

Moveo-sovellusprojekti

Karoliina Lappalainen

Tuomas Moisio

Visa Nykänen

Petra Puumala

Moveatis-sovellusprojekti

Jarmo Juujärvi

Sami Kallio

Kai Korhonen

Juha Moisio

Ilari Paananen



Sovellusraportti

Julkinen

Versio 1.0.0

6.6.2019

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.20__		
Tilaaaja	__.__.20__		
Ohjaaja	__.__.20__		

Tietoja dokumentista

Tekijät:

Karoliina Lappalainen (KL)	karoliina.a.lappalainen@student.jyu.fi	040-7719392
Tuomas Moisio (TM)	tuomas.s.moisio@student.jyu.fi	040-8449568
Visa Nykänen (VN)	visa.k.nykanen@student.jyu.fi	044-0151096
Jarmo Juujärvi	jarmojuujarvi@gmail.com	050-3651136
Kai Korhonen	kaivkorhonen@gmail.com	045-6362355
Sami Kallio	sami.m.j.kallio@student.jyu.fi	046-6418125
Juha Moisio	juha.pa.moisio@student.jyu.fi	040-7314776
Ilari Paananen	ilari.k.paananen@student.jyu.fi	044-9746743

Dokumentin nimi: Moveo-projekti, Sovellusraportti

Sivumäärä: 58

Tiedosto: Moveo_Sovellusraportti_1.0.0.docx

Tiivistelmä: Moveo-sovellusprojekti kehitti Moveatis-sovellukseen lisäosana opettajan palautteen luokittelutyökalun Jyväskylän yliopiston liikuntatieteelliselle tiedekunnalle. Sovellusraportti kuvaa Moveatis- ja Moveo-projekteissa kehitetyn tietojärjestelmän käyttöliittymää ja kokonaisrakennetta, tavoitteiden toteutumista, tietokannan rakennetta sekä jatkokehitysideoita.

Avainsanat: Jatkokehitys, kokonaisrakenne, komponentit, käyttöliittymä, ohjeet, sovellus, tavoitteet, tietojärjestelmä, tietokanta, toteutusratkaisut, ylläpito.

Muutoshistoria

Versio	Päivä	Muutokset	Tekijä
0.0.1	14.5.2019	Sovellusraportin laatiminen aloitettiin. Luku 2 kirjoitettiin.	KL
0.0.1	15.5.2019	Luvun 3 kirjoittaminen aloitettiin.	TM
0.0.2	16.5.2019	Luvun 3 kirjoittamista jatkettiin.	TM
0.0.3	16.5.2019	Lukujen 1 ja 4–8 kirjoittaminen aloitettiin.	VN
0.0.4	17.5.2019	Luvun 3 kirjoittamista jatkettiin.	TM
0.0.5	17.5.2019	Lukujen 4–8 kirjoittamista jatkettiin.	VN
0.0.6	19.5.2019	Lukujen 4–8 kirjoittamista jatkettiin.	VN
0.0.7	20.5.2019	Kieli- ja ulkoasua korjattiin.	KL
0.1.0	23.5.2019	Tehtiin kieli- ja ulkoasukorjauksia palautteen mukaan.	VN
0.1.1	24.5.2019	Muokattiin lukua 3 palautteen mukaan.	TM
0.1.2	25.5.2019	Kirjoitettiin jatkokehitysideat ja tehtiin muutamia muita pienempiä lisäyksiä palautteen mukaan.	VN
0.1.3	26.5.2019	Viimeisteltiin ulkoasua.	VN
0.2.1	29.5.2019	Tehtiin kieli- ja ulkoasukorjauksia palautteen mukaan.	TM
0.3.0	29.5.2019	Tehtiin kieli- ja ulkoasukorjauksia palautteen mukaan.	VN
0.4.0	1.6.2019	Tehtiin kieli- ja ulkoasukorjauksia palautteen mukaan.	TM
0.4.1	3.6.2019	Tehtiin kieli- ja ulkoasukorjauksia Visan palautteen mukaan.	TM
1.0.0	6.6.2019	Muutettiin versionumero ja tehtiin viimeiset ulkoasupäivitykset.	TM

Tietoja projektista

Moveo-projekti kehitti Moveatis-sovellukseen opettajan palautteen luokittelutyökalun Jyväskylän yliopiston liikuntatieteelliselle tiedekunnalle, jossa sitä käytetään muun muassa opettajaopiskelijoiden toiminnan tarkkailuun opetustilanteessa. Ohjelma kehitettiin WWW-sovelluksena Java-ohjelmointikielellä.

Tekijät:

Karoliina Lappalainen (KL)	<code>karoliina.a.lappalainen@student.jyu.fi</code>	040-7719392
Tuomas Moisio (TM)	<code>tuomas.s.moisio@student.jyu.fi</code>	040-8449568
Visa Nykänen (VN)	<code>visa.k.nykanen@student.jyu.fi</code>	044-0151096
Petra Puumala (PP)	<code>petra.p.puumala@student.jyu.fi</code>	040-0741288

Tilaaajan edustajat:

Pilvikki Heikinaro-Johansson	<code>pilvikki.heikinaro-johansson@jyu.fi</code>	040-7243352
Nelli Lyyra	<code>nelli.lyyra@jyu.fi</code>	040-8427624

Ohjaajat:

Jani Saareks	<code>jani.j.saareks@student.jyu.fi</code>	040-0859341
Jukka-Pekka Santanen	<code>santanen@mit.jyu.fi</code>	050-5504666

Yhteystiedot:

- Sähköpostilistat `moveo@korppi.fi`,
`moveo_opetus@korppi.fi`

- **Sähköpostiarkistot** [https://korppi.jyu.fi/kotka/servlet/list-archive/moveo,](https://korppi.jyu.fi/kotka/servlet/list-archive/moveo)
https://korppi.jyu.fi/kotka/servlet/list-archive/moveo_opetus
- **WWW-sivusto** <http://sovellusprojektit.it.jyu.fi/moveo/>

Sisältö

1 Johdanto.....	1
2 Termit.....	2
2.1 Kohdealueen termejä.....	2
2.2 Sovelluksen käyttäjäroolit.....	3
2.3 Käyttöliittymän termejä.....	4
3 Sovelluksen käyttöliittymä ja käyttö.....	6
3.1 Sovelluksen käyttöliittymän rakenne ja näkymäkartta	6
3.2 Sovelluksen etusivu	8
3.3 Komentovalikon toiminnot (Moveatis)	8
3.4 Kirjautumisvalinnat ja niiden vaikutus käyttöön (Moveatis).....	9
3.5 Menetelmätyypin valintasivu (Moveo).....	10
3.6 Hallintasivun tapahtumat (Moveatis)	11
3.7 Hallintasivun kategoriaryhmät ja kategoriat (Moveatis).....	14
3.8 Observoinnissa käytettävien kategorioiden valinta (Moveatis)	17
3.9 Observoinnin suorittaminen (Moveatis)	19
3.10 Observoinnin yhteenvetosivu (Moveatis).....	20
3.11 Observoinnin kirjausten ja yhteenvedon tallennus (Moveatis)	22
3.12 Palauteanalyysissä käytettävien kategorioiden valinta (Moveo)	23
3.13 Palauteanalyysin suorittaminen (Moveo)	25
3.14 Palauteanalyysin raporttisivu (Moveo).....	27
3.15 Palauteanalyysin yhteenvetosivu (Moveo-projekti)	28
3.16 Palauteanalyysin kirjausten ja yhteenvedon tallennus.....	31
4 Sovelluksen rakenne ja toteutusratkaisut.....	33
4.1 Poikkeukset Moveatis-projektin toteutusratkaisuihin (Moveo)	33
4.2 Kehitystyökalut	33
4.3 Sivujen toiminnallisuus	34
4.4 Toimintalogiikka	35

4.5 Sovelluspalvelin.....	36
5 Sovelluksen tietokanta	37
5.1 Käyttäjiä ja rooleja koskevat taulut (Moveatis).....	38
5.2 Kategorioihin, tapahtumiin ja observointeihin liittyvät taulut (Moveatis).....	40
5.3 Taulujen käyttötarkoitukset (Moveatis)	43
5.4 Palauteanalyysiin liittyvät taulut (Moveo).....	45
5.5 Palauteanalyysiin liittyvien taulujen käyttötarkoitus (Moveo).....	46
5.6 Palauteanalyysiin liittyvien taulujen ER-kaaviot (Moveo).....	46
6 Moveo-projektin sovelluksen tavoitteiden toteutuminen	49
6.1 Moveo-projektissa toteutetut vaatimukset	49
6.2 Jatkokehitykseen sovitut vaatimukset	50
7 Ohjeita ylläpitoon ja jatkokehitykseen	51
7.1 Sovelluspalvelimen asennusohjeet	51
7.2 Tietokantapalvelimen asennusohjeet	52
7.3 Tietokantataulujen automaattinen muodostus.....	52
7.4 Sovelluksen toimivuus palvelin- ja selainohjelmistoilla.....	53
7.5 Ohjeita jatkokehittäjälle	53
7.6 Ideoita jatkokehitykseen	55
7.7 Ohjeita tuotantopalvelimen ylläpitoon.....	56
8 Yhteenveto	57
Lähteet	58

1 Johdanto

Jyväskylän yliopiston liikuntatieteellisessä tiedekunnassa 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 vuonna 2016 opetustapahtuman tarkkailuun WWW-sovelluksen nimeltään Moveatis, joka perustuu kerta- ja kestorekisteröintiin. Moveo-projekti jatkoi Moveatis-sovelluksen kehitystä ja lisäsi siihen palauteanalyysin toiminnallisuuden, jolla voidaan opetustilanteesta kirjata opettajan antamaa palautetta opetustapahtumasta. Sovellukseen on mahdollista määritellä opettajan palautteesta seurattavia kategoriaryhmiä ja niihin kategorioita. Sovellusta käytetään pääosin edellä mainitulla kurssilla, mutta sitä voidaan hyödyntää myös muilla kursseilla sekä tutkimuskäytössä vuorovaikutustilanteita arvioidessa.

Sovellusraportti kuvaa toteutetun sovelluksen toteutusratkaisuja ja käyttöliittymää Moveo- sekä Moveatis-projektien toteuttamien ominaisuuksien suhteen. Asetettujen tavoitteiden toteutumista sekä olennaisimpia puutteita, virheitä ja jatkokehitysideoita käydään Moveo-projektin toteuttamien ominaisuuksien osalta. Raportissa mainitaan alalukujen otsikoissa suluissa, koskeeko alaluku Moveo- vai Moveatis-projektin kehitettäviä ominaisuuksia. Sovellusraporttia kirjoitettaessa on hyödynnetty Moveo-projektin vaatimusmäärittelydokumenttia [1], projektiraporttia [2], järjestelmätestausraportteja [3], [4] ja [8] sekä asennus- ja ylläpito-ohjeita [5] ja [6]. Lisäksi sovellusraportti nojaa hyvin suurelta osin Moveatis-projektin laatimaan sovellusraporttiin [7].

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 ja tauluja. 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- ja Moveo-projektien kehittämiin ominaisuuksiin liittyviä termejä, jotka on jaoteltu erillisiin alalukuihin kohdealueen, käyttäjäroolien ja käyttöliittymän mukaan.

2.1 Kohdealueen termejä

Analyysi (analyzation)	on yksittäisen opetustapahtuman purkaminen havaintojen pohjalta <i>kirjauksiksi</i> .
Analyysidata (analyzation data)	on yhdestä tapahtumasta kirjausten avulla kerätty tieto. Analyysidata voidaan lähettää sähköpostiin, tallentaa sovelluksen tietokantaan tai tallentaa laitteelle CSV-tiedostona.
Analyysitulokset (analyzation results)	on <i>analyysidatasta</i> muodostettava yhteenveto, joka esitetään yhteenvetosivulla. Analyysitulokset voidaan lähettää sähköpostiin, tallentaa sovelluksen tietokantaan tai tallentaa laitteelle CSV-tiedostona.
Kategoria (category)	on <i>kirjaukseen</i> liitetty palautteen laatu, kuten yleinen tai spesifi.
Kategoriaryhmä (category set)	on <i>kategorioista</i> koostuva ryhmä. <i>Kategoriaryhmiä</i> voi ottaa käyttöön <i>tapahtumaryhmän avaimella</i> tai muodostamalla ne <i>hallintasiivulla</i> .
Kirjaus (recording)	on yksi <i>observoinnin</i> tai <i>analyysin</i> aikana tehty merkintä. <i>Kirjauksella</i> on ajankohta ja <i>kategoria</i> , sekä siihen voi liittää tekstikommentteja.
Observointi (observation)	on yksittäisen <i>tapahtuman</i> tarkkailukerta, jonka aikana tehdyistä <i>kirjauksista</i> muodostetaan <i>observointidata</i> .
Observointidata (observation data)	on yhdestä <i>observoinnista kirjausten</i> avulla kerätty tieto. <i>Observointidata</i> voidaan lähettää sähköpostiin, tallentaa laitteelle ja

tallentaa sovelluksen tietokantaan.

Observointitulokset
(**observation results**) on *observointidatasta* muodostettava yhteenveto, joka esitetään yhteenvetosivulla.

Systemaattiset
observointimenetelmät
(**methods of systematic**
observation) ovat keinoja tarkastella ja havainnoida observoinnin *tapahtumaa* ennalta asetettujen *kategorioiden* mukaan.

Systemaattiset analyso-
intimenetelmät
(**methods of systematic**
analyzation) ovat keinoja tarkastella ja havainnoida analyysin *tapahtumaa* ennalta asetettujen *kategorioiden* mukaan.

Tapahtuma
(**event**) on *observoinnin* tai *analyysin* kohteena oleva vuorovaikutustilanne. Esimerkiksi kurssin *Tutkiva opettaja* jokainen opetustapahtuma on sovelluksessa yksi *tapahtuma*.

2.2 Sovelluksen käyttäjäroolit

Julkinen käyttäjä
(**public user**) käyttää sovellusta kirjautumatta JYU-tunnuksilla. Hän voi käyttää sovellusta antamalla *kategoriat* itse tai hyödyntämällä *ryhmävaimella* saatavia *kategoriaryhmiä*. Hän voi tallentaa observoinnin tulokset käyttämäänsä laitteeseen.

Kirjautunut käyttäjä
(**logged-in user**) on kirjautunut sovellukseen JYU-tunnuksilla. Hän voi julkisen käyttäjän toimintojen lisäksi luoda ja hallita omistamiaan *tapahtumia*, sekä lähettää observoinnin tulokset sähköpostiin tai tallentaa ne sovelluksen tietokantaan.

Pääkäyttäjä
(**superuser**) on kirjautunut sovellukseen JYU-tunnuksilla, joille on määritetty *pääkäyttäjän* oikeudet. *Pääkäyttäjille* näytetään edellä mainittujen *kirjau-*

tuneen käyttäjän toimintojen lisäksi pääkäyttäjän toiminnot, kuten pääkäyttöoikeuksien lisääminen ja poistaminen sekä kaikkien tapahtumaryhmien ja tapahtumien tarkastelu.

Tapahtuman omistaja (event owner) on *käyttäjä*, joka on luonut kyseisen *tapahtuman* tai jolle on annettu omistajan oikeudet *pääkäyttäjän* toimesta. *Tapahtumaa* voi *tapahtuman omistajien* lisäksi hallita *pääkäyttäjät*.

2.3 Käyttöliittymän termejä

Analysoinnin yhteenvetosivu (analysis summary) on sovelluksen sivu, jossa näytetään *analyysitulokset analyysin* aikana tehdyistä *kirjauksista*. Sivun tärkein sisältö on *analyysitulokset* ja niistä muodostettu *kaavio*.

Analysointityypin valintasivu (analysis selection page) on *julkisen käyttäjän* sivu, jossa käyttäjä valitsee, haluaako hän *analysoida* vai *observoida*.

Etusivu (front page) on sovelluksen aloitussivu, jossa voi kirjautua sisään, syöttää *ryhmäavaimen* tai siirtyä käyttämään sovellusta kirjautumatta.

Hallintasivu (control page) on sivu, jolla *kirjautunut käyttäjä* voi muun muassa luoda ja hallita omia *tapahtumia*, sekä selata tallentamia *analyysidatoja*.

Kaavio / diagrammi (figure) on *analyysin* ja *observoinnin kirjauksista* muodostettu visuaalinen kuvaaja, joka esitetään *yhteenvetosivulla*.

Observoinnin yhteenvetosivu on sovelluksen sivu, jossa näytetään *observointitulokset observoinnin* aikana tehdyistä *kirjauksista*. Sivun tärkein sisältö on

(observation summary)	<i>observointitulokset</i> ja niistä muodostettu kaavio.
Palauteanalyysisivu (feedback recording sheet)	on sovelluksen sivu, jossa kirjataan palaute ja valitaan palautteen mukaiset <i>kategoriat</i> . Sivun tärkein sisältö ovat kategoriaryhmittäiset painikkeet eri <i>kategorioiden kirjauksille</i> .
Palautekirjaus (feedback recording)	on <i>analysoitavan</i> palautteen kirjaus tekstiksi.
Raporttisivu (report page)	on sivu, jossa näkyvät kootut tiedot <i>palauteanalyysisivun kirjauksista</i> . Raportista pääsee tarkastelemaan <i>kirjauksia</i> yksityiskohtaisesti.
Valikko (menu)	on sovelluksen oikeassa ylälaudassa sijaitseva painike, joka sisältää pääsyn asetusten hallintaan, <i>hallintasivulle</i> ja kielivalintaan.

3 Sovelluksen käyttöliittymä ja käyttö

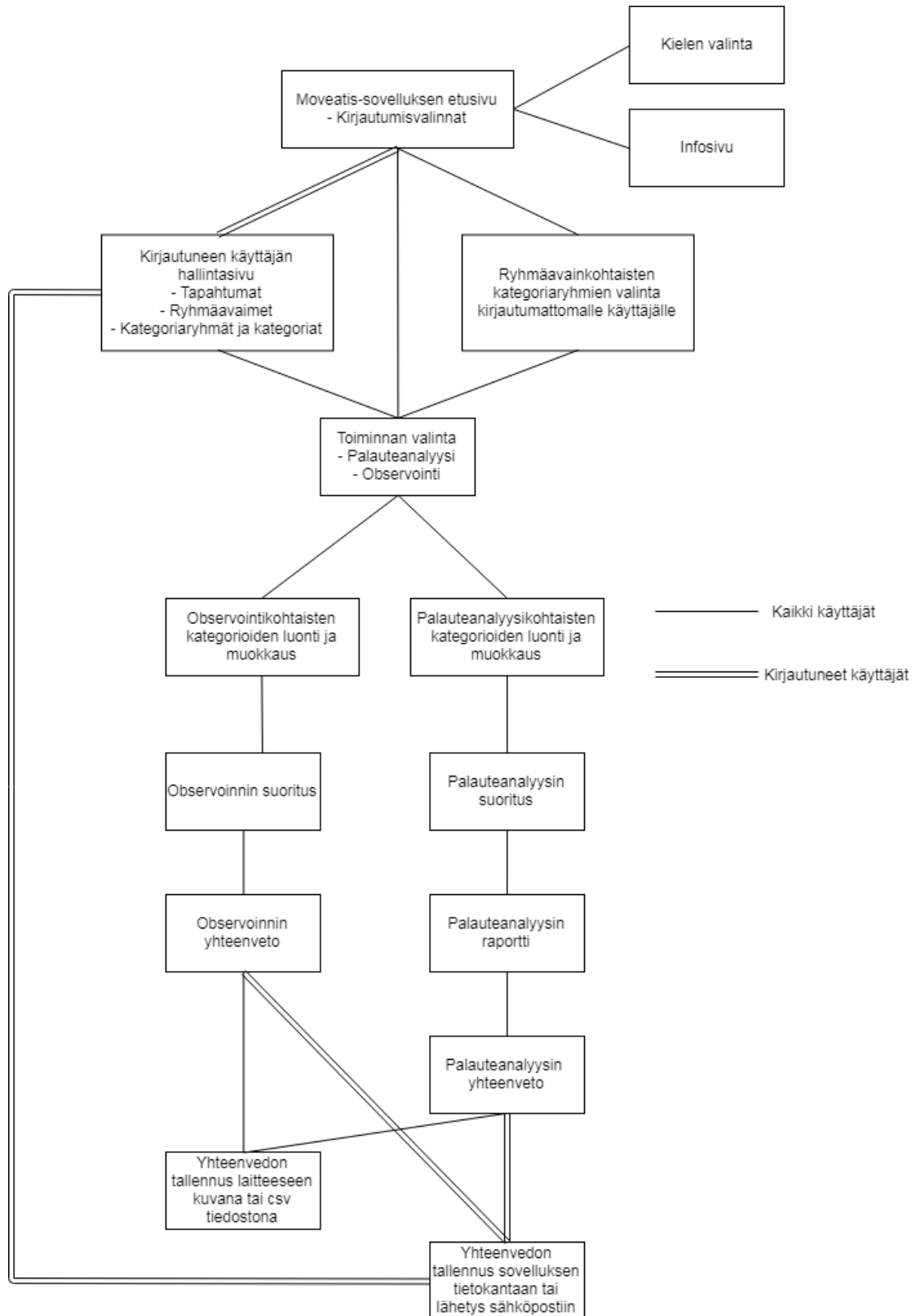
Luvussa esitellään sovelluksen käyttöliittymä ja sovelluksen käyttöä. Sovellus on toteutettu sekä suomeksi että englanniksi sisällöltään identtisenä. Raportissa käytetään esimerkkinä suomenkielistä kieliversiota.

3.1 Sovelluksen käyttöliittymän rakenne ja näkymäkartta

Sovelluksen käyttöliittymän sivut ovat kolmiosaisia (katso kuva 2). Ylinnä olevasta yläotsakkeesta löytyvät valikkopainikkeet. Alinna olevassa alaotsakkeessa sijaitsee tekijänoikeuslauseke, joka toimii myös linkkinä infosivulle. 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 1 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 kyseistä sivua.



Kuva 1: Moveo-sovelluksen näkymien riippuvuussuhteet.

3.2 Sovelluksen etusivu

Etusivu koostuu kirjautumisvalintapainikkeista sekä tervetulo- ja pikaohjetekstistä kuvan 2 mukaisesti. Kirjautumispainikkeilla valitaan sovelluksen käyttötapa, joita ovat *ryhmävain*, *kirjautuminen Jyväskylän yliopiston käyttäjätunnuksilla* ja *julkinen käyttö*. Sovelluksen käyttötavat on kuvattu tarkemmin luvussa 3.4.



Kuva 2: Etusivu.

3.3 Komentovalikon toiminnot (Moveatis)

Yläotsakkeen muodostavat värillinen vaakapalkki, sovelluksen nimi ja komentovalikko (katso kuva 3). 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 3: Yläpalkki ja komentovalikko kirjautuneen käyttäjän tapauksessa.

Kirjautuneen käyttäjän komentovalikon kautta (katso kuva 4) käyttäjä voi siirtyä hallintasivulle tai kirjautua ulos sovelluksesta. Jos käyttäjä ei ole kirjautunut sovellukseen, kirjautuneen käyttäjän valikko ei ole näkyvässä. Kirjautuneen käyttäjän nimi näkyy myös osana komentovalikkoa.



Kuva 4: Kirjautuneen käyttäjän valikko.

Asetusvalikosta (katso kuva 5) 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 5: Asetusvalikko.

3.4 Kirjautumisvalinnat ja niiden vaikutus käyttöön (Moveatis)

Etusivun (katso kuva 2) kirjautumisvalikossa on kuvan 6 mukaisesti valittavissa Moveatis-sovellukselle kolme eri käyttötapaa: ryhmäavain, JYU-kirjautuminen ja julkinen käyttö.



Kuva 6: 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 tai palauteanalyysin tulokset paikallisesti omalle päätelaitteelleen yhteenvetosivulla kuvana tai CSV-tiedostomuodossa.

JYU-kirjautuminen tapahtuu Jyväskylän yliopiston käyttäjätunnuksilla antaen sovelluksen käyttäjälle oikeuden luoda tapahtumia, kategoriaryhmiä, kategorioita ja ryhmäavaimia. Kirjautunut käyttäjä voi paikallisen tallennuksen lisäksi lähettää valitsemansa analyysityypin tulokset syöttämäänsä sähköpostiosoitteeseen tai tallentaa ne sovelluksen tietokantaan. Tietokantaan tallennetut tulokset ovat myös myöhemmin katseltavissa. Käyttäjä myös näkee kaikki luomallaan ryhmäavaimella suoritettut observoinnit ja palauteanalyysit 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 palauteanalyysin tai observoinnin tiedot ainoastaan paikallisesti omalle päätelaitteelle joko kuvana tai CSV-tiedostomuodossa.

3.5 Menetelmätyypin valintasivu (Moveo)

Painettuaan *Julkinen käyttö* -painiketta (katso kuva 6) tai kirjaututtaan sovellukseen ryhmäavaimella, käyttäjä ohjataan toiminnan valintasivulle (katso kuva 7). Siinä käyttäjä suorittaa valinnan, haluaako hän käyttää menetelmänä observointia vai palauteanalyysia. Sivun sisältää lisäksi käyttäjälle ohjetekstit palauteanalyysin ja observoinnin suorittamiseen.

Moveatis ⚙️

Palauteanalyysi


Observointi

Moveatis 2.0 sisältää kaksi erilaista opetustilanteen analysointimenetelmää: puheen luokitteluun perustuvan *Palauteanalyysin* sekä toiminnan kestoa tai esiintymiskertoja mittaavan *Observoinnin*.

Palauteanalyysi on menetelmä, jonka avulla puhetta voidaan luokitella ennalta määriteltyihin kategorioihin. Sovelluksessa puhe kirjataan tekstilaatikkoon, jonka jälkeen kirjoitettu teksti luokitellaan. Palauteanalyysia on mahdollista käyttää myös kirjaamatta tekstiä, jolloin analysoija valitsee kuulemansa perusteella puhetta tai palautetta kuvaavan kategorian.

Observointi on menetelmä, jonka avulla oppitunnilla esiintyviä tilanteita, tapahtumia tai käyttäytymistä voidaan mitata. Mittayksikkönä voi olla joko aika tai esiintymiskertojen lukumäärä. Observointi-menetelmällä saat yhteenvetona aikajanan, josta toiminnan tai tapahtumien ajallinen sijoittuminen tarkkailtavalla jaksolla on nähtävissä.

[Lisätietoa sovelluksesta löytyy Moveatis 2.0 käyttäjän oppaasta.](#)



© 2016 Moveatis-projekti, 2019 Moveo-projekti

Kuva 7: Menetelmätyypin valintasivu.

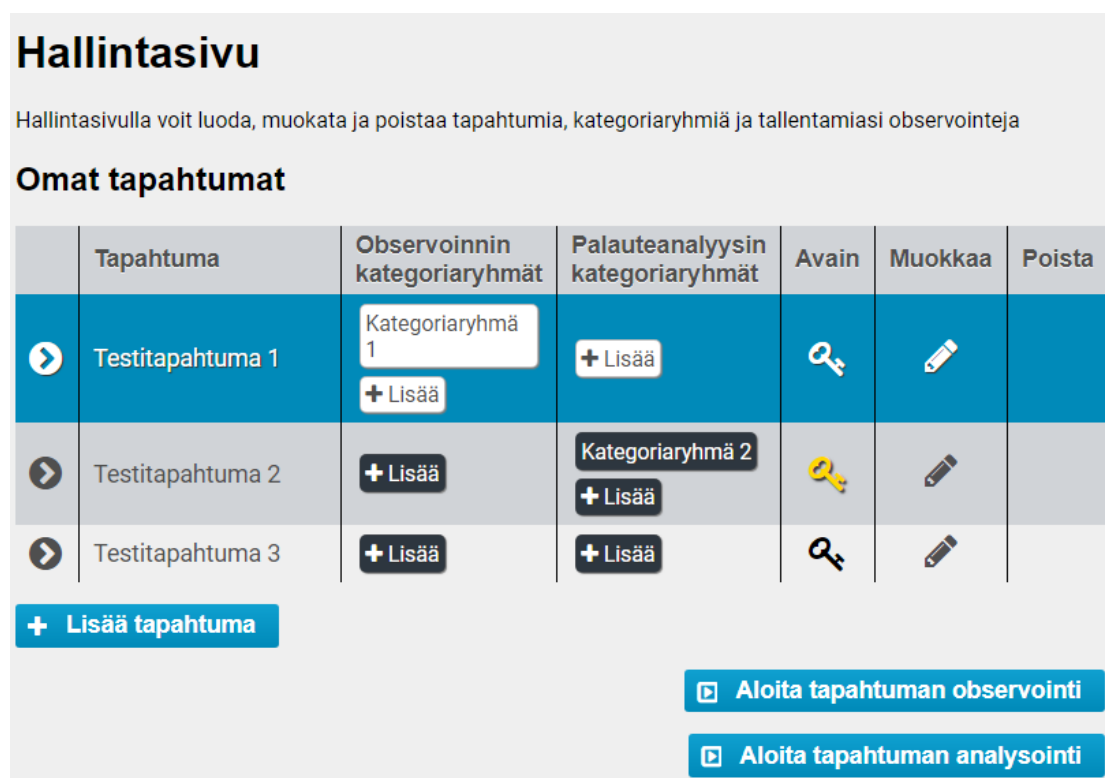
3.6 Hallintasivun tapahtumat (Moveatis)

Kirjaututtuaan sovellukseen JYU-tunnuksilla käyttäjä ohjataan automaattisesti kuvissa 8 ja 9 esitetyille hallintasivulle. Hallintasivulla 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.



Kuva 8: Hallintasivu ilman olemassa olevia tapahtumia.



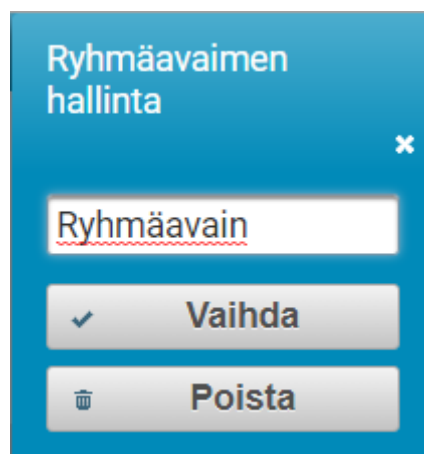
Kuva 9: Hallintasivu, jossa on listattuna luotuja tapahtumia.

Tapahtumataulukossa (katso kuva 9) kategoriaryhmän nimen edessä olevaa nuolta painamalla käyttäjä saa näkyviin tapahtumalle jo suoritettut observoinnit ja palauteanalyysit, 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

lisäyspainikkeilla. Tapahtuman ollessa aktiivisena, voi käyttäjä siirtyä uuteen observointiin *Aloita tapahtuman observointi* -painiketta painamalla tai uuteen palauteanalyysiin *Aloita tapahtuman analysointi* -painiketta painamalla.

Rekisteröitynyt käyttäjä voi hallintasivulla myös asettaa tapahtumalle ryhmävaimen harmaasta avainkuvakkeesta (katso kuva 9) avautuvasta dialogista (katso kuva 10). Jos tapahtumalle on jo asetettu ryhmävain, tapahtuman avainkuvake näkyy keltaisena (katso kuva 9). Ryhmävainta käyttämällä voi kirjautumatonkin käyttäjä saada käyttöönsä tapahtumaan ennalta määritellyt kategoriat. Kirjautuneen käyttäjän suorittamat observoinnit ja palauteanalyysit ovat nähtävissä myös ryhmävaimen luojan hallintäkymässä, jos käyttäjä käyttää ryhmävainta ja tallentaa observoinnista tai palauteanalyysistä saadun datan sovelluksen tietokantaan.



Kuva 10: Ryhmävaimen syöttöikkuna.

Mikäli kirjautunut käyttäjä suorittaa observoinnin tai palauteanalyysin suoraan etusivulta ryhmävaimella tai julkisena käyttäjänä tallentaen suorittamansa analyysit, tulevat ne näkyviin hallintasivulle omien tapahtumien alle otsikolla *Muut suoritetut observoinnit* tai *Muut suoritetut palauteanalyysit* kuvan 11 mukaisesti.

Hallintasivu

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

Omat tapahtumat

Tapahtuma	Observoinnin kategoriaryhmät	Palauteanalyysin kategoriaryhmät	Avain	Muokkaa	Poista
➔ Testitapahtuma 1	+ Lisää	+ Lisää	🔑	✎	

+ Lisää tapahtuma

Aloita tapahtuman observointi

Aloita tapahtuman analysointi

Muut suoritettut observoinnit

- Näytä/Piilota

Nimi	Tapahtuma
Observointi - 14.5.2019 15:50	

Muut suoritettut analysoinnit

- Näytä/Piilota

Nimi	Tapahtuma
Analization - Tue May 14 15:48:53 EEST 2019	

Kuva 11: Tapahtumiin liittymättömät tallennetut observoinnit ja palauteanalyysit.

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

3.7 Hallintasivun kategoriaryhmät ja kategoriat (Moveatis)

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




Moveo-projekti kehitti kuvan 11 näkymään *Palauteanalyysin kategoriaryhmät* -sarakkeen, jonka *Lisää*-painikkeesta (katso kuva 11) avautuvassa dialogissa (katso kuva 13) luodaan tapahtumakohtaiset kategoriaryhmät ja niiden kategoriat palauteanalyysille. Kategoriaryhmän kuvaus ei ole pakollinen tieto. Kategorioita lisätään *Lisää kategoria*

-painikkeella ja poistetaan *Poista kategoria* -painikkeella. Kategorian muokkaaminen tapahtuu kynäkuvakkeesta.

Kategoriaryhmän tiedot

Kategoriaryhmän nimi
Testikategoriaryhmä

Kategoriaryhmän kuvaus
Testikategoriaryhmä

Kategoria	Tyyppi	
Kategoria 1	Kesto	
Kategoria 2	Kerta	
Kategoria 3	Kesto	

+ Lisää kategoria **- Poista kategoria**

✓ Tallenna **× Peruuta**

Kuva 12: Observoinnin kategoriaryhmien ja kategorioiden luontidialogi.

Kategoriaryhmän tiedot

Kategoriaryhmän nimi
Testikategoriaryhmä

Kategoriaryhmän kuvaus
Testikategoriaryhmä

Kategoria

Testikategoria 1

Testikategoria 2

Testikategoria 3

+ Lisää kategoria - Poista kategoria

✓ Tallenna × Peruuta

Kuva 13: Palauteanalyysin kategoriaryhmien ja kategorioiden luontidialogi.

Kategoriaryhmien lisäämisen jälkeen, käyttäjä ohjataan lisäämään tapahtumakohtainen kategoriaryhmä tai -ryhmiä ja vähintään yksi kategoria jokaiseen käytettävään ryhmään (katso kuva 14 ja 15). Nämä jälkikäteen lisätyt kategoriaryhmät ja kategoriat eivät ole käytettävissä muissa observoinneissa tai palauteanalyysissä.

Tapahtuman 'Testitapahtuma 1' observoinnin kategoriat

Observointiin lisätyt kategoriaryhmät

Observointiin ei ole lisätty vielä yhtään kategoriaryhmää.

Lisää kategoriaryhmiä observointiin

Uusi kategoriaryhmä:

Kuva 14: Observoinnin kategoriaryhmän lisäyssivu.

Tapahtuman 'Testitapahtuma 1' palauteanalyysin kategoriat

Palauteanalyysiin lisätyt kategoriaryhmät

Palauteanalyysiin ei ole lisätty vielä yhtään kategoriaryhmää.

Lisää kategoriaryhmiä palauteanalyysiin

Uusi kategoriaryhmä:

Kuva 15: Palauteanalyysin kategoriaryhmän lisäyssivu.

3.8 Observoinnissa käytettävien kategorioiden valinta (Moveatis)

Yksittäisessä observoinnissa on käyttäjäroolista riippuen käytettävissä erilaisia kategoriaryhmä- ja kategoriavalintoja. Kirjautuneella käyttäjällä on käytettävissään kaikki luomansa kategoriaryhmät ja kyseiseen observointiin määrittämänsä ryhmät. Ryhmäavaimella 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 16), jolloin ryhmät ovat myös jatkossa käytettävissä. Observointikohtaisten muutosten teko on myös mahdollista.

Tapahtuman 'Testitapahtuma 1' observoinnin kategoriat

Observointiin lisätyt kategoriaryhmät

Kategoriaryhmä2 ✕
Kerta ✕
Kesto ✕
Kerta ✕
Kesto ✕
+ Lisää uusi kategoria

Kategoriaryhmä1 ✕
Kerta ✕
Kesto ✕
Kerta ✕
+ Lisää uusi kategoria

Lisää kategoriaryhmiä observointiin

Uusi kategoriaryhmä: **Lisää**

Omat kategoriaryhmät: **Lisää**

Oletuskategoriaryhmät: **Lisää**

Jatka observointiin

Kuva 16: Observoinnin tapahtuman kategoriaryhmien valintasivu.

Ryhmäavaimella kirjautuneella käyttäjällä on käytettävissään avaimen sidotut kategoriaryhmät. Käyttäjä voi valita yhden tai useamman näistä ryhmistä ja myös lisätä omia observointikohtaisia kategorioita.

Julkisella käyttäjällä on käytettävissään vain itse määrittelemänsä observaatiokohtaiset kategoriaryhmät ja kategoriat kuvan 17 mukaisesti. Niitä ei voi tallentaa sovelluksen tietokantaan.

Observoinnin kategoriat

Observointiin lisätyt kategoriaryhmät

Observointiin ei ole lisätty vielä yhtään kategoriaryhmää.

Lisää kategoriaryhmiä observointiin

Uusi kategoriaryhmä: **Lisää**

Jatka observointiin

Kuva 17: Julkisen käyttäjän kategorialuontisivu.

3.9 Observoinnin suorittaminen (Moveatis)

Observointinäkylässä on kuvien 18 ja 19 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 18: Observointinäkylä ennen aloitusta.

Observoinnin aikana aloitettujen kategorioiden rekisteröintipainikkeet muuttuvat punaisiksi kuvan 19 mukaisesti. Kestokategorioiden rekisteröinnit vaativat aloitus- ja lopetuspainallukset, mutta kertakategorioiden rekisteröinnit vain yhden painalluksen.

Kertakategoriapainikkeet eivät jää punaiseksi, vaan ainoastaan käyvät hetken punaisena.





Kuva 19: Observointinäkymä observoinnin aikana.

Observointi lopetetaan *Lopeta*-painikkeella, joka myös automaattisesti avaa kuvan 20 observoinnin yhteenvetosivun. Observoinnin lopetus on tilaajan toiveesta päätetty pitää peruuttamattomana toimenpiteenä, eli sille ei pyydetä käyttäjältä vahvistusta.

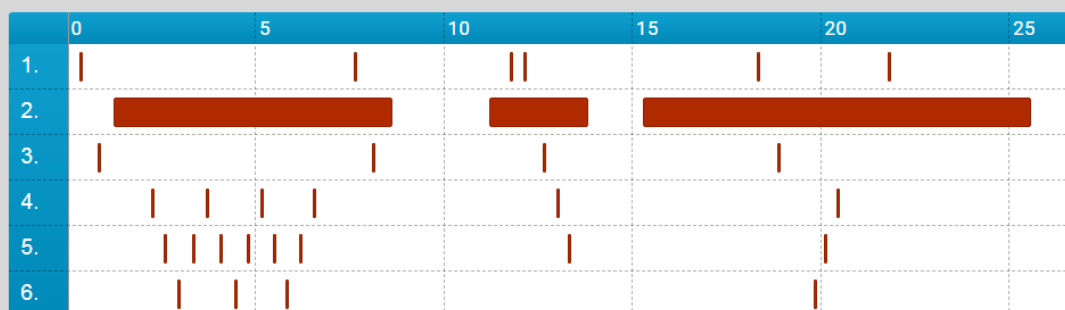
3.10 Observoinnin yhteenvetosivu (Moveatis)

Observoinnin yhteenvetosivulla on kuvan 20 mukaisesti nähtävillä suoritettujen observoinnin 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 kokonaiskesto.

Yhteenveto : Observointi - 15.5.2019 11:29


Kirjaukset alkaen 00:00:00  päättyen 00:00:25 

	Määrä (kpl)	Kesto
1. Katgoria 1.1	6 (19%)	0 s (0%)
2. Katgoria 2.1	3 (10%)	20 s (79%)
3. Katgoria 3.1	4 (13%)	0 s (0%)
4. Katgoria 1.2	6 (19%)	0 s (0%)
5. Katgoria 2.2	8 (26%)	0 s (0%)
6. Katgoria 3.2	4 (13%)	0 s (0%)
Yhteensä	31	25 s





Tallenna observointi

Lataa kuvana

Takaisin ylös 

Kuva 20: Observoinnin yhteenvetosivu.

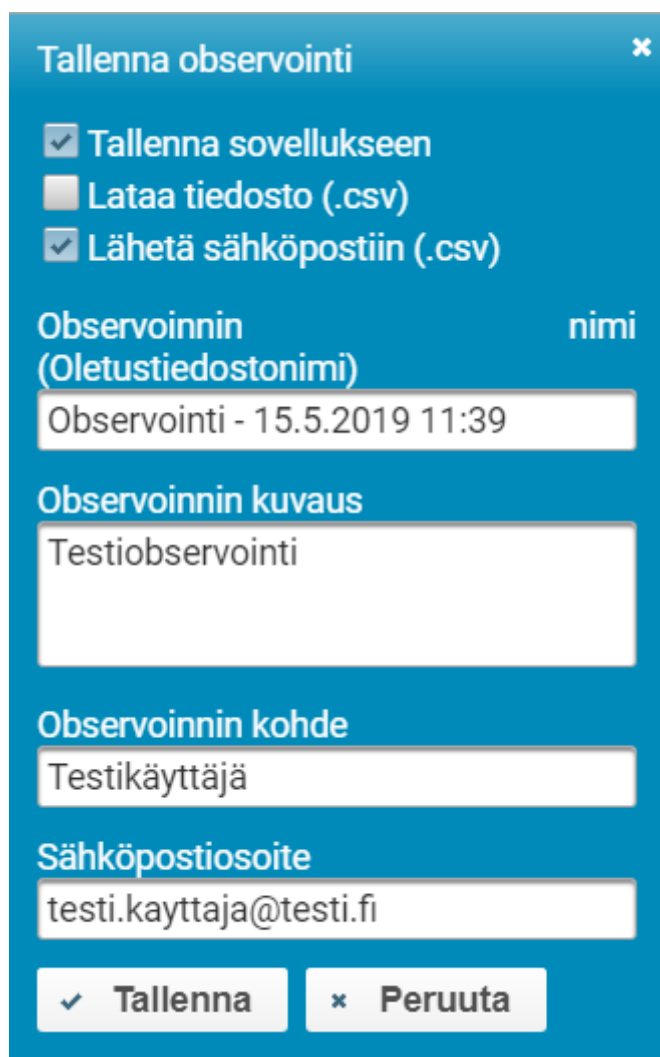
Kirjausten rajauskentillä (katso kuva 21) voidaan rajata syötteillä tai kellopainikkeista avautuvilla liukusäätimillä tarkasteltava observoinnin aikajänne. Tämä rajaus ei vaikuta tallennettavaan dataan, vaan se sisältää aina koko observoinnin aikajakson.

Kirjaukset alkaen 00:00:00  päättyen 00:00:25 

Kuva 21: Observoinnin yhteenvedon aikajänneen rajaustyökalurivi.

3.11 Observoinnin kirjausten ja yhteenvedon tallennus (Moveatis)

Observoinnin kirjaukset ja yhteenvedo voidaan tallentaa käyttäjäroolista riippuen. JYU-tunnistautuneella käyttäjällä on käytettävissään kuvan 23 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 - 15.5.2019 11:39

Observoinnin kuvaus
Testiobservointi

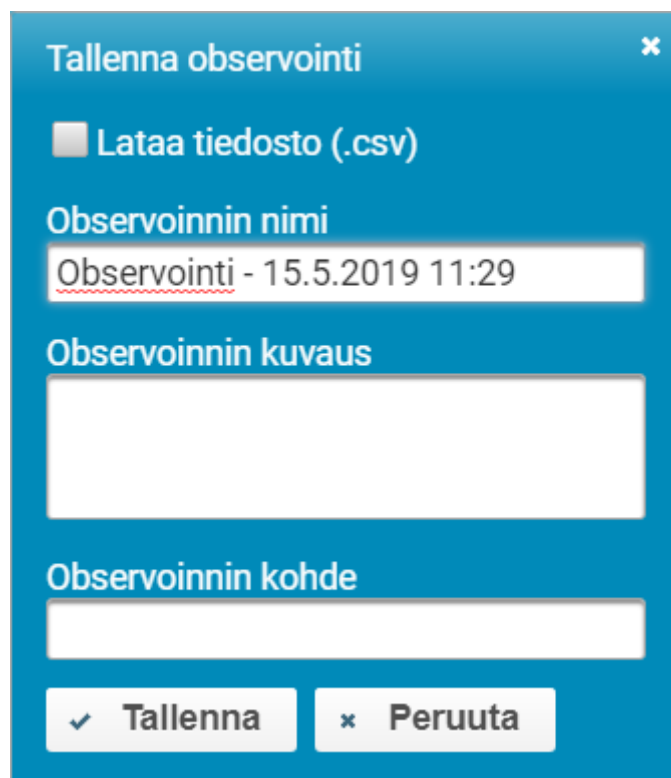
Observoinnin kohde
Testikäyttäjä

Sähköpostiosoite
testi.kayttaja@testi.fi

✓ Tallenna × Peruuta

Kuva 22: Kirjautuneen käyttäjän observoinnin tallennusikkuna.

Ryhmäavaimella kirjautunut tai julkinen käyttäjä voi tehdä ainoastaan paikallisen tallennuksen (katso kuva 23). Jotkin mobiililaitteet eivät tue CSV-tiedoston paikallista tallentamista. Moveo-projektissa havaittiin ainakin iPhonesta puuttuvan kyseinen tallennusmahdollisuus.



Kuva 23: Ryhmäavaimella kirjautuneen tai julkisen käyttäjän observoinnin tallennusikkuna.

Moveo-projekti kehitti toiminnon, jonka avulla observoinnin yhteenvedon voi halutessaan tallentaa myös kuvana painamalla kuvan 20 yhteenvetosivulta löytyvää *Lataa kuva* -painiketta. Tällöin käyttäjä saa png-tiedoston yhteenvetosivusta.

3.12 Palauteanalyysissä käytettävien kategorioiden valinta (Moveo)

Yksittäisessä palauteanalyysissä on käyttäjäroolista riippuen käytettävissä erilaisia kategoriaryhmä- ja kategoriavalintoja. Kirjautuneella käyttäjällä on käytettävissään kaikki luomansa kategoriaryhmät ja kyseiseen palauteanalyysiin määrittämänsä kategoriaryhmät. Ryhmäavaimella palauteanalyysiin siirtynyt käyttäjä voi valita avainkohtaiset kategoriaryhmät ja lisäksi luoda palauteanalyysikohtaisia kategoriaryhmiä. Julkisella käyttäjällä on käytössään vain palauteanalyysikohtaiset kategoriaryhmät.

Kirjautuneella käyttäjällä on käytettävissään koko kategoriavalintojen kirjo. Kirjautunut käyttäjä voi luoda hallintasivulla tapahtumakohtaisia kategoriaryhmiä (katso kuva 24), jolloin ryhmät ovat myös jatkossa käytettävissä. Palauteanalyysikohtaisten muutosten teko on myös mahdollista.

Moveo Tuomas Moisio

Tapahtuman 'Testitapahtuma 1' palauteanalyysin kategoriat

Palauteanalyysiin lisätyt kategoriaryhmät

Laatu	Kohde	Taso
Positiivinen	Yksilö	Spesifi
Neutraali	Ryhmä	Yleinen
Negatiivinen		
+ Lisää uusi kategoria	+ Lisää uusi kategoria	+ Lisää uusi kategoria

Lisää kategoriaryhmiä palauteanalyysiin

Uusi kategoriaryhmä: [Lisää](#)

Omat kategoriaryhmät: [Lisää](#)

Oletuskategoriaryhmät: [Lisää](#)

Näytä ajastin palauteanalyysissa.

[Jatka palauteanalyysiin](#)

© 2016 Moveo-projekti, 2019 Moveo-projekti

Kuva 24: Palauteanalyysin tapahtuman kategoriaryhmien määrittelysivu.

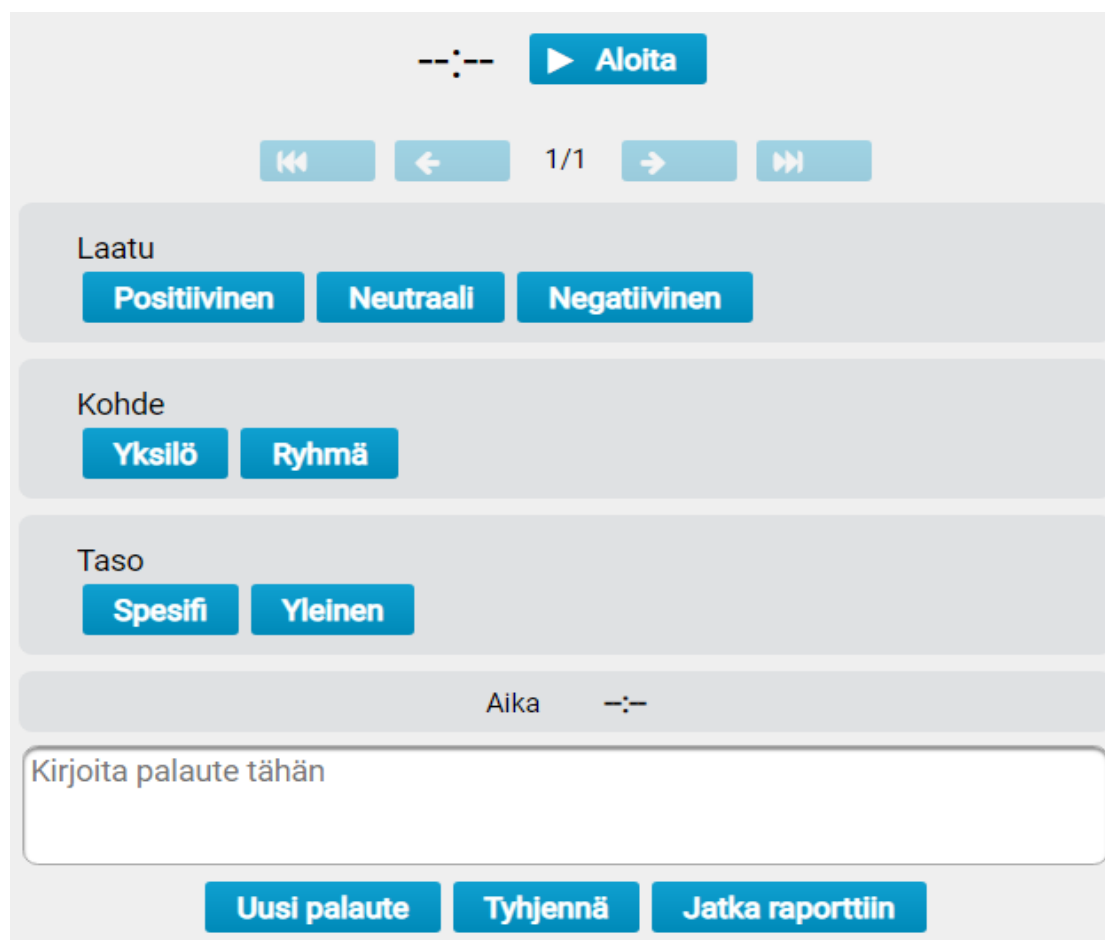
Ryhmäavaimella kirjautuneella käyttäjällä on käytettävissään avaimeen sidotut kategoriaryhmät. Käyttäjä voi valita yhden tai useamman näistä ryhmistä ja myös lisätä omia palauteanalyysikohtaisia kategorioita (katso kuva 24).

Julkisella käyttäjällä on käytettävissään vain itse määrittelemänsä palauteanalyysikohtaiset kategoriat. Niitä ei voi tallentaa sovelluksen tietokantaan.

Palauteanalyysin kategorioiden määrittelysivulla käyttäjä voi valita myös, haluaako hän ajastimen käyttöönsä palauteanalyysin suorituksen ajaksi. Ajastimen valitsemalla jokaiselle kirjaukselle tulee aikaleima kuvaamaan ajankohtaa, jolloin palaute on annettu oppitunnin aikana.

3.13 Palauteanalyysin suorittaminen (Moveo)

Palauteanalyysinäkymä avattaessa on kuvan 25 mukaisesti näkyvillä ajastin (mikäli valittu), kirjausten selauspainikkeet, kategoriaryhmien nimet, kategorioiden rekisteröinti-painikkeet, palautteen ajankohta, sanallisen palautteen kirjaukseen tarkoitettu tekstiruutu sekä painikkeet *Uusi palaute*, *Tyhjennä* ja *Jatka raporttiin*.



The screenshot shows the Moveo feedback analysis interface. At the top, there is a play button labeled "Aloita" and a timer icon. Below it are navigation buttons: a double left arrow, a single left arrow, "1/1", a single right arrow, and a double right arrow. The interface is divided into three main sections: "Laatu" (Quality) with buttons for "Positiivinen", "Neutraali", and "Negatiivinen"; "Kohde" (Target) with buttons for "Yksilö" and "Ryhmä"; and "Taso" (Level) with buttons for "Spesifi" and "Yleinen". Below these sections is a field labeled "Aika" (Time) with a timer icon. At the bottom, there is a text input field with the placeholder "Kirjoita palaute tähän" (Write feedback here). Below the input field are three buttons: "Uusi palaute" (New feedback), "Tyhjennä" (Clear), and "Jatka raporttiin" (Continue to report).

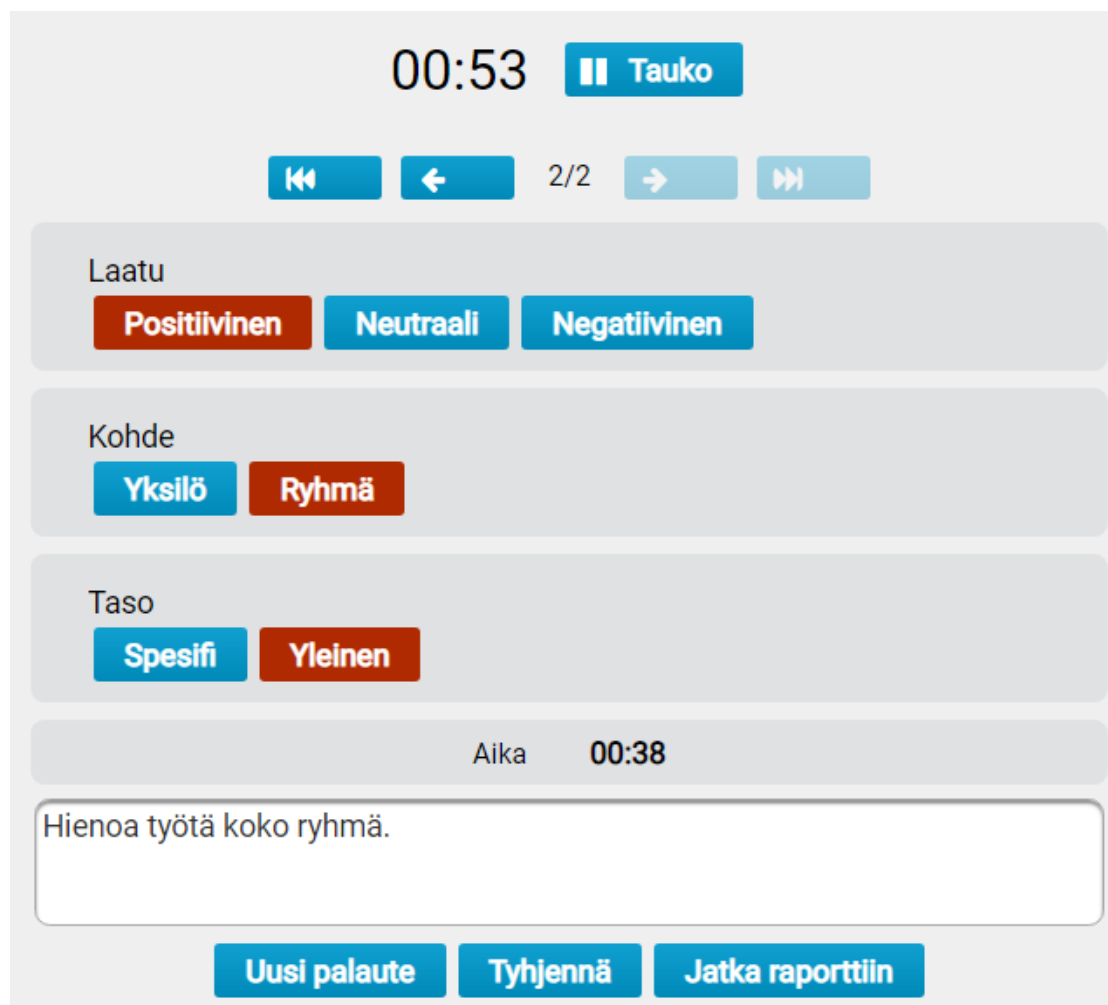
Kuva 25: Palauteanalyysisivu ennen ensimmäistä kirjausta.

Ylimpänä sijaitsevan ajastimen vieressä olevasta *Aloita*-painikkeesta ajastin käynnistetään. Se näyttää minutteina ja sekunteina kuluneen ajan ensimmäisestä kirjauksesta. Palauteanalyysin ollessa käynnissä voidaan se keskeyttää tilapäisesti *Aloita*-painikkeen tilalle tulevalla *Tauko*-painikkeella ja jatkaa samalle paikalle vaihtuvalla *Aloita*-painikkeella.

Käyttäjä voi selata tekemiään kirjauksiaan ajastimen alapuolella sijaitsevilla painikkeilla. Keskellä oleva numero näyttää kyseisen kirjauksen järjestysnumeron ja kaikkien käyttäjän tekemien kirjausten lukumäärän. Vasemmanpuoleisin painike (kaksi nuolta

osoittaa vasemmalle) vie käyttäjän ensimmäiseen palautteeseen. Käyttäjän vie edelliseen kirjaukseen painike, jossa on kuvattu yksi nuoli osoittamaan vasemmalle. Järjestysnumeron oikealla puolella olevat painikkeet toimivat samalla tavalla vieden käyttäjän seuraavaan kirjaukseen ja viimeisimpään kirjaukseen.

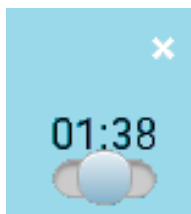
Palauteanalyysin aikana valitut kategorioiden rekisteröintipainikkeet muuttuvat punaisiksi kuvan 26 mukaisesti. Jokaisesta kategoriaryhmästä voi valita enintään yhden kategorian ja vähintään yhdestä kategoriaryhmästä tulee valita kategoria. Kunkin kategoriaryhmän kategoriat osoitetaan taustavärillä. Kun käyttäjä valitsee kategorian tai hän alkaa kirjoittaa kuultua palautetta, palautteen ajankohdaksi valikoituu ajastimen sen hetkinen aika (katso kuva 26). Palautteen ajankohtaa klikkaamalla käyttäjälle avautuu kuvan 27 mukainen dialogi, josta käyttäjä voi muuttaa palautteen ajankohtaa haluamakseen.



The screenshot shows a user interface for recording and analyzing feedback. At the top, a timer displays '00:53' and a 'Tauko' (Pause) button. Below the timer are navigation buttons: a double left arrow, a single left arrow, '2/2', a single right arrow, and a double right arrow. The interface is divided into three sections for filtering feedback: 'Laatu' (Quality) with buttons for 'Positiivinen' (Positive), 'Neutraali' (Neutral), and 'Negatiivinen' (Negative); 'Kohde' (Target) with buttons for 'Yksilö' (Individual) and 'Ryhmä' (Group); and 'Taso' (Level) with buttons for 'Spesifi' (Specific) and 'Yleinen' (General). Below these sections, the current time 'Aika 00:38' is shown. A text input field contains the feedback text 'Hienoa työtä koko ryhmä.' At the bottom, there are three buttons: 'Uusi palaute' (New feedback), 'Tyhjennä' (Clear), and 'Jatka raporttiin' (Continue to report).

Kuva 26: Palauteanalyysisivu kirjauksen aikana.

Uusi palaute -painikkeesta käyttäjä voi siirtyä kirjaamaan uuden palautteen. *Tyhjennä*-painike tyhjentää sen hetken kirjauksen kaikki valinnat, tekstiruudun ja kirjaushetken. *Jatka raporttiin* -painikkeesta käyttäjä lopettaa palauteanalyysin ja jatkaa raporttisivulle.



Kuva 27: Palautteen ajankohdan muokkausdialogi.

3.14 Palauteanalyysin raporttisivu (Moveo)

Palauteanalyysin raporttisivu listaa kuvan 28 mukaisesti jokaisen käyttäjän tekemän kirjauksen taulukkoon. Taulukon ensimmäinen sarake on varattu järjestysnumerolle, toinen sarake mahdolliselle sanalliselle palautteelle sekä seuraavat kategoriaryhmille ja palautteen ajankohdalle. Viimeisessä kahdessa sarakkeessa ovat *muokkaa*- ja *poista*-painikkeet.

Mikäli käyttäjä ei ole valinnut kirjauksessa yhtään kategoriala, se esitetään taulukossa pitkänä viivana (katso kuvan 28 neljäs kirjaus). *Muokkaa*-painike ohjaa käyttäjän takaisin palauteanalyysisivulle sen kirjauksen kohdalle, jonka riviltä käyttäjä painiketta painoi. *Poista*-painiketta painamalla käyttäjä poistaa kyseisellä rivillä olevan kirjauksen.

Lisäksi taulukon ulkopuolella ovat painikkeet *Lataa kuvana* ja *Jatka yhteenvetoon*. *Lataa kuvana* -painikkeesta käyttäjä saa taulukon itselleen png-tiedostona. *Jatka yhteenvetoon* -painikkeen kautta käyttäjä ohjataan yhteenvetosivulle.

Raportti

[Jatka yhteenvetoon](#)

Nro	Sanallinen palaute	Laatu	Kohde	Taso	Aika	Muokkaa	Poista
1	Hieno lyönti Jari!	Positiivinen	Yksilö	Spesifi	00:23		
2	Hienoa työtä koko ryhmä.	Positiivinen	Ryhmä	Yleinen	00:38		
3	Keskity suoritukseen Jorma, älä vain huido.	Neutraali	Yksilö	Spesifi	00:57		
4	Tarkkuutta kaikki.	Neutraali	Ryhmä	----	01:30		

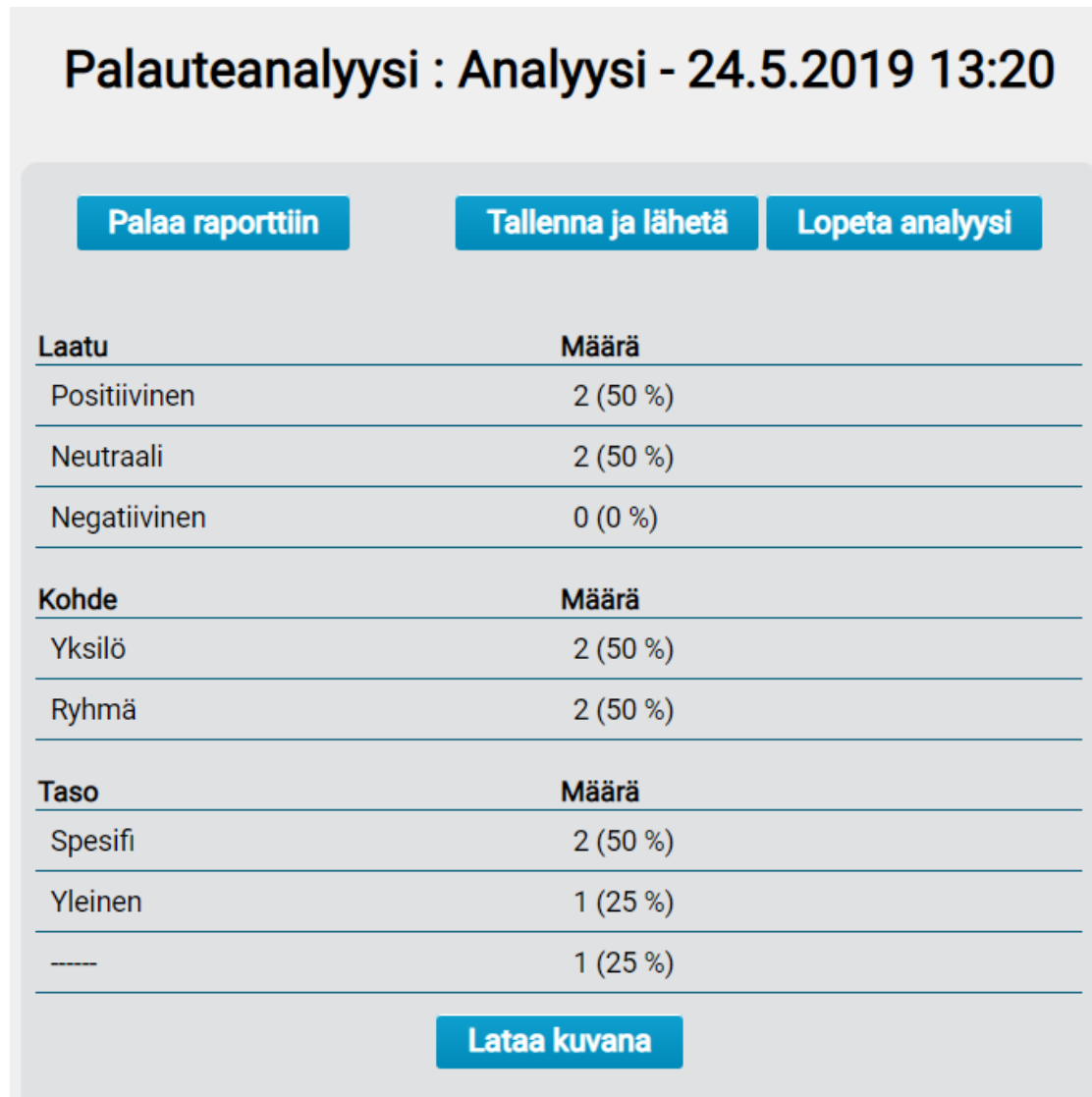
[Lataa kuvana](#)

[Jatka yhteenvetoon](#)

Kuva 28: Palauteanalyysin raporttisivu.

3.15 Palauteanalyysin yhteenvetosivu (Moveo)

Palauteanalyysin yhteenvetosivulla on kuvan 29 mukaisesti nähtävillä kategorioittain eriteltynä suoritettujen palauteanalyysin kirjausten määrät ja prosenttiosuudet sekä tarvittaessa kuvan 30 mukaisesti pylväs- ja piirakkadiagrammit.

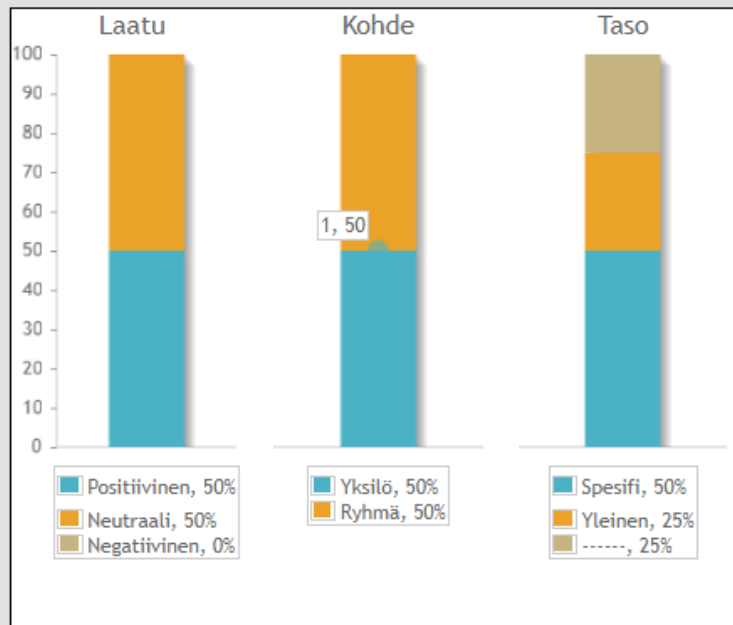


Kuva 29: Palauteanalyysin yhteenvetosivun taulukko

Näytä kuvaajat

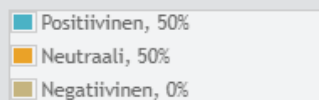
Pylväskuvaajat

Piirakkakuvaajat

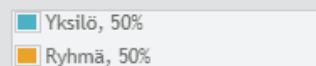


Lataa kuvana

Laatu



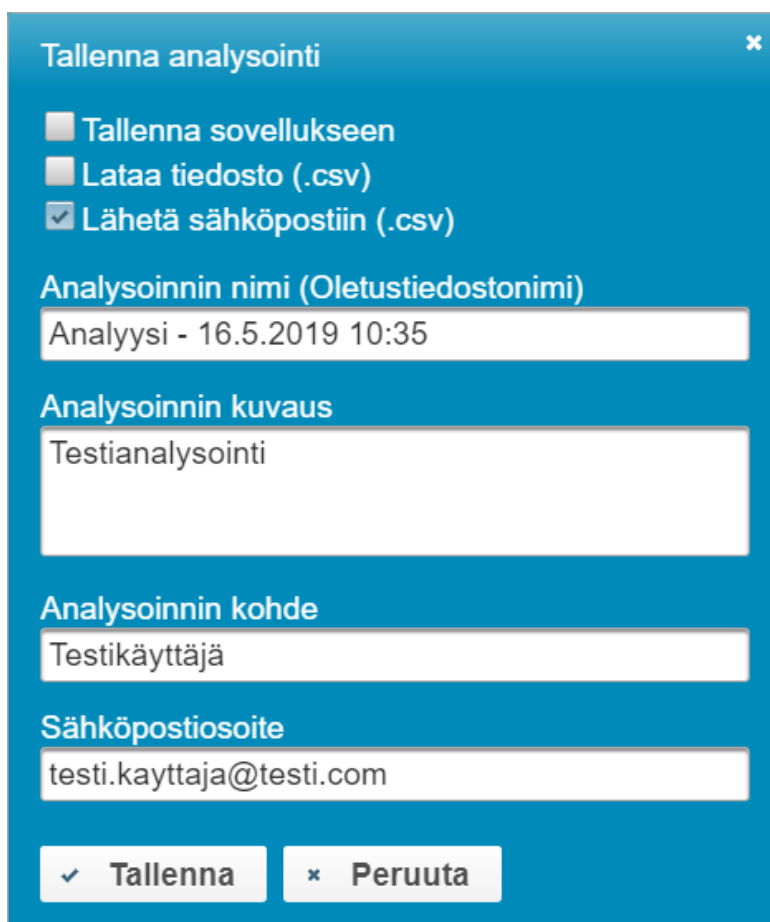
Kohde



Kuva 30: Palauteanalyysin yhteenvetosivun kuvaajat

3.16 Palauteanalyysin kirjausten ja yhteenvedon tallennus

Palauteanalyysin kirjaukset ja yhteenvedo voidaan tallentaa käyttäjäroolista riippuen. JYU-tunnistautuneella käyttäjällä on käytettävissään kuvan 31 mukaisesti paikallinen tallennus omalle päätelaitteelle, sähköpostiin lähetys tai sovelluksen tietokantaan tallentaminen.



Tallenna analysointi

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

Analysoinnin nimi (Oletustiedostonimi)

Analyysi - 16.5.2019 10:35

Analysoinnin kuvaus

Testianalysointi

Analysoinnin kohde

Testikäyttäjä

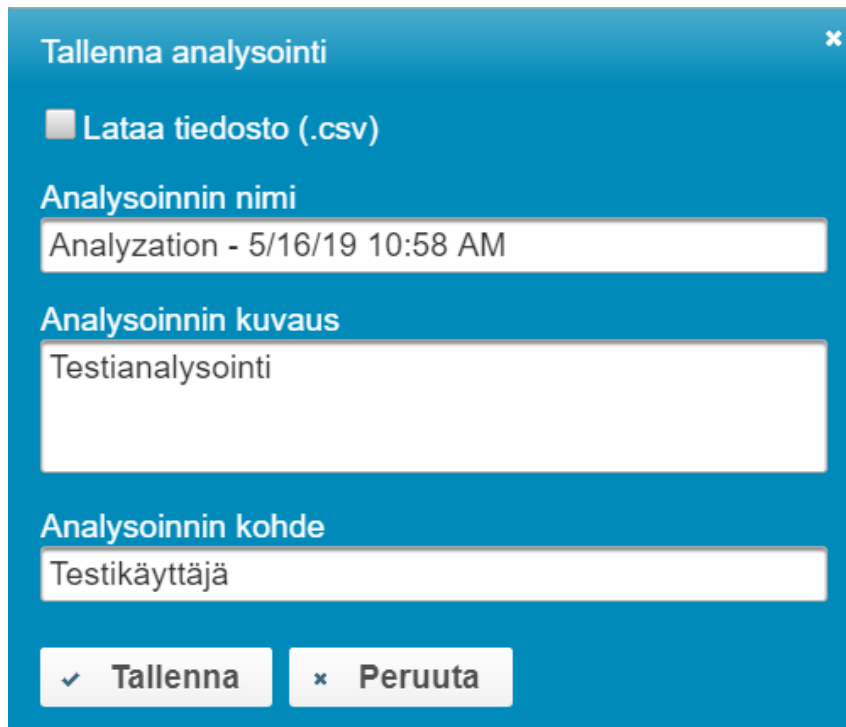
Sähköpostiosoite

testi.kayttaja@testi.com

✓ Tallenna ✕ Peruuta

Kuva 31: Kirjautuneen käyttäjän palauteanalyysin tallennusikkuna.

Ryhmäavaimella kirjautunut tai julkinen käyttäjä voi tehdä ainoastaan paikallisen tallennuksen (katso kuva 32). Jotkin mobiililaitteet eivät tue CSV-tiedoston paikallista tallentamista. Moveo-projektissa havaittiin ainakin iPhonesta puuttuvan kyseinen tallennusmahdollisuus.



Tallenna analysointi

Lataa tiedosto (.csv)

Analysoinnin nimi
Analyzation - 5/16/19 10:58 AM

Analysoinnin kuvaus
Testianalysointi

Analysoinnin kohde
Testikäyttäjä

✓ Tallenna ✕ Peruuta

Kuva 32: Julkisen käyttäjän palauteanalyysin tallennusikkuna.

Kaikki käyttäjät voivat ladata yhteenvedon taulukon ja kuvaajat omina kuvatiedostoinaan laitteeseen painamalla taulukon tai kuvaajan vieressä sijaitsevaa *Lataa kuvana* -painiketta (katso kuva 30).

4 Sovelluksen rakenne ja 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.

4.1 Poikkeukset Moveatis-projektin toteutusratkaisuihin (Moveo)

Moveo-projektin toteutusratkaisut vastaavat hyvin pitkälti Moveatis-projektin toteutusratkaisuja. Merkittävimmät poikkeukset sovelluksen toteutusratkaisussa liittyivät Moveatis-projektin toteuttamien komponenttien käyttöön. Alun perin oli tarkoitus myös palauteanalyysin kirjaussivulla käyttää Moveatis-projektin JavaScriptillä toteuttamia painikekomponentteja, joita hyödynnetään observointisivulla. Niitä ei kuitenkaan saatu toimimaan halutulla tavalla, joten toteutuksessa käytettiin PrimeFacesin valmiskomponentteja. Myöskään kuvan luominen Moveatis-projektin käyttämästä aikajana kuvista ei onnistunut suunnitellulla tavalla vaan ratkaisuna käytettiin JavaScript-kirjastoa, joka tekee HTML-sivusta `canvas`-elementin ja luo kuvan sen perusteella.

4.2 Kehitystyökalut

Palvelinpuolella sovelluksen ohjelmointikielenä käytetään Javaa, jonka tukena käytetään Maven-ohjelmistokehitystyökalua. Mavenin avulla hallitaan esimerkiksi sovelluksen tarvitsemia kirjastoriippuvuuksia. Sovelluksen WWW-käyttöliittymässä ohjelmointikielenä toimii JavaScript. Lisäksi käyttöliittymässä hyödynnetään HTML5- ja CSS3-merkintäkieliä sekä Primefaces-komponenttikirjastoa ja JavaServer Faces -sovelluskehystä. Sovelluksen lähdekoodi on sijoitettuna avoimen lähdekoodin BSD 3-Clause -lisenssin alaisuuteen.

Sovelluksen sivujen kehitysalustana (engl. *framework*) käytetään JSF:ää (Java Server Faces), joka on Javan päällä toimiva WWW-näyttöteknologia. JSF tukee malleihin (engl. *templates*) ja komponentteihin perustuvaa käyttöliittymäkehitystä. Sivut on toteutettu JSF:n XML-pohjaisina Facelet-sivuina, jotka JSF kääntää XHTML-sivuuksi. Moveatis-sovelluksen käyttöliittymän komponenteissa hyödynnetään PrimeFaces-

käyttöliittymäkirjastoa, joka on rakennettu JSF:n päälle. Moveatis-projektiryhmän päätös käyttää PrimeFaces-kirjastoa perustui sen monipuolisuuteen, laajuuteen ja yhtenäisyyteen. PrimeFaces on käytettävissä avoimella Apache-lisenssillä.

4.3 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. Esimerkiksi hallintasivulla (katso kuva 9) painettaessa *lisää*-painiketta palauteanalyysin kategoriaryhmät-sarakkeessa, luodaan ControlManagedBean-käsittelijän instanssiin uusi tyhjä CategorySetEntity-olio ja se näytetään kategoriaryhmän tietojen syöttämiseen tarkoitettussa dialogissa. Dialogin syötekenttien avulla voidaan muokata luodun kategoriaryhmäolion vastaavia kenttiä ja painettaessa *tallenna*-painiketta viedään luotu kategoriaryhmä tietokantaan käyttäen kategoriaryhmän tietokantayhteyttä hallinnoivaa CategorySetBean-oliota.

Sovellus hyödyntää REST-rajapintaa asiakassovellukselta saatavien tietojen välittämiseen palvelimelle. REST mahdollistaa palvelujen esittämisen resurssien muodossa HTTP:tä käyttäen. Resurssien yleinen esitysmuoto on JSON tai XML. Moveatis-sovelluksessa REST ei ole olennaisessa roolissa, eli palvelu ei ole ns. RESTful. Moveo-projektin toteuttamassa lisäosassa REST mahdollistaa esimerkiksi raporttisivun taulukosta JavaScriptillä luodun CSV-datan lähettämisen palvelinpuolelle.

JavaScriptiä hyödynnetään sovelluksessa erinäisiin asiakaspuolen tehtäviin. Suurin osa Moveo-projektin lisäosassa käyttämistä JavaScript-ominaisuuksista hyödyntää Primefacesin ja JSF:n valmiita JavaScript-koodeja ja AJAX-kutsuja. Omia JavaScript-koodeja käytetään sivujen toiminnallisuuden toteuttamiseen huomattavasti vähemmän kuin Moveatis-projektin toteuttamassa toiminnallisuudessa. Esimerkiksi analyysisivun

(katso kuvat 29 ja 30) dynaamiset ominaisuudet on toteutettu hyvin pitkälti JSF:ää, PrimeFacesia sekä analyysisivua vastaavaa ManagedBean-oliota hyödyntäen siinä missä Moveatis-projektin toteuttaman observointisivun lataaminen ja dynaaminen toiminnallisuus nojaa hyvin pitkälti JavaScriptiin. Merkittävimmät JavaScriptillä toteutetut toiminnallisuudet Moveo-projektin toteuttamassa lisäosassa ovat kuvien ja CSV-tiedostojen luonti raportti- ja yhteenvetosivuista (katso kuvat 28-30). JavaScript-koodit suoritetaan käyttäjän näkymässä, eli ne ovat ns. JS-tulkin selaimessa suorittamia asiakassovelluksia (engl. *client-side*). JavaScript on kirjoitettu jQuery-kirjastoa käyttäen.

Moveatis-sovelluksen sivujen rakenne on määritelty HTML:llä ja esitysstyyli on määritelty CSS3:a käyttäen. Tyylin perusasetukset ovat määriteltyinä yhdessä CSS-tiedostossa (`base.css`) ja sivukohtaisesti niitä on täydennetty sivujen omissa tyyli-tiedostoissa. Myös käyttöliittymäkirjasto PrimeFaces tarjoaa sovellukselle jonkin verran valmiita CSS-määrittelyksiä.

4.4 Toimintalogiikka

Taustaohjelmisto on toteutettu Java EE -ohjelmistoalustalla ja se tarjoaa Moveatis-sovelluksen tarvitsemat taustatoiminnot. Taustaohjelmisto huolehtii käyttäjän tunnistamisesta rajapinnan avulla JYU-tunnistautumispalvelussa, 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. Moveatis-sovelluksessa 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

oliassa. Moveatis-sovelluksessa `ManagedBean`-luokat tunnistaa `ManagedBean`-pääteestä, ja ne sijaitsevat paketissa `com.moveatis.managedbeans`.

Control- eli hallintakerros hallitsee tietokantayhteyttä sekä toimii näkymäkerroksen ja mallikerroksen välisenä kerroksena. Hallintaoliot hakevat, luovat, muokkaavat tai poistavat entiteettejä. Kukin hallintaolio vastaa yhdestä entiteetistä, ja vastaavasti kullakin entiteetillä on vain yksi hallintaolio. Hallintaolioiden toteutus Moveatis-sovelluksessa jakautuu kahteen osaan: liittymäosioihin ja niitä toteuttaviin Enterprise Java Bean-luokkiin (EJB). Liittymäosiossa määritellään sovelluspalvelimen ylläpitämän EJB-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 suorittainakaan tarvitse tuhlata kenttien ja yhteyden alustamiseen. Liittymällä täytyy olla sen toteuttava luokka, joka suorittaa ne operaatiot, joihin se on liittymän toteuttamisessa sitoutunut. Moveatis-sovelluksessa liittymät on sijoitettu pakettiin `com.moveatis.interfaces` ja liittymän toteuttavat EJB-luokat tunnistaa niiden nimen `Bean`-pääteestä.

4.5 Sovelluspalvelin

Moveatis-sovellus on sijoitettu Wildfly-sovelluspalvelimelle hakemistoon `standalone/deployments`, jossa tiedosto `ROOT.war` sisältää Moveatis-sovelluksen 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. Moveatis-sovelluksen hakemistorakenne ei sisällä sellaisia asetustiedostoja, joita käsittelemällä Moveatis-sovelluksen toimintaa voisi muokata sovelluksen ollessa käynnissä. Moveatis-sovelluksen päivitys vaatii aina sovelluksen pysäyttämisen ja uuden tiedoston `ROOT.war` kopioinnin Wildflyn hakemistoon `standalone/deployments`.

5 Sovelluksen tietokanta

Tässä luvussa esitetään Moveatis-tietojärjestelmän tietokannan taulut. Alkuperäisiin tauluihin ei Moveo-projektissa tehty muutoksia, joten niiden esittely on siirretty tähän sellaisenaan Moveatis-projektin sovellusraportista [7]. Jokaiseen tietokannan tauluun, joka vastaa Java-koodissa määriteltyä entiteettiä määritellään tietokannassa seuraavat kentät:

ID-kenttä Määrittää riville yksilöivän tunniste.

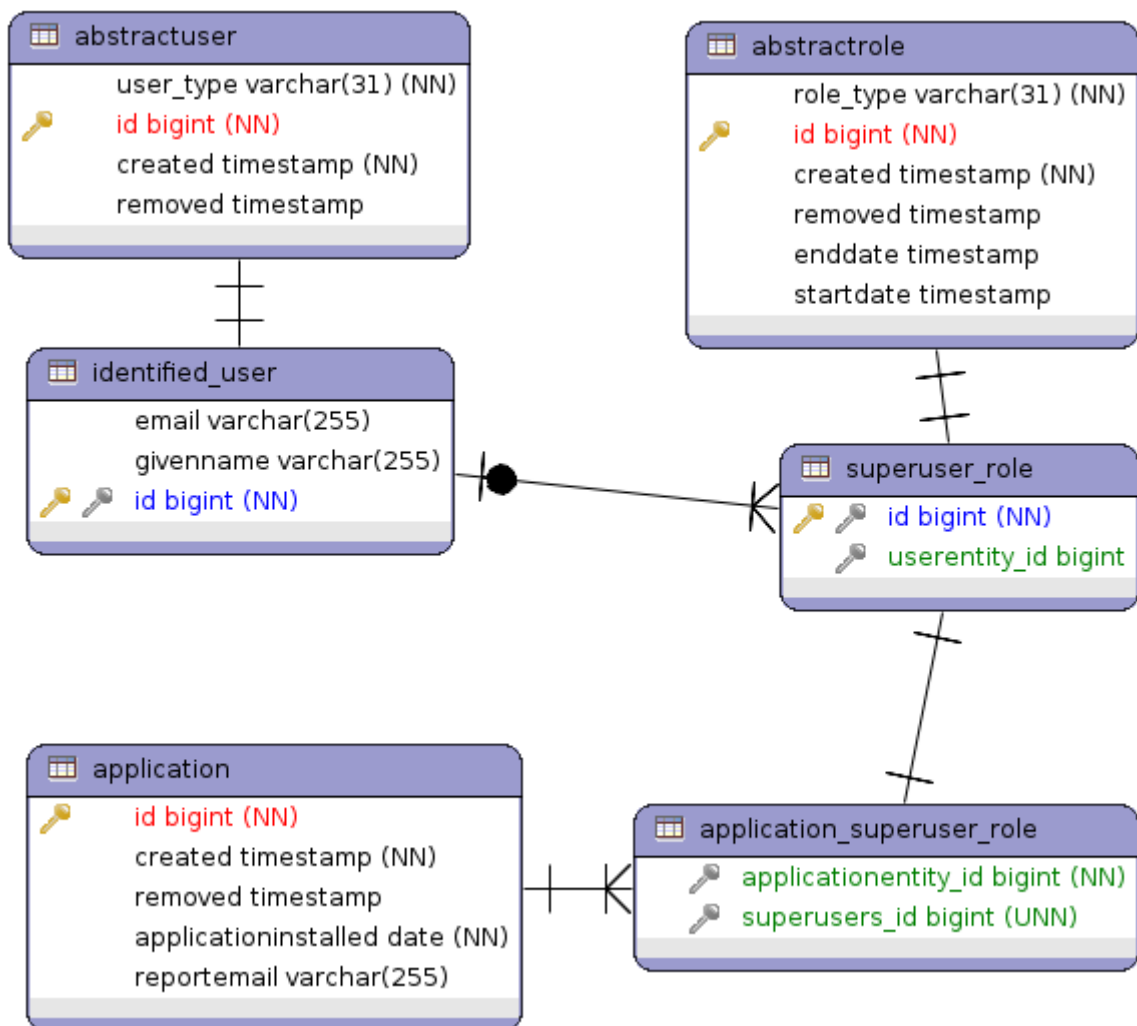
Created-kenttä Kertoo milloin rivi on luotu.

Removed-kenttä Kertoo milloin rivi on poistettu.

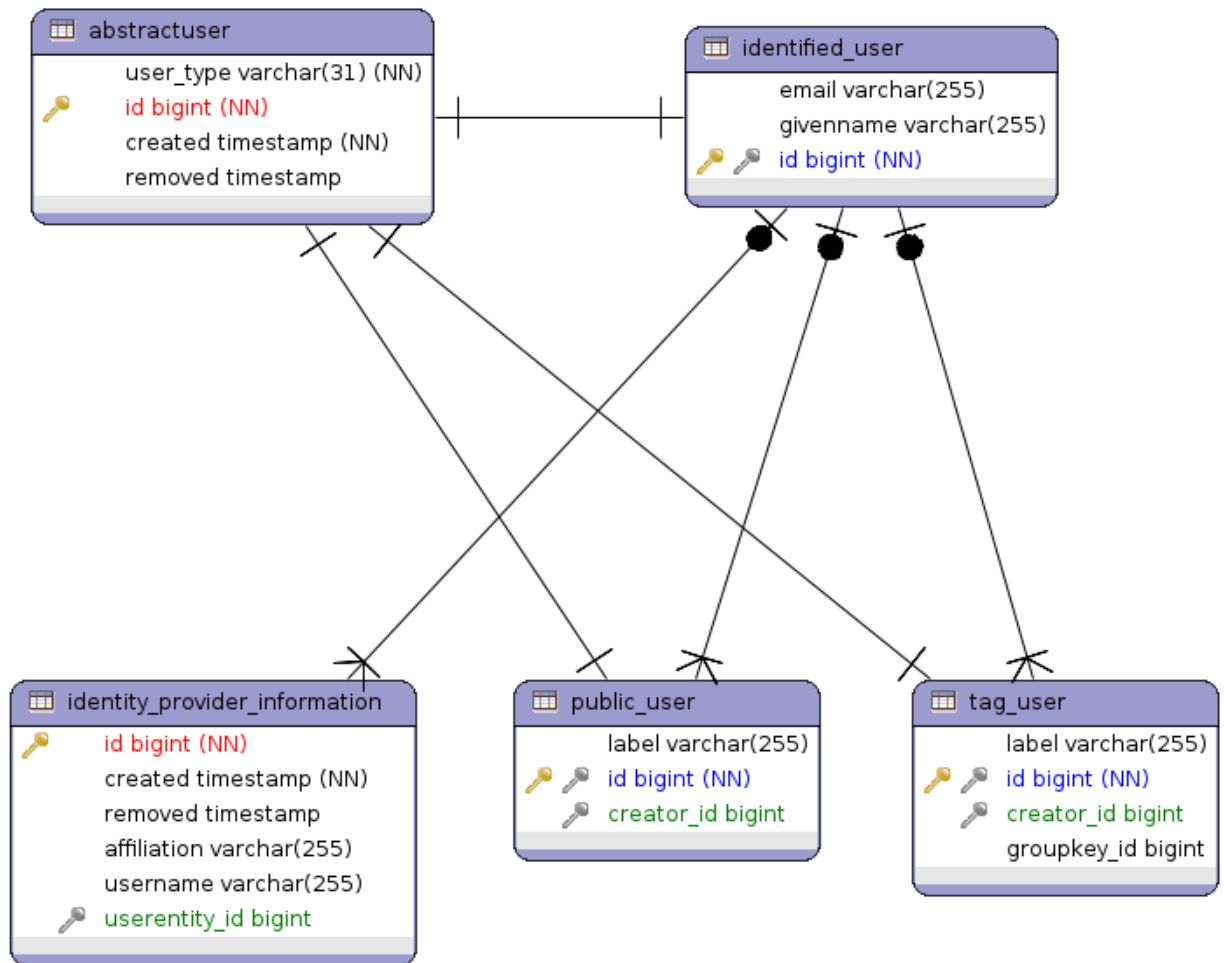
Sovelluksen käyttämästä tietokannasta ei käytännössä poisteta mitään, vaan poistetut rivit ainoastaan merkitään poistetuiksi eikä niitä enää näytetä sovelluksessa.

5.1 Käyttäjiä ja rooleja koskevat taulut (Moveatis)

Kuva 33 kuvaa käyttäjät ja heidän rooleihinsa liittyvät tietokantataulut. Kuvassa 34 kuvataan käyttäjän tietoihin liittyvät tietokantataulut.



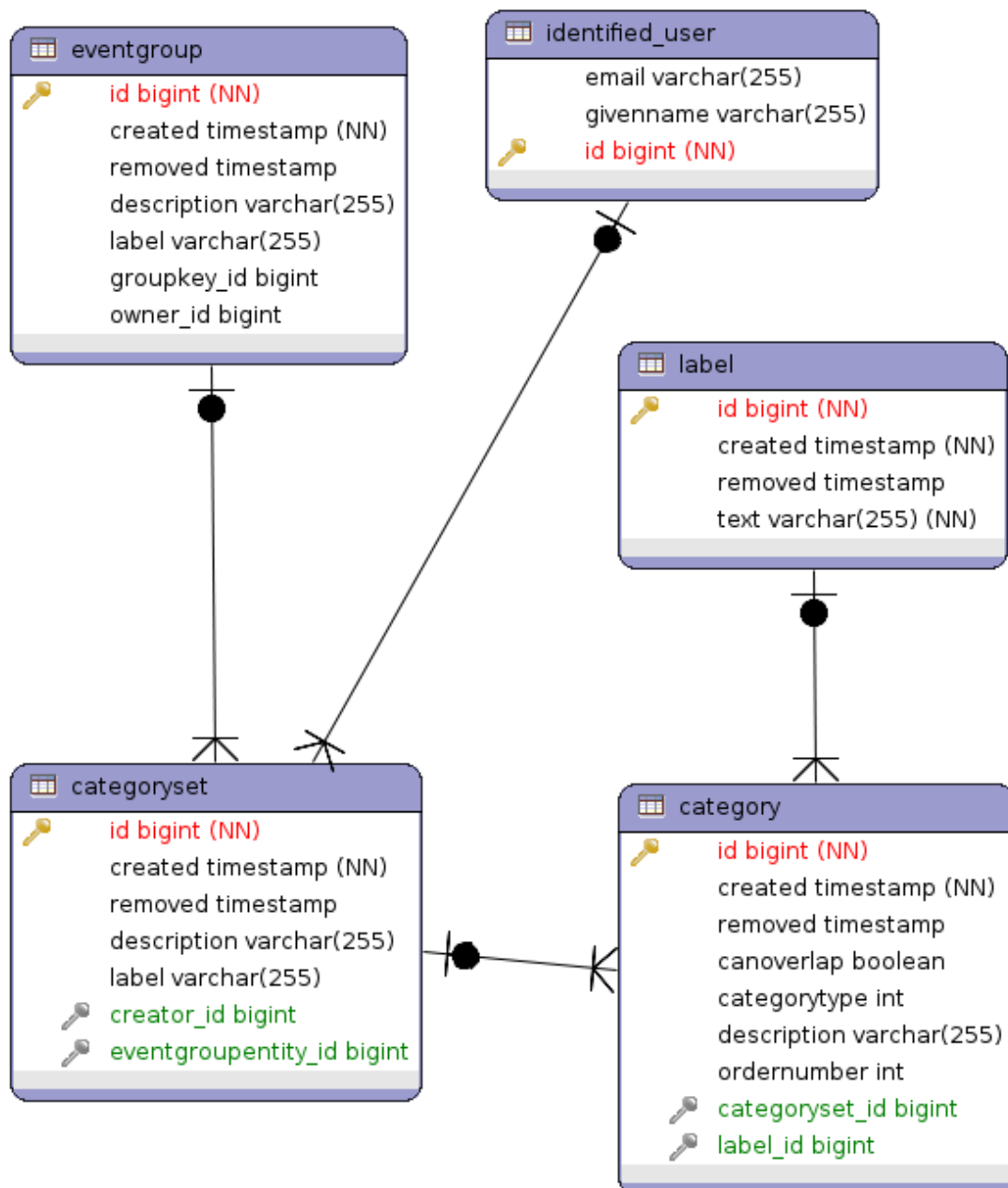
Kuva 33. Rooleihin ja sovellukseen liittyvät taulut.



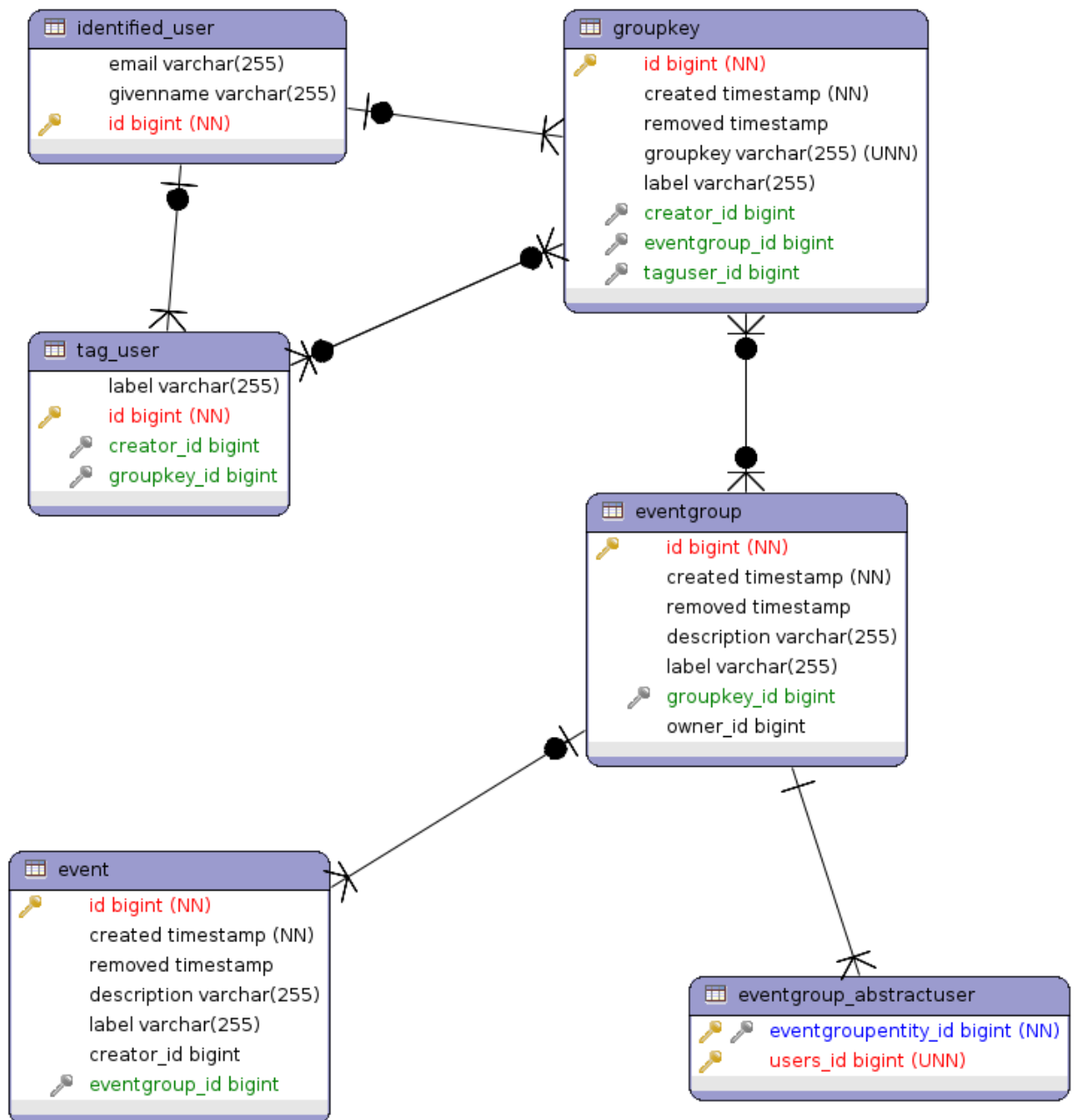
Kuva 34. Käyttäjien tietoihin liittyvät taulut.

5.2 Kategorioihin, tapahtumiin ja observointeihin liittyvät taulut (Moveatis)

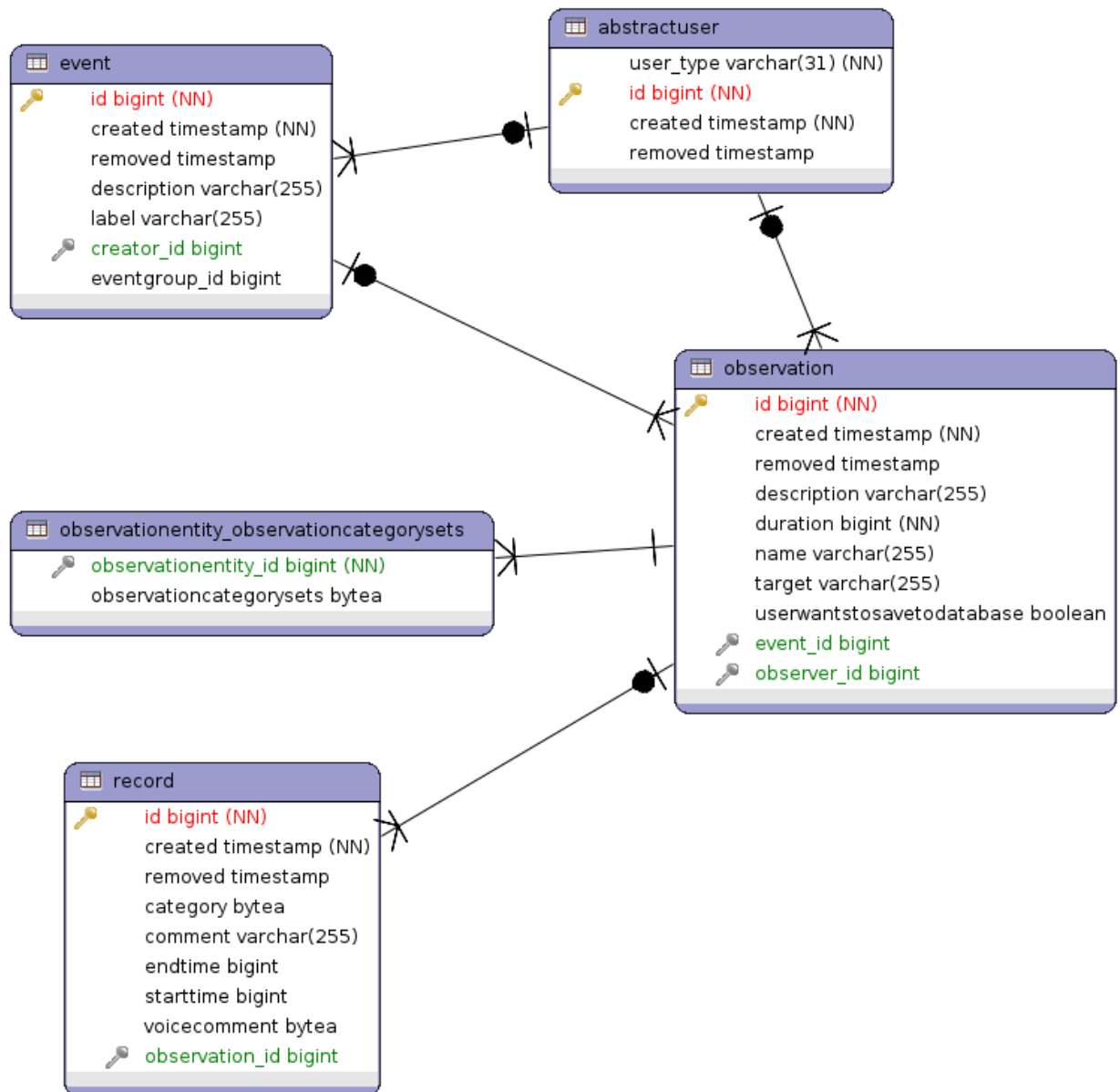
Kuvassa 35 kuvataan kategorioiden, kategoriaryhmien ja tapahtumaryhmien väliset yhteydet. Kuvassa 36 kuvataan tapahtumien, tapahtumaryhmien ja avainkäyttäjien yhteydet ja kuvassa 37 observointien, kirjausten ja tapahtumien yhteydet.



Kuva 35. Kategorioiden, kategoriaryhmien ja tapahtumaryhmien yhteydet.



Kuva 36. Tapahtumien, tapahtumaryhmien ja avainkäyttäjien yhteydet.



Kuva 37. Observointien, kirjausten ja tapahtumien yhteydet.

5.3 Taulujen käyttötarkoitukset (Moveatis)

Moveatis-sovelluksen tietokannan taulujen käyttötarkoitusta on kuvattu taulukossa 1.

Taulu (ja vastaava Java-entiteetti)	Kuvaus
abstractrole (AbstractRole)	Roolien ylliluokka. Uusia rooleja lisätään Moveatukseen tätä luokkaa laajentamalla
abstractuser (AbstractUser)	Moveatoksen kolmen käyttäjäroolin ylliluokka.
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.
identity_provider_information (IdentityProviderInformationEntity)	Luo yhteyden JYU-tunnistautumistietojen ja Moveatoksen tunnistetun käyttäjän tietojen välille.

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

Taulukko 1. Tietokannan taulujen käyttötarkoitus.

5.4 Palauteanalyysiin liittyvät taulut (Moveo)

Moveo-projektin toteuttaman lisäosan analysointiominaisuutta varten tietokantaan lisäämistä tauluista kaikki vastaavat kentiltään hyvin läheisesti observointipuolella käytössä olevia tauluja. Ohjelmassa ei kuitenkaan päädytty käyttämään palauteanalyysin ja observoinnin tietojen tallentamiseen yhteistä taulua, koska kyseisiä tauluja vastaavia luokkia halutaan käyttää eri tehtäviin. Tällöin tulisi tauluihin lisätä vähintään joku lip-pumuuttuja kertomaan, kuuluuko taulun rivi analysointipuolelle vai observointipuolelle. Uusien kenttien lisääminen vanhoihin tauluihin aiheuttaa JavaEE:n taulujen generoimistavassa sen, että vanhat kentät jouduttaisiin ylikirjoittamaan ja tätä pyrittiin välttämään. Lisäksi ohjelmakoodissa koettiin selkeämmäksi, että molemmilla on omat luokkansa. Näin ollen myös observointipuolen lähdekoodi on voitu jättää hyvin pitkälti ennalleen.

Yhteisen taulun käyttäminen voisi mahdollistaa ylläpidettävämmän lähdekoodin, yksinkertaisemmat kyselyt ja paremman optimoinnin. Arvioitiin kuitenkin, että edut tulisivat esiin vasta suuremman mittakaavan ohjelmassa ja mikäli sovelluksessa tehtäisiin enemmän SQL-kyselyjä. Koska JavaEE pitkälti hallitsee tietokantayhteyttä ja oliot luodaan automaattisesti tietokannan rivien perusteella, koettiin erillisten taulujen käyttö kätevämmäksi.

Kuvissa 38 ja 39 esitellään tietokannan uusien taulujen rakenne ER-kaavioina. Taulukossa 2 kuvataan taulujen käyttötarkoitukset.

5.5 Palauteanalyysiin liittyvien taulujen käyttötarkoitus (Moveo)

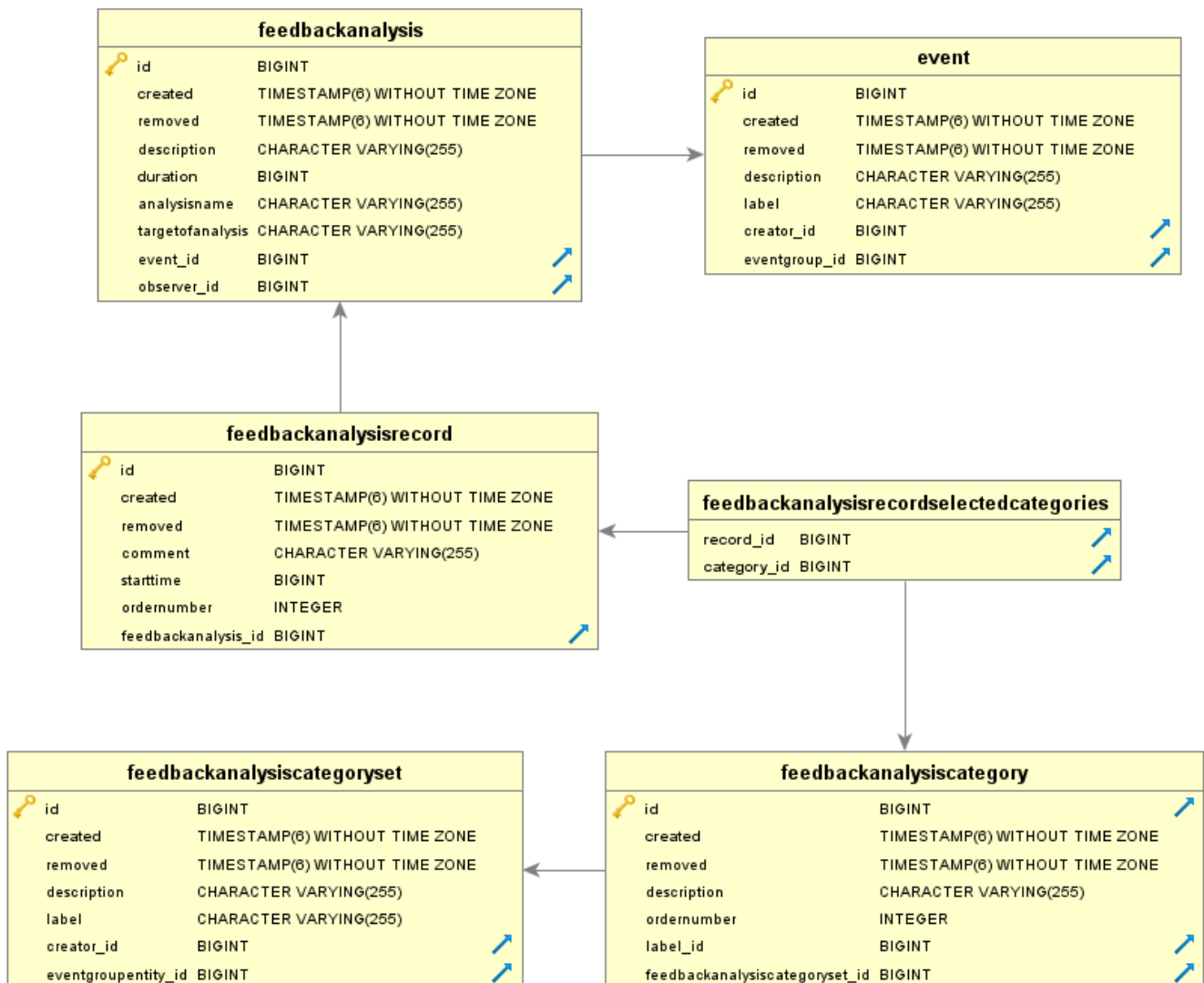
Taulukossa 2 kuvataan palauteanalyysiin liittyvien taulujen käyttötarkoitukset.

Taulukko 2: Taulujen käyttötarkoitukset.

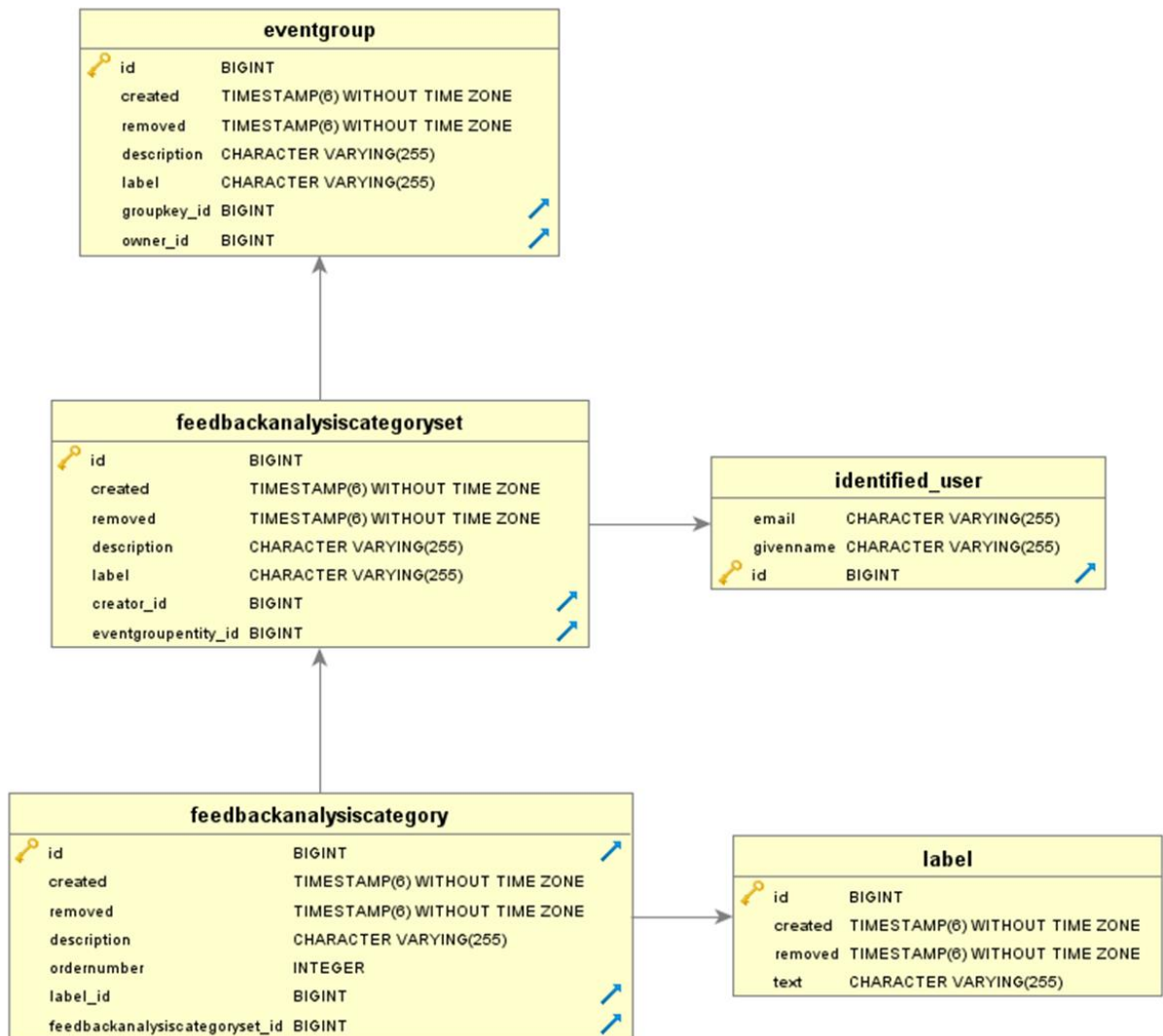
Taulu (ja vastaava Java-entiteetti)	Kuvaus
feedbackanalysis (FeedbackAnalysisEntity)	Kokoaa yhden palauteanalyysitapahtuman kirjaukset sekä palauteanalyysitapahtumaa kuvailevia kenttiä, kuten palauteanalyysin kesto, kohde, analysoija, nimi ja kuvaus. Kertoo myös, mihin tapahtumaan analyysi kuuluu.
feedbackanalysisrecord (FeedbackAnalysisRecordEntity)	Kirjaus palauteanalyysitapahtumassa pitää kirjaa siitä, mihin analyysiin kirjaus kuuluu sekä kirjauksen järjestysnumerosta analyysin sisällä, ajanhetkestä sekunteina ja palautteesta.
feedbackanalysiscategory (FeedbackAnalysisCategoryEntity)	Pitää kirjaa palauteanalyysin kategorian tiedoista, kuten kategorian nimestä, kuvauksesta, järjestysnumerosta kategoriaryhmän sisällä ja siitä, mihin kategoriaryhmään kategoria kuuluu.
feedbackanalysiscategoryset (FeedbackAnalysisCategorySetEntity)	Pitää kirjaa palauteanalyysin kategoriaryhmän tiedoista, kuten kategoriaryhmän nimestä ja kuvauksesta. Kertoo myös, mihin tapahtumaryhmään kategoriaryhmä kuuluu ja kuka sen on luonut.
feedbackanalysisrecordselectedcategories (Ei vastaavuutta)	JavaEE:n automaattisesti luoma liitostaulu palauteanalyysikirjausten ja -kategorioiden välillä. Pitää kirjaa siitä, mitkä kategoriat ovat valittuna missäkin kirjauksessa. Kussakin kirjauksessa voi olla valittuna monta kategoriaa, ja kukin kategoria voi sisältyä moneen eri kirjaukseen.

5.6 Palauteanalyysiin liittyvien taulujen ER-kaaviot (Moveo)

Kuvassa 38 on kuvattu palauteanalyysin kirjausten taulut ER-kaaviona. Kuvassa 39 on kuvattu tapahtumien, kategoriaryhmien ja kategorioiden ER-kaavio palauteanalyysiin liittyen.



Kuva 38: Palauteanalyysin kirjausten taulut ER-kaaviona.



Kuva 39: Tapahtumien, kategoriaryhmien ja kategorioiden ER-kaavio palauteanalyysiin liittyen.

6 Moveo-projektin sovelluksen tavoitteiden toteutuminen

Moveo-projektin Moveatis-tietojärjestelmään kehittämiä tietoja ja toimintoja kuvataan tarkemmin vaatimusmäärittelydokumentissa [1], jossa käydään läpi jokaisen vaatimuksen tila projektin päättyessä. Vaatimusmäärittelyssä tehdyn priorisoinnin mukaan pakollisista vaatimuksista saatiin toteutettua kaikki.

6.1 Moveo-projektissa toteutetut vaatimukset

Palauteanalyysin toiminnallisuus saatiin onnistuneesti kehitettyä Moveatis-tietojärjestelmään observointitoiminnallisuuden rinnalle. Molemmissa on nyt vastaavat toiminnot kategorioiden muokkauksen ja aiempien palauteanalyysien ja observointien kirjausten tarkastelun osalta. Sovelluksen vaatimukseen uuden toiminnallisuuden kehittämisen lisäksi kuului myös vaatimus sovelluksen ulkoasun kehittamisestä, joka myös projektissa toteutui.

Sovelluksen käyttöliittymän toteutuksessa pyrittiin ottamaan huomioon se, että sovellusta käytetään myös mobiililaitteilla. Erityisen haasteellista tämä oli palauteanalyysin raporttisivulla, sillä kirjausten esittäminen selkeästi taulukossa vaatii enemmän tilaa kuin mobiililaitteen näytöt välttämättä sallivat. Toteutusratkaisuna sovelluksessa esiintyvät taulukot rivittyvät eri tavalla, mikäli näyttö on liian kapea. Mobiililaitetuen toteuttamisessa onnistuttiin kuitenkin varsin hyvin, eikä sovelluksen järjestelmätestauksissa älypuhelimilla (katso raportit [3] ja [4]) ilmennyt ongelmia.

Sekä sovelluksen käyttöliittymää että lähdekoodia koskien tavoitteena oli, että sovelluksen jatkokehittäminen tehtäisiin mahdollisimman helpoksi. Lähdekoodin osalta tähän pyrittiin Moveatis-projektin tavoin pitämällä ohjelmiston rakenne MVC-arkkitehtuurin mukaisena, jolloin ohjelmiston eri osat ovat erillään toisistaan. Sovelluskoodissa pyrittiin myös yleisten toiminnallisuuksien helppoon uudelleenkäytettävyyteen, ettei saman kaltaisia ominaisuuksia tarvitse kirjoittaa uudelleen. Esimerkiksi tiedostojen lataamista varten lähdekoodissa on toteutettuna `DownloadTools`-niminen luokka, jonka avulla sovellus toteuttaa tiedoston lataamisen useilla eri sivuilla ja jota käyttäen tiedostojen lataaminen on myös jatkokehityksessä helppo toteuttaa. Lisäksi lähdekoodissa pyrittiin kuvaavaan ja yhtenevään nimeämisen ja kommentointikäytäntöön.

Käyttöliittymän osalta tarkoitus oli ottaa huomioon se, että jatkokehityksessä sovellukseen saatetaan lisätä vielä kolmas tai useampi menetelmätyyppi. Menetelmätyypinvalintasivun (katso kuva 7) osalta tämä ei tuottane ongelmia, mutta hallintasivun (katso kuvat 8, 9 ja 11) tapahtumat kokoava taulukko käy liian leveäksi, mikäli siihen vielä lisätään sarakkeita. Taulukon rakennetta täytynee siinä tapauksessa kehittää siten, että joko osa taulukon tiedoista täytyy erikseen valita näkyville tai taulukossa esiintyvät menetelmätyypit esitettäisiin taulukossa allekkain rinnakkaisuuden sijaan.

6.2 Jatkokehitykseen sovitut vaatimukset

Tärkeiksi määritellyistä vaatimuskokonaisuuksista kokonaan jatkokehitykseen sovittiin tilaajan kanssa jo Moveatis-projektin aikana määritellyt vaatimukset pääkäyttäjä-roolista ohjelmassa sekä videon upotuksesta sovellukseen. Videon upotus koettiin haastavaksi erityisesti sen sivulta vaatiman tilan vuoksi. Ilman videon ohjaamista sovelluksesta ajastimen käyttö videon kanssa samaan aikaan on hankalaa. Mikäli video halutaan pysäyttää kirjauksen tekemisen ajaksi, tulee myös ajastin pysäyttää erikseen.

Pääkäyttäjäroolin toteuttaminen sovellukseen sovittiin myös jatkokehitykseen. Nykyisellään sovelluksessa kaikilla kirjautuneilla käyttäjillä on samat oikeudet hallita omia tapahtumiaan. Pääkäyttäjän oikeuksiin kuuluisi lisäksi mahdollisuus hallita myös muiden käyttäjien tapahtumia. Näiden lisäksi jatkokehitykseen sovittiin erilaisia suurimmaksi osaksi sovelluksen käytettävyyteen liittyviä ominaisuuksia, joista voi lukea vaatimusmäärittelydokumentista [1].

7 Ohjeita ylläpitoon ja jatkokehitykseen

Luvussa annetaan yleisiä ohjeita liittyen Moveatis-sovelluksen ylläpitoon ja jatkokehitykseen. Myös tarkemmat vaiheittaiset asennusohjeet [5] on laadittu.

Moveatis-sovelluksen lähdekoodi on sijoitettu GitHub-palveluun Moveatis-tilille. Moveatis 1.0:n lähdekoodi löytyy tietovarastosta Moveatis ja Moveatis 2.0:n varastosta Moveo. Tilin tunnukset ovat Nelli Lyyralla, Jukka-Pekka Santasella ja Pilvikki Heikinaro-Johanssonilla. Tili ei kuitenkaan ole yksityinen, eli kuka tahansa voi ladata lähdekoodin kehitettäväkseen. Vaatimusmäärittelydokumenttiin [1] 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 Sovelluspalvelimen asennusohjeet

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 luvussa voida esittää tyhjentävää ohjeistusta niitä kaikkia varten. Tarkista käyttämästäsi sovelluspalvelimen ohjeistuksesta, mikä on sen käyttämä asennushakemisto. Esimerkiksi **Wildfly**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-sovellus käyttää tietokantaa tietojen tallentamiseen, ja Java EE:ssä sovelluspalvelin tarjoaa **tietokantayhteyden** sovelluksen käyttöön. Oletusasetuksena Moveatis-sovellus odottaa, että sovelluspalvelin tarjoaa tietokantayhteyden polussa `java:jboss/datasources/MoveatisDS`. Polkua voi muuttaa tiedostossa `pom.xml`, mutta nimeä muutettaessa pitää Moveatis-sovellus kääntää uudelleen. Tarkista käyttämäsi sovelluspalvelimen ohjeista, miten siinä asetetaan tietokantayhteyksiä tarjolle.

7.2 Tietokantapalvelimen asennusohjeet

Moveatis-sovellus 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 sekä tuotantoympäristössä Wildfly-sovelluspalvelinta ja PostgreSQL-tietokantapalvelinta. Wildflyhyn piti asentaa jar-tiedostoksi pakattu **PostgreSQL-ajuri** ja määrittellä MoveatisDS-niminen tietokantayhteys käyttämällä `jboss-cli.sh`-komentorivityökalua. Moveatis-projektiryhmän kehitystyössä käyttämän Docker-imagen rakennustiedosto `Dockerfile` löytyy Githubista osoitteesta <https://github.com/phinaliumz/moveatis-docker>. Siitä voi ottaa vaikutteita, sillä se sisältää Wildfly 9:n asennuksen sekä PostgreSQL:n skriptin, joka asentaa PostgreSQL-ajurin sekä määrittelee MoveatisDS-tietokantayhteyden. Moveo-projektissa kehitystyössä käytettiin Dockerin sijaan Virtualbox-ohjelmistossa käynnissä olevaa Ubuntu-server-virtuaalikonetta.

7.3 Tietokantataulujen automaattinen muodostus

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`. Moveatis-sovelluksessa avain määritellään tiedostossa `persistence.xml`, joka sijaitsee hakemistossa `src/main/resources/META-INF`. Tuotantokäytössä asetusta on `none`, mutta kehitystyössä saattaa olla hyödyllistä käyttää asetusta `drop-and-create`, jolloin sovelluspalvelin poistaa vanhat taulut tietokannasta ja luo ne uudelleen. Tämä asetusta siis tyhjentää tietokannan tiedoista ja luo uudet tyhjät taulut. Tämä on syytä huomioida kehitystyössä, ettei poista säilytettäviä tietoja väärään aikaan.

Sovelluspalvelin suorittaa edellä mainitut toimenpiteet avaimella `javax.persistence.schema-generation.database.action` sille määritetyt tietokantaoperaatiot sovellusta asennettaessa, esimerkiksi siirtämällä `war`-tiedosto sovelluspalve-

limen asennushakemistoon. Jos Moveatis-sovellus asennetaan uudelleen tyhjään tietokantaan, tulee tiedoston `persistence.xml` avain `javax.persistence.schema-generation.database.action` muuttaa arvoon `create`. Ilman kyseistä toimenpidettä sovelluspalvelin ei luo tietokantatauluja, eikä Moveatis-sovellus ole käytettävissä. Luonnin jälkeen avain on syytä muuttaa takaisin arvoon `none`. Huomaa, että arvoa muutettaessa, tulee kirjoitushetken tilanteen mukaisesti Moveatis-sovellus kääntää ja paketoita uudelleen war-tiedostoksi.

7.4 Sovelluksen toimivuus palvelin- ja selainohjelmistoilla

Sovellusta asennettaessa on huomioitava seuraavat rajoitteet:

- Sovelluksen on testattu toimivan Java EE:n versiolla 7, Java 8 -virtuaalikooneessa, Wildfly sovelluspalvelimen versioilla 10 ja 14 sekä PostgreSQL tietokantapalvelinten versioilla 9.4 ja 11.0. Tuotantopalvelimella on käytössä PostgreSQL 9.4 ja Wildfly 14.
- Sovellus ei toimi ilman Javaa eikä ilman tietokantaa.

Sovellusta käyttöön otettaessa on syytä huomioida seuraavat rajoitteet:

- Sovellusta on testattu yleisimmillä, keväällä 2019 käytettävissä olevilla WWW-selainten versioilla. Suositellut selaimet ovat Microsoft Internet Explorer 11, Mozilla Firefox 67, Google Chrome 76 ja Apple Safari 12.
- JavaScript on olennainen osa sovellusta, eikä sovellus toimi oikein, mikäli JavaScript on selaimessa kytketty pois päältä.

7.5 Ohjeita jatkokehittäjälle

Kaikki käyttöliittymään liittyvä koodi löytyy hakemistosta `src/main/webapp`. JavaScript, CSS ja kuvat löytyvät hakemistosta `webapp/META-INF/resources`. Jokaisella HTML-sivulla ovat ylä- ja alapalkki 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.

Raportin 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.

Virheenjäljitys onnistunee parhaiten sovelluskehitysympäristön, kuten Netbeansin tai IntelliJ IDEAn, debuggaustyökaluilla. Esimerkiksi Netbeansin komennolla *Attach debugger* voi liittää debuggerin sovelluspalvelimella pyörivään Moveatis-sovellukseen. Virheenjäljityksessä hyödyllistä lienee myös lukea sovelluspalvelimen **lokitytiedostoja**, Wildflyn tapauksessa ne löytyvät hakemistosta `<Wildflyn juurihakemisto>/standalone/logs`. Käyttöliittymään liittyviä JavaScripti- ja CSS-tiedostoja voidaan debugata selaimien tarjoamilla työkaluilla.

Moveatis-sovelluksessa **kirjautuminen** toteutetaan 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. Sovellus tunnistaa itse automaattisesti, kun sitä ajetaan localhostissa ja käyttää kirjautumistoimintoon tällöin kyseistä sivua.

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 raportin kirjoitushetkellä Githubista löytyvä Dummy-SMTP on testikäyttöön riittävä sähköpostipalvelin ja se luodaan Python-skriptillä.

7.6 Ideoita jatkokehitykseen

Luvussa 6 esiteltyjen jatkokehitykseen sovittujen vaatimusten lisäksi sovelluksen jatkokehityksessä voitaneen helpottaa sovelluksen käytettävyyttä **sivujen toimintalogiikkaa** hiomalla. Esimerkiksi sovellukseen voisi lisätä muokattavuutta tarjoamalla käyttäjälle mahdollisuus **muuttaa kategorioiden ja kategoriaryhmien järjestystä** hallintasivulla sekä palauteanalyysin ja observoinnin kategoriaryhmien määrittelysivulla. Myös kategoriaryhmän nimen muuttaminen kategoriaryhmien määrittelysivuilla olisi hyvä mahdollistaa. Mikäli käyttäjä tekee virheen kategoriaryhmän nimeä määrittäessään, joutuu hän nyt poistamaan koko kategoriaryhmän ennen sen uudelleen määrittelyä.

Palauteanalyysin raporttisivun taulukossa voitaisiin korostaa muokkauksen ja poiston yhteydessä käsiteltävää kirjausta koskeva rivi, jolloin käyttäjän ei tarvitsisi luottaa muistiinsa niin paljon. Hallintasivun ja kategoriaryhmien määrittelysivujen **validoinnin** logiikan pohtiminen lienee myös jatkokehityksessä huomionarvoinen seikka. Validoinnin toiminnassa on tällä hetkellä jonkin verran epä johdonmukaisuutta. Esimerkiksi duplikaattikategorioista kategoriaryhmien määrittelysivulla annetaan virheilmoitus vasta käyttäjän halutessa edetä analyysisivulle, eikä niiden syötön yhteydessä.

Ajastimen toiminnassa palauteanalyysissä on myös jonkin verran mahdollisuuksia kehitykseen. Koska kirjausten ajankohtien haluttiin vastaavan kirjausten kirjausjärjestystä, ei aiempien kirjausten väliin lisätyille kirjauksille kirjata ajankohtaa lainkaan ja ajankohdan muokkaamiseen tarkoitettu työkalu sallii ajankohdan muokkaamisen ainoastaan kirjausta edeltävän ja seuraavan kirjauksen ajankohdan välillä. Käyttäjälle voisi olla hyvä antaa mahdollisuus järjestää kirjatukset raporttisivulla valintansa mukaan kirjausjärjestyksen tai ajankohdan mukaan ja mahdollisesti myös muuttaa järjestysnumerot ajankohdan mukaisiksi. Myös ajankohdan muokkaustyökalun toteuttaminen syöteläatikkona nykyisen liukusäätimen sijaan voisi olla käytettävyyden kannalta parempi ratkaisu.

Myös sovelluksen **tietokannan** rakennetta saattaa olla tarpeen pohtia uudelleen, mikäli jatkokehityksessä sovellukseen lisätään useampia menetelmätyyppejä opetustapahtumien tarkkailuun. Taulujen yhdistäminen saattaa nimittäin siinä tapauksessa olla hyvä tapa vähentää sovelluksessa tehtävien kyselyjen määrää.

7.7 Ohjeita tuotantopalvelimen ylläpitoon

Myös tuotantopalvelimella käytetään sovelluspalvelimena Wildflyta, joten suuri osa luvuissa 7.1–7.2 sekä asennusohjeissa [5] esitetyistä ohjeista pätee myös tuotantopalvelimen hallintaan. Tuotantopalvelimen hallintaan on laadittu myös erilliset ohjeet [6].

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

Moveatis-sovelluksessa **poisto-ominaisuus** on toteutettu siten, että käyttäjän poistamat rivit eivät poistu **tietokannasta**. Ne vain merkitään poistetuiksi, eikä niitä enää näytetä sovelluksessa. Moveatis-sovellus ei tarjoa ylläpitäjälle toimintoa, jonka avulla tietokannan tauluja voisi tarkastella. **Tietokantaan liittyvät ylläpitotoimet** tulee tehdä tietokannan omilla työkaluilla tai muilla, tietokannan hallinta ohjelmistoilla, kuten pgAdminilla.

Sovellukseen kirjautuminen Jyväskylän yliopiston tunnuksilla tapahtuu siten, että Moveatis-sovelluksen kanssa samalla palvelimella on asennettuna Apache httpd -palvelin, joka hoitaa liikenteen Moveatis-sovelluksen ja tunnistautumispalvelun välillä. Moveatis-sovellus ja Apache httpd kommunikoivat AJP-protokollalla Java EE -sovelluspalvelimen portin 8009 kautta. Kommunikaatio asetetaan Apachen ja sovelluspalvelimen asetuksia muuttamalla, eikä Moveatis-sovelluksen lähdekoodiin tarvitse koskea. Tunnistautumispalvelun käyttöönotto on oma taiteenlajinsa, jota varten luvussa ei voida antaa tyhjentävää ohjeistusta. Mikäli mahdollista, tutustu Moveatis-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.

8 Yhteenveto

Moveo-projekti kehitti Moveatis-sovellukseen palauteanalyysitoiminnallisuuden, jonka avulla voidaan kirjata ja raportoida opetustapahtuman aikana annettua palautetta. Sovellusta tullaan käyttämään Jyväskylän yliopiston liikuntatieteellisen tiedekunnan opettajakoulutuksessa erityisesti *Tutkiva opettaja* -opintojaksolla opettajaopiskelijoiden palautteen annon arviointiin. Moveatis-sovellusta on käytetty opintojaksolla opetustapahtumien tarkkailuun jo aiempina vuosina, ja Moveo-projektin kehittämän lisäosan ansiosta sitä voidaan nyt käyttää entistä laajemmin. Moveatis-sovelluksen observointiominaisuus on suorittanut hyvin tavoitteensa korvata opetustapahtumien observointiin aiemmin käytetyn Lotas-observer-ohjelmiston. Vastaavasti toivotaan palauteanalyysitoiminnallisuuden kykenevän korvaamaan tehtävään aiempina vuosina käytetyn Excel-kaavakkeen. Moveatis-sovelluksen ansiosta palauteanalyysi on helposti suoritettavissa käytännössä millä tahansa laitteella, jolla on pääsy Internetiin. Jatkokehitetty Moveatissovellus mahdollistaa myös analyysin tulosten jatkokäsittelyn, sillä tulokset saadaan sovelluksesta tallennettua taulukko-ohjelmistojen yleisesti tuntemassa CSV-muodossa.

Sovellus vastaa sille määritettyjä pakollisia vaatimuksia. Vastaavasti kuin sovellukseen Moveatis-projektin toteuttamassa observointitoiminnallisuudessa, voidaan sovellukseen tallentaa Moveo-projektin toteuttamilla ominaisuuksilla palauteanalyysin kategoriaryhmiä ja kategorioita. Niihin pääsee käsiksi niiden JYU-tunnuksilla kirjautuneen omistajan lisäksi ryhmäavaimella kuka tahansa sovelluksen käyttäjä. Sovellusta voi käyttää myös kirjautumatta, jolloin kategoriaryhmät ja kategoriat täytyy määrittellä jokaista analyysia varten erikseen. Sovellus tarjoaa kirjautuneelle käyttäjälle myös mahdollisuuden tarkastella ja muokata vanhoja analyysitapahtumien kirjauksia.

Moveo-projektissa toteutusratkaisut pyrittiin tekemään yhdenmukaisesti Moveatis-projektin kanssa. Ainoat poikkeavat toteutusratkaisut liittyivät Moveatis-projektin kehittämisiin komponentteihin, joita ei jatkokehityksessä saatu toimimaan halutulla tavalla.

Lisäksi Moveo-projekti päätti luoda uudet tietokantataulut palauteanalyysiä varten. Tämän ratkaisun avulla Moveatis-tietojärjestelmän tietokantaan aiemmin tallennetut tiedot pystyttiin helposti säilyttämään tietokannassa muuttumattomina.

Lähteet

- [1] Petra Puumala, "Moveo-projekti, vaatimusmäärittely", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2019, <http://sovellusprojektit.it.jyu.fi/moveo/vaatimusmäärittely>.
- [2] Karoliina Lappalainen, "Moveo-projekti, projektiraportti", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2019, <http://sovellusprojektit.it.jyu.fi/moveo/dokumentit/projektiraportti>.
- [3] Karoliina Lappalainen, "Moveo-projekti, Järjestelmätestausraportti Android", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2019, <http://sovellusprojektit.it.jyu.fi/moveo/testaus/testausraportit/>.
- [4] Karoliina Lappalainen, "Moveo-projekti, Järjestelmätestausraportti Iphone", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2019, <http://sovellusprojektit.it.jyu.fi/moveo/testaus/testausraportit>.
- [5] Visa Nykänen, "Moveo-projekti, Asennusohjeet", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2019, <http://sovellusprojektit.it.jyu.fi/moveo/dokumentit/ohjeet>.
- [6] Visa Nykänen, "Moveo-projekti, Tuotantopalvelimen ylläpito-ohjeet", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2019, <http://sovellusprojektit.it.jyu.fi/moveo/dokumentit/ohjeet>.
- [7] Sami Kallio, Kai Korhonen, Ilari Paananen, Juha Moisio, Jarmo Juujärvi, "Moveatis-projekti, Sovellusraportti", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2016, <http://sovellusprojektit.it.jyu.fi/moveatis/dokumentit/sovellusraportti/>.
- [8] Karoliina Lappalainen, "Moveo-projekti, Järjestelmätestausraportti PC", Jyväskylän yliopisto, informaatioteknologian tiedekunta, 2019, <http://sovellusprojektit.it.jyu.fi/moveo/testaus/testausraportit/>.