

Tietokantasovellus

Hiihtokisojen tulospalvelu

Jani Luukko

Sisällysluettelo

Johdanto.....	3
Tavoitteet.....	3
Käyttöohje.....	3
Käyttäjärühmät.....	3
Järjestelmän yleisrakenne.....	3
Asennuksen tiedot.....	4
Asennusohjeet omalle palvelimelle.....	4
Käyttötapaukset.....	4
Jokamiehen käyttötapaukset.....	4
Kisojen järjestäjän käyttötapaukset.....	5
Kirjaajan käyttötapaukset.....	5
Järjestelmän tietosisältö.....	6
Kisa.....	6
Kilpailija.....	6
KisaLahtolista.....	7
KisaAika.....	7
Testaus, bugit & jatkokehitys.....	7
Omat kokemukset.....	7
Käyttöliittymä ja järjestelmän komponentit.....	8

Johdanto

Harjoitustyöni aiheena on hiihtokisojen tulospalvelu, joka on otettu Advanced Kittenry-sivuston valmiista aiheista. Tulospalvelun tarkoitus välittää sen käyttäjille väli- ja lopputuloksia hiihtokisoista reaaliajassa sekä antaa kisojen järjestäjille mahdollisuus ilmoittaa kisojen tuloksista nopeasti ja vaivattomasti sieltä missä kisat ovat.

Tavoitteet

- Järjestelmä antaa hiihtokisojen ystäville helpon tavan saada selville eri kisojen tulokset
- Järjestelmä antaa hiihtokisojen järjestäjille sulavan tavan ilmoittaa netissä kisojen väli- ja lopputuloksia

Järjestelmä toteutetaan PHP:llä. Tällä hetkellä suunnitelmissa on toteuttaa järjestelmälle rajapinta, jota sitten AngularJS-kirjastolla toteutettu käyttöliittymä.

Se on suunniteltu toimimaan PostgreSQL-tietokannan kanssa. Lisäksi koska järjestelmän käyttöliittymä hyödyntää AngularJS:ää, selaimen pitää tukea JavaScriptiä.

Käyttöohje

Järjestelmä on testattavissa osoitteessa <http://jjluukko.users.cs.helsinki.fi/tulospalvelu/>

Se toimii sekä tietokoneilla että moderneilla mobiililaitteilla.

Käyttäjryhmät

Järjestelmällä on kolme käyttäjäryhmää: Jokamiehet (ts. kisojen seuraajat), kisojen järjestäjät ja kirjaajat.

Jokamiehellä tarkoitetaan ketä tahansa, joka käyttää järjestelmää anonyyminä. (ts. ei-kirjautuneena)

~~Kisojen järjestäjä on rekisteröitynyt henkilö, joka luo mm. kisan järjestelmään.~~

~~Kirjaaja on myös rekisteröitynyt henkilö, joka pystyy kirjaamaan aikoja järjestelmässä oleviin kisoihin joihin kisojen järjestäjä on antanut oikeuden.~~

(Koska en ehtinyt perehtyä asiaan, järjestelmässä ei ole rekisteröityneitä käyttäjiä. Kuka tahansa voi käyttää järjestelmää ja sen toimintoja miten haluaa.)

Järjestelmän yleisrakenne

Tietokantasovellusta tehdessä on noudatettu MVC-mallia. Kontrollerit, näkymät ja mallit sijaitsevat hakemistoissa controllers, views- ja models. Käytetyt apukirjastot on sijoitettu hakemistoon lib ja asetukset ovat tiedostossa settings.php.

Järjestelmä on luotu niin että kaikki operaatiot tehdään rajapintaan, joka ottaa ja palauttaa JSON-dataa. Tätä hyödynnetään sitten frontend-puolella siten että Angularin kontrollerit hakevat kaiken tarvittavan datan rajapinnasta ja renderöi näkymät itse. Järjestelmää voi täten helposti jatkokehittää

esimerkiksi luomalla kännykkäsovelluksen joka hyödyntää samaa rajapintaa.

Juuresta löytyvästä assets-kansiosta löytyy staattisen sisällön (kuvat, tyylitiedostot, javascript) myös Angularin käyttämät partial-templatet, joihin rajapinnasta saatava data renderöidään.

Asennuksen tiedot

Järjestelmä on nyt asennettu laitoksen users-palvelimelle Kalle Ilveksen ohjeita käyttämällä. Tähän on sisältynyt muun muassa Tsoha-Bootstrapista löytyneiden bash-skriptojen (bootstrap.sh, deploy.sh) käyttö asennus- ja kehitysvaiheessa. Tietokantaan liittyvien asetusten konfiguraatio löytyy config/database.php-tiedostosta.

Asennusohjeet omalle palvelimelle

Ottaaksesi järjestelmän käyttöön, sinun pitää tehdä seuraavat toimenpiteet. Nämä ohjeet saattavat vaihdella riippuen siitä miten olet konfiguroinut palvelimesi, mutta nämä vaiheet on syytä pitää muistissa.

1. Pura repo kansioon johon www-palvelimellasi on pääsy
2. Konfiguroi config/database.php-tiedostoon tietokantasi asetukset. Huomioi, että järjestelmää on vain testattu PostgreSQL:llä mutta saattaa toimia myös MySQL:llä. (Ei ole testattu)
3. Asenna Composer (kts. <https://getcomposer.org/>)
4. Kun olet asentanut Composerin, lataa vaadittavat kirjastot menemmällä repon juureen ja suorita Composerin install-komento joka lataa tarvittavat kirjastot.
5. Tämän jälkeen siirry sql-kansioon ja suorita komento `psql < create_tables.sql` (tämä luo tarvittavat taulut)
6. Järjestelmä on käytettävissä! Jos haluat testidataa, suorita vielä `psql < add_test_data.sql`

Käyttötapaukset

Jokamiehen käyttötapaukset

Kisojen listaaminen

Kuka tahansa pystyy saamaan järjestelmästä listan järjestelmässä olevista kisoista.

Lähtölistan katselu

Kuka tahansa pystyy saamaan kisojen lähtölistan järjestelmästä.

Väliaikapisteen tilanteen selvittäminen

Kuka tahansa pystyy selvittää kisojen tilanteen tietyllä väliaikapisteellä. Käyttäjä saa selville myös ketkä kilpailijoista eivät ole vielä saapuneet haetulle väliaikapisteelle.

Lopputulosten selvittäminen

Kuka tahansa saa haettua järjestelmästä kisojen lopputulokset.

Kilpailijan sijoittumisen selvittäminen

Kuka tahansa pystyy selvittämään miten kilpailija on edennyt matkallaan. (ts. sijoitukset lähdöstä maaliin eri väliaikapisteiden kohdilla)

Kisojen järjestäjän käyttötapaukset

(ts. ”jokamiehen”, koska rekisteröityneitä käyttäjiä ja ryhmiä ei ole)

Kilpailijan lisääminen

Järjestäjä pystyy lisäämään kilpailijan järjestelmään.

Kilpailijan muokkaus

Järjestäjä pystyy muokkaamaan kilpailijan tietoja järjestelmässä.

Kilpailijan poisto

Järjestäjä pystyy poistamaan kilpailijan järjestelmästä.

Kisan luonti

Järjestäjä pystyy luomaan kisan järjestelmään. Järjestäjä voi kisojen nimen lisäksi määrittää missä ne pidetään ja milloin.

Kisan muokkaus

Järjestäjä pystyy muokkaamaan hänen järjestelmään luoman kisan.

Kilpailijan lisääminen kisaan

Järjestäjä pystyy lisäämään kilpailijan hänen järjestelmään luomaan kisaan.

Kisan lähtölistan laatiminen

Järjestäjä pystyy laatimaan kisan lähtölistan.

Kirjaajan lisääminen kisaan

Järjestäjä pystyy määrittämään kisalle kirjaajia, jotka voivat kirjata kisan tuloksia järjestelmään.

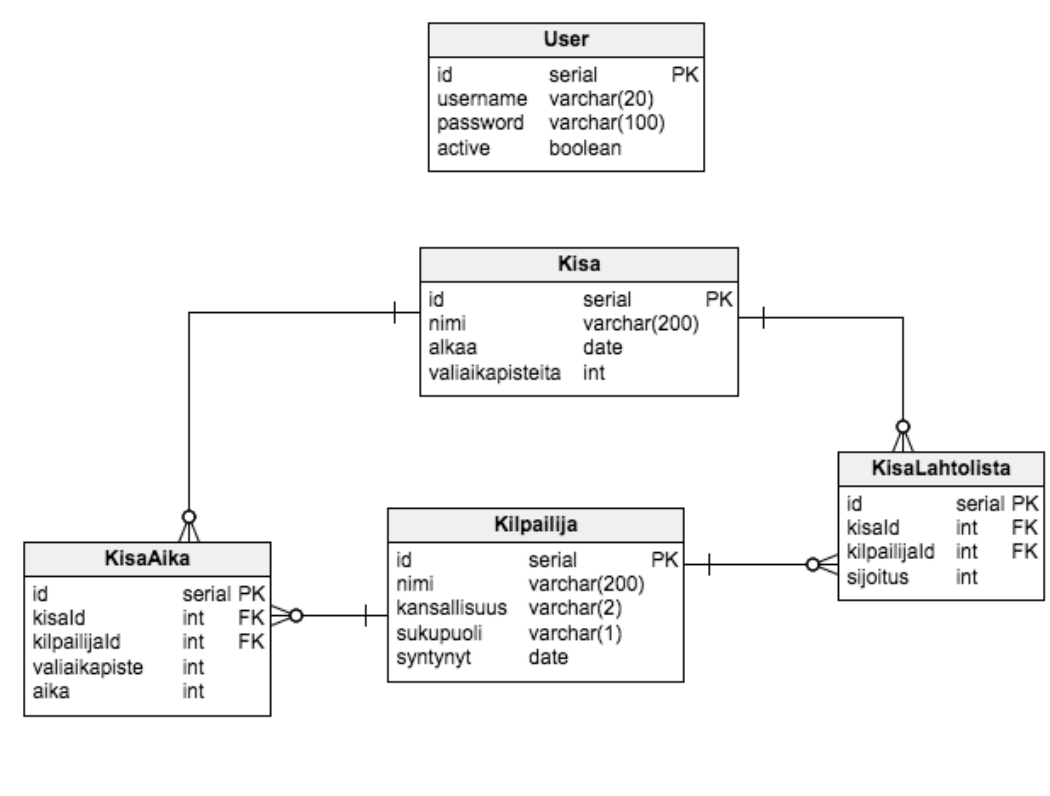
Kirjaajan käyttötapaukset

(ts. ”jokamiehen”, koska rekisteröityneitä käyttäjiä ja ryhmiä ei ole)

Väliajan/loppuajan kirjaus

Kirjaaja pystyy kirjaamaan kisan väli- ja loppuaikoja järjestelmään.

Järjestelmän tietosisältö



Kisa

Itse hiihtokisa. Hiihtokisalla on lähtölistan sijoituksia sekä aikoja, joihin kumpaankin on liitetty myös Kilpailija. (kuvailtu alempana)

Attribuutti	Arvojoukko	Kuvailu
id	serial	Kisan id
nimi	Merkkijono, max 200 merkkiä	Kisan nimi, esim. ”Kumpulan hiihtokisat”
alkaa	Päivämäärä	Kisojen kisapäivä
valiaikapisteita	Kokonaisluku	Kuinka monta väliaikapistettä kisassa on

Kilpailija

Kilpailijalla voi olla useita aikoja sekä useita lähtösijoituksia taulussa KisaLahtolista.

Attribuutti	Arvojoukko	Kuvailu
id	serial	Kilpailijan id
nimi	Merkkijono, max 200 merkkiä	Kilpailijan nimi, esim. ”Matti Meikäläinen”
kansallisuus	Merkkijono, max 2 merkkiä	Kilpailijan kansallisuus, esim. ”FI” (Suomi)

Attribuutti	Arvojoukko	Kuvailu
sukupuoli	Merkkijono, max 1 merkki	Kilpailijan sukupuoli, esim. ”M” (mies)
syntynyt	Päivämäärä	Kilpailijan syntymäpäivä

KisaLahtolista

KisaLahtolista sisältää kisojen lähtölistat. Jokainen rivi vastaa yhden kilpailijan lähtösijoitusta liitettyssä kisassa.

Attribuutti	Arvojoukko	Kuvailu
id	serial	Kisan lähtölistan sijoituksen id
kisaId	Kokonaisluku	Kisa johon viitataan (FK)
kilpailijaId	Kokonaisluku	Kilpailija johon viitataan (FK)
sijoitus	Kokonaisluku	Kilpailijan sijoitus lähtölistassa

KisaAika

Kilpailija aika tietyllä väliaikapisteellä tietyssä kisassa.

Attribuutti	Arvojoukko	Kuvailu
id	serial	Kisan id
kisaId	Kokonaisluku	Kisa johon viitataan (FK)
kilpailijaId	Kokonaisluku	Kilpailija johon viitataan (FK)
valiaikapiste	Kokonaisluku	Monennen väliaikapisteen ajasta on kyse
aika	Kokonaisluku	Kilpailijan aika (millisekunteina) lähdöstä rivin väliaikapisteeseen

Testaus, bugit & jatkokehitys

Järjestelmää on testattu ja debugattu Google Chromella ja järjettömällä järjestelmän kliksetelulla. Järjestelmässä tällä hetkellä voi huomata että sijoitus saattaa ilmestyä listaan kahdesti tietyissä tapauksissa.

Rajapinnan ansiosta järjestelmän ympärille on mahdollista luoda muita samaa tietokantaa hyödyntäviä sovelluksia eri alustoille. Nykyiseen järjestelmään voisi tosin lisätä graaffin, joka näyttää kilpailijoiden edistymisen eri välipisteillä.

Omat kokemukset

Tällaisen sovelluksen teko oli jo tuttua puuhaa, joten halusin pientä haastetta tähän ottamalla mukaan Angularin. Valitettavasti tämä toi myös sen haasteen että autentikointi olisi pitänyt hoitaa muulla tavalla kuin sessioilla. Ajatuksena oli käyttää tähän JSON Web Tokeneita, mutta valitettavasti aika ei vain riittänyt siihen.

(Jätetty viimeiselle sivulle sen tarvittavan tilan vuoksi)

