

Kodavi-sovellusprojekti

**Lauri Antila
Outi Hilola
Antti Kauppi
Nuutti Rantanen
Anne Vaarala**

Sovellusraportti

**Julkinen
Versio 0.1.0
4.12.2020**

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

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

Tietoja dokumentista

Tekijät:

Lauri Antila (LA) lauri.s.antila@student.jyu.fi

Outi Hilola (OH) outi.m.hilola@student.jyu.fi

Antti Kauppi (AK) antti.p.kauppi@student.jyu.fi

Nuutti Rantanen (NR) l.nuutti.a.rantanen@student.jyu.fi

Anne Vaarala (AV) anne.m.vaarala@student.jyu.fi

Dokumentin nimi: Kodavi-projekti, Sovellusraportti

Sivumäärä: 43

Tiedosto: kodavi_sovellusraportti_0.1.0.docx

Tiivistelmä: Kodavi-sovellusprojektin sovellusraportti kuvaa Kodavi-projektissa kehitetyn tietojärjestelmän käyttöliittymää, toteutusratkaisuja, tietokannan rakennetta, tavoitteiden toteutumista sekä jatkokehitysideoita.

Avainsanat: data, HBSC, jatkokehitys, koululaistutkimus, käyttöliittymä, liikuntatieteellinen tiedekunta, ohjeet, sovellus, sovellusprojekti, tavoitteet, tietojärjestelmä, tietokanta, toteutusratkaisut, visualisointi, WHO,

Muutoshistoria

Versio	Päivä	Muutokset	Tekijä
0.0.1	12.11.2020	Sovellusraportin laatiminen aloitettiin. Luvut 4, 5 ja 7.8 lisättiin.	AK
0.0.2	25.11.2020	Luku 7.9 lisättiin.	AV
0.0.3	27.11.2020	Muokattiin luvun 6 ja 7 otsikoita ja poistettiin turhia tekstejä.	OH
0.0.4	27.11.2020	Lisätty luku 3. Muokattu lukua 4. Muokattu yleisesti.	LA
0.0.5	2.12.2020	Lisätty yhteystiedot.	LA
0.0.6	3.12.2020	Jatkettu lukua 3. Lisätty johdanto. Lisätty luku 2.	LA
0.0.7	3.12.2020	Lisättiin tekstiä lukuihin 6, 7 ja 7.1–7.3.	OH
0.0.8	3.12.2020	Lisättiin lähteet ja korjattiin viittausten numeroita.	OH
0.0.9	3.12.2020	Muokattiin tekijöiden tietoja. Muokattu yleisesti.	AV
0.0.10	4.12.2020	Lisätty luku 8. Lisätty tiivistelmä ja avainsanat.	LA
0.0.11	4.12.2020	Muotoiltu tekstiä yleisesti.	AV
0.0.12	4.12.2020	Etusivun tyylejä muutettiin ja korjattiin tekstistä havaittuja virheitä.	OH
0.1.0	4.12.2020	Versionumero vaihdettiin.	OH

Tietoja projektista

Kodavi-projektin Study Health with HBSC on Jyväskylän yliopiston liikuntatieteellisen tiedekunnan ja informaatioteknologian tiedekunnan yhteistyönä kehittämä WWW-sovellus, joka avaa julkiseen käyttöön WHO-Koululaistutkimuksen datan vuosilta 2002–2018. Sovellus käyttöliittymä toteutettiin suomeksi, englanniksi ja ruotsiksi.

Tekijät:

Lauri Antila (LA)	lauri.s.antila@student.jyu.fi
Outi Hilola (OH)	outi.m.hilola@student.jyu.fi
Antti Kauppi (AK)	antti.p.kauppi@student.jyu.fi
Nuutti Rantanen (NR)	l.nuutti.a.rantanen@student.jyu.fi
Anne Vaarala (AV)	anne.m.vaarala@student.jyu.fi

Tilaajan edustajat:

Nelli Lyyra	nelli.lyyra@jyu.fi
Kristiina Ojala	kristiina.ojala@jyu.fi
Jorma Tynjälä	jorma.a.tynjala@jyu.fi

Ohjaajat:

Jukka-Pekka Santanen	santanen@mit.jyu.fi
Enni Stylman	enni.m.stylman@jyu.fi
Arttu Ylä-Sahra	arttu.e.yla-sahra@student.jyu.fi

Yhteystiedot:

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

Sisältö

1 Johdanto	1
2 Termit.....	2
2.1 Käyttöliittymän termejä	2
2.2 Kohdealueen ja raportin termejä.....	4
3 Sovelluksen käyttöliittymä ja käyttö.....	8
3.1 Sovelluksen käyttöliittymän rakenne ja mobiilikäytön erot	8
3.2 Sovelluksen etusivun näkymä.....	9
3.3 Yläotsake ja navigointipalkki	10
3.4 Etusivun sisältö	11
3.5 Hakusivu	12
3.6 Hakusivun hakuuehtopalkki.....	12
3.7 Hakutulosten esittäminen.....	13
4 Sovelluksen rakenne ja toteutusratkaisut	17
4.1 Kehitystyökalut	17
4.2 Sovelluksen toiminnallisuus ja toimintalogiikka	18
4.3 Sovelluspalvelin.....	20
5 Sovelluksen tietokanta	21
5.1 Tutkimusdatan tallentamiseen liittyvät taulut.....	21
5.2 Käyttäjät ja roolit	26
5.3 Tapahtumiin liittyvät taulut.....	28
5.4 Tehtäviin, vastauksiin ja esitystapoihin liittyvät taulut.....	30
6 Sovellukselle asetettujen vaatimusten toteutuminen	33
6.1 Sovellusprojektissa toteutetut vaatimukset	33
6.2 Jatkokehitykseen sovitut vaatimukset.....	34

7 Ideoita sovelluksen jatkokehitykseen	35
7.1 Hakusivun jatkokehitysideat	35
7.2 Käyttäjäroolien jatkokehitysideat	36
7.3 Tapahtumien ja tehtävien jatkokehitysideat.....	37
7.4 Tietokannan jatkokehitysideat	37
7.5 Sovelluksen lähdekoodin jatkokehitysideat	40
8 Yhteenveto	41
Lähteet.....	42

1 Johdanto

Kodavi-sovellusprojektissa suunniteltiin ja toteutettiin vuonna 2020 Jyväskylän yliopiston liikuntatieteelliselle tiedekunnalle WWW-sovellus, jonka avulla käyttäjät voivat hakea WHO-Koululaistutkimuksen tuloksia ja tarkastella niitä visuaalisesti. WHO-Koululaistutkimuksessa on kerätty tietoa 5.-, 7.- ja 9.-luokkalaisten terveyden kokemuksista ja terveystottumuksista 1980-luvulta lähtien. Tutkimus on kansainvälinen, mutta sovellusprojekti rajattiin kansallisen, vuodesta 2002 alkaen kerätyn datan hakuun ja esittämiseen.

Sovellusraportti kuvaa toteutetun sovelluksen käyttöliittymää, rakennetta, toteutusratkaisuja sekä tietokantaa. Lisäksi raportti kuvaa projektin tavoitteiden toteutumista sekä jatkokehitykseen liittyviä asioita. Sovellusraporttia kirjoitettaessa on hyödynnetty Kodavi-projektin vaatimusmäärittelydokumenttia [4], projektiraporttia [7], järjestelmätestausraportteja [1], [8], [9] ja [10] sekä palvelimen asennusohjetta [3]. Sovellusraportti on laadittu Moveo-projektin sovellusraportin pohjalta [6].

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 kuvataan jatkokehitykseen sovitut ominaisuudet. Luvussa 7 on ideoita sovelluksen jatkokehitystä varten.

2 Termit

Luvussa kuvataan Kodavi-projektin kehittämään sovellukseen liittyviä termejä. Käyttöliittymän termit sekä kohdealueen ja raportin termit on jaoteltu omiin alalukuihinsa.

2.1 Käyttöliittymän termejä

Etusivu (front page)	on sovelluksen <i>etusivu</i> , josta <i>käyttäjä</i> voi siirtyä <i>hakusivulle</i> .
Graafi (graph)	on numeerisista tiedoista muodostettu kuvaaja, eli kaavio.
Hakuehdot (search rules)	käsittävät <i>teeman</i> , <i>muuttujan</i> sekä <i>luokittelevat tekijät</i> , jotka ovat olleet valittuina, kun <i>käyttäjä</i> on painanut <i>Hae</i> -painiketta.
Hakutulokset (search results)	tarkoittavat <i>kuvaajia</i> ja <i>taulukoita</i> , jotka esitetään <i>käyttäjälle</i> valittujen <i>hakuehto</i> jen perusteella.
Haun tekeminen	tarkoittaa <i>hakuehto</i> jen eli <i>teeman</i> , <i>muuttujan</i> ja <i>luokittelevien tekijöiden</i> valitsemista sekä <i>Hae</i> -painikkeen painamista.
Hampurilaisvalikko (hamburger menu)	on usein mobiilikäyttöliittymissä esiintyvä valikkopainike, joka piilottaa valikon vaihtoehdot ennen kuin sitä painetaan.
Kielivalinta (language selection)	koostuu kieliä kuvastavista kahden merkin mittaisista lyhennöksistä, joita painamalla voi vaihtaa <i>käyttöliittymän</i> kielen.
Käyttäjä (user)	on henkilö, joka käyttää sovellusta.
Käyttöliittymä (user interface)	on sovelluksen osa, jonka <i>käyttäjä</i> näkee ja jonka kautta <i>käyttäjä</i> käyttää sovellusta.

Luokitteleva tekijä (classifying variable)	on muuttuja, jonka avulla hakutulokset ryhmitellään ja esitetään sovelluksessa. Sovelluksessa valittavissa olevat luokittelevat tekijät ovat luokkataso, sukupuoli, vastausvuosi, opetuskieli ja alue.
Mobiilinäkymä (mobile view)	on mobiililaitteiden pienemmille ruuduille sopivammaksi skaalattu näkymä, joka eroaa <i>työpöytänäkymästä</i> .
Muuttuja (variable)	vastaa WHO-Koululaistutkimuksen kysymystä tai väittämää, esim. <i>päänsäryn useus viimeisen 6kk:n aikana</i> tai <i>luotan paljon opettajiini</i> .
Navigointipalkki (navigation bar)	on palkki, jolla navigoinnista vastaavat painikkeet sijaitsevat. <i>Navigointipalkin</i> painikkeilla voi siirtyä <i>etusivun</i> ja <i>hakusivun</i> välillä.
Päätelaite (end device)	<i>käyttäjän</i> käyttämä laite sovelluksen käytössä.
Taulukko (table)	on <i>taulukko</i> , jossa esitetään <i>hakutulokset</i> prosentuaalisesti.
Työkaluvihje (tooltip)	tulee esiin, kun pidät hiiren osoitinta graafin vastauspalkin päällä. Näyttää värikoodit, <i>muuttujat</i> ja prosenttiosuudet.
Työpöytänäkymä (desktop view)	on palkki, jolla navigoinnista vastaavat painikkeet sijaitsevat. on näkymä, jonka <i>käyttäjä</i> näkee tietokoneen ruudulla. <i>Työpöytänäkymä</i> eroaa hieman <i>mobiilinäkymästä</i> .
Vastausvaihtoehdot	ovat vaihtoehtoja, joista WHO-Koululaistutkimukseen vastannut henkilö on valinnut itseään ja tilannettaan parhaiten vastaavan vaihtoehdon kunkin kysymyksen osalta. Tietokantaan tallennetussa dataassa erilaisia vastausvaihtoehtoja on kussakin muuttujassa 2-4.
Yläotsake (header)	on <i>käyttöliittymän</i> yläosassa näkyvä palkki, josta löytyy sovelluksen nimi, HBSC:n logo sekä grafiikkaa. Tämän alapuolella on <i>navigointipalkki</i> .

2.2 Kohdealueen ja raportin termejä

Apache Tomcat	on yliopiston tuotantopalvelimella toimiva sovelluspalvelin.
Babel	on <i>JSX-syntaksilaajennoksiin</i> käytetty kääntäjä. Kääntää <i>React</i> -koodin <i>JavaScriptiksi</i> .
Backend	on sovelluksen taustalogiikka, joka toimii palvelimella.
Bootstrap	on sovelluskehys, jota on hyödynnetty <i>käyttöliittymän</i> komponenteissa.
Create React App	on työkalu, jolla sovelluksen <i>React.js</i> -projekti pystytettiin.
CSS (cascading style sheets)	on sovelluksen käyttöliittymässä käytetty tyyliohjejärjestelmä käyttöliittymän tyylittelyyn.
Frontend	on sovelluksen käyttäjälle näkyvä osa, jossa käyttäjä on tekemisissä sovelluksen kanssa.
Esitystavat	tarkoittavat tapoja, joilla hakutulokset esitetään käyttäjälle. Projektissa esitystavoista toteutettiin taulukko ja pinottu pylväskaavio. Muita esitystapoja voisivat olla esimerkiksi trendiviiva ja piirakkakaavio.
HTML (hypertext markup language)	on kuvauskieli, jota on käytetty sovelluksen käyttöliittymässä.
HTTP-pyyntö (HTTP request)	on <i>käyttöliittymän</i> lähettämä pyyntö, johon <i>backend</i> vastaa tuomalla dataa <i>tietokannasta käyttöliittymän</i> käsiteltäväksi ja esitettäväksi.
i18next-react	on sovelluksessa käytetty <i>JavaScript</i> -pohjainen käännöskirjasto.
Java	on sovelluksen <i>backendin</i> kehityksessä käytetty ohjelmointikieli.

JavaScript	on sovelluksen käyttöliittymän ohjelmoinnissa käytetty ohjelmointikieli.
JSON (JavaScript Object Notation)	on yksinkertainen tiedostomuoto tiedonvälitykseen. <i>Backend</i> ja <i>Frontend</i> keskustelevat käyttäen <i>JSONia</i> .
JSX-syntaksilaajennokset	ovat syntaksilaajennoksia <i>JavaScriptiin</i> , joita käytetään <i>React.js</i> -ohjelmoinnissa.
Julkinen käyttäjä (public user)	on käyttäjä, joka ei ole kirjautunut sovellukseen.
Katselmointi (review)	on teknisen ohjaajan suorittama <i>lähdekoodin</i> arviointi- ja kommentointitilaisuus.
Kerrosarkkitehtuuri	tarkoittaa sitä, kuinka sovellus jakautuu kerroksiin, jotka toimivat vuorovaikutuksessa toistensa kanssa.
Kirjautunut käyttäjä	tarkoittaa <i>Peda.net</i> -tunnuksella tai <i>ryhmävaimella kirjautunutta käyttäjää</i> .
Luokiteltu ryhmä	on valittujen <i>luokittelevien tekijöiden</i> mukaan muodostettu vastaajajoukko. Esimerkiksi jos luokittelevaksi tekijäksi on valittu sukupuoli, saadaan kaksi ryhmää, tytöt ja pojat. Jos valitaan lisäksi toinen <i>luokitteleva tekijä</i> , molemmat ryhmät jakautuvat pienempiin ryhmiin.
Lähdekoodi	on projektiryhmän jäsenten tuottama koodi, mistä sovellus koostuu.
Maven	on ohjelmistokehitystyökalu, jota käytettiin taustaohjelmistoprojektin hallinnassa.
Nginx	on palvelimella toimiva välityspalvelin.
npm (node package manager)	on paketinhallintajärjestelmä, jota käytettiin <i>frontendin</i> kehityksessä.

Peda.net	on verkkoympäristö, jota hyödynnetään opetuksessa.
PostgreSQL	on projektissa käytetty tietokannanhallintajärjestelmä.
Pääkäyttäjä	on <i>Peda.net</i> -tunnuksella kirjautunut <i>käyttäjä</i> , jolle on myönnetty oikeudet määritellä muuttujien näkyvyys sekä myöntää <i>pääkäyttäjän</i> oikeuksia muille <i>käyttäjille</i> .
React.js	on <i>JavaScript</i> kirjasto, joka käyttää <i>JSX</i> -syntaksia. Sovelluksen <i>käyttöliittymä</i> kehitettiin käyttäen <i>Reactia</i> .
Recharts.js	on visualisointikirjasto, jota käytettiin <i>käyttöliittymän graafien</i> toteuttamisessa.
REST-rajapinta	on rajapinta, jonka kautta sovelluksen <i>frontend</i> ja <i>backend</i> keskustelvat käyttäen <i>HTTP</i> -pyyntöjä.
Ryhmäavaimella kirjautunut käyttäjä	on kirjautunut tapahtumaan liitettyllä aktiivisella ryhmäavaimella.
Spring Boot	on sovelluskehys, jota käytettiin sovelluksen <i>backendin</i> kehityksessä.
Sovelluslogiikka	tarkoittaa sovelluksen taustaohjelmistoa ja sen toimintoja.
Tapahtuma	on <i>Peda.net</i> -tunnuksilla kirjautuneen <i>käyttäjän</i> luoma kokoelma, johon voi olla liitettynä <i>muuttujia</i> , <i>luokittelevia tekijöitä</i> ja <i>tehtäviä</i> . <i>Tapahtumaan</i> liitetyn ryhmäavaimen kautta muut <i>käyttäjät</i> voivat kirjautua tapahtumaan tekemään hakuja <i>tapahtumaan</i> liitettyillä <i>muuttujilla</i> ja tallentamaan vastauksia <i>tehtäviin</i> .
Tapahtuman luonut käyttäjä	on <i>Peda.net</i> -tunnuksella kirjautunut käyttäjä, joka on luonut tietyn tapahtuman.

Tapahtuman muokausoikeuden omaava käyttäjä	on Peda.net-tunnuksella kirjautunut käyttäjä, joka on luonut tietyn tapahtuman tai jolle tapahtuman luonut käyttäjä on antanut oikeudet muokata tapahtumaa.
Tehtävä	on tapahtumaan liitetty sanallinen <i>tehtävä</i> , johon <i>tapahtumaan</i> kirjautunut voi vastata.
Tehtävän vastaus	sisältää ryhmäavaimella kirjautuneen käyttäjän tallentaman kirjallisen vastauksen <i>tapahtumaan</i> liitettyyn <i>tehtävään</i> sekä <i>hakuehdot</i> , joilla tallennusta edeltävä haku on tehty.
Tietokanta	on palvelimella sijaitseva dataa sisältävä kanta, josta sovelluksen taustaohjelmisto hakee dataa.
Tietokannanhallintajärjestelmä	katso <i>PostgreSQL</i> .
Webpack	on <i>Create React Appin</i> sisältämä moduulipaketoija.

3 Sovelluksen käyttöliittymä ja käyttö

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

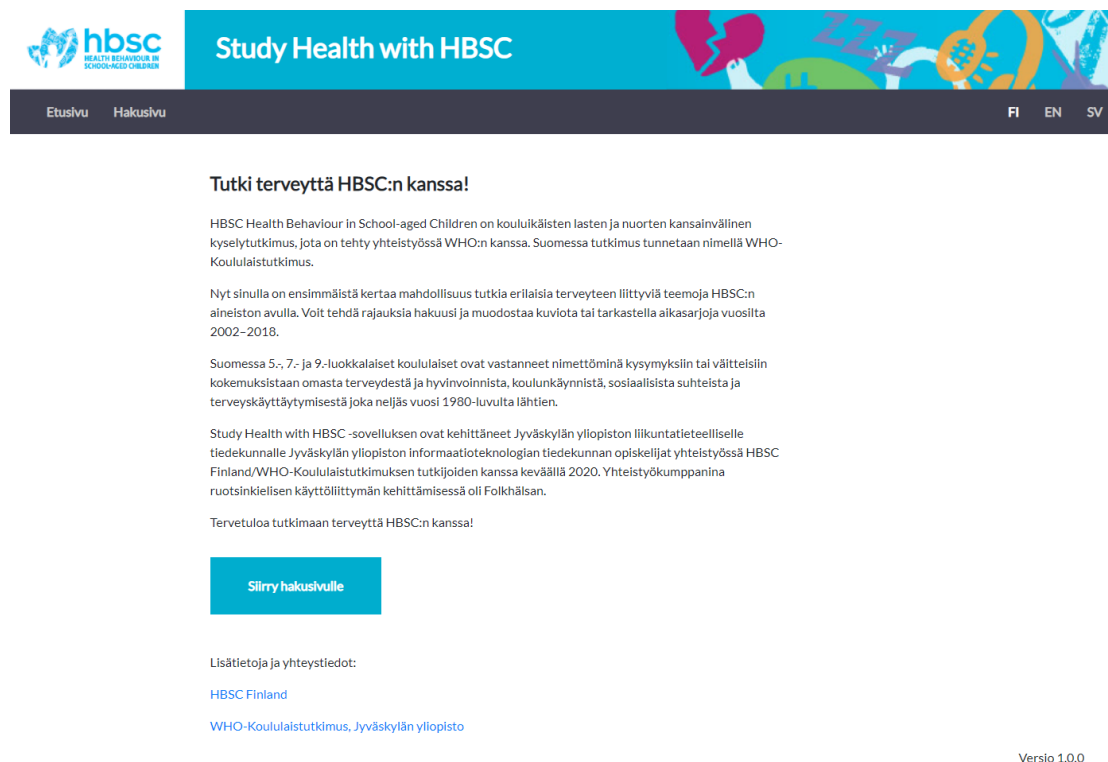
3.1 Sovelluksen käyttöliittymän rakenne ja mobiilikäytön erot

Sovelluksen käyttöliittymä koostuu etusivusta ja hakusivusta. Ylhäällä oleva yläotsake on molemmilla sivuilla samanlainen. Yläotsakkeen alla on navigointipalkki, jossa voi vaihtaa sivua ja kielivalintaa. Varsinainen sovellussisältö on keskellä. Sovelluksen sivut esitetään tarkemmin seuraavissa luvuissa.

Työpöytänäkymästä poiketen navigointipalkki muuntuu mobiilinäkymässä niin kutsuksi hampurilaisvalikoksi, jota klikkaamalla saa vaihtoehdot esiin. Mobiilikäytössä ja pienemmän resoluution monitoreilla hakusivun hakupalkki siirtyy vasemmasta reunasta sovelluksen näkymän yläosaan, navigointipalkin alle. Hakutulokset ja graafi esitetään tämän alla. Sovelluksen toiminnallisuudet ovat oleellisesti samat päätelaitteesta riippumatta. Raportissa käsitellään sovelluksen toimintaa ja näkymiä työpöytäkäytössä.

3.2 Sovelluksen etusivun näkymä

Etusivu koostuu esittelytekstistä, hakusivulle siirtymisen painikkeesta sekä lisätieto- ja yhteystietolinkeistä kuvan 3.2.1 mukaisesti. Oikealla alakulmassa näkyy sovelluksen versionumero. Siirry hakusivulle -painikkeella voidaan siirtyä sovelluksen hakusivulle. Sovelluksen käyttötavat on kuvattu tarkemmin seuraavissa luvuissa.



Tutki terveyttä HBSC:n kanssa!

HBSC Health Behaviour in School-aged Children on kouluikäisten lasten ja nuorten kansainvälinen kyselytutkimus, jota on tehty yhteistyössä WHO:n kanssa. Suomessa tutkimus tunnetaan nimellä WHO-Koululaistutkimus.

Nyt sinulla on ensimmäistä kertaa mahdollisuus tutkia erilaisia terveyteen liittyviä teemoja HBSC:n aineiston avulla. Voit tehdä rajauksia hakuasi ja muodostaa kuviota tai tarkastella aikasarjoja vuosilta 2002–2018.

Suomessa 5., 7- ja 9-luokkalaisten koululaiset ovat vastanneet nimettöminä kysymyksiin tai väitteisiin kokemuksistaan omasta terveydestä ja hyvinvoinnista, koulunkäynnistä, sosiaalisista suhteista ja terveyskäyttäytymisestä joka neljäs vuosi 1980-luvulta lähtien.

Study Health with HBSC -sovelluksen ovat kehittäneet Jyväskylän yliopiston liikuntatieteelliselle tiedekunnalle Jyväskylän yliopiston informaatioteknologian tiedekunnan opiskelijat yhteistyössä HBSC Finland/WHO-Koululaistutkimuksen tutkijoiden kanssa keväällä 2020. Yhteistyökumppanina ruotsinkielisen käyttöliittymän kehittämisessä oli Folkhälsan.

Tervetuloa tutkimaan terveyttä HBSC:n kanssa!

[Siirry hakusivulle](#)

Lisätietoja ja yhteystiedot:

[HBSC Finland](#)

[WHO-Koululaistutkimus, Jyväskylän yliopisto](#)

Versio 1.0.0

Kuva 3.2.1: Sovelluksen etusivu.

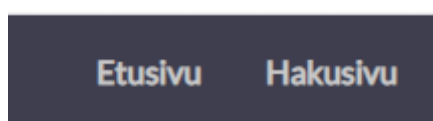
3.3 Yläotsake ja navigointipalkki

Yläotsakkeen vasemmassa laidassa on HBSC:n kansainvälinen logo. Logon oikealla puolella on sovelluksen nimi sekä grafiikkaa (katso kuva 3.3.1). HBSC:n logo toimii myös linkkinä takaisin sovelluksen etusivulle. Yläotsakkeen alla on navigointipalkki.



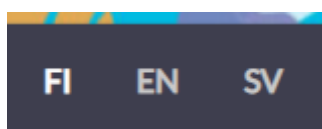
Kuva 3.3.1: Yläotsake ja navigointipalkki.

Käyttäjä voi siirtyä sivulta toiselle käyttäen navigointipalkin sivujen navigointipainikkeita (katso kuva 3.3.2). Kun kursorin siirtää painikkeen kohdalle, painikkeen teksti muuttuu hohtavan valkoiseksi. Tämä antaa käyttäjälle visuaalista palautetta kursorin sijainnista.



Kuva 3.3.2: Sivujen navigointi.

Kielivalinnasta (katso kuva 3.3.3) käyttäjä voi vaihtaa sovelluksen kieltä. Valittavissa on suomi, englanti sekä ruotsi. Nämä on kuvattu kielten ensimmäisellä kahdella kirjaimella. Valittuna kielenä on oletusarvoisesti suomi. Valittu kieli erottuu hohtavan valkoisena muista vaihtoehdoista.



Kuva 3.3.3: Kielivalinta.

3.4 Etusivun sisältö

Etusivun tekstissä esitellään HBSC:n tutkimus sekä tämä sovellus (katso kuva 3.4.1). Tervetulotekstin alla on *Siirry hakusivulle* -painike sekä lisätietoja ja yhteystietoja antavat linkit (katso kuva 3.4.2). Ensimmäinen linkki vie HBSC:n suomalaiselle sivustolle ja toinen linkki Jyväskylän yliopiston sivuille, jossa kerrotaan WHO-Koululaistutkimuksesta. Sovelluksen kieltä vaihtamalla etusivun kaikki teksti sekä myös linkit vaihtuvat halutulle kielelle.

Tutki terveyttä HBSC:n kanssa!

HBSC Health Behaviour in School-aged Children on kouluikäisten lasten ja nuorten kansainvälinen kyselytutkimus, jota on tehty yhteistyössä WHO:n kanssa. Suomessa tutkimus tunnetaan nimellä WHO-Koululaistutkimus.

Nyt sinulla on ensimmäistä kertaa mahdollisuus tutkia erilaisia terveyteen liittyviä teemoja HBSC:n aineiston avulla. Voit tehdä rajauksia hakuusi ja muodostaa kuviota tai tarkastella aikasarjoja vuosilta 2002–2018.

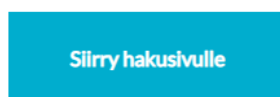
Suomessa 5.-, 7.- ja 9.-luokkalaiset koululaiset ovat vastanneet nimettöminä kysymyksiin tai väitteisiin kokemuksistaan omasta terveydestä ja hyvinvoinnista, koulunkäynnistä, sosiaalisista suhteista ja terveystietoisuudesta joka neljäs vuosi 1980-luvulta lähtien.

Study Health with HBSC -sovelluksen ovat kehittäneet Jyväskylän yliopiston liikuntatieteelliselle tiedekunnalle Jyväskylän yliopiston informaatioteknologian tiedekunnan opiskelijat yhteistyössä HBSC Finland/WHO-Koululaistutkimuksen tutkijoiden kanssa keväällä 2020. Yhteistyökumppanina ruotsinkielisen käyttöliittymän kehittämisessä oli Folkhälsan.

Tervetuloa tutkimaan terveyttä HBSC:n kanssa!

Kuva 3.4.1: Esittelyteksti.

Tervetuloa tutkimaan terveyttä HBSC:n kanssa!



Lisätietoja ja yhteystiedot:

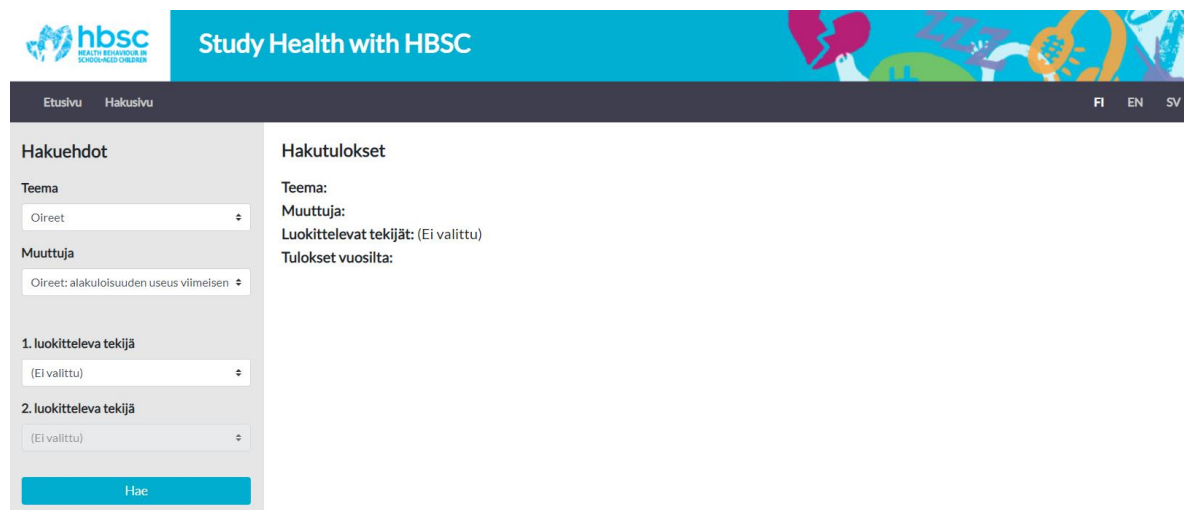
[HBSC Finland](#)

[WHO-Koululaistutkimus, Jyväskylän yliopisto](#)

Kuva 3.4.2: Siirry hakusivulle -painike sekä linkit.

3.5 Hakusivu

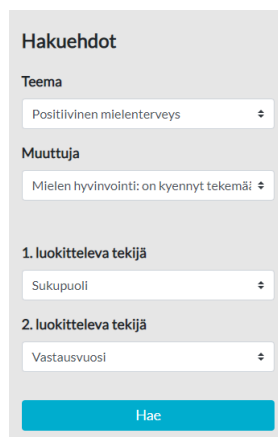
Painettuaan *Siirry hakusivulle* -painiketta (katso kuva 3.4.2), käyttäjä ohjataan hakusivulle (katso kuva 3.5.1). Käyttäjä näkee vasemmalla puolella palkin, jossa esitetään hakuehdot. Palkin oikealla puolella näkyvät hakutulokset. Hakutuloksia ei esitetä ennen kuin käyttäjä on valinnut hakuehdot ja klikannut *Hae*-painiketta (katso kuva 3.6.1).



Kuva 3.5.1: Hakusivu ilman tuloksia.

3.6 Hakusivun hakuehtopalkki

Näkymän vasemmassa reunassa sijaitsevassa hakuehtopalkissa (katso kuva 3.6.1) voi tehdä erilaisia valintoja hakuun. Haku käynnistetään painamalla *Hae*-painiketta.

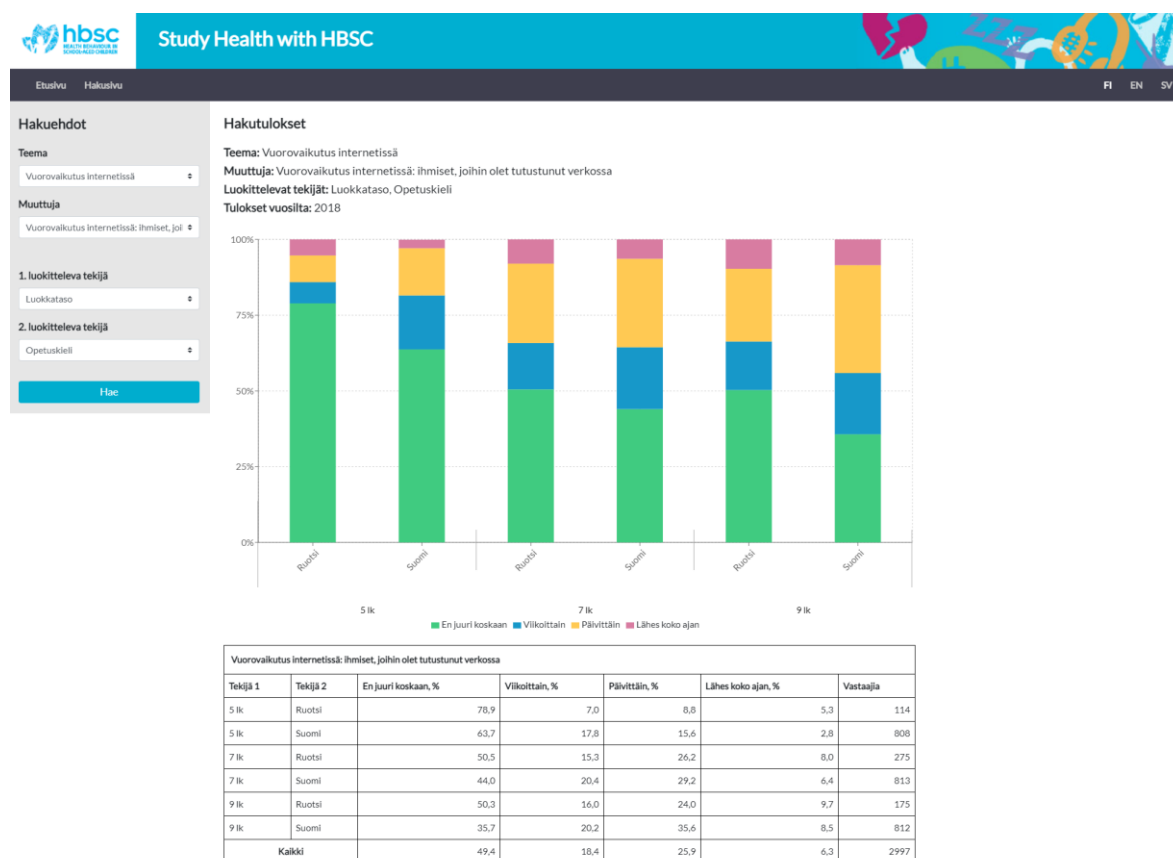


Kuva 3.6.1: Hakuehtopalkki ja Hae-painike.

Hakuehdoissa ylimpänä on Teema. Teemaa vaihtamalla käyttäjä määrittää, miltä tutkimuksen osa-alueelta tilastoja halutaan tarkastella. Toiseksi valitaan Muuttuja. Muuttuja-valikossa esitetään valittuun Teemaan liittyvät Muuttujat. Muuttujat vastaavat tutkimuksen osa-alueeseen, eli Teemaan liittyviä kysymyksiä.

3.7 Hakutulosten esittäminen

Käyttäjän painettua Hae-painiketta (katso kuva 3.6.1), hakutulokset esitetään hakuehtopalkin oikealla puolella (katso kuva 3.7.1). Hakutuloksissa esitetään ensin valitut hakuehdot, kuten teema, muuttuja(t), luokittelevat tekijät sekä vuosi tai vuodet, joilta tulokset ovat.

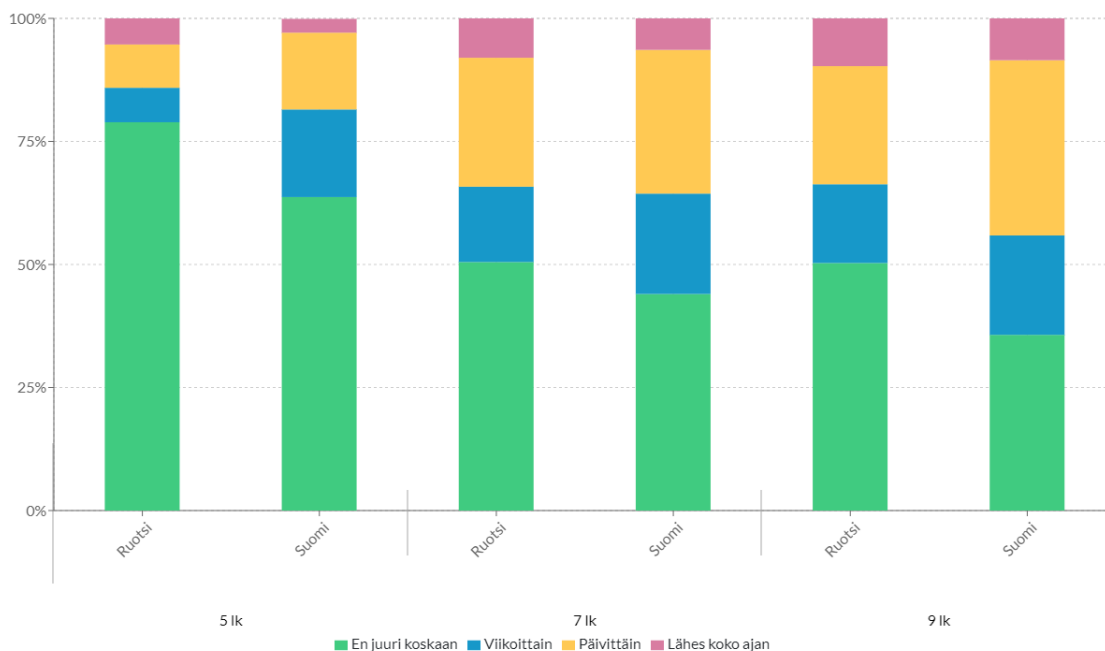


Kuva 3.7.1: Hakesivun näkymä hakutuloksilla.

Hakutuloksissa esitettävien hakuehtojen alla esitetään graafi (katso kuva 3.7.2), johon piirtyvät hakutulosten mukaiset tulokset pinoavat palkit. Palkkien tulokset on värikoodattu siten, että jokainen väri vastaa eri vastausvaihtoehtoa. Palkkien eri väristen osien

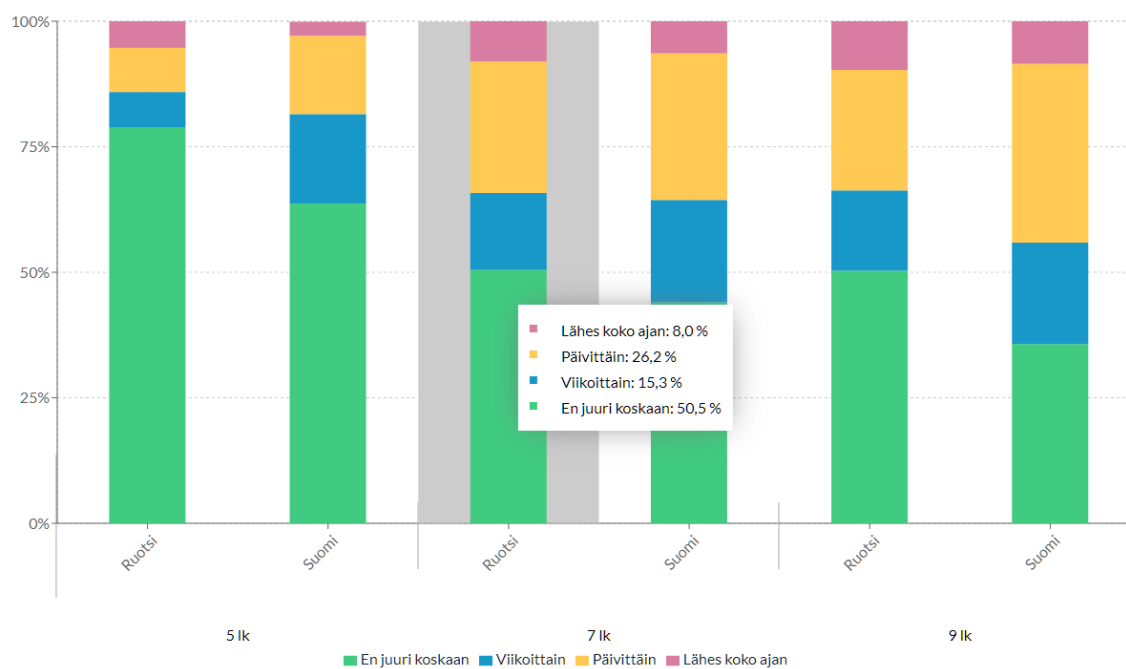
koko vastaa vastauksien prosenttiosuuksia. Vastausvaihtoehtojen värien selitteet ovat graafin alla.

Vastauspalkit jaotellaan luokittelevien muuttujien mukaan siten, että ensimmäinen luokitteleva muuttuja ryhmittää palkit. Jos on valittu toinen luokitteleva muuttuja, se lukee kunkin palkin alla. Kuvassa 3.7.2 on ensimmäisenä luokittelevana tekijänä luokkataso. Luokkatasot esitetään graafin alla (5., 7. ja 9. luokka). Jokaisen luokkatason sisällä on jaoteltuna palkit toiselle valitulle luokittelevalle tekijälle, eli opetuskielille (ruotsi ja suomi).



Kuva 3.7.2: Hakutulosten palkeista koostuva graafi.

Vastauspalkkeja voi tutkia tarkemmin siirtämällä kursori halutun vastauspalkin päälle. Kursorilla osoitetun palkin tausta muuttuu harmaaksi ja kursorin kohdalle aukeaa työkaluvihje (katso kuva 3.7.3). Työkaluvihjeessä kerrotaan vastausvaihtoehdot, joista palkki koostuu sekä lisäksi vastausvaihtoehtojen osuus prosentteina. Kuvassa 3.7.3 työkaluvihje näyttää, että suurin osa palkista koostuu vastauksesta ”En juuri koskaan” osuudella 50,5 prosenttia. Tämän vastauksen väri on vihreä, joten palkin koko pituudesta yli puolet on piirretty vihreällä.



Kuva 3.7.3: Palkin päälle on piirtynyt työkaluvihje.

Graafin alla on taulukko (katso kuva 3.7.4), jossa hakutulokset on eritelty. Taulukon yläosassa kerrotaan valittu Muuttuja. Taulukon sisältö on jaettu sarakkeisiin. Vasemmalta lähtien sarakkeina ovat ensin luokittelevat tekijät, sitten vastausvaihtoehdot ja oikealla vastaajien määrän ilmoittava sarake.

Tulokset on ryhmitelty siten, että ne kuvaavat jokaisen tekijäkombinaation omalla rivillään. Yhden rivin tulokset muodostavat yhden vastauspalkin graafiin. Kuvan 3.7.3 kursorin osoittamassa graafin kolmannessa vastauspalkissa näkyy vastaukset 7. luokan oppilailta, joiden opetuskielenä on ruotsi. Samat vastaukset löytyvät taulukon kolmannelta vastausriviltä.

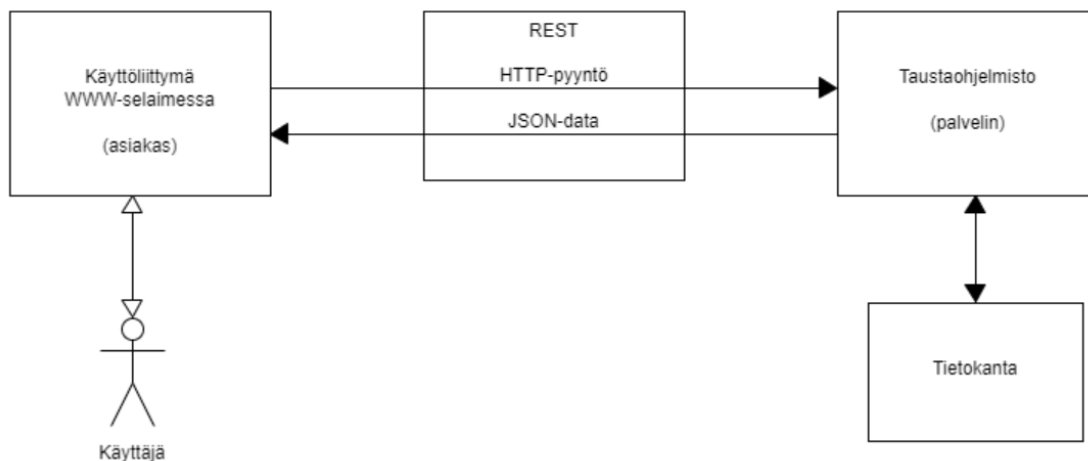
Vuorovaikutus internetissä: ihmiset, joihin olet tutustunut verkossa						
Tekijä 1	Tekijä 2	En juuri koskaan, %	Viikoittain, %	Päivittäin, %	Lähes koko ajan, %	Vastaajia
5 lk	Ruotsi	78,9	7,0	8,8	5,3	114
5 lk	Suomi	63,7	17,8	15,6	2,8	808
7 lk	Ruotsi	50,5	15,3	26,2	8,0	275
7 lk	Suomi	44,0	20,4	29,2	6,4	813
9 lk	Ruotsi	50,3	16,0	24,0	9,7	175
9 lk	Suomi	35,7	20,2	35,6	8,5	812
Kaikki		49,4	18,4	25,9	6,3	2997

Kuva 3.7.4: Hakutuloksista muodostuva taulukko.

Taulukon Vastaajia-sarakkeesta ilmenee, että vastanneita 7. luokan oppilaita, joiden opetuskielenä oli ruotsi, on 275 kappaletta. Taulukon viimeinen rivi, ”Kaikki”, esittää prosenttilukemina kunkin vastausvaihtoehdon prosentuaalisen osuuden kaikista vastausvaihtoehdoista. Lisäksi rivin lopussa kerrotaan kaikkien valituilla hakuehdoilla saattujen yhteenlaskettu määrä, tässä tapauksessa 2997 vastaajaa.

4 Sovelluksen rakenne ja toteutusratkaisut

Sovelluksen arkkitehtuurimallina käytettiin kerrosarkkitehtuuria. Kerrosarkkitehtuurin mukaisesti sovellus jakautuu itsenäisiin kerroksiin, jotka toimivat vuorovaikutuksessa toistensa kanssa. Kerrosarkkitehtuuri mahdollistaa käyttöliittymän ja sovelluslogiikan eristämisen toisistaan, mikä helpottaa muutosten tekemistä ja jatkokehittämistä. Sovelluksen kokonaisrakenne jakautuu asiakaspuolen käyttöliittymäkerrokseen sekä palvelinpuolen ohjain-, palvelu-, tietolähde- ja mallikerrokseen. Kerrosten keskinäistä vuorovaikutusta ja toimintaa on kuvattu luvussa 4.2. Sovelluksen kokonaisrakenne on esitetty kuvassa 4.1.



Kuva 4.1. Sovelluksen kokonaisrakenne.

4.1 Kehitystyökalut

Palvelinpuolen taustaohjelmisto ohjelmoitiin Java-kielellä, ja sen toteutuksessa hyödynnettiin Spring Boot -sovelluskehystä. Spring Bootin avulla kehittäjän on helppo hallita esimerkiksi ohjelmiston tarvitsemia kirjastoriippuvuuksia. Maven-ohjelmistokehitystyökalua käytettiin koko taustaohjelmistoprojektin hallinnassa.

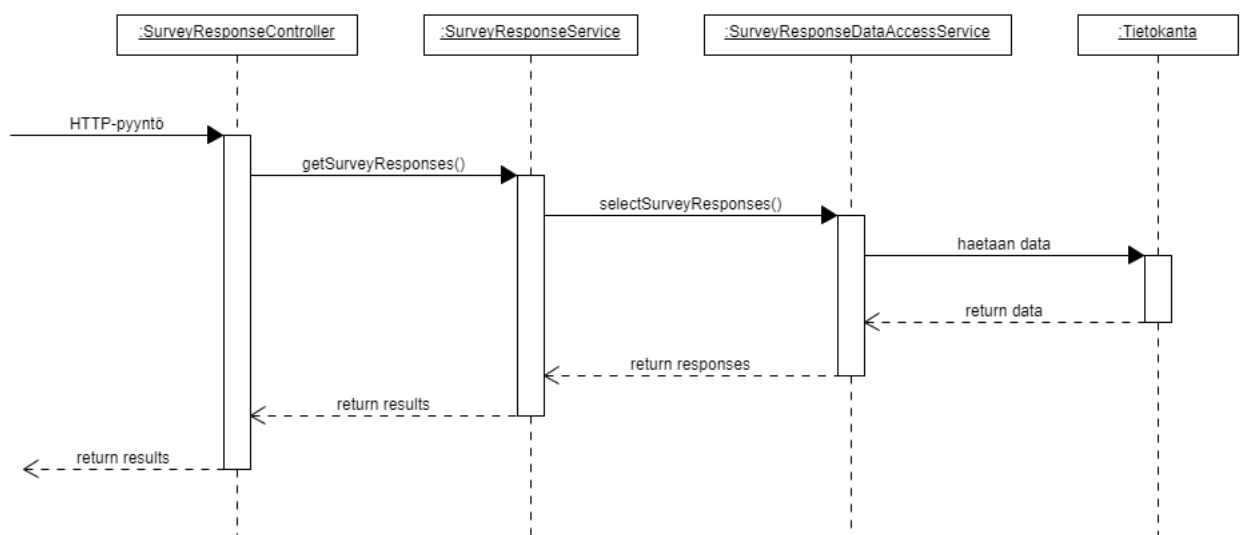
Sovelluksen käyttöliittymän toteutuksessa päätettiin hyödyntää React.js-kirjastoa, joka on hyvin suosittu käyttöliittymien ja niiden komponenttien kehityksessä. React-projekti pystytettiin käyttämällä Create React App -työkalua. Työkaluun sisältyy muun muassa paketinhallintajärjestelmä npm, webpack-moduulipaketoija sekä Babel, jota käytetään JSX-syntaksilaajennoksiin.

Käyttöliittymän ohjelmointikielenä käytettiin JavaScriptia, ja se toteutettiin Node.js-sovellusalustalla. Käyttöliittymän rakenne ja ulkoasu määriteltiin HTML5- ja CSS3-merkintäkielillä. Käyttöliittymän komponenttien luomisessa hyödynnettiin React Bootstrap -sovelluskehystä. Käyttöliittymässä esitettävien kuvaajien toteutuksessa käytettiin apuna hyvin tuettua Recharts.js-visualisointikirjastoa. Käyttöliittymän käännökset toteutettiin i18next-react-kirjastojen avulla.

4.2 Sovelluksen toiminnallisuus ja toimintalogiikka

Käyttäjän tekemät valinnat **käyttöliittymässä** välitetään palvelimelle HTTP-pyyntön mukana REST-rajapintaa hyödyntäen. Pyyntössä välitetään tieto valitusta vastemuuttujasta ja mahdollisista luokittelevista tekijöistä, ja pyyntö lähetetään *Hae*-painiketta painettaessa. Palvelin palauttaa vastauksena käsitellyn datan JSON-muodossa, ja datan perusteella käyttöliittymä luo siitä kuvaajan ja taulukon.

Palvelinpuolen taustaohjelmisto tarjoaa sovelluksen tarvitsemat taustatoiminnot. Taustaohjelmisto suorittaa haut PostgreSQL-tietokantaan käyttäjän käyttöliittymässä tekemien valintojen perusteella, ja se muokkaa hakutulokset käyttöliittymälle sopivaan muotoon. Taustaohjelmisto jakautuu rakenteeltaan ohjain-, palvelu-, tietolähde- ja mallikerrokseen.



Kuva 4.2. Taustaohjelmiston kerrosten välinen vuorovaikutus.

Ohjainkerros vastaanottaa ja ohjaa eteenpäin REST-rajapinnasta saapuvat HTTP-pyynnöt. Pyyntöä käsittelevän jälkeen se palauttaa vastauksen kutsujalleen. Ohjainkerroksen Java-luokista muodostettavat Controller-oliot vastaavat kukin tietyn tehtävän suorittamisesta. Esimerkiksi käyttöliittymän *Hae*-painikkeen klikkausta kuuntelee tietty Controller-olio. Kuvassa 4.2 on esitetty HTTP-pyyntöä eteneminen taustaohjelmistossa, kun käyttöliittymässä on painettu *Hae*-painiketta. Kuvassa 4.2 esitettyjä metodikutsuja on lyhennetty ja parametrit on poistettu, koska ne voisivat muuten kohtuuttomasti tilaa. Ohjainkerroksen Java-luokat tunnistaa luokan `Controller`-pääteestä, ja luokat on sijoitettu pakettiin `com.kodavi.api`.

Palvelukerros tarjoaa palveluita ohjainkerrokselle, ja se sisältää tietojärjestelmän sovelluslogiikan. Kukin palvelukerroksen Java-luokista muodostettu Service-olio tarjoaa palveluita tietylle Controller-oliolle. Palvelukerros välittää pyynnöt tietokantayhteyksistä vastaavalle kerrokselle, ja se muokkaa vastauksena saadut tiedot oikeaan muotoon ennen niiden palauttamista ohjainkerrokselle. Palvelukerroksen luokat tunnistaa `Service`-pääteestä, ja ne on sijoitettu pakettiin `com.kodavi.service`.

Tietolähdekerros tarjoaa palveluita palvelukerrokselle suorittamalla haut sovelluksen tietokantaan. Kukin tietolähdekerroksen Java-luokista muodostettu `DataService`-olio tarjoaa palveluita tietylle Service-oliolle. Tietolähdekerroksen luokat tunnistaa `DataService`-pääteestä, ja ne on sijoitettu pakettiin `com.kodavi.dao`. Sovelluksen tietokannan rakenne on kuvattu luvussa 5.

Mallikerroksessa Java-luokat toteuttavat mallinnettavan maailman käsitteet. Luokat on johdettu suoraan tietokannan tauluista. Mallikerros sisältää myös luokkia, joille ei ole vastinetta tietokannassa. Näitä luokkia tarvitaan tietokannasta haettujen tietojen käsittelyssä palvelu- ja tietolähdekerroksella. Tietokannasta johdetut luokat ovat paketissa `com.kodavi.model.db`. Muut luokat ovat paketissa `com.kodavi.model`.

4.3 Sovelluspalvelin

Sovelluksen taustaohjelmisto on sijoitettu tuotantopalvelimelle Apache Tomcat -sovelluspalvelimen `webapps`-kansioon, jossa tiedosto `ROOT.war` sisältää taustaohjelmiston pakatun tiedostorakenteen. Tomcat purkaa `war`-tiedoston automaattisesti kansioon `ROOT` ja käynnistää taustaohjelmiston. Taustaohjelmiston päivitys vaatii aina uuden `ROOT.war`-tiedoston kopioinnin `webapps`-kansioon.

Sovelluksen käyttöliittymä on sijoitettu tuotantopalvelimelle Nginx-välityspalvelimen `nginx.conf`-tiedoston `root`-määritelmän osoittamaan kansioon. Sovelluksen asennusohjeet on kuvattu kokonaisuudessaan palvelimen asennusohjeet -dokumentissa [3].

5 Sovelluksen tietokanta

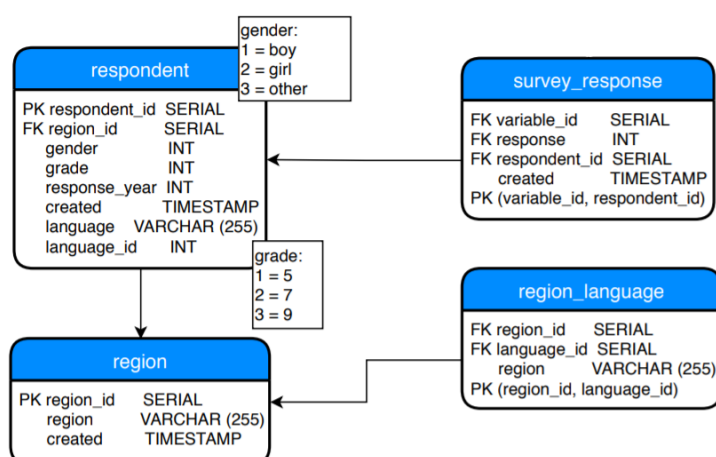
Sovelluksen tietokannan rakennekaavio on selvyyden vuoksi jaettu moneen osaan taulukokonaisuuksittain, jotta datajoukkojen rakenne hahmottuu selkeämmin. Taulujen ja sarakkeiden tekstisisällöt ovat näin ollen myös helpommin luettavissa. Osa tauluista esiintyy useammassa kuvassa.

Kodavi-projektissa kehitetyssä sovelluksessa ei hyödynnetty kaikkia kuvissa näkyviä tauluja, koska osa suunnitelluista ominaisuuksista sovittiin toteutettavaksi sovelluksen jatkokehityksessä. Kodavi-projektin kehittämä sovellus ei hyödynnä käyttäjätietoihin, tapahtumiin, tehtäviin, vastauksiin ja esitystapoihin liittyviä tauluja. Nämä taulut on esitetty lukujen 5.2–5.4 kuvissa 5.4–5.8.

Tietojärjestelmän Java-luokkien nimet on johdettu tietokannan taulujen nimistä siten, että sanojen ensimmäiset kirjaimet on kirjoitettu suuraakkosin ja sanoja toisistaan erottavat alaviivat on poistettu (esim. taulua *survey_response* vastaava Java-luokka on *SurveyResponse*).

5.1 Tutkimusdatan tallentamiseen liittyvät taulut

Luvussa kuvataan tutkimusdatan tallentamiseen liittyvät taulut ja niiden sarakkeet. Taulujen väliset suhteet esitetään kuvissa 5.1–5.3.



Kuva 5.1. Kyselyn vastaajien tietoihin liittyvät taulut.

Tauluun **region** tallennetaan suomenkieliset tiedot maantieteellisistä alueista, joista tutkimusdataa kerätään. Taulu sisältää seuraavat sarakkeet:

region_id	Alueen yksilöivä tunniste.
region	Alueen nimi.
created	Aikaleima, jolloin rivi on lisätty tauluun.

Tauluun **region_language** tallennetaan muut kuin suomenkieliset tiedot maantieteellisistä alueista, joista tutkimusdataa kerätään. Taulu sisältää seuraavat sarakkeet:

region_id	Alueen yksilöivä tunniste.
language_id	Kielen yksilöivä tunniste.
region	Alueen nimi.

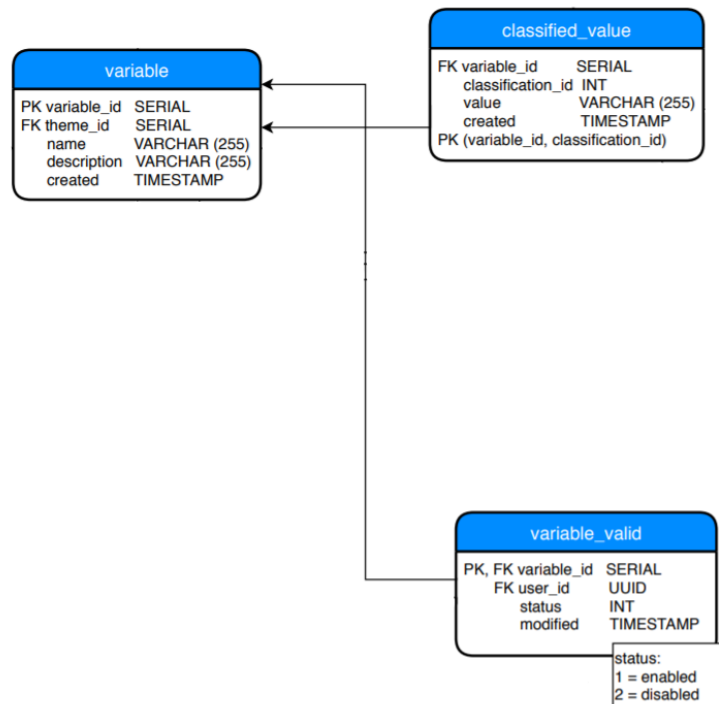
Tauluun **respondent** tallennetaan kyselyn vastaajien tiedot, ja se sisältää seuraavat sarakkeet:

respondent_id	Kyselyn vastaajan yksilöivä tunniste.
region_id	Kyselyn vastaajan asuinalueen yksilöivä tunniste.
gender	Kyselyn vastaajan sukupuoli (kokonaisluku, jonka arvo tulkitaan sovelluksen lähdekoodissa).
grade	Kyselyn vastaajan luokkataso (kokonaisluku, jonka arvo tulkitaan sovelluksen lähdekoodissa).
response_year	Vastausvuosi.
created	Aikaleima, jolloin rivi on lisätty tauluun.
language	Kyselyn vastaajan kieli.
language_id	Kyselyn vastaajan kielen tunniste.

Tauluun **survey_response** tallennetaan tiedot kyselyn vastaajien vastauksista muuttujiin. Taulu sisältää seuraavat sarakkeet:

variable_id	Muuttujan yksilöivä tunniste.
response	Muuttujan vastausvaihtoehdon järjestysnumero.

respondent_id Vastaajan yksilöivä tunniste.
created Aikaleima, jolloin rivi on lisätty tauluun.



Kuva 5.2. Muuttujiin ja niiden vastausvaihtoehtoihin liittyvät suomenkieliset tiedot.

Tauluun **classified_value** tallennetaan suomenkieliset tiedot muuttujien vastausvaihtoehtoista. Taulu sisältää seuraavat sarakkeet:

variable_id Muuttujan yksilöivä tunniste.
classification_id Muuttujan vastausvaihtoehdon yksilöivä tunniste.
value Vastausvaihtoehdon kuvaus.
created Aikaleima, jolloin rivi on lisätty tauluun.

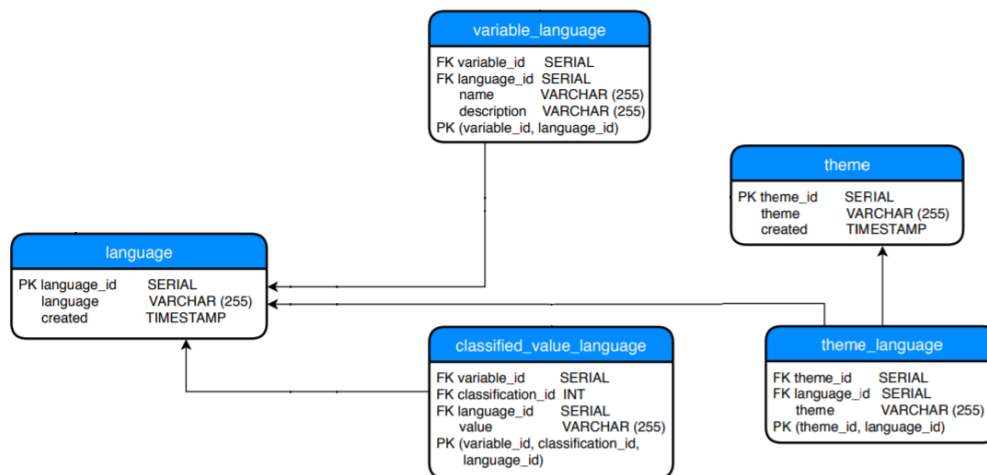
Tauluun **variable** tallennetaan suomenkieliset tiedot muuttujista, ja se sisältää seuraavat sarakkeet:

variable_id Muuttujan yksilöivä tunniste.
theme_id Teeman yksilöivä tunniste.

name	Muuttujan nimi.
description	Muuttujan kuvaus (kysymys, joka näytetään käyttöliittymässä).
created	Aikaleima, jolloin rivi on lisätty tauluun.

Tauluun **variable_valid** tallennetaan tiedot muuttujien voimassaolosta (ts. näkyvyydestä käyttöliittymässä). Taulu sisältää seuraavat sarakkeet:

variable_id	Muuttujan yksilöivä tunniste.
user_id	Muuttujan voimassaoloa viimeksi muokanneen käyttäjän yksilöivä tunniste.
status	Muuttujan voimassaolon tila (kokonaisluku, jonka arvo tulkitaan sovelluksen lähdekoodissa).
modified	Aikaleima, jolloin riviä on muokattu.



Kuva 5.3. Teemojen, muuttujien ja vastausvaihtoehtojen muunkieliset tiedot sekä kielitiedot.

Tauluun **classified_value_language** tallennetaan muut kuin suomenkieliset tiedot muuttujien vastausvaihtoehdoista. Taulu sisältää seuraavat sarakkeet:

variable_id	Muuttujan yksilöivä tunniste.
--------------------	-------------------------------

classification_id	Muuttujan vastausvaihtoehdon yksilöivä tunniste.
language_id	Kielen yksilöivä tunniste.
value	Vastausvaihtoehdon kuvaus.

Tauluun **language** tallennetaan tiedot sovelluksessa käytettävistä kielistä, ja se sisältää seuraavat sarakkeet:

language_id	Kielen yksilöivä tunniste.
language	Kieli.
created	Aikaleima, jolloin rivi on lisätty tauluun.

Tauluun **theme** tallennetaan suomenkieliset tiedot teemoista, ja se sisältää seuraavat sarakkeet:

theme_id	Teeman yksilöivä tunniste.
theme	Teeman kuvaus.
created	Aikaleima, jolloin rivi on lisätty tauluun.

Tauluun **theme_language** tallennetaan muut kuin suomenkieliset tiedot teemoista, ja se sisältää seuraavat sarakkeet:

theme_id	Teeman yksilöivä tunniste.
language_id	Kielen yksilöivä tunniste.
theme	Teeman kuvaus.

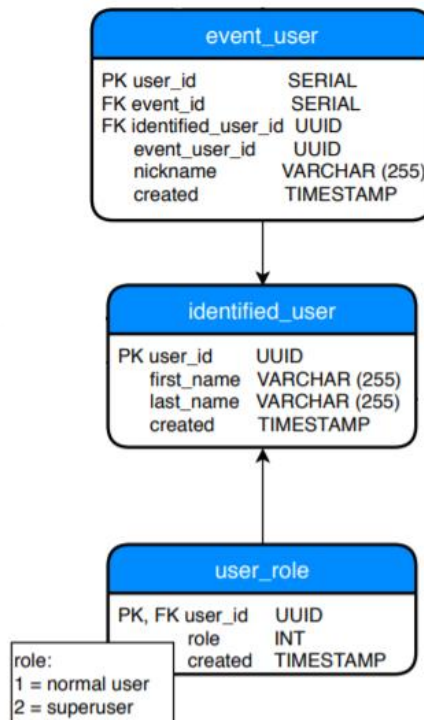
Tauluun **variable_language** tallennetaan muut kuin suomenkieliset tiedot muuttujista.

Taulu sisältää seuraavat sarakkeet:

variable_id	Muuttujan yksilöivä tunniste.
language_id	Kielen yksilöivä tunniste.
name	Muuttujan nimi.
description	Muuttujan kuvaus (kysymys, joka näytetään käyttöliittymässä).

5.2 Käyttäjät ja roolit

Luvussa kuvataan käyttäjätietojen ja -roolien tallentamiseen liittyvät taulut ja niiden sarakkeet. Taulujen väliset suhteet esitetään kuvassa 5.4.



Kuva 5.4. Käyttäjien tietoihin liittyvät taulut.

Tauluun **event_user** tallennetaan tiedot tapahtumiin kirjautuneista käyttäjistä (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

user_id	Taulun rivin yksilöivä tunniste.
event_id	Tapahtuman yksilöivä tunniste.
identified_user_id	Peda.net-tunnuksilla ja ryhmäavaimella kirjautuneen käyttäjän yksilöivä tunniste.
event_user_id	Pelkällä ryhmäavaimella kirjautuneen käyttäjän yksilöivä tunniste.
nickname	Käyttäjän nimimerkki.
created	Aikaleima, jolloin rivi on lisätty tauluun.

Tauluun **identified_user** tallennetaan tiedot Peda.net-tunnuksilla kirjautuneista käyttäjistä (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

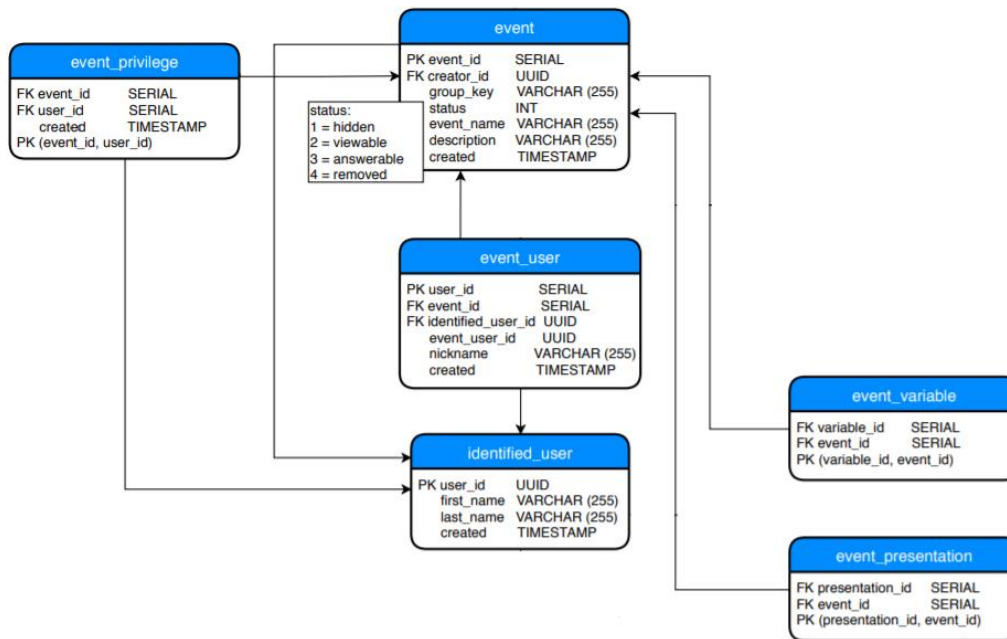
user_id	Käyttäjän yksilöivä tunniste.
first_name	Käyttäjän etunimi.
last_name	Käyttäjän sukunimi.
created	Aikaleima, jolloin rivi on lisätty tauluun.

Tauluun **user_role** tallennetaan tiedot Peda.net-tunnuksilla kirjautuneiden käyttäjien käyttäjärooleista (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

user_id	Käyttäjän yksilöivä tunniste.
role	Käyttäjärooli (kokonaisluku, jonka arvo tulkitaan sovelluksen lähdekoodissa).
created	Aikaleima, jolloin rivi on lisätty tauluun.

5.3 Tapahtumiin liittyvät taulut

Luvussa kuvataan tapahtumien tietojen tallentamiseen liittyvät taulut ja niiden sarakkeet. Taulujen väliset suhteet esitetään kuvassa 5.5.



Kuva 5.5. Tapahtumiin liittyvät tiedot.

Tauluun **event** tallennetaan tiedot tapahtumista (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

event_id	Tapahtuman yksilöivä tunniste.
creator_id	Tapahtuman luoneen käyttäjän yksilöivä tunniste.
group_key	Ryhmäavain, jolla käyttäjät kirjautuvat tapahtumaan.
status	Tapahtuman voimassaolon tila (kokonaisluku, jonka arvo tulkitaan sovelluksen lähdekoodissa).
event_name	Tapahtuman nimi.

description	Tapahtuman kuvaus.
created	Aikaleima, jolloin rivi on lisätty tauluun.

Tauluun **event_presentation** tallennetaan tiedot tapahtumissa käytössä olevista hakutulosten esitystavoista (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

presentation_id	Esitystavan yksilöivä tunniste.
event_id	Tapahtuman yksilöivä tunniste.

Tauluun **event_privilege** tallennetaan tiedot käyttäjien oikeuksista muokata tapahtumien tietoja (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

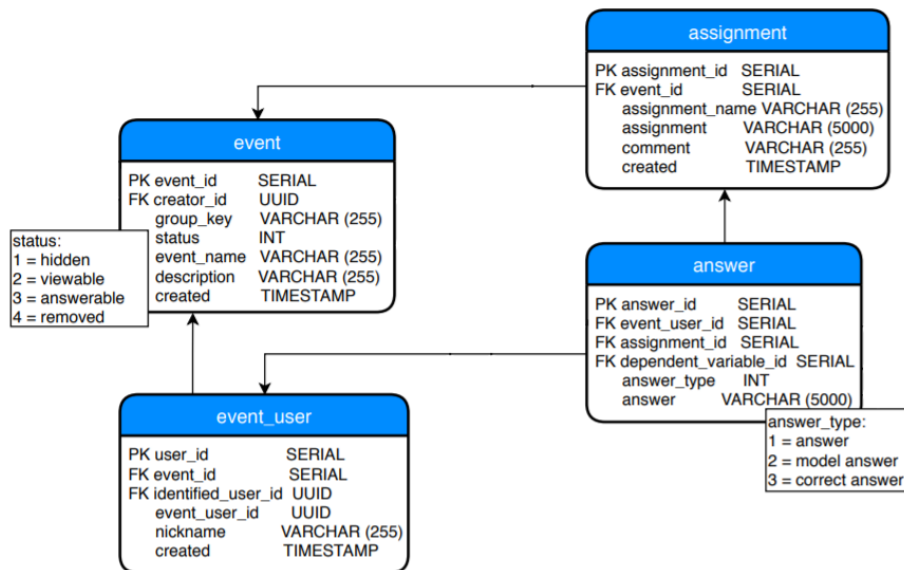
event_id	Tapahtuman yksilöivä tunniste.
user_id	Tapahtuman muokkaamiseen oikeutetun käyttäjän yksilöivä tunniste.
created	Aikaleima, jolloin rivi on lisätty tauluun.

Tauluun **event_variable** tallennetaan tiedot tapahtumissa käytössä olevista muuttujista (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

variable_id	Muuttujan yksilöivä tunniste.
event_id	Tapahtuman yksilöivä tunniste.

5.4 Tehtäviin, vastauksiin ja esitystapoihin liittyvät taulut

Luvussa kuvataan tehtävien, vastausten ja esitystapojen tietojen tallentamiseen liittyvät taulut ja niiden sarakkeet. Taulujen väliset suhteet esitetään kuvissa 5.6–5.8.



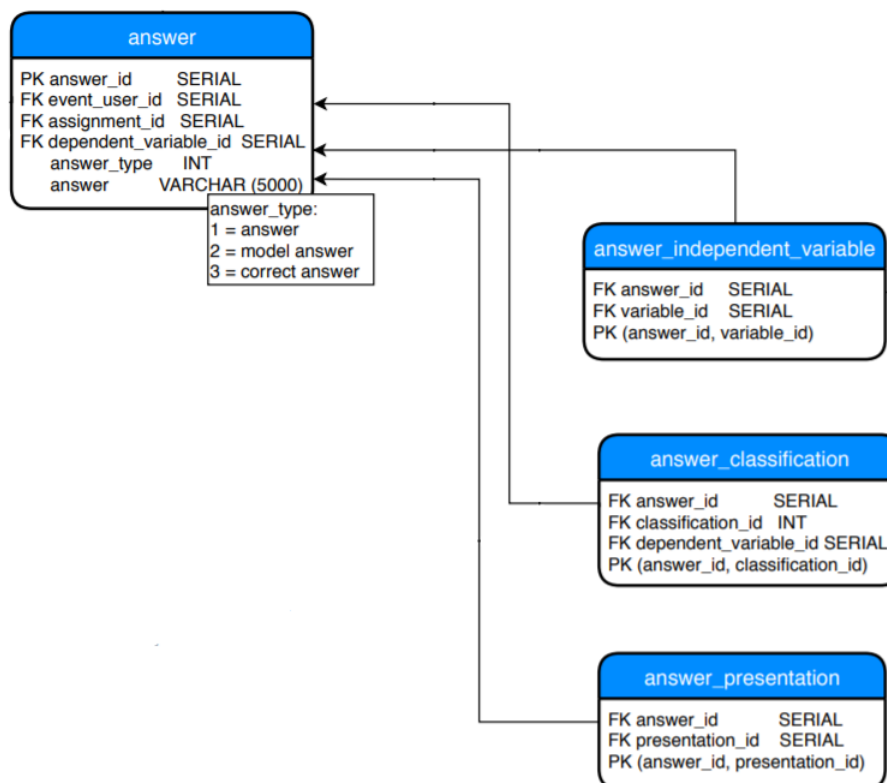
Kuva 5.6. Tehtäviin ja vastauksiin liittyvät tiedot.

Tauluun **answer** tallennetaan tiedot käyttäjien vastauksista tapahtumiin liittyviin tehtäviin (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

answer_id	Vastauksen yksilöivä tunniste.
event_user_id	Tapahtumaan kirjautuneen vastaajan yksilöivä tunniste.
assignment_id	Tehtävän yksilöivä tunniste.
dependent_variable_id	Vastemuuttujan yksilöivä tunniste.
answer_type	Vastauksen tyyppi (kokonaisluku, jonka arvo tulkitaan sovelluksen lähdekoodissa).
answer	Vastaus tehtävään.

Tauluun **assignment** tallennetaan tiedot tapahtumiin liittyvistä tehtävistä (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

assignment_id	Tehtävän yksilöivä tunnus.
event_id	Tapahtuman yksilöivä tunnus.
assignment_name	Tehtävän nimi.
assignment	Tehtävän kuvaus.
comment	Kommenttikenttä, johon tehtävän laatija voi kirjata esim. muistiinpanoja.
created	Aikaleima, jolloin rivi on lisätty tauluun.



Kuva 5.7. Vastauksiin liittyvät tiedot.

Tauluun **answer_presentation** tallennetaan tiedot esitystavoista, jotka käyttäjä on valinnut tehtävän vastaukseensa (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

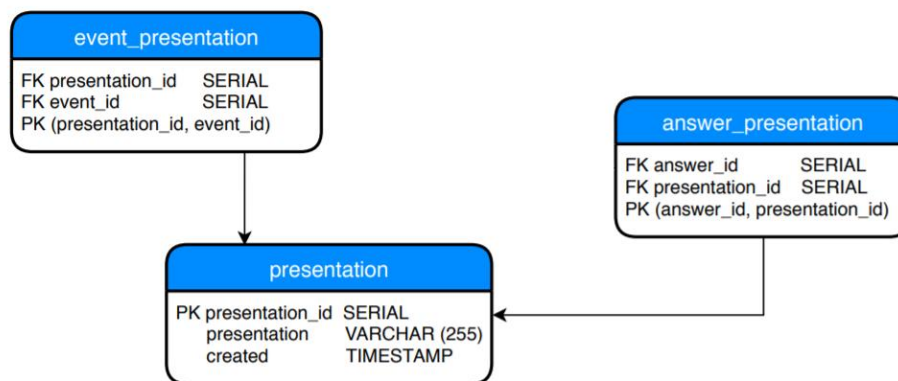
answer_id	Vastauksen yksilöivä tunniste.
presentation_id	Esitystavan yksilöivä tunniste.

Tauluun **answer_classification** tallennetaan tiedot siitä, mitkä vastemuuttujan vastausvaihtoehdot käyttäjä on valinnut näytettäväksi hakutuloksen kuvaajassa (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

answer_id	Vastauksen yksilöivä tunniste.
classification_id	Vastausvaihtoehdon yksilöivä tunniste.
dependent_variable_id	Vastemuuttujan yksilöivä tunniste.

Tauluun **answer_independent_variable** tallennetaan tiedot vastauksiin sisältyvistä luokittelevista muuttujista (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

answer_id	Vastauksen yksilöivä tunniste.
variable_id	Luokittelevan muuttujan yksilöivä tunniste.



Kuva 5.8. Esitystapoihin liittyvät tiedot.

Tauluun **presentation** tallennetaan tiedot hakutulosten esitystavoista (taulu ei ole käytössä sovelluksen versiossa 1.0). Taulu sisältää seuraavat sarakkeet:

presentation_id	Esitystavan yksilöivä tunniste.
presentation	Esitystapa.
created	Aikaleima, jolloin rivi on lisätty tauluun.

6 Sovellukselle asetettujen vaatimusten toteutuminen

Luvussa kerrotaan projektissa toteutuneista vaatimuksista ja vaatimuskokonaisuuksista. Kodavi-sovellusprojektissa toteutettiin yhteensä 23 vaatimusta, joista 17 olivat prioriteetiltaan pakollisia, viisi tärkeitä ja yksi mahdollinen. Sovellusprojektin aikana kirjattiin kaikkiaan 116 vaatimusta, jatkokehitykseen sovittiin 86 vaatimusta ja seitsemän vaatimusta hylättiin. Kaikki projektissa määritellyt vaatimukset on listattu *Vaatimusmäärittelydokumentissa* [4], joka sisältää myös vaatimuksille määritetyt prioriteetit ja projektin päättymishetkeä vastaavat tilat.

6.1 Sovellusprojektissa toteutetut vaatimukset

Sovellusprojektin alussa sovellukseen suunniteltiin useita toiminnallisuuksia ja erilaisia käyttäjärooleja. Projektille varatut resurssit eivät kuitenkaan riittäneet kaikkien vaatimusten toteuttamiseen. Projektin aikana toteutettiin tilaajan edustajien mielestä tärkeimmät vaatimuskokonaisuudet eli hakusivun perustoiminnot ja käyttöliittymän erikieliset versiot.

Projektissa toteutettiin eniten hakusivuun liittyviä vaatimuksia. Muuttujan ja luokittelevien tekijöiden valitsemisen, hakuehtojes esittämisen sekä hakutulosten esitystapojen vaatimuskokonaisuuksista toteutettiin suurin osa prioriteetiltaan pakollisiksi tai tärkeiksi määritellyistä vaatimuksista. Sovellukseen toteutettiin hakuun liittyvät perustoiminnot, jotka mahdollistavat muuttujan ja luokittelevien tekijöiden valitsemisen sekä hakutulosten esittämisen.

Hakusivuun liittyvien vaatimusten ohella toteutettiin suurin osa tietokannan ja käyttöliittymän vaatimuksista. Tavoite käyttöliittymän erikielisistä versioista toteutui. Käyttöliittymää voidaan käyttää suomeksi, ruotsiksi tai englanniksi. Sovelluksen kehittämisessä on huomioitu käytön mahdollisuus myös mobiililaitteilla kuten tablettitietokoneilla ja älypuhelimilla.

Tärkeä tavoite projektissa oli suunnitella tietojärjestelmään kuuluva tietokanta mahdollisimman monikäyttöiseksi mahdollista jatkokehitystä silmällä pitäen. Tietokantaan tallennettiin projektissa vain WHO-Koululaistutkimuksen esikäsitelty kansallinen data.

Samalla otettiin kuitenkin tietokannan rakenteessa huomioon mahdollisuus tallentaa myöhemmin myös muiden maiden dataa.

6.2 Jatkokehitykseen sovitut vaatimukset

Jatkokehitykseen sovittiin kirjautumisen, kaikkien käyttäjäroolien sekä tapahtumiin ja tehtäviin liittyvät vaatimuskokonaisuudet kokonaisuudessaan. Kyseisten vaatimuskokonaisuuksien toteuttaminen lisäisi sovelluksen tarjoamia toimintoja huomattavasti.

Hakusivuun liittyvistä vaatimuksista sovittiin kokonaisuudessaan jatkokehitykseen esitystapojen valintoja, muuttujaan liitettäviä lisätietoja ja hakutulosten tallentamista koskevat vaatimuskokonaisuudet. Myös osa muuttujan ja luokittelevien tekijöiden valitsemisen, hakutulosten esittämisen sekä hakutulosten esitystapojen vaatimuskokonaisuuksien vaatimuksista sovittiin jatkokehitykseen. Hakusivuun liittyvien, jatkokehitykseen sovittujen vaatimusten ja vaatimuskokonaisuuksien toteuttaminen lisäisi käyttäjän mahdollisuuksia valita hakuehdot nykyistä monipuolisemmin ja tarkemmin.

Tietokantaan liittyvänä tavoitteena oli myös, että käyttäjä ei pystyisi näkemään sovelluksen avulla tietokantaan tallennettua alkuperäistä tutkimusdataa, vaan pelkästään tilaajan valitsemat ja luokittelemat muuttujat. Toisen koodikatselmoinnin yhteydessä havaittiin, että vaatimus ei ollut toteutunut tarkoitetulla tavalla. Alkuperäisten muuttujien tietoja on mahdollista hakea kehittäjän työkaluilla. Tilaajan edustajat eivät pitäneet kyseisen vaatimuksen toteutumatta jäämistä kriittisenä, ja kyseinen tavoite sovittiin siirrettäväksi jatkokehitykseen.

7 Ideoita sovelluksen jatkokehitykseen

Luvussa kuvataan Kodavi-sovellusprojektin aikana sovelluksen jatkokehitystä varten kerättyjä ideoita. Ideat liittyvät käyttöliittymän ominaisuuksien lisäämiseen sekä tietokannan rakenteen ja lähdekoodin muutoksiin.

7.1 Hakusivun jatkokehitysideat

Kodavi-sovellusprojektin aikana jatkokehitykseen sovitut vaatimukset ovat luettavissa *Vaatimusmäärittelydokumentista* [4]. Jatkokehitysideoita kerättiin myös käytettävyystestauksen yhteydessä koehenkilöiltä. Heidän kirjaamansa ideat ovat luettavissa *Käytettävyystestausraportista* [2].

Kodavi-sovellusprojektissa toteutettiin hakuehtojen valitsemisen ja hakutulosten esittämisen perustoiminnallisuudet. Kyseisiä toiminnallisuuksia voitaisiin jatkokehittää nykyistä monipuolisemmiksi esimerkiksi lisäämällä seuraavat ominaisuudet:

- Käyttäjä voisi valita, minkä vuoden tulokset hän haluaa nähdä. Nyt käyttäjälle näytetään automaattisesti tulokset viimeisimmältä tutkimusvuodelta, jolloin valittua muuttujaa on tutkittu.
- Hakuehtojen valitsemisen yhteydessä voitaisiin esittää muuttujien lyhyet nimet ja hakutulosten yhteydessä muuttujien pidemmät kuvaukset. Nyt esitetään molemmissa yhteyksissä pitkät kuvaukset.
- Valittujen hakuehtojen yhteydessä voitaisiin esittää valitun muuttujan vastausvaihtoehdot ja värikoodit, jotka esiintyvät myös kuvaajassa. Myös taulukossa tulokset voitaisiin merkitä värikoodein.
- Käyttäjä voisi valita vain osan hakutuloksista esitettäväksi kuvaajassa. Hän voisi valita joko tietyn vastausvaihtoehdon tulokset tai tietyt luokitellut vastausjaryhmät, joiden tulokset esitetään kuvaajassa.
- Käyttäjä voisi valita haluamansa värit kuvaajan palkkeihin tai muuttaa taulukon rivien ja sarakkeiden järjestystä.
- Taulukon tuloksia selkeyttämään voitaisiin lisätä sarake, jossa näkyisi luokitellun ryhmän tulosten prosenttiosuudet yhteen laskettuna. Laskutapaa olisi myös hyvä muuttaa niin, että summa olisi aina tasan 100 %, eivätkä yksittäisten lukujen pyöristykset vaikuttaisi siihen.

- Hakutulosten esitystapoja voisi olla useampia ja aikasarjan tulokset voitaisiin esittää trendiviivana.
- Käyttäjä saisi tallennettua hakutulokset taulukkona omalle laitteelleen.
- Samaan näkymään voisi saada useamman kuin yhden haun tulokset näkyviin samanaikaisesti, jotta niitä olisi helppo vertailla. Toiminnallisuuden tarpeellisuutta kannattaa harkita ennen sen toteuttamista, koska sovelluksen yhtenä haasteena on käytön hankaluus pienikokoisilla näytöillä. Näkymien jakamisen mahdollistaminen todennäköisesti pahentaisi ongelmaa entisestään. Käyttäjän on myös mahdollista kiertää toiminnallisuuden puuttuminen avaamalla useampi selainikkuna ja jakamalla näyttötila niiden kesken.

7.2 Käyttäjäroolien jatkokehitysideat

Kodavi-sovellusprojektissa sovittiin jatkokehitykseen kaikki käyttäjäroolien ja kirjautumisen ominaisuudet. Niille kirjattiin kuitenkin projektin aikana useita vaatimuksia, jotka on listattu *Vaatimusmäärittelydokumentissa* [4]. Käyttäjärooleja suunniteltiin olevan neljä: pääkäyttäjä, Peda.net-tunnuksella kirjautunut käyttäjä, ryhmävaimella kirjautunut käyttäjä ja julkinen käyttäjä.

Pääkäyttäjällä suunniteltiin olevan laajimmat oikeudet. Pääkäyttäjä voisi esimerkiksi valita muiden käyttäjien käytössä olevat muuttujat, luokittelevat tekijät ja esitystavat sekä luokittelevien tekijöiden maksimimäärät. Seuraavaksi eniten oikeuksia olisi Peda.net-tunnuksella kirjautuneella käyttäjällä. Peda.net-tunnuksella kirjautunut käyttäjä voisi esimerkiksi luoda tapahtumia ja tehtäviä, joista kerrotaan luvussa 7.3. Ryhmävaimella kirjautuneet käyttäjät voisivat kirjautua näihin tapahtumiin ja vastata tehtäviin. Vähiten oikeuksia olisi julkisella käyttäjällä. Julkinen käyttäjä ei voisi tehdä sovelluksessa muuta kuin suorittaa hakuja ja tarkastella niiden tuloksia. Hakujen tekeminen ja hakutulosten tarkastelu kuuluvat kaikkien käyttäjäroolien oikeuksiin, vaikka käyttäjäroolien valittavina olevat hakuehdot ja hakutulosten esitystavat voivat erota toisistaan.

7.3 Tapahtumien ja tehtävien jatkokehitysideat

Kodavi-sovellusprojektissa sovittiin jatkokehitykseen kaikki tapahtumiin ja tehtäviin liittyvät ominaisuudet. Niille kirjattiin projektin aikana useita vaatimuksia, jotka on lisätty *Vaatimusmäärittelydokumentissa* [4].

Pääpiirteittäin tapahtumia ja tehtäviä voitaisiin hyödyntää niin, että pääkäyttäjä tai Peda.net-tunnuksella kirjautunut käyttäjä voisi luoda tapahtuman ja liittää siihen haluamansa muuttajat, luokittelevat tekijät ja hakutulosten esitystavat sekä määrittellä luokittelevien tekijöiden maksimimäärän. Tapahtumassa voitaisiin tehdä hakuja vain siihen liitetyillä hakuehdoilla. Tapahtumaan voitaisiin liittää myös tehtäviä, joihin etsitään vastauksia esimerkiksi suorittamalla hakuja sekä ryhmäavain, jonka kautta muut henkilöt pääsisivät tarkastelemaan tapahtumaa ja vastaamaan siihen liitettyihin tehtäviin. Tapahtuman lisännyt käyttäjä voisi antaa myös toiselle käyttäjälle oikeuksia muokata tapahtumaa. Käyttäjät, joilla on oikeus muokata tapahtumaa, pääsisivät katsелеmaan tapahtuman tehtäviin tallennettuja vastauksia.

7.4 Tietokannan jatkokehitysideat

Kodavi-projektin kehittämä sovellus ei hyödynnä kaikkia tietokannan tauluja, koska osa suunnitelluista ominaisuuksista sovittiin toteutettaviksi sovelluksen jatkokehityksessä. Luvussa 5 mainitaan taulut ja taulukokonaisuudet, joita sovellus ei vielä hyödynnä lainkaan.

Etenkin tauluihin, joita sovellus ei vielä toistaiseksi hyödynnä, voi olla tarpeen tehdä muutoksia jatkokehityksessä. Muutoksia on mahdollisesti järkevää tehdä myös jo sovelluksen käyttämiin tauluihin. Jatkokehityksessä kannattaa pohtia ainakin seuraavien muutosten tarpeellisuutta:

- Tietokannasta ei tulisi mielellään poistaa mitään fyysisesti, vaan poistettava taulun rivi tulisi vain merkitä poistetuksi. Kannattaisiko siis tauluun **event_privilege** lisätä sarake `removed`? Onko aktiivisen ja poistetun lisäksi tarvetta myös muille tiloille, jolloin sarakkeen nimi kuvaisi paremmin tilaa tai voimassaoloa?
- Kaikkien taulujen osalta kannattaa pohtia tarvetta sarakkeelle, jossa on tieto taulun rivin tilasta tai voimassaolosta.

- Kannattaisiko tauluun **event_privilege** lisätä sarake, jossa on tieto käyttäjän oikeudesta tapahtumaan (esim. onko muita oikeuksia kuin muokkaaminen)? Kannattaisiko samaan tauluun lisätä sarake, jossa on tieto oikeuden lisääjästä?
- Taulun **event_user** pääavainsarakkeen nimi voisi olla taulua paremmin kuvaava `event_user_id` ja sarakkeen `event_user_id` nimeksi voisi muuttaa dataa kuvaavammin `group_key_user_id`.
- Kannattaisiko tauluun **user_role** lisätä sarake käyttäjän kirjautumistavalle? Voiko esimerkiksi pääkäyttäjä kirjautua sovellukseen muutoin kuin Peda.net-rajapinnan kautta? Alkuperäisen suunnitelman mukaan tauluun **identified_user** tallennetaan tiedot Peda.net-rajapinnan kautta kirjautuneista käyttäjistä.
- Kannattaisiko tauluihin **event_presentation** ja **event_variable** lisätä tieto rivin muokkaajasta, tilasta ja muutoshetkestä?
- Kannattaisiko taulussa **assignment** olla tieto tehtävän lisääjästä, jos useampi käyttäjä voi lisätä tapahtumaan tehtäviä? Tarvitaanko kenttää tehtävän tilalle/voimassaololle?
- Tarvitaanko taulussa **answer** saraketta, jossa on tieto rivin lisäyshetkestä?
- Tulisiko taulussa **presentation** huomioida esitystapojen muunkieliset tiedot suomen ohella?
- Taulussa **respondent** on sarakkeet `language` ja `language_id`. Sovellus ei tällä hetkellä hyödynnä sarakkeessa `language_id` olevia tietoja. Jatkokehityksessä kannattaa pohtia ko. sarakkeen poistamista taulusta, vaikka sen poistaminen ei olekaan välttämätöntä.

Tietokanta sisältää muutamia taulupareja, joissa suomen- ja muunkieliset tiedot on erotettu omiin tauluihinsa. Esimerkiksi tauluun **region** on tallennettu suomenkieliset tiedot kyselyn vastaajien asuinalueista, ja tauluun **region_language** on tallennettu muunkieliset tiedot asuinalueista. Kyseiseen ratkaisuun päädyttiin siksi, koska tietokannanhallintajärjestelmä ei salli niin sanottuja osittaisia viittauksia taulujen välillä. Jos esimerkiksi tauluun **region** tallennettaisiin asuinalueiden tiedot kaikilla kielillä, muodostuisi taulun perusavain sarakkeista `region_id` ja `language_id`. Näin ollen vastaajien taulun **respondent** sarake `region_id` viittaisi taulun **region** perusavaimen osaan eli sarakkeeseen `region_id`. Tietokannanhallintajärjestelmä ei salli tällaisia viittauksia.

Ongelma olisi ollut helppo kiertää lisäämällä esimerkiksi tauluun **region** erillinen pääavainsarake (ts. uusi keinoavain), joka yksilöi taulun rivin. Näin ollen taulua **region_language** ei olisi tarvittu. Kodavi-projektin loppuvaiheessa kyseistä muutosta ei kuitenkaan lähdetty enää tekemään, koska viime hetken muutokset olisivat voineet aiheuttaa virheitä sovelluksen toiminnassa. Jatkokehityksessä kannattaa kuitenkin pohtia, olisiko kyseisen korjauksen tekeminen mielekästä. Uuden pääavainsarakkeen lisäämisen lisäksi taulurakenteet saattavat kaivata myös muita muokkauksia, mitä kannattaa pohtia tarkemmin jatkokehityksessä. Edellä kuvattu ongelma koskee seuraavia taulupareja:

- **classified_value** ja **classified_value_language**,
- **region** ja **region_language**,
- **theme** ja **theme_language** sekä
- **variable** ja **variable_language**.

7.5 Sovelluksen lähdekoodin jatkokehitysideat

Sovelluksen lähdekoodin loppukatselmointiraportissa [11] on kirjattuna lähdekoodiin liittyviä huomioita. Näistä osa on jo korjattu, mutta joitakin raportissa esiintyviä huomioita ei ole lähdekoodin muokkauksessa huomioitu, koska viime hetken muutokset olisivat voineet aiheuttaa virheitä sovelluksen toiminnassa. Jatkokehityksessä olisi suositeltavaa **backendin lähdekoodin** osalta tarkastella seuraavien muutosten tarpeellisuutta:

- Monet `SurveyResponseService`-luokan metodit ovat huomattavan pitkiä. Kannattaisiko niitä pilkkoa pienempiin osiin?
- `@CrossOrigin`-annotaatio ei välttämättä ole sovelias tuotantoympäristöön. Olisiko annotaatio syytä poistaa?

Frontendin lähdekoodin osalta jatkokehityksessä kannattaa tarkastella ainakin seuraavien muutosten tarpeellisuutta:

- `actions/index.js`-tiedostossa on useita samankaltaisia funktioita. Kannattaisiko niitä yhdistää?
- `searchpage/index.jsx`-tiedoston `return()`-osassa on sisäkkäisiä ternäri- ja `map/reduce/filter`-rakenteita. Kannattaisiko niitä jakaa funktioihin?

Katselmointiraportissa [11] on huomautettu myös muuttujista, joiden tietoja pääsee tarkastelemaan selaimen kehitystyökalujen avulla, vaikka näitä muuttujia ei ole tarkoitettu nähtäviksi. Jatkokehityksessä muuttujien tietokantahakua voitaisiin muuttaa esimerkiksi siten, että tarkastettaisiin `VariableValid`-taulusta, onko kyseinen muuttuja käytössä.

Katselmointiraportissa on huomautettu myös siitä, ettei vääriä osoitteita käsitellä hallitusti, vaan sovellus tuottaa `500 Internal Error`-virheen. Sovelluksen jatkokehityksessä voisi olla kannattavaa käsitellä nämä virhetilanteet jollain tavalla.

8 Yhteenveto

Kodavi-projektin Study Health with HBSC on Jyväskylän yliopiston liikuntatieteellisen tiedekunnan ja informaatioteknologian tiedekunnan yhteistyönä kehittämä verkkosovellus, joka avaa julkiseen käyttöön WHO-Koululaistutkimuksen datan vuosilta 2002–2018. Sovellus kehitettiin ensisijaisesti terveystiedon opiskelu- ja opetuskäyttöön. Sovelluksen julkisen luonteen vuoksi kuka tahansa voi tarkastella WHO-Koululaistutkimuksen keräämää dataa ja tutkia erilaisia ilmiöitä. Pääasiallinen kehitys kohdistui työpöytäkäyttöön. Sovellus kuitenkin toimii tabletilla ja puhelimella siedettävällä tasolla. Sovelluksen käyttöliittymä toteutettiin suomen lisäksi englanniksi ja ruotsiksi.

Sovellusprojektin aikana toteutettiin tilaajan edustajien mielestä tärkeimmät vaatimuskokonaisuudet eli hakusivun perustoiminnot ja käyttöliittymän erikieliset versiot. Sovellusprojektin alkupuolella sovellukseen suunnitelluista toiminnallisuuksista suuri osa päätettiin siirtää jatkokehitykseen. Projektille varatut resurssit eivät riittäneet kaikkien vaatimusten toteuttamiseen tämän projektin aikana. Tilaajan edustajat ovat todenneet sovelluksen kuitenkin toimivaksi ja hyödylliseksi.

Jatkokehitykseen siirrettyjen ominaisuuksien määrän takia sovelluksen jatkokehitys vaikuttaa Kodavi-projektin päättyessä todennäköiseltä. Kodavi-projektin toteutusratkaisut perustuivat vuoden 2020 suositeltuihin ja suosittuihin teknologioihin. Käytetyt ja yleiset teknologiat edesauttavat sovelluksen päivitettävyyttä ja pitkäikäisyyttä. Lisäksi valittujen teknologioiden parissa työskentelyn pitäisi olla mielekästä myös sovelluksen jatkokehitystä ajatellen. Myös sovelluksen tietokanta on rakennettu sovelluksen jatkokehitystä silmällä pitäen. Dokumentointiin ja ohjeisiin on myös kiinnitetty erityistä huomiota, jotta sovelluksen jatkokehitys onnistuisi luontevasti.

Lähteet

- [1] Antila Lauri. (2020). *Kodavi-sovellusprojekti, Järjestelmätestausraportti Mac, Safari*. Jyväskylän yliopisto, informaatioteknologian tiedekunta. Saatavilla osoitteessa http://sovellusprojektit.it.jyu.fi/kodavi/Testaus/J%c3%a4rjestelm%c3%a4testaus/kodavi_jarjestelmatestausraportti_mac_safari_2_0_0.pdf.
- [2] Hilola Outi. (2020). *Kodavi-sovellusprojekti, Käytettävyytestausraportti*. Jyväskylän yliopisto, informaatioteknologian tiedekunta. Saatavilla osoitteessa http://sovellusprojektit.it.jyu.fi/kodavi/Testaus/Kaytettavyystestaus/kodavi_kaytettavyystestausraportti_1_0_0.pdf.
- [3] Hilola Outi. (2020). *Kodavi-sovellusprojekti, Palvelimen asennusohje*. Jyväskylän yliopisto, informaatioteknologian tiedekunta. Saatavilla osoitteessa http://sovellusprojektit.it.jyu.fi/kodavi/Ohjeet/kodavi_palvelimen_asennusohje_1_0_0.pdf.
- [4] Hilola Outi. (2020). *Kodavi-sovellusprojekti, Vaatimusmäärittelydokumentti*. Jyväskylän yliopisto, informaatioteknologian tiedekunta. Saatavilla osoitteessa http://sovellusprojektit.it.jyu.fi/kodavi/Vaatimukset/kodavi_vaatimusmaarittely_1_0_0.pdf.
- [5] Kauppi Antti. (2020). *Kodavi-sovellusprojekti, Järjestelmätestaussuunnitelma*. Jyväskylän yliopisto, informaatioteknologian tiedekunta. Saatavilla osoitteessa http://sovellusprojektit.it.jyu.fi/kodavi/Testaus/J%c3%a4rjestelm%c3%a4testaus/kodavi_jarjestelma-testaussuunnitelma_1_0_0.pdf.
- [6] Lappalainen Karoliina, Moisio Tuomas, Nykänen Visa, Juujärvi Jarmo, Korhonen Kai, Kallio Sami, Moisio Juha, Paananen Ilari. (2019). *Moveo-projekti, Sovellusraportti*. Jyväskylän yliopisto, informaatioteknologian tiedekunta. Saatavilla osoitteessa http://sovellusprojektit.it.jyu.fi/moveo/dokumentit/sovellusraportti/Moveo_Sovellusraportti_1.0.0.pdf.

- [7] Rantanen Nuutti. (2020). *Kodavi-sovellusprojekti, Projektiraportti*. Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [8] Vaarala Anne. (2020). *Kodavi-sovellusprojekti, Järjestelmätestausraportti iPad, Chrome*. Jyväskylän yliopisto, informaatioteknologian tiedekunta. Saatavilla osoitteessa http://sovellusprojektit.it.jyu.fi/kodavi/Testaus/J%c3%a4rjestelm%c3%a4testaus/kodavi_jarjestelmatestausraportti_iPad_chrome_1_0_0.pdf.
- [9] Vaarala Anne. (2020). *Kodavi-sovellusprojekti, Järjestelmätestausraportti PC, Chrome*. Jyväskylän yliopisto, informaatioteknologian tiedekunta. Saatavilla osoitteessa http://sovellusprojektit.it.jyu.fi/kodavi/Testaus/J%c3%a4rjestelm%c3%a4testaus/kodavi_jarjestelmatestausraportti_PC_chrome_2_0_0.pdf.
- [10] Vaarala Anne. (2020). *Kodavi-sovellusprojekti, Järjestelmätestausraportti PC, Firefox*. Jyväskylän yliopisto, informaatioteknologian tiedekunta. Saatavilla osoitteessa http://sovellusprojektit.it.jyu.fi/kodavi/Testaus/J%c3%a4rjestelm%c3%a4testaus/kodavi_jarjestelmatestausraportti_PC_firefox_2_0_0.pdf.
- [11] Ylä-Sahra Arttu. (2020). *Loppukatselmoinnin raportti*. Saatavilla osoitteessa http://sovellusprojektit.it.jyu.fi/kodavi/Palaverit/Katselmoinnit/kodavi_2_koodikatselmointi_raportti.md.