvan 11 mukaisesti.

## Hallintasivu

Hallintasivulla voit luoda, muokata ja poistaa tapahtumia, kategoriaryhmiä ja tallentamiasi observointeja

### Omat tapahtumat

	Tapahtuma	Kategoriaryhmät	Avain	Muokkaa	Poista
	Testitapahtuma 1	 Lisää			

 Lisää tapahtuma

 Aloita tapahtuman observointi

### Muut suoritettut observoinnit

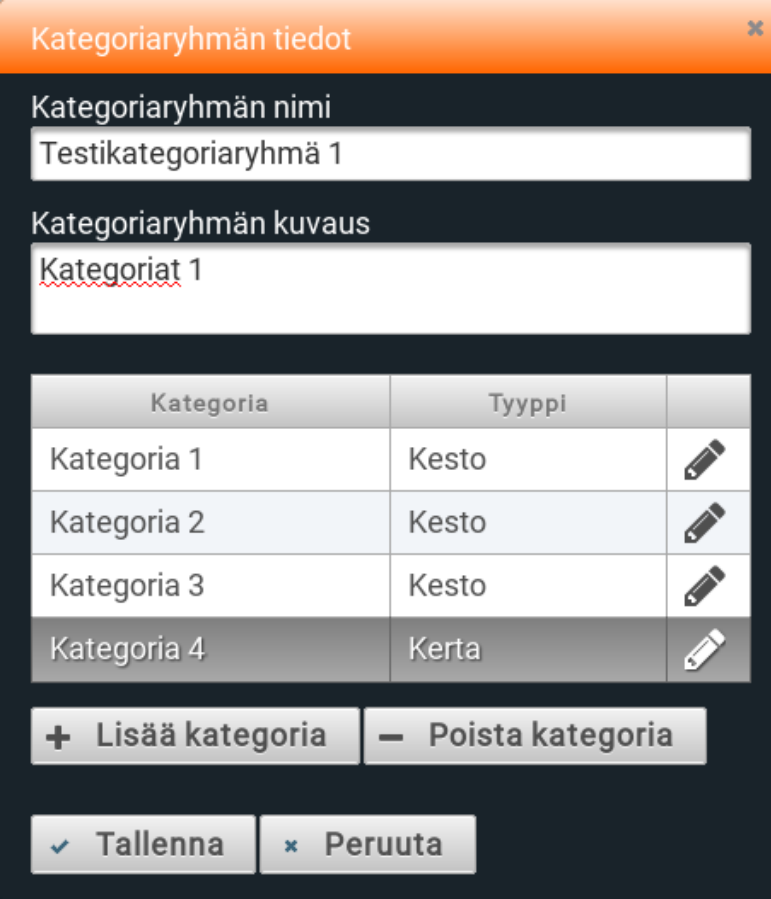
 Näytä/Piilota

Nimi	Tapahtuma
TestiObservointi 3	

Kuva 11. Tapahtumiin liittymättömien tallennettujen observointien näkymä.

Tapahtumataulukossa listatun tapahtuman tietoja voi muokata kynäkuvakkeesta aukeavassa dialogissa, joka on sisällöltään tapahtuman luontidialogin mukainen. Muokkaustilassa tapahtuman voi poistaa kokonaan roskakorikuvakkeella.





Kategoriaryhmäsarakkeen *Lisää*-painikkeesta (katso kuva 11) avautuvassa dialogissa (katso kuva 12) luodaan tapahtumakohtaiset kategoriaryhmät ja niiden kategoriat. Kategoriaryhmän kuvaus ei ole pakollinen tieto. Kategorioita lisätään *Lisää kategoria*-painikkeella ja poistetaan *Poista kategoria*-painikkeella. Kategorian muokkaaminen tapahtuu kynä-kuvakkeesta.



Kategoriaryhmän tiedot

Kategoriaryhmän nimi  
Testikategoriaryhmä 1

Kategoriaryhmän kuvaus  
Kategoriat 1

Kategoria	Tyyppi	
Kategoria 1	Kesto	
Kategoria 2	Kesto	
Kategoria 3	Kesto	
Kategoria 4	Kerta	

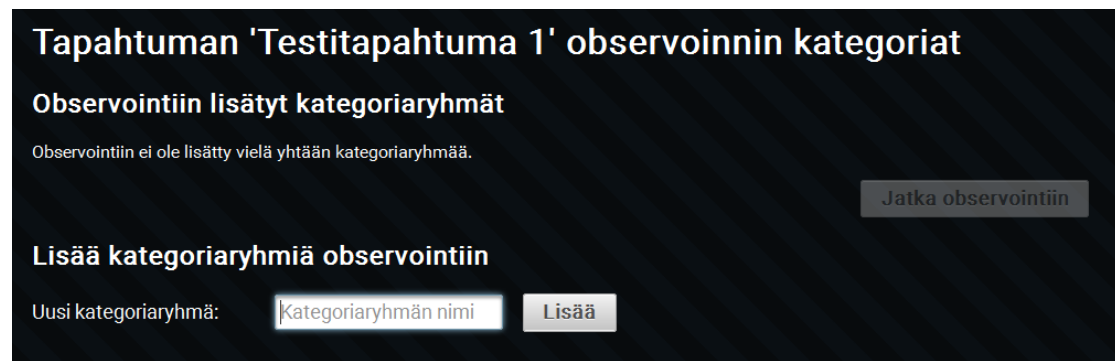
+ Lisää kategoria    - Poista kategoria

✓ Tallenna    ✕ Peruuta

Kuva 12. Kategoriaryhmien ja kategorioiden luontidialogi.

Mikäli tapahtumaan ei hallintasivulla liitetä kategoriaryhmää (katso kuva 8), ohjataan käyttäjä ennen observointiin siirtymistä lisäämään observointikohtainen kategoriaryhmä tai -ryhmiä ja vähintään yksi kategoria jokaiseen käytettävään ryhmään (katso kuva 13). Observoinnin kategoriaryhmien ja kategorioiden muokkaussivulla lisätyt kategoriaryhmät eivät ole käytettävissä muissa observoinneissa.





Kuva 13. Observointikohtaisen kategoriaryhmän lisäyssivu.

### 3.6 Observoinnissa käytettävien kategorioiden valinta

Yksittäisessä observoinnissa on käyttäjäroolista riippuen käytettävissä erilaisia kategoriaryhmä- ja kategoriavalintoja. Kirjautuneella käyttäjällä on käytettävissään kaikki luomansa kategoriaryhmät ja kyseiseen observointiin määrittämänsä ryhmät. Ryhmävaimella observointiin siirtynyt käyttäjä voi valita avainkohtaiset kategoriaryhmät ja lisäksi luoda observointikohtaisia kategoriaryhmiä. Julkisella käyttäjällä on käytössään vain observointikohtaiset kategoriat.

**Kirjautuneella käyttäjällä** on käytettävissään koko kategoriavalintojen kirjo. Kirjautunut käyttäjä voi luoda hallintasivulla tapahtumakohtaisia kategoriaryhmiä (katso kuva 14), jolloin ryhmät ovat myös jatkossa käytettävissä. Observointikohtaisten muutosten teko on myös mahdollista.

## Tapahtuman 'Testitapahtuma 1' observoinnin kategoriat

### Observointiin lisätyt kategoriaryhmät

Testikategoriaryhmä 1 ✕

Kesto	Kategoria 1	✕
Kesto	Kategoria 2	✕
Kesto	Kategoria 3	✕
Kerta	Kategoria 4	✕

+ Lisää uusi kategoria

Jatka observointiin

### Lisää kategoriaryhmiä observointiin

Uusi kategoriaryhmä:

Omat kategoriaryhmät:

Oletuskategoriaryhmät:

Kuva 14. Tapahtuman kategoriaryhmien valintasivu.

**Ryhmäavaimella** kirjautuneella käyttäjällä on käytettävissään avaimeen sidotut kategoriaryhmät. Käyttäjä voi valita yhden tai useamman näistä ryhmistä ja myös lisätä omia observointikohtaisia kategorioita (katso kuva 15).

**Tapahtuman 'Testitapahtuma 2' observoinnin kategoriat**

Observointiin lisätyt kategoriaryhmät

Kategoriaryhmä 1

Kesto Kategoria 1

Kerta Kategoria 2

+ Lisää uusi kategoria

Jatka observointiin

**Lisää kategoriaryhmiä observointiin**

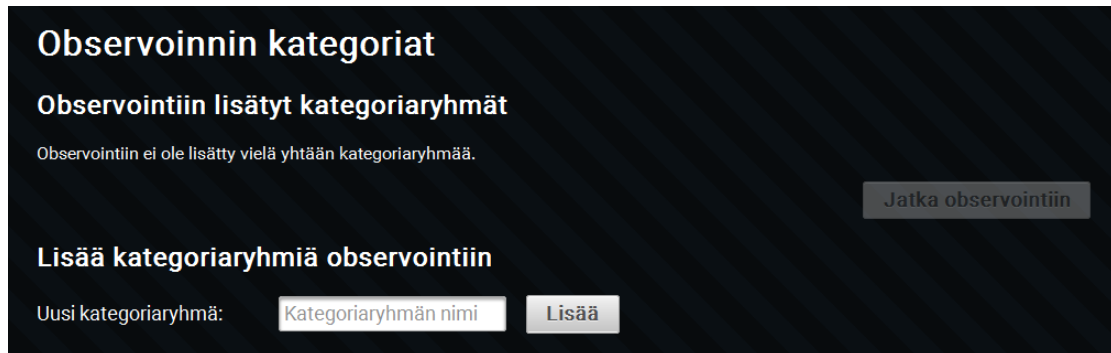
Uusi kategoriaryhmä: Kategoriaryhmän nimi Lisää

Omat kategoriaryhmät: Testikategoriaryhmä Lisää

Oletuskategoriaryhmät: Kategoriaryhmä 1 Lisää

Kuva 15. Ryhmäavaimella kirjautuneen käyttäjän kategoriavalintasivu.

**Julkisella käyttäjällä** on käytettävissään vain itse määrittelemänsä observaatiokohtaiset kategoriat kuvan 16 mukaisesti. Niitä ei voi tallentaa sovelluksen tietokantaan.



Kuva 16. Julkisen käyttäjän kategorialuontisivu.

### 3.7 Observoinnin suorittaminen

Observointinäkylässä on kuvien 17 ja 18 mukaisesti näkyvillä observoinnin ohjauspainikkeet sekä kategorioiden rekisteröintipainikkeet. Kerta- tai kestokategorialuokituksen mukaan on kullekin rekisteröintipainikkeelle joko oma kello tai kertalaskuri. Lisäksi yläosassa oleva observointikello näyttää observoinnin kokonaiskestoja. Observointi aloitetaan *Aloita*-painikkeella. Observoinnin ollessa käynnissä voidaan se keskeyttää tilapäisesti *Aloita*-painikkeen tilalle tulevalla *Tauko*-painikkeella ja jatkaa samalle paikalle vaihtuvalla *Jatka*-painikkeella.



Kuva 17. Observointinäkymä ennen aloitusta.

Observoinnin aikana aloitettujen kategorioiden rekisteröintipainikkeet muuttuvat keltaisiksi kuvan 18 mukaisesti. Kestokategorioiden rekisteröinnit vaativat aloitus- ja lopetuspainallukset, mutta kertakategorioiden rekisteröinnit vain yhden painalluksen. Kertakategoriapainikkeet eivät jää keltaiseksi, vaan ainoastaan käyvät hetken keltaisena.

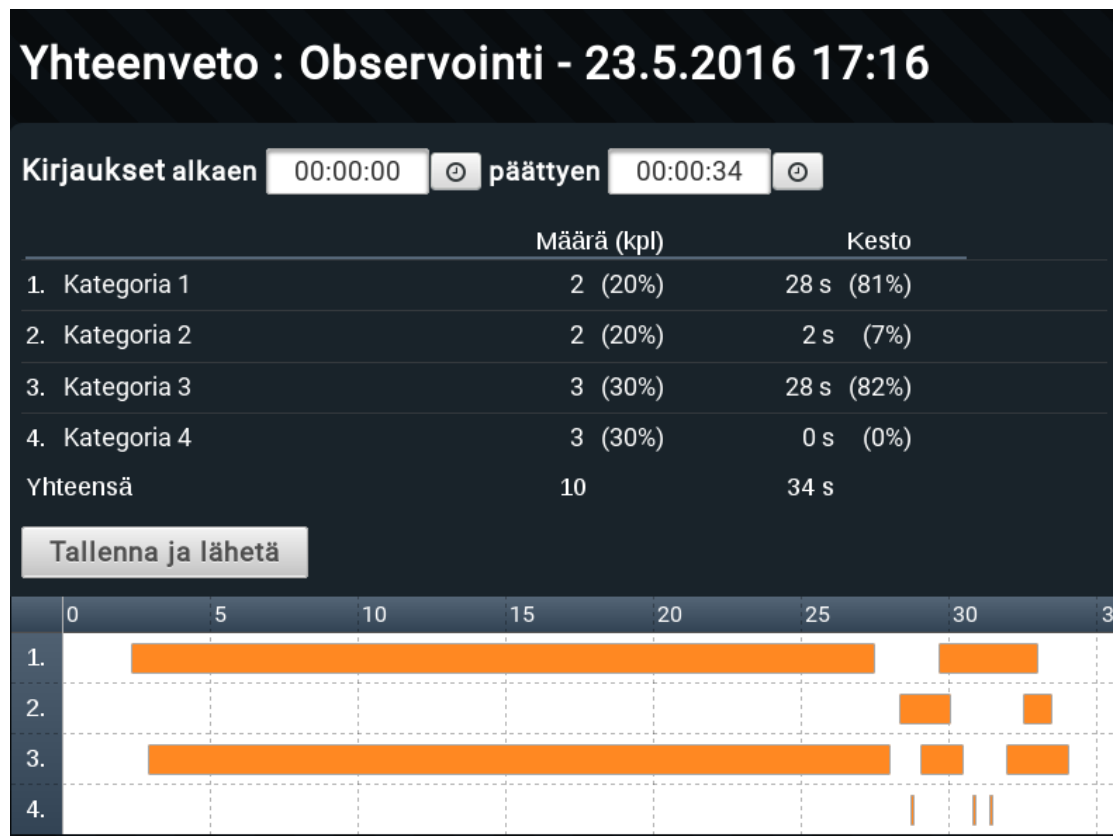


Kuva 18. Observointinäkymä observoinnin aikana.

Observointi lopetetaan *Lopeta*-painikkeella, joka myös automaattisesti avaa kuvan 19 observoinnin yhteenvetosivun. Observoinnin lopetus on tilaajan toiveesta päätetty pitää peruuttamattomana toimenpiteenä, eli sille ei pyydetä käyttäjältä vahvistusta.

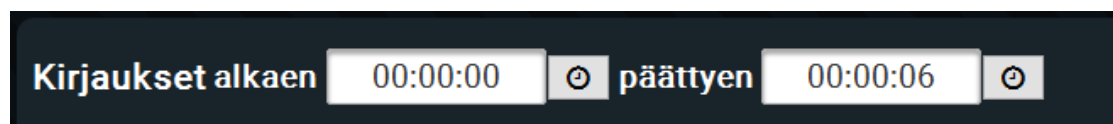
### 3.8 Observoinnin yhteenvetosivu

Observoinnin yhteenvetosivulla on kuvan 19 mukaisesti nähtävillä suoritettujen observointien numeerinen data kategorioittain eriteltyinä sekä aikajanakaavio sisältäen kunkin kategorian aikajänteet. Klikkaamalla aikajanan palkkia tulevat esille myös kyseisen kirjauksen alku- ja loppuaika sekä kesto. Kaikki ajat ovat aikoja observoinnin alkuhetkestä laskettuna. Numeerisessa datassa on myös prosenttiosuudet jokaisen kategorian kirjausten yhteenlasketun keston suhteesta kaikkien kategorioiden kokonaiskeston.



Kuva 19. Observoinnin yhteenvetosivu.

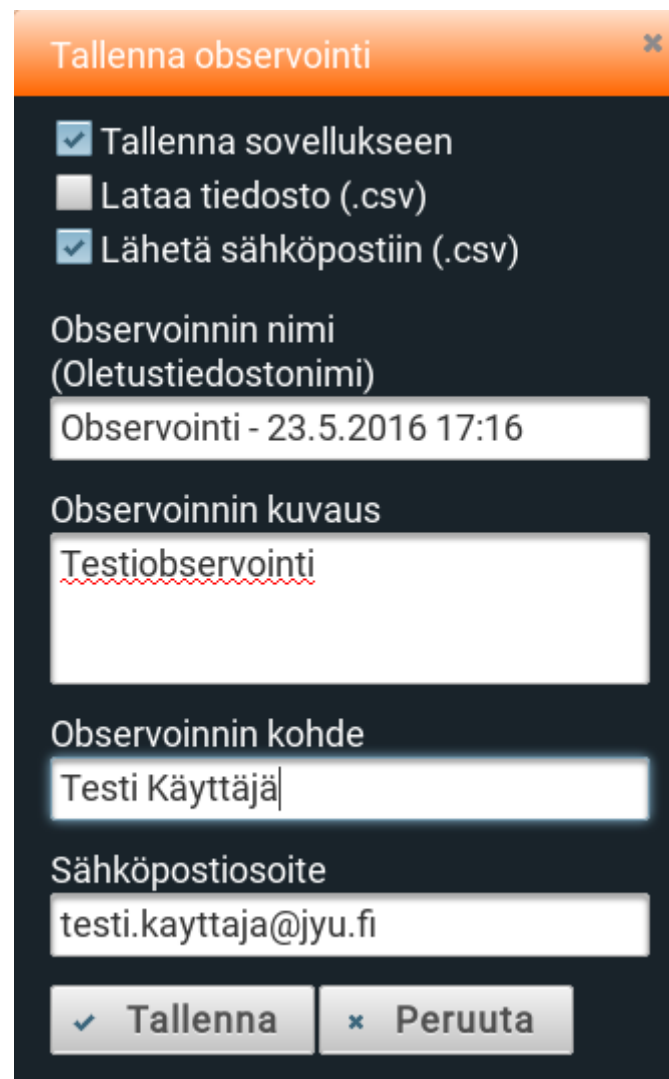
Kirjausten rajauskentillä (katso kuva 20) voidaan rajata syötteillä tai kellopainikkeista avautuvilla liukusäätimillä tarkasteltava observoinnin aikajänne. Tämä rajaus ei vaikuta tallennettavaan dataan, vaan se sisältää aina koko observoinnin aikajakson.



Kuva 20. Observoinnin yhteenvedon aikajänneen rajaustyökalurivi.

### 3.9 Observoinnin kirjausten ja yhteenvedon tallennus

Observoinnin kirjaukset ja yhteenvedo voidaan tallentaa käyttäjäroolista riippuen. JYU-tunnistautuneella käyttäjällä on käytettävissään kuvan 21 mukaisesti paikallinen tallennus omalle päätelaitteelle, sähköpostiin lähetys tai sovelluksen tietokantaan tallentaminen.



Tallenna observointi

- Tallenna sovellukseen
- Lataa tiedosto (.csv)
- Lähetä sähköpostiin (.csv)

Observoinnin nimi  
(Oletustiedostonimi)

Observointi - 23.5.2016 17:16

Observoinnin kuvaus

Testiobservointi

Observoinnin kohde

Testi Käyttäjä

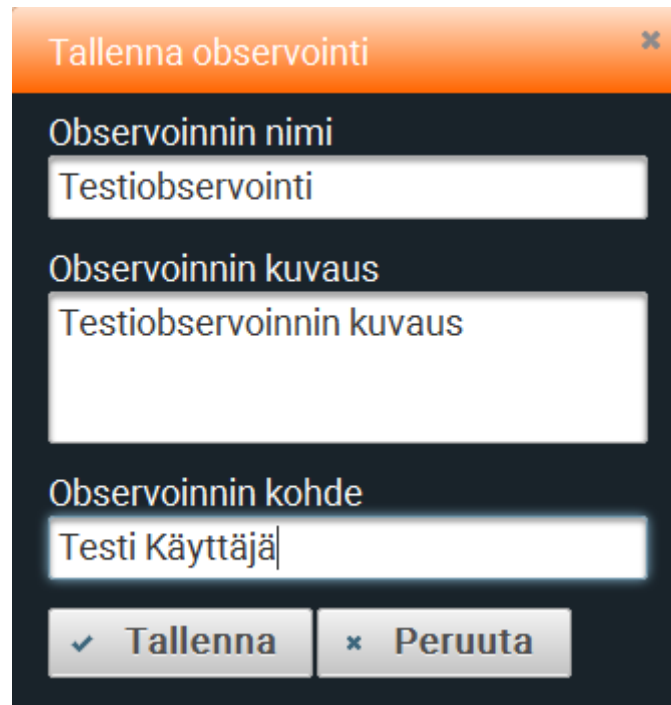
Sähköpostiosoite

testi.kayttaja@jyu.fi

Kuva 21. Kirjautuneen käyttäjän observoinnin tallennusikkuna.



Ryhmäavaimella kirjautunut tai julkinen käyttäjä voi tehdä ainoastaan paikallisen tallennuksen (katso kuva 22). Jotkin mobiililaitteet eivät tue CSV-tiedoston paikallista tallentamista. Projektissa havaittiin ainakin iPhonesta puuttuvan kyseinen tallennusmahdollisuus.



Tallenna observointi

Observoinnin nimi  
Testiobservointi

Observoinnin kuvaus  
Testiobservoinnin kuvaus

Observoinnin kohde  
Testi Käyttäjä

✓ Tallenna    ✕ Peruuta

Kuva 22. Julkisen käyttäjän observointidatan ja yhteenvedon tallennusikkuna.

## 4 Sovelluksen toteutusratkaisut

Moveatis-sovellus on toteutettu MVC-ohjelmistoarkkitehtuuria noudattaen. MVC (Model-View-Controller) perustuu ohjelmiston arkkitehtuurin jakamiseen kolmeen tasoon eli malleihin, näkymiin ja käsittelijöihin. Tällä mahdollistetaan käyttöliittymän ja sovelluslogiikan eristäminen ja yleistäminen, jolloin muutokset ja jatkokehitys helpottuvat merkittävästi.

### 4.1 Kehitystyökalut

Sovelluskehityksen palvelinpuolen sovelluksen pääasiallisena ohjelmointikielenä käytettiin Javaa, jonka tukena käytettiin Maven-ohjelmistokehitystyökalua. Sovelluksen WWW-käyttöliittymässä pääasiallisena ohjelmointikielenä toimi Javascript. Lisäksi käytettiin HTML5- ja CSS3-ohjelmointikieliä sekä Primefaces-komponenttikirjastoa ja JavaServer Faces -sovelluskehystä. Sovelluksen lähdekoodin projektiryhmä sijoitti avoimen lähdekoodin BSD 3-Clause -lisenssin alaisuuteen.

**Sovelluksen sivujen kehitysalustana** (engl. *framework*) on käytetty JSF:ää (Java Server Faces), joka on Javan päällä toimiva WWW-näyttötekniologia. JSF tukee malleihin (engl. *templates*) ja komponentteihin perustuvaa käyttöliittymäkehitystä. Sivut on toteutettu JSF:n XML-pohjaisina Facelet-sivuina, jotka JSF kääntää XHTML-sivuiksi.

Moveatis-sovelluksen käyttöliittymän komponenteissa hyödynnettiin PrimeFaces-**käyttöliittymäkirjastoa**, joka on rakennettu JSF:n päälle. PrimeFaces valikoitui käyttöön monipuolisuutensa, laajuutensa ja yhtenäisyytensä vuoksi. PrimeFaces on käytettävissä avoimella Apache-lisenssillä.

## 4.2 Sivujen toiminnallisuus

Sivujen toiminnallisuudet on rakennettu noudattamaan MVC:n ja JSF:n toimintalogiikkaa sisältäen tasoina mallit (*backend*), käsittelijät (*logiikka*) ja näkymät (*sivut*). Java-luokissa eli puvuissa on vastaavasti käytössä kolmitasorakenne Backing Beans, Managed Beans ja HTML. JSF ja PrimeFaces käyttävät AJAX-kutsuja käyttäjän tilan ylläpitämiseen, kuten esim. lomakkeen lähetykseen. Ajax-kutsu välittyy käsittelijöille (Managed Beans), jotka päivittävät sivua. Näkymää päivitetään dynaamisesti, eli vain kutsuun osallistuvat näkymäkomponentit päivittyvät. Käyttäjän sivulle tekemät muutokset puolestaan päivittävät käsittelijöiden kautta mallia ja siten mm. tietokantaa.

REST-rajapinta on avattu observointidatan vastaanottamiselle. REST mahdollistaa palvelujen esittämisen resurssien muodossa HTTP:tä käyttäen. Resurssien yleinen esitysmuoto on JSON tai XML. Moveatis-palvelussa REST ei ole olennaisessa roolissa, eli palvelu ei ole ns. RESTful.

JavaScriptiä on käytetty sivujen dynaamisen toiminnallisuuden rakentamiseen. Yhteenvetosivulla on toteutettu JavaScriptillä mm. yhteenvetotietojen rajausta ja päivitystä sekä aikajanakomponentin dynaamiset toiminnallisuudet. Observointisivun toimintalogiikka on myös toteutettu JavaScriptillä. JavaScriptit suoritetaan käyttäjän näkymässä, eli ne ovat ns. JS-tulkin selaimessa suorittamia asiakassovelluksia (engl. *client-side*). JavaScriptit on kirjoitettu jQuery-kirjastoa käyttäen.

Moveatis-sovelluksen sivujen ulkoasu on määritelty HTML:llä ja esitysmuoto on määritelty CSS3:a käyttäen. Perusasetukset on määritelty yhdessä CSS-tiedostossa ja sivukohtaisesti niitä on täydennetty.

### 4.3 Toimintalogiikka

Backend on Java EE -ohjelmistoalustalla toteutettu ratkaisu, joka tarjoaa Moveatiksien tarvitsemat taustatoiminnot. Backend huolehtii käyttäjän tunnistamisesta rajapinnan avulla JYU-tunnistautumispalveluun, käytön kannalta olennaisten tietojen tallentamisesta PostgreSQL-tietokantaan, sähköpostien lähettämisestä sähköpostipalvelimen välityksellä, observointien datan muokkaamisesta koneellisesti käsiteltävään tekstimuotoon sekä käyttäjän istunnon hallinnasta.

Java EE -ohjelmistoalustan mukainen ratkaisu jakautuu kolmeen kerrokseen MVC-periaatteen mukaisesti. **Model- eli mallikerroksessa** mallinnettavan reaali maailman käsitteet toteuttavat Java-entiteetit, ja ne abstrahoidaan tietojärjestelmän tauluiksi. Moveatiksessa entiteetit tunnistaa luokan `Entity`-pääteestä sekä `@Entity`-määreistä. Java EE -palvelin rakentaa määritellyistä entiteeteistä automaattisesti tietokantataulut.

**View- eli näkymäkerroksen** `JavaServerFaces`-sivuista rakentuu käyttäjän selaimella havainnoima käyttöliittymä, ja `ManagedBeans`-oliot rakentavat yhteyden backendin sekä JSF-sivujen välille. Esimerkiksi käyttöliittymässä olevaa painiketta kuuntelee jokin tietty `ManagedBean`-olio, ja painikkeen aktivointi saa aikaan toiminnot kyseisessä oliossa. Moveatiksessa `ManagedBean`-luokat tunnistaa `ManagedBean`-pääteestä, ja ne sijaitsevat paketissa `com.moveatis.managedbeans`.

**Control- eli hallintänäkymä** hallitsee tietokantayhteyttä sekä toimii näkymäkerroksen ja tietokannan välisenä kerroksena. Hallintaoliot hakevat, luovat, muokkaavat tai poistavat entiteettejä. Kukin hallintaolio vastaa yhdestä entiteetistä, ja vastaavasti kullakin entiteetillä on vain yksi hallintaolio. Hallintaolioiden toteutus Moveatiksessa jakautuu kahteen osaan: liittymäosioihin ja niitä toteuttaviin EJB-luokkiin. Liittymäosiossa määritellään sovelluspalvelimen ylläpitämien Enterprise JavaBeanien varaston tarpeita varten se liittymä, jonka jokin hallintaluokka toteuttaa. Sovelluspalvelin ylläpitää useita instansseja hallintaluokasta ja valitsee liittymän perusteella oikean, kun liittymän palveluita tarvitaan. Näin tietokantayhteys on

valmiina, eikä suoritinaikaa tarvitse tuhata kenttien ja yhteyden alustamiseen. Liittymällä täytyy olla sen toteuttava luokka, joka suorittaa ne operaatiot, joihin se on liittymän toteuttamisessa sitoutunut. Moveatiksessa liittymät on sijoitettu pakettiin `com.moveatis.interfaces` ja liittymän toteuttavat EJB-luokat tunnistaa niiden nimen Bean-päätteestä.

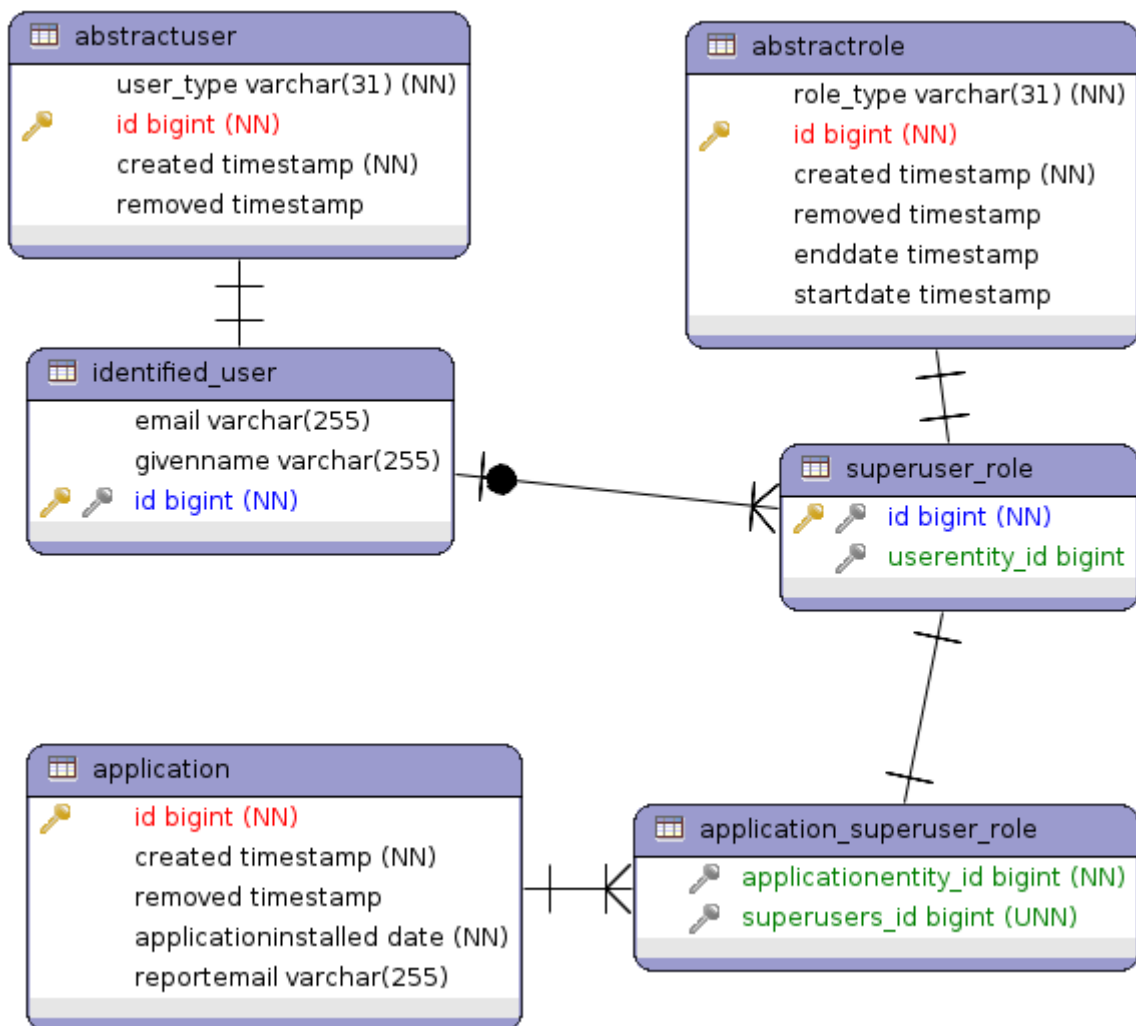
#### 4.4 Sovelluspalvelin

Moveatis on sijoitettu Wildfly-sovelluspalvelimelle hakemistoon `standalone/deployments`, jossa tiedosto `ROOT.war` sisältää Moveatiksien pakatun tiedostorakenteen. WAR-formaatti on standardin mukainen tapa jakaa Java EE -sovelluksia. Tiedoston nimi määrittelee sen URL-polun, jolle sovelluspalvelin sijoittaa kyseisen sovelluksen. `ROOT` on erikoisnimi, joka sijoittaa sovelluksen palvelimen juureen. Moveatiksien hakemistorakenne ei sisällä sellaisia asetustiedostoja, joita käsittelemällä Moveatiksien toimintaa voisi muokata sovelluksen ollessa käynnissä. Moveatiksien päivitys vaatii aina sovelluksen pysäyttämisen ja uuden tiedoston `ROOT.war` kopioinnin Wildflyn hakemistoon `standalone/deployments`.

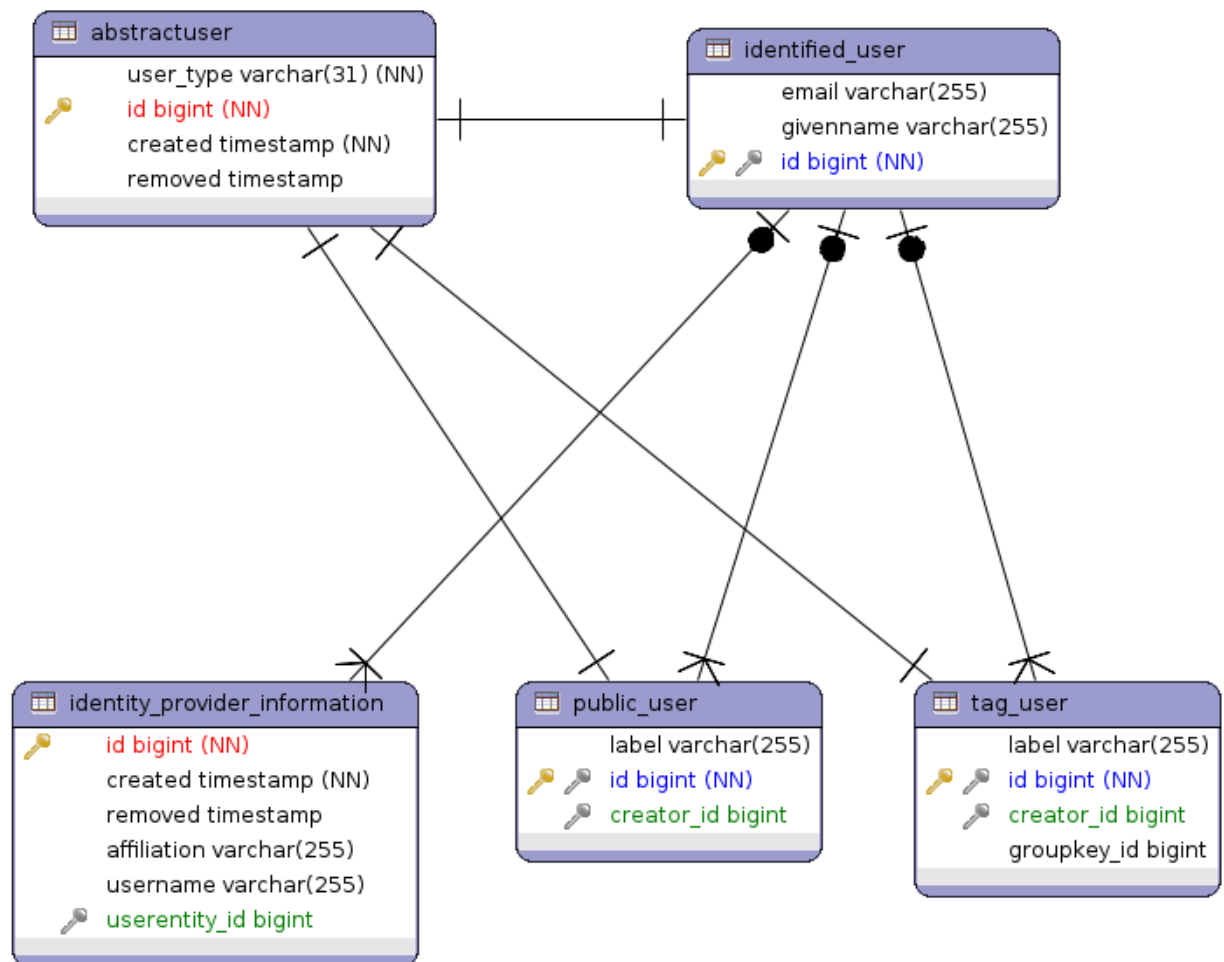
## 5 Sovelluksen tietokannan rakenne

Moveatiksien tietokannan rakennekaavio on selvyyden vuoksi jaettu moneen osaan, jotta datajoukkojen rakenne hahmottuu selkeämmin sekä taulujen ja kenttien nimet ovat tekstikooltaan luettavampia. Taulut esiintyvät useassa kaaviossa, koska moni tauluista liittyy niin vahvasti toisiinsa. Taulut on esitetty kuvissa 23–27 ja selitetty taulukossa 1.

### 5.1 Käyttäjät ja roolit

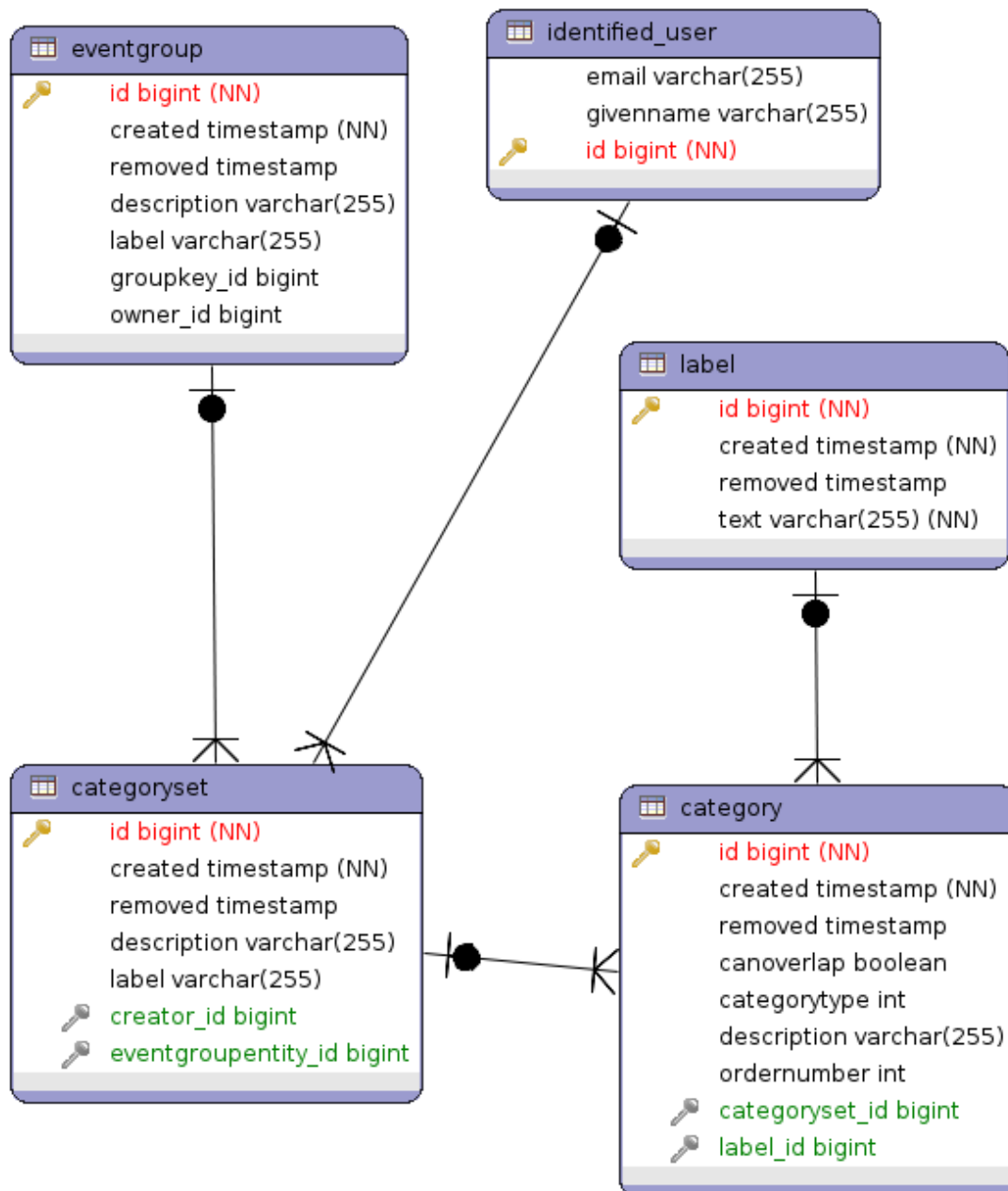


Kuva 23. Rooleihin ja sovellukseen liittyvät taulut.



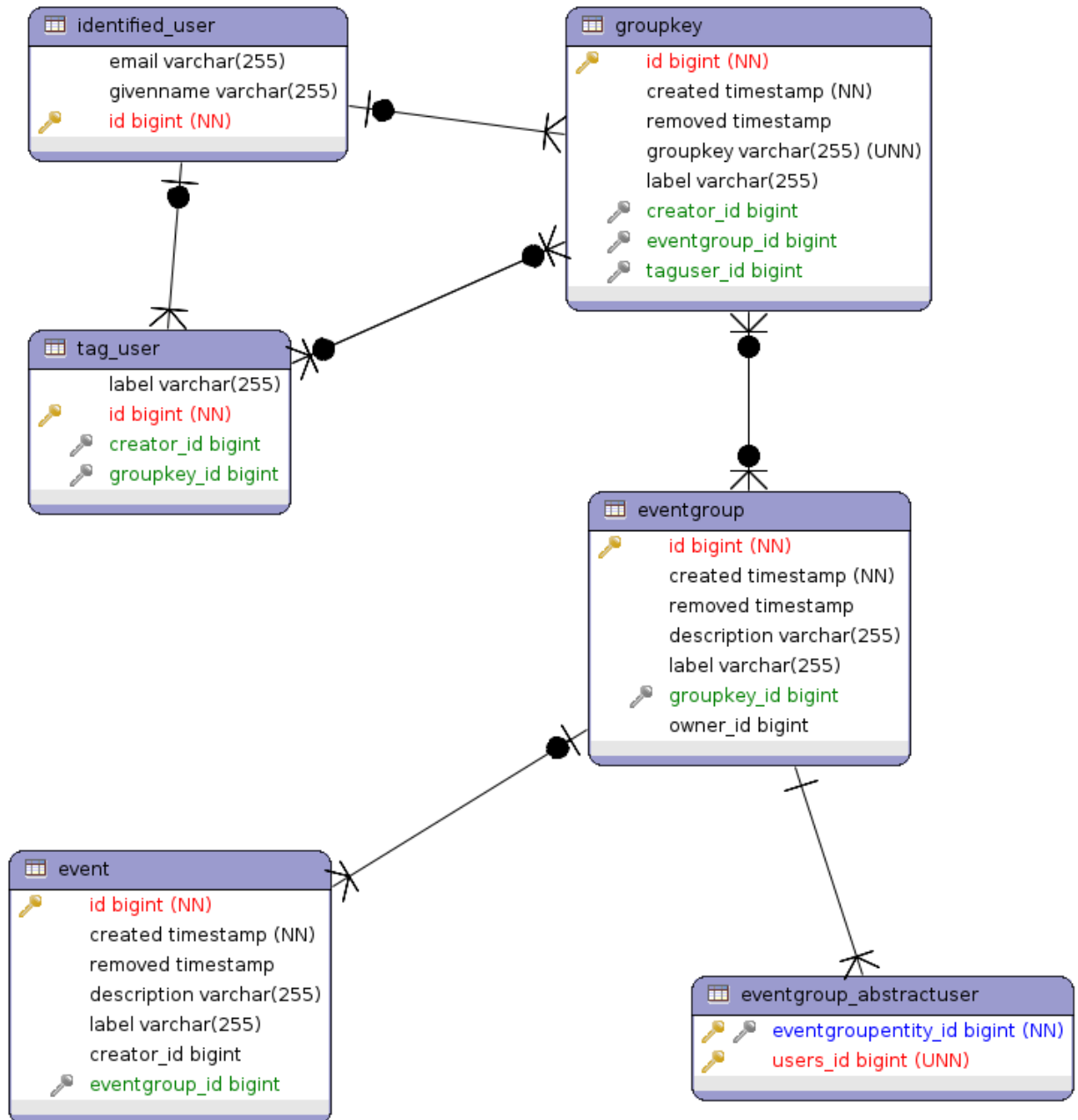
Kuva 24. Käyttäjien tietoihin liittyvät taulut.

## 5.2 Kategoriat, tapahtumat ja observoinnit

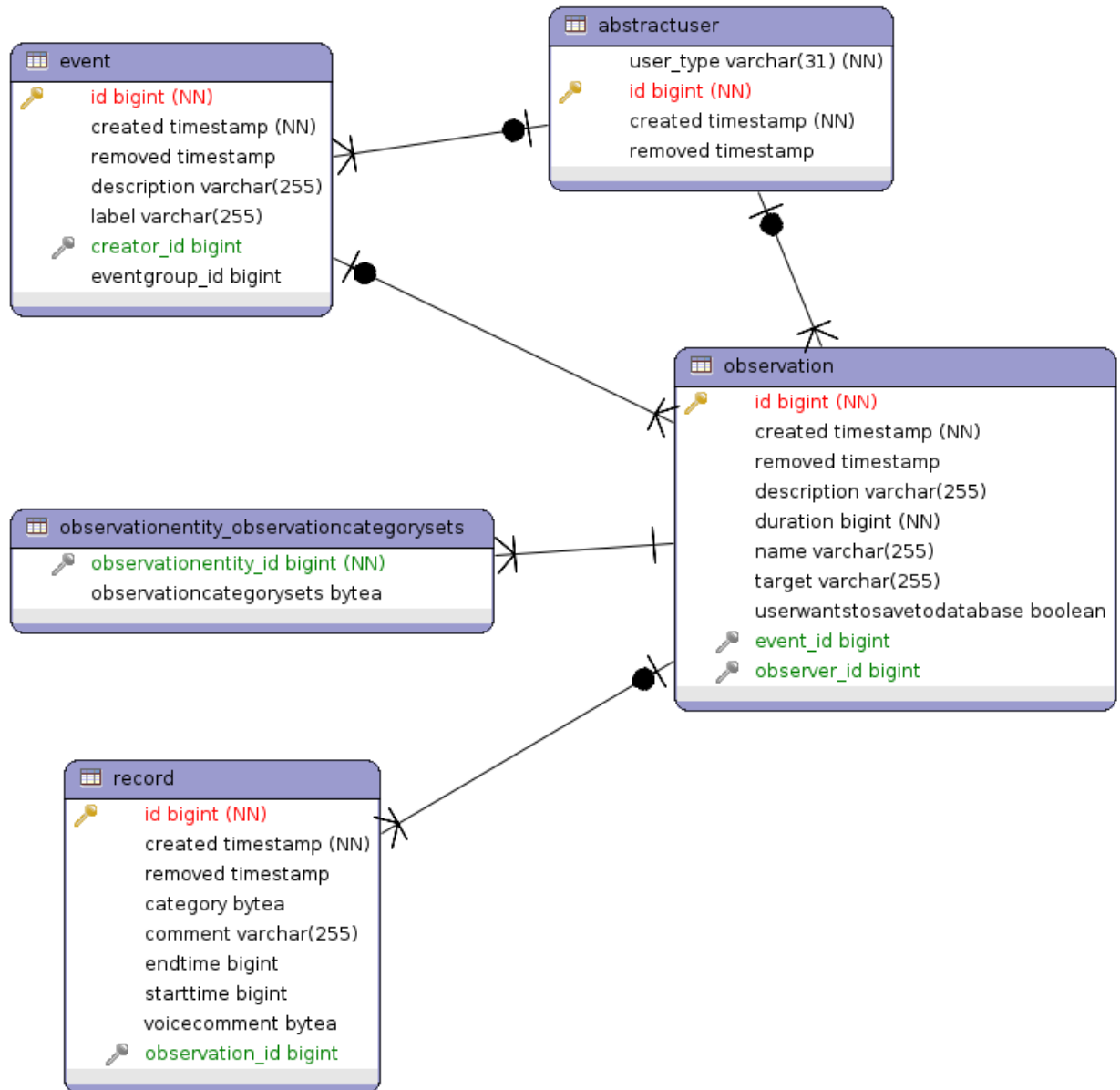


Kuva 25. Kategorioiden, kategoriaryhmien ja tapahtumaryhmien yhteydet.





Kuva 26. Tapahtumien, tapahtumaryhmien ja avainkäyttäjien yhteydet.



Kuva 27. Observointien, kirjausten ja tapahtumien yhteydet.

### 5.3 Taulujen käyttötarkoitukset

Moveatis-sovelluksen tietokannan taulujen käyttötarkoitusta on kuvattu taulukossa 1. Tarkemmin tietokannan käyttöä ja rakennetta on kuvattu Moveatis-projektin tietokantakuvausessa [6].

<b>Taulu (ja vastaava Java-entiteetti)</b>	<b>Kuvaus</b>
abstractrole (AbstractRole)	Roolien yliluokka. Uusia rooleja lisätään Moveatukseen tätä luokkaa laajentamalla
abstractuser (AbstractUser)	Moveatoksen kolmen käyttäjäroolin yliluokka.
application (ApplicationEntity)	Sisältää sovelluksen omaseurantaan käyttämiä tietoja.
application_superuser_role (Ei vastaavuutta.)	Ohjelmistoalustan automaattisesti luomassa taulussa pidetään kirjaa sovelluksen pääkäyttäjistä.
category (CategoryEntity)	Kategoriaan liittyvien tietojen tallennus.
categoryset (CategorySetEntity)	Kategoriaryhmään liittyvien tietojen tallennus.
event (EventEntity)	Tapahtumien tietojen tallennus. Ei käyttöä Moveatoksen versiossa 1.0, sillä kaikilla tapahtumaryhmillä on vain yksi tapahtuma.
eventgroup (EventGroupEntity)	Tapahtumaryhmien tietojen tallennus.
eventgroup_abstractuser (Ei vastaavuutta.)	Ohjelmistoalustan automaattisesti luoma taulu, jossa pidetään kirjaa tapahtumaryhmien käyttöoikeuksista.
groupkey (GroupKeyEntity)	Tapahtumaryhmien ryhmäavainten tietojen tallennus.
identified_user (IdentifiedUserEntity)	JYU-tunnistautuneiden käyttäjien tietojen tallentamiseen tarkoitettu taulu.

<b>Taulu (ja vastaava Java-entiteetti)</b>	<b>Kuvaus</b>
identity_provider_information (IdentityProviderInformationEntity)	Luo yhteyden JYU-tunnistautumistietojen ja Moveatiksens tunnistetun käyttäjän tietojen välille.
label (LabelEntity)	Kategorioilla on käyttäjälle näkyvä tunniste, esimerkiksi ”Tarkkailee”. Tunnisteet tallennetaan label-tiluun, jotta samaa tunnistetta voidaan hyödyntää useassa kategoriassa.
observation (ObservationEntity)	Observaatioiden tietojen tallennus.
observationentity_observation- categorysets (ObservationCategorySet)	Taluun tallennetaan observaatiokohtaiset kategoriaryhmät, jotta käyttäjä voi vapaasti muokata kategorioita ennen observaation suorittamista.
public_user (AnonUserEntity)	Talu sisältää yhden yhteisen julkisen käyttäjän tiedot. Niitä käytetään backendin toiminnoissa, jotka liittyvät sivujen käyttöoikeuksiin ja näytettäviin toimintoihin.
record (RecordEntity)	Observaatioiden kirjaukset tallennetaan taluun. Byte-muotoista tietokenttää oli tarkoitus käyttää äänikommenttien tallentamiseen, mutta sitä ei kuitenkaan käytetä versiossa 1.0.
superuser_role (SuperUserRoleEntity)	Pääkäyttäjien roolissa käyttäjälle sallitaan Moveatiksens ylläpitotoimien suoritusoikeus. Talu ei ole käytössä versiossa 1.0
tag_user (TagUserEntity)	Moveatista voi käyttää myös ennakkoon määritellyn ryhmävaimen avulla. Taluun tallennetaan avaimeen liitettävän ryhmävainkäyttäjän tiedot.

Taulukko 1. Tietokannan taulujen käyttötarkoitus.

## 6 Sovelluksen tavoitteiden toteutuminen

Kehitetyn sovelluksen taustaa, sillä tuettavaa prosessia ja asetettuja tavoitteita on kuvattu projektiraportissa [2]. Sovellukselle laadittiin sovellussuunnitelmia. Vaatimuksia hallittiin Trello-palvelulla. Vaatimusmäärittelydokumenttiin [5] on Trellostä koostettu toteutetut ja jatkokehitykseen siirretyt ominaisuudet. Jatkokehityssuunnitelmaa luotaessa on syytä ottaa huomioon myös sovelluksen tämänhetkisen version varsinaisen käytön aikana kertyvät kokemukset.

### 6.1 Toteutetut ominaisuudet

Olennaisimmat ominaisuudet on toteutettu vaatimusmäärittelyn mukaisesti. Tärkeimmät toiminnallisuudet, kuten Jyväskylän yliopiston tunnuksilla kirjautuminen ja ryhmävaimen käyttö toimivat tilaajan toivomalla tavalla. Observoinnin yhteenveto sisältää numeerisen yhteenvedon lisäksi kuvaajan, jota on mahdollista rajata halutun tarkastelujakson mukaisesti. Observointidata on mahdollista tallentaa sovelluksen tietokantaan, paikallisesti omalle laitteelle sekä lähettää sähköpostitse.

Yksi aiemman sovelluksen suurimmista ongelmista oli observointidatan jatkokäytön vaikeus. Moveatis-sovelluksessa observointidata tallennetaan CSV-muodossa, joka on yleisesti käytössä oleva esitysmuoto. Mm. Excel ja SPSS tunnistavat ko. tiedostomuodon.

Omien kategoriaryhmien ja tapahtumien hallinta pyrittiin myös tekemään aiempaa sovellusta helpommaksi. Saavutetun tason katsottiin olevan varsin riittävä. Esimerkiksi ryhmävaimen luoja saa nyt kaikki ryhmävaimella suoritettut observoinnit omaan näkymäänsä riippumatta siitä, kuka sen on tehnyt.

Vaatimusmäärittelyä [5] laadittaessa vaatimusten priorisoinnissa käytettiin seuraavaa jaottelua: *Pakollinen*, *Tärkeä*, *Mahdollinen*, *Idea* ja *Ei toteuteta*. Vaatimuksen

*Hyväksytyy*-tila tarkoittaa, että ominaisuus on testattu onnistuneesti ja hyväksytyt tilaajan toimesta. *Sovittu jatkokehitykseen* -tila tarkoittaa, ettei ominaisuutta toteutettu, vaan se sovittiin tilaajan kanssa jätettäväksi mahdolliseen jatkokehitykseen. Tila voi myös olla *Toteutettu, ei käytössä*, jolloin kyseinen toiminto on toteutettu, mutta se on otettu pois käytöstä.

Taulukoissa 2 ja 3 on esitelty yhteenveto vaatimusten toteutumisesta prioriteetteihin ja kokonaisuuksiin jaoteltuna. Pakolliset vaatimukset yhtä lukuun ottamatta saatiin toteutettua. Vaatimusten ainoa pakolliseksi määritelty, mutta toteuttamaton ominaisuus oli kategoriaryhmän kategorioiden valinta ei-aktiiviseksi observointikohtaisesti. Toteutusratkaisuna päädyttiin kategorian poistoon kokonaan, ja valintamahdollisuus sovittiin tilaajan kanssa jatkokehitykseen. Lisäksi toteutettiin myös joitain alemman prioriteetin vaatimuksia. Pääkäyttäjäsivu ja video-observointi sovittiin tilaajan kanssa kokonaan jatkokehitykseen.

<b>Prioriteetti</b>	<b>Vaatimuksia</b>	<b>Toteutettu</b>
Pakollinen	63	62
Tärkeä	16	5
Mahdollinen	31	3
Idea	15	2
Ei toteuteta	8	0
<b>Yhteensä</b>	<b>133</b>	<b>71</b>

Taulukko 2. Vaatimukset prioriteeteittain.

<b>Kokonaisuus</b>	<b>Vaatimuksia</b>	<b>Toteutettu</b>
Käyttöliittymä	15	7
Kategoriaryhmien valintasivu	20	12
Observointisivu	14	12
Yhteenvetosivu	23	12
Julkinen käyttö	12	1
Hallintasivu	36	23
Asetukset	9	2
Rajoitukset	3	3
Pääkäyttäjäsivu	6	0
Video-observointi	4	0
<b>Yhteensä</b>	<b>133</b>	<b>71</b>

Taulukko 3. Vaatimukset kokonaisuuksittain.

## 6.2 Jatkokehitykseen sovitut ominaisuudet

Sovellusprojektin aikana ideoitiin useita ominaisuuksia ja toiminnallisuuksia, jotka rajatusta työmäärästä ja aikajänteestä johtuen sovittiin tilaajan kanssa jatkokehitykseen. Olennaisimpia niistä ovat observointidatan tuonti sovellukseen, useamman yhtäaikaisen observoinnin vertailu sovelluksessa sekä observointitapahtuman teksti- ja äänikommentointi. Sovelluksen integrointi Jyväskylän yliopiston moniviestimen videotoistoon oli myös esillä varhaisessa vaiheessa. Koska sovelluksen käyttöä on tarkoitus laajentaa yliopiston ulkopuolelle, ei tätä ominaisuutta katsottu oleelliseksi. Yliopiston ulkopuolinen käyttö edellyttää myös valmiutta sovelluksen toimintaan jonkin muun kirjautumistavan, esimerkiksi HAKA-kirjautumisen, kanssa.

Sovellus on toteutettu tällä hetkellä suomeksi ja englanniksi. Kieliversiot on kovakoodattu sisään ohjelmaan siten, että kaikki saman kieliversioon tekstit ovat omassa määrittelytiedostossaan. Kielitukea laajennettaessa voi olla tarpeen määrittellä kieliversioille erilainen toteutustapa ylläpidon helpottamiseksi muutostilanteissa.

Käytettävyyteen liittyviä jatkokehitystarpeita ovat mm. observoinnin yhteenvedon kuvaajan aikayksiköiden esityksen yhdenmukaistaminen. Käytettävissä olleen ajan

puitteissa ei ko. valmiskomponentin toimintaa saatu selkeimmäksi mahdolliseksi. Tapahtumien, kategoriaryhmien ja kategorioiden hallintasivun käyttölogiikkaa on myös mahdollista parantaa yhtenäistämällä esitystapaa ja vähentämällä erillisiä dialogeja.

Jatkokehitykseen siirrettiin myös pääkäyttäjäroolien toiminnallisuudet ja hallinta. Kaikilla kirjautuneilla käyttäjillä on nykyisessä versiossa samat oikeudet, eli oikeus omien tapahtumien, kategoriaryhmien ja observointien hallintaan. Jatkokäyttöä ajatellen voi olla tarpeen toteuttaa normaalin kirjautuneen käyttäjän yläpuolelle ns. pääkäyttäjärooli laajemmilla käyttöoikeuksilla. Samalla on syytä pohtia normaalien kirjautuneiden käyttäjien toimintoja ja niiden mahdollista rajaamista.



## 7 Ohjeita sovelluksen ylläpitoon ja jatkokehitykseen

Moveatis-sovelluksen lähdekoodi on sijoitettu GitHub-palveluun Moveatis-tilille. Tilin tunnukset ovat Nelli Lyyralla. Tili ei kuitenkaan ole yksityinen, eli kuka tahansa voi ladata lähdekoodin kehitettäväkseen. Trelloon määritellyt vaatimukset sisältävät myös projektin jälkeiseen jatkokehitykseen sovitut tiedot ja toiminnot.

Muiden sovellusten tavoin Moveatis tarvitsee ylläpitäjiltään tiettyjä toimia. Projektiryhmä toivoo Moveatiksena olevan käyttäjilleen niin hyödyllinen, että sitä halutaan myös jatkokehittää. Luvussa ohjeistetaan sekä ylläpitäjiä että jatkokehittäjiä.

### 7.1 Sovelluksen asennus- ja käyttöönotto-ohjeet

Moveatis on Java EE -verkkosovellus, joten se tarvitsee Java EE:tä tukevan sovelluspalvelimen asennuskohteekseen. Java EE -verkkosovellukset paketoidaan war-tiedostoksi, joka sijoitetaan sovelluspalvelimen asennushakemistoon. Erilaisia Java EE -sovelluspalvelimia on paljon, eikä tässä voida esittää tyhjentävää ohjeistusta niitä kaikkia varten. Tarkista käyttämästäsi sovelluspalvelimen ohjeistuksesta, mikä on sen käyttämä asennushakemisto. Esimerkiksi Wildfly 10:ssä asennushakemisto on `<wildflyn juurihakemisto>standalone/deployments`. Katso myös ohjeista, miten käyttämäsi sovelluspalvelin asetetaan ottamaan haluttu war-tiedosto käyttöön. Wildflyssä riittää, että tiedosto siirretään asennushakemistoon.

Moveatis käyttää tietokantaa tietojen tallentamiseen, ja Java EE:ssä sovelluspalvelin tarjoaa tietokantayhteyden sovelluksen käyttöön. Oletusasetuksena Moveatis odottaa, että sovelluspalvelin tarjoaa tietokantayhteyden polussa `java:jboss/datasources/MoveatisDS`. Polkua voi muuttaa tiedostossa `pom.xml`, mutta nimeä muutettaessa pitää Moveatis kääntää uudelleen. Tarkista käyttämästäsi sovelluspalvelimen ohjeista, miten siinä asetetaan tietokantayhteyksiä tarjolle.

Moveatis vaatii Java EE -sovelluspalvelimen lisäksi tietokantapalvelimen. Tietokantapalvelimen tulee olla sellainen, jota valittu Java EE -sovelluspalvelin tukee. Turvallinen valinta on MySQL:n tai PostgreSQL:n kaltainen suosittu tietokantapalvelin. Tutustu valitsemasi Java EE -sovelluspalvelimen ohjeisiin siitä, mitä toimenpiteitä se vaatii toimiakseen käyttämäsi tietokantapalvelimen kanssa. Esimerkiksi projektiryhmä käytti kehitystyössä Wildfly 9 -sovelluspalvelinta ja PostgreSQL 9.4 -tietokantapalvelinta. Wildflyhyn piti asentaa jar-tiedostoksi pakattu PostgreSQL-ajuri ja määrittellä MoveatisDS-niminen tietokantayhteys käyttämällä `jboss-cli.sh`-komentorivityökalua. Projektiryhmän kehitystyössä käyttämän Docker-imagin rakennustiedosto Dockerfile löytyy Githubista osoitteesta <https://github.com/phinaliumz/moveatis-docker>. Siitä voi ottaa vaikutteita, sillä se sisältää Wildfly 9:n asennuksensa sekä PostgreSQL:n skriptin, joka asentaa PostgreSQL-ajurin sekä määrittelee MoveatisDS-tietokantayhteyden.

Java EE sovelluksissa sovelluspalvelin rakentaa tietokantataulut automaattisesti sovelluksen asennuksen yhteydessä, jos niin halutaan. Tämän ominaisuuden saa sovelluskohtaisesti päälle määrittelemällä avaimen `javax.persistence.schema-generation.database.action` arvoksi `create`. Moveatiksessa avain määritellään tiedostossa `persistence.xml`, joka sijaitsee hakemistossa `src/main/resources/META-INF`. Tuotantokäytössä asetus on `none`, mutta kehitystyössä saattaa olla hyödyllistä käyttää asetusta `drop-and-create`, jolloin sovelluspalvelin poistaa vanhat taulut tietokannasta ja tekee ne uudelleen. Tämä asetus siis tyhjentää tietokannan tiedoista ja luo uudet tyhjät taulut. Tämä on syytä huomioida kehitystyössä, ettei poista säilytettäviä tietoja väärään aikaan.

Sovelluspalvelin suorittaa edellä mainitut tietokantaoperaatiot, kun sovellusta asennetaan, esimerkiksi siirtämällä war-tiedosto sovelluspalvelimen asennushakemistoon. Jos Moveatis asennetaan uudelleen tyhjään tietokantaan, tulee tiedoston `persistence.xml` avain `javax.persistence.schema-generation.database.action` muuttua arvoon `create`, tai sovelluspalvelin ei luo tietokantatauluja eikä Moveatis ole käytettävissä. Luonnin jälkeen avain on syytä muuttaa takaisin arvoon `none`. Huomaa, että kun arvoa muutetaan, tulee

kirjoitushetken tilanteen mukaisesti Moveatis kääntää ja paketoita uudelleen war-tiedostoksi.

Kirjoitushetkellä kirjautuminen Jyväskylän yliopiston tunnuksilla tapahtuu siten, että Moveatiksensa kanssa samalla palvelimella on asennettuna Apache httpd -palvelin, joka hoitaa liikenteen Moveatiksensa ja tunnistautumispalvelun välillä. Moveatis ja Apache httpd kommunikoivat AJP-protokollalla Java EE -sovelluspalvelimen portin 8009 kautta. Kommunikaatio asetetaan Apachen ja sovelluspalvelimen asetuksia muuttamalla, eikä Moveatiksensa lähdekoodiin tarvitse koskea. Tunnistautumispalvelun käyttöönotto on oma taiteenlajinsa, jota varten tässä ei voida antaa tyhjentävää ohjeistusta. Mikäli mahdollista, tutustu Moveatiksensa projektiryhmän asennuksen Apache httpd - ja Wildfly -asetustiedostoihin. Pyydä apua Jyväskylän yliopiston IT-palveluilta, mutta varaudu mahdollisesti pitkään viiveeseen vastauksen saannissa ja ylimalkaiseen ohjeistukseen. Kehitystyön aikana on tarkoituksenmukaista käyttää Jyväskylän yliopiston kirjautumista ainoastaan silloin, kun siihen liittyvää toimintoa kehitetään.

#### **Sovellusta asennettaessa on huomioitava seuraavat rajoitteet:**

- Sovelluksen on testattu toimivan Java EE 7 -versiolla, Java 8 -virtuaalikoneessa, Wildfly 10 -sovelluspalvelimella ja PostgreSQL 9.4 -tietokantapalvelimella.
- Sovellus ei toimi ilman Javaa eikä ilman tietokantaa.

#### **Sovellusta käyttöön otettaessa on syytä huomioida seuraavat rajoitteet:**

- Sovellusta on testattu yleisimmillä, valmistajien tukemilla selaimilla, keväällä 2016 käytettävissä olevilla versioilla. Suositellut selaimet ovat Microsoft Internet Explorer 10, Mozilla Firefox 45.0, Google Chrome 49.0 ja Apple Safari 9 (tai näiden selainten uudemmat versiot).
- JavaScript on olennainen osa käyttöliittymää, eikä käyttöliittymä toimi oikein, mikäli JavaScript on selaimessa kytketty pois päältä.

## 7.2 Ohjeita jatkokehittäjälle

Kaikki käyttöliittymään liittyvä koodi löytyy hakemistosta `src/main/webapp`. Javascript, CSS ja kuvat löytyvät hakemistosta `webapp/META-INF/resources`. Jokaisella HTML-sivulla on ylä- ja alapalkki, jotka määritellään hakemistosta `src/main/webapp/WEB-INF/` löytyvillä `xhtml`-tiedostoilla. Sovelluksen JavaServer Faces -teknologialla toteutetut HTML-sivut rakennetaan XHTML-tiedostoina, joissa on HTML-koodin lisäksi erilaisia määreitä sivun rakentamista varten. JavaServer Faces -teknologiaan kuuluu FacesServlet, joka rakentaa XHTML-sivuista varsinaiset näytettävät HTML-sivut.

Java EE -teknologiassa tietokantataulut rakennetaan sovelluspalvelimen toimesta sovellusta asennettaessa. Tietokannan taulu tulee kustakin Java-luokasta, jolla on `@Entity`-määre. Entiteetit on hajautettu käyttötarkoituksensa mukaisesti eri paketteihin, jotka ne jakavat yhdessä niihin liittyvien Enterprise JavaBeanien kanssa. Esimerkiksi käyttäjään liittyvä entiteetti `IdentifiedUserEntity` sijaitsee paketissa `com.moveatis.user`. Entiteetit tunnistaa `Entity`-loppuosasta luokan nimessä.

Kirjoitushetkellä Java EE -sovelluspalvelin ei luo entiteeteistä tietokantatauluja, jos samannimisiä tauluja on jo olemassa. Jos siis jonkun entiteetin luokkaa muutetaan, täytyy tietokanta tyhjentää aluksi kokonaan. Tämä on syytä huomioida jatkokehityksessä, eikä tietokantatauluja tule muuttaa kevyin perustein. Tietokannasta ei tulisi muutenkaan poistaa mitään, vaan vain merkitä tieto poistetuksi.

Virheenjäljitys onnistunee parhaiten sovelluskehitysympäristön, kuten Netbeansin tai IntelliJ IDEAn, debuggaustyökaluilla. Esimerkiksi Netbeansin komennolla *Attach debugger* voi liittää debuggerin sovelluspalvelimella pyörivään Moveatikseen. Käyttöliittymään liittyvää JavaScriptia ja CSS:ää voi debugata selaimien tarjoamilla työkaluilla.

Moveatis odottaa, että sitä käytetään JYU-tunnuksilla. Luokkaa `src/main/java/com/moveatis/devel/DevelJYULoginBean` kannattaa hyödyntää kehitystyössä, sillä se antaa mahdollisuuden testata sovellusta dummy-tunnuksilla. Tällöin kirjautuminen tehdään `/jyutesting`-sivun kautta.

Sähköpostien lähetystä voi kehityskoneella testata asentamalla jonkin dummy-SMTP-palvelimen omalle koneelleen. Oletuksena Moveatis käyttää localhostia SMTP-palvelimena, joten muuta säätöä ei tarvitse tehdä kuin asentaa jokin dummy-SMTP-palvelin. Esimerkiksi kirjoitushetkellä Githubista löytyvä Dummy-SMTP on testikäyttöön riittävä sähköpostipalvelin, joka luodaan Python-skriptillä.

### 7.3 Ohjeita ylläpitäjälle

Java EE -sovelluspalvelimella tulee olla tietokantayhteys tarjolla samalla määrittelyllä, joka Moveatiksien tiedostossa `pom.xml` on määriteltä. Tutustu sovelluspalvelimen ohjeisiin tietokantayhteyksien määrittelystä. Moveatis odottaa, että tietokantayhteys löytyy osoitteesta `java:jboss/datasources/MoveatisDS`. Mikäli osoitetta halutaan muuttaa, pitää Moveatis käynnä uudelleen.

Moveatis antaa käyttäjilleen mahdollisuuden poistaa observaatioita. Observaatiot eivät poistu tietokannasta, vaan ne vain merkitään poistetuiksi. Kirjoitushetkellä Moveatis ei tarjoa ylläpitäjälle toimintoa, jonka avulla tietokannan tauluja voisi tarkastella. Tietokantaan liittyvät ylläpitotoimet tulee tehdä tietokannan omilla työkaluilla tai muilla, tietokantaan erikoistuneilla ohjelmistoilla, kuten pgAdminilla.



## 8 Yhteenveto

Moveatis-projekti kehitti kevään 2016 Sovellusprojekti-kurssilla Lotas Observer -ohjelman korvaavan WWW-sovelluksen Jyväskylän yliopiston liikuntakasvatuksen laitokselle. Sovelluksen avulla seurataan opettajaopiskelijoiden toimintaa *Tutkiva opettaja* -kurssilla, minkä lisäksi sovellusta pyritään jatkossa hyödyntämään myös muilla kursseilla ja laitoksen tutkimuksissa. Sovellusprojektin tuloksena valmistuneen version voidaan katsoa olevan valmis tuotantokäyttöön. Tuotantoon ottamista edellä mainitulla kurssilla syksyllä 2016 tukee kaksi ryhmän jäsentä. Sovellus on sijoitettu Jyväskylän yliopiston palvelimelle.

Sovellusprojektissa kehitettyä WWW-sovellusta voi käyttää kaikilla laitteilla, joilla on pääsy Internetiin. Saatavuus oli yksi edellisen sovelluksen suurimpia ongelmia, sillä se oli muutamalle tietokoneelle asennettu työpöytäsovellus. Observointinäkymän ja yhteenvedon osilta Moveatis-sovelluksen toimintaperiaatteet vastaavat edeltäjää. Lisäksi sovellukseen kehitettiin rajapinta JYU-tunnistautumista varten. Hallinta- ja pääkäyttäjäsivuilla käyttäjät voivat luoda omia tapahtumaryhmiä, tapahtumia ja kategoriaryhmiä. Sovellusta voi käyttää myös kirjautumatta, jolloin observointidatan tallentaminen on mahdollista ainoastaan asiakaslaitteelle tiedostoksi.

Jatkokehitykseen sovittiin täydentäviä ominaisuuksia, kuten esimerkiksi observointidatan tuonti sovellukseen ja vaihtoehtoinen kirjautumistapa. Vaatimusten ainoa pakolliseksi määritelty, mutta toteuttamaton ominaisuus oli kategoriaryhmän kategorioiden valinta ei-aktiiviseksi observointikohtaisesti. Toteutusratkaisuna päädyttiin kategorian poistoon kokonaan, ja valintamahdollisuus sovittiin tilaajan kanssa jatkokehitykseen. Ominaisuuksien täydentämisen lisäksi sovelluksen käytettävyyttä on mahdollista parantaa dialogeja vähentämällä ja toimintalogiikkaa yhdenmukaistamalla.

## Lähteet

- [1] Joonas Konki, Anu Koskela, Mikko Kuhno, Henrik Paananen ja Atte Rätty, ”Kepler-projekti, sovellusraportti”, Jyväskylän yliopisto, tietotekniikan laitos, 2015.
- [2] Jarmo Juujärvi, Sami Kallio, Kai Korhonen, Juha Moisio ja Ilari Paananen, ”Moveatis-projekti, projektiraportti”, Jyväskylän yliopisto, tietotekniikan laitos, 2016.
- [3] Jarmo Juujärvi, Sami Kallio, Kai Korhonen, Juha Moisio ja Ilari Paananen, "Moveatis Project, Moveatis Java Class Documentation", University of Jyväskylä, Department of Mathematical Information Technology, 2016.
- [4] Jarmo Juujärvi, Sami Kallio, Kai Korhonen, Juha Moisio ja Ilari Paananen, "Moveatis Project, Moveatis JavaScript Class Documentation", University of Jyväskylä, Department of Mathematical Information Technology, 2016.
- [5] Kai Korhonen ja Juha Moisio, ”Moveatis-projekti, vaatimusmäärittely”, Jyväskylän yliopisto, tietotekniikan laitos, 2016.
- [6] Sami Kallio, ”Moveatis-projekti, tietokantakuvaus”, Jyväskylän yliopisto, tietotekniikan laitos, 2016.
- [7] Kai Korhonen, ”Moveatis-projekti, käytettävyydestauksen muistio”, Jyväskylän yliopisto, tietotekniikan laitos, 2016.
- [8] Kai Korhonen, ”Moveatis-projekti, Moveatis-käytettävyysskysely”, Jyväskylän yliopisto, tietotekniikan laitos, 2016.
- [9] Kai Korhonen, ”Moveatis-projekti, Moveatis järjestelmätestausraportti 1”, Jyväskylän yliopisto, tietotekniikan laitos, 2016.
- [10] Kai Korhonen, ”Moveatis-projekti, Moveatis järjestelmätestausraportti 2 Iphone”, Jyväskylän yliopisto, tietotekniikan laitos, 2016.



- [11] Kai Korhonen, ”Moveatis-projekti, Moveatis järjestelmätestausraportti 2 PC”, Jyväskylän yliopisto, tietotekniikan laitos, 2016.