

Mika Huttunen

# *Tietokantasovellus*

1. Johdanto	s. 3
2. Käyttötapaukset	s. 3
3. Tietosisältö	s. 6
4. Relaatiotietokantakaavio	s. 9
5. Järjestelmän yleisrakenne	s. 9
6. Käyttöliittymä	s. 10
7. Asennusohjeet	s. 11
8. Käynnistysohje	s. 11
9. Testaus, bugit ja jatkokehitysideat	s. 11
10. Omat kokemukset	s. 12
11. Liitteet	s. 12

## Johdanto

Työn aiheenani on Hiihtokisojen tulospalvelu, joka toteutetaan kurssisivuilla olevan ko. esimerkkiaiheen mukaisesti.

Tulospalvelutietokannasta voivat tavalliset käyttäjät selvittää monenlaisia kilpailuun liittyviä tietoja; mm. kisan osallistujat, heidän sijoittumisensa tiettyjen väliaikapisteiden kohdalla sekä tietysti lopputulokset.

Tavallisille käyttäjille kuuluvien toimintojen lisäksi tietokantaan voi tietysti lisätä tarvittavaa tietoa em. asioihin liittyen. Nämä toiminnot vaativat kuitenkin järjestelmään kirjautumisen.

-Toteutusympäristönä käytössä on Tomcat.

-Web-sovelluksen alustajärjestelmän tulee tukea Java-ohjelmointikieltä.

-Ohjelmisto edellyttää PostgreSQL tietokannan käyttöä.

## Käyttötapaukset

### Käyttäjät

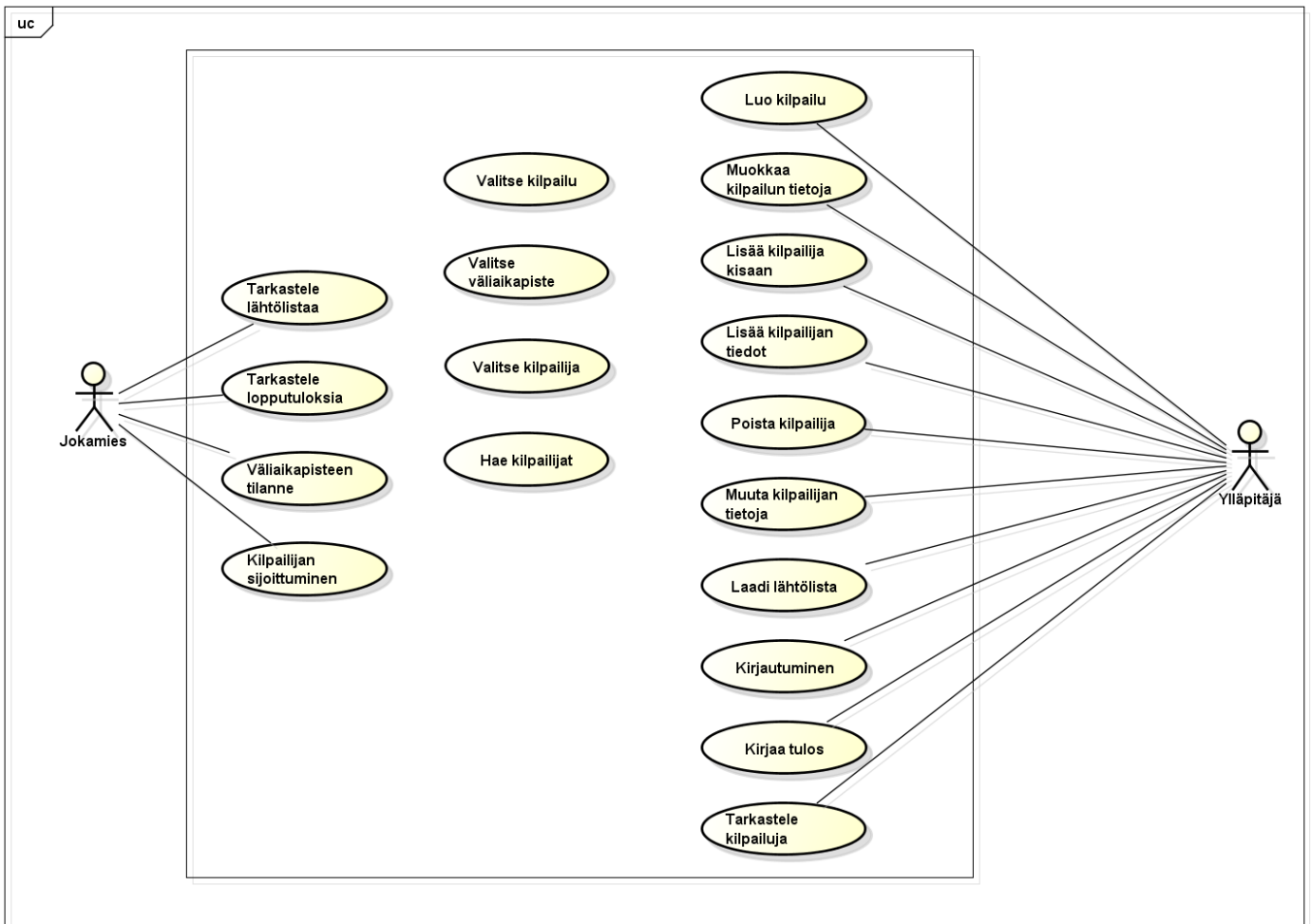
#### Jokamies

Jokamiehellä tarkoitetaan ketä tahansa, joka Internetin välityksellä tulee katselemaan Hiihtokisojen tulospalvelun www-sivuja.

#### Ylläpitäjä

Ylläpitäjä(t) omaavat käyttäjätunnuksen, jolla voivat kirjautua järjestelmään ja tehdä tätä kautta järjestelmään muutoksia. He voivat lisätä kilpailuista tietoa ja päivittää niitä. Ylläpitäjä voi myös tarkastella sivustoa Jokamiehen tavoin.

## Käyttötapaukset



powered by Astah

### Jokamies

#### **Väliaikapisteiden tilanne**

- Esiehto: Väliaikapiste valittu
- Käyttäjä pääsee tarkastelemaan tilannetta valitulla väliaikapisteellä

#### **Kilpailijan sijoittuminen**

- Esiehto: Kilpailija valittu
- Käyttäjä pääsee tarkastelemaan valitun kilpailijan sijoitusta valitulla väliaikapisteellä. Mikäli väliaikapistettä ei ole valittu, näytetään kilpailijan sijoitus "uusimmalla" väliaikapisteellä, missä kilpailija on ollut (mahd. maali).

Muita käyttötapauksia: tarkastele lähtölistää, tarkastele lopputuloksia

## Ylläpitäjä

### **Luo kilpailu**

- Esiehto: Kirjautuminen
- Ylläpitäjä syöttää kilpailun tiedot ja kilpailu tulee näkyviin ylläpitäjälle järjestelmään

### **Laadi lähtölista**

- Esiehto: Kilpailu valittu
- Ylläpitäjä laatii lähtölistan ko. kilpailun osallistujista. Lähtölista tulee julkisesti näkyviin järjestelmään.

### **Kirjaa tulos**

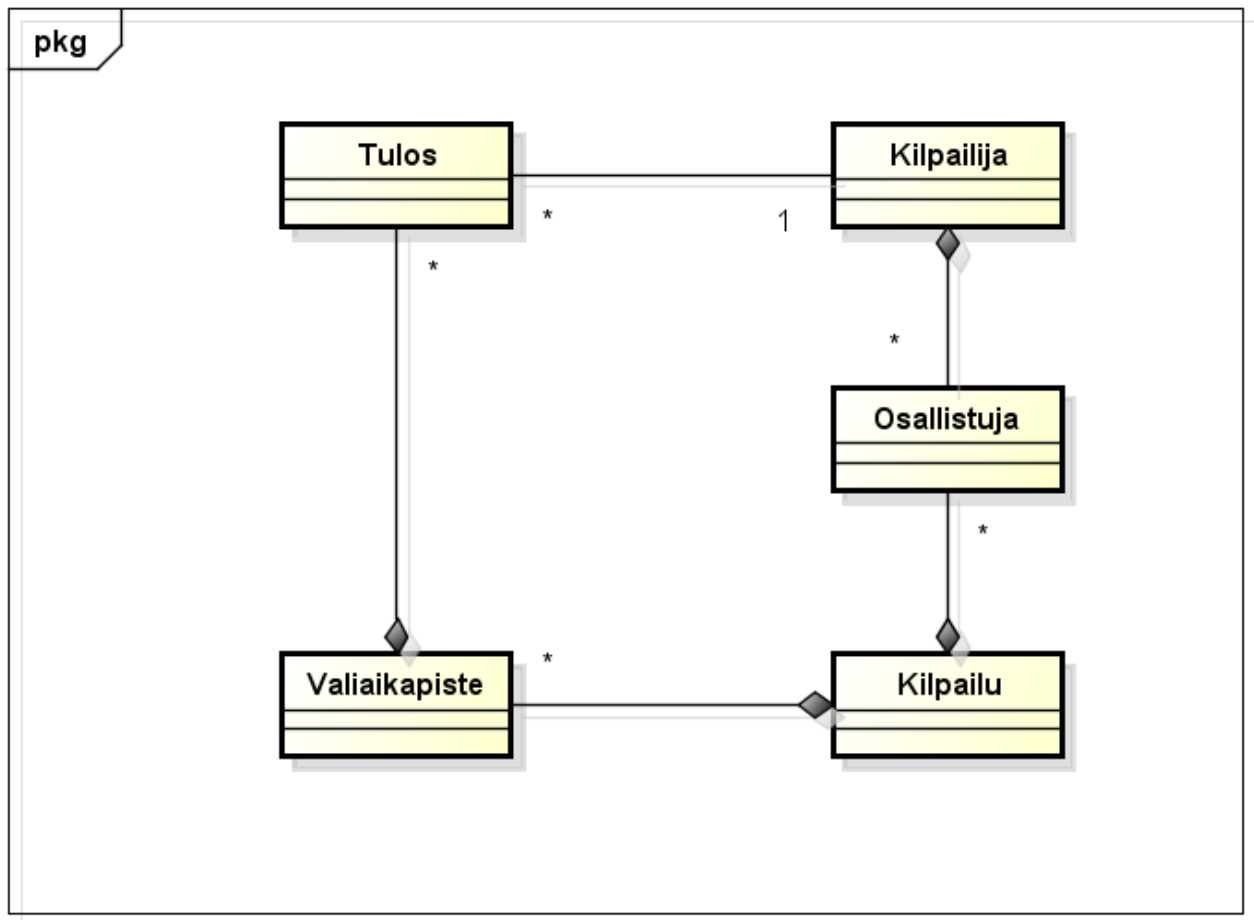
- Esiehto: Kilpailija valittu
- Ylläpitäjä kirjaa valitun kilpailijan tuloksen koskien valittua väliaikapistettä. Jos väliaikapistettä ei ole valittu, tulos on lopputulos.

Muita käyttötapauksia: muokkaa kilpailun tietoja, lisää kilpailija kisaan, lisää kilpailijan tiedot, poista kilpailija, muuta kilpailijan tietoja, kirjautuminen, tarkastele kilpailuja

Yleisiä käyttötapauksia, jotka ainoastaan esiehtoina muihin käyttötapauksiin: valitse kilpailu, valitse väliaikapiste, valitse kilpailija, hae kilpailijat

## Tietosisältö

### Tietokantakaavio



powered by Astah

### Taulujen määrittely

#### Tietokohde: kilpailu

Attribuutti	Arvojoukko	Kuvaus
id	Integer	kilpailu(id)
nimi	String	kilpailun nimi

Kilpailulla voi olla useita osallistujia sekä väliaikapisteitä.

**Tietokohde: kilpailija**

Attribuutti	Arvojoukko	Kuvaus
id	Integer	kilpailija(id)
nimi	String	kilpailijan nimi

Kilpailija voi osallistua kilpailuihin roolissa "osallistuja". Kilpailija voi osallistua näin useisiin eri kilpailuihin.

**Tietokohde: tulos**

Attribuutti	Arvojoukko	Kuvaus
id	Integer	tulos(id)
aika	String	Kilpailijan aika ko. väliaikapisteellä
kilpailija	Integer	viiteavain kilpailijaan
valiaikapiste	Integer	viiteavain valiaikapisteeseen

Jokainen tulos kuuluu aina yhdelle kilpailijalle ja kilpailijalla voi olla useita eri tuloksia. Tulos koskee aina tiettyä väliaikapistettä ja on tämän olemassaolosta riippuvainen.

**Tietokohde: valiaikapiste**

Attribuutti	Arvojoukko	Kuvaus
id	Integer	valiaikapiste(id)
numero	Integer	monesko väliaikapiste kilpailussa
kilpailu	Integer	viiteavain kilpailuun

Väliaikapisteeseen liittyy tuloksia ja väliaikapiste on olemassaoloriippuvainen kilpailustaan.

**Tietokohde: osallistuja**

Attribuutti	Arvojoukko	Kuvaus
kilpailu	Integer	viiteavain kilpailuun
kilpailija	Integer	viiteavain kilpailijaan

Osallistuja on olemassaoloriippuvainen sekä kilpailusta että kilpailijasta. Osallistuja on "kilpailijan rooli ko. kilpailussa".

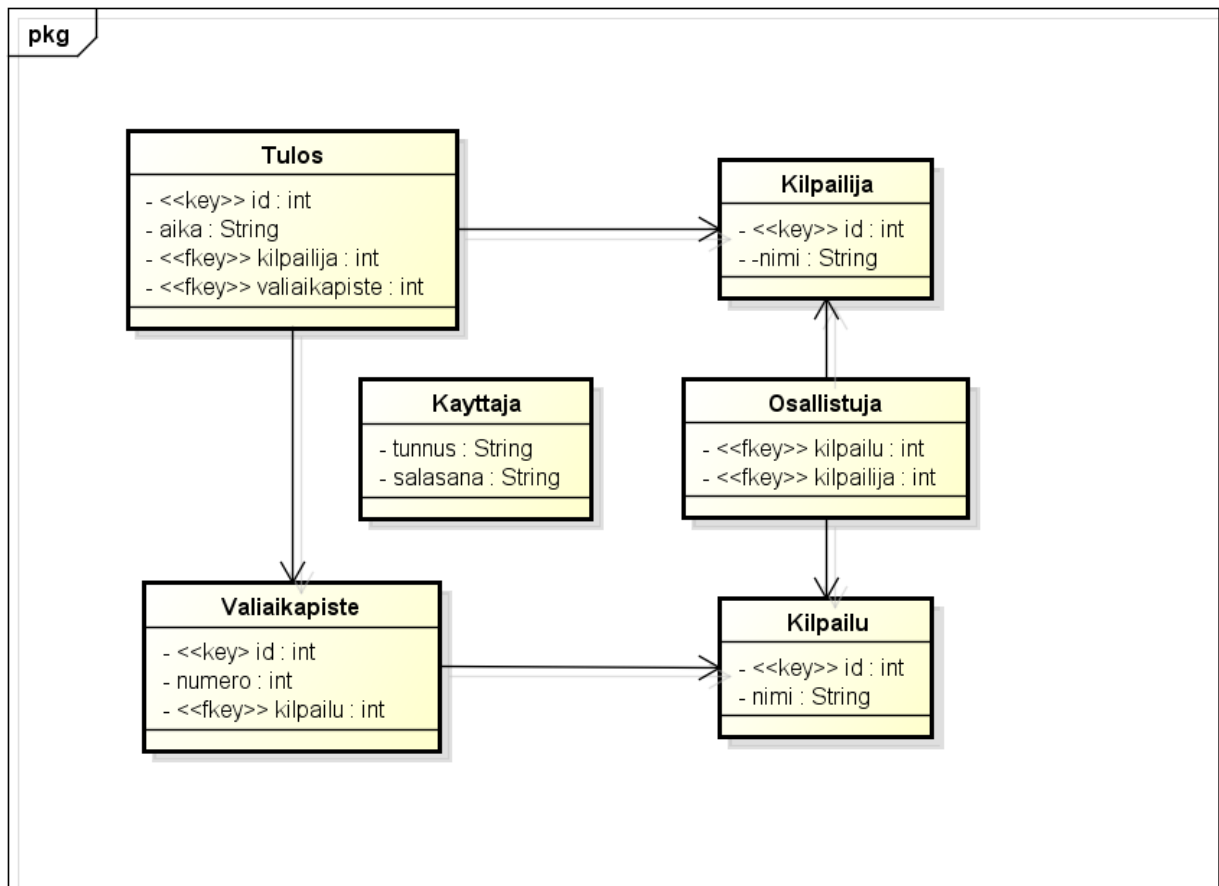
**Tietokohde: käyttaja**

Attribuutti	Arvojoukko	Kuvaus
tunnus	String	käyttäjätunnus
salasana	String	salasana

Tauluun lisätään tunnukset ja salasanat, joilla järjestelmään pääsee kirjautumaan ja tietoja muokkaamaan.



## Relaatiotietokantakaavio



powered by Astah

## Järjestelmän yleisrakenne

Tietokantasovellus on toteutettu MVC-arkkitehtuurin mukaan. Mallit ja servletit sijaitsevat hakemistoissa /src/java/Mallit ja /src/java/Servletit ja näkymät on sijoitettu hakemistoon /web. Servletit on lisäksi jaeteltu pakkauksiin sen mukaan, ovatko ne esim. muokkaus-, poisto- tai lisäystoimintoja varten.

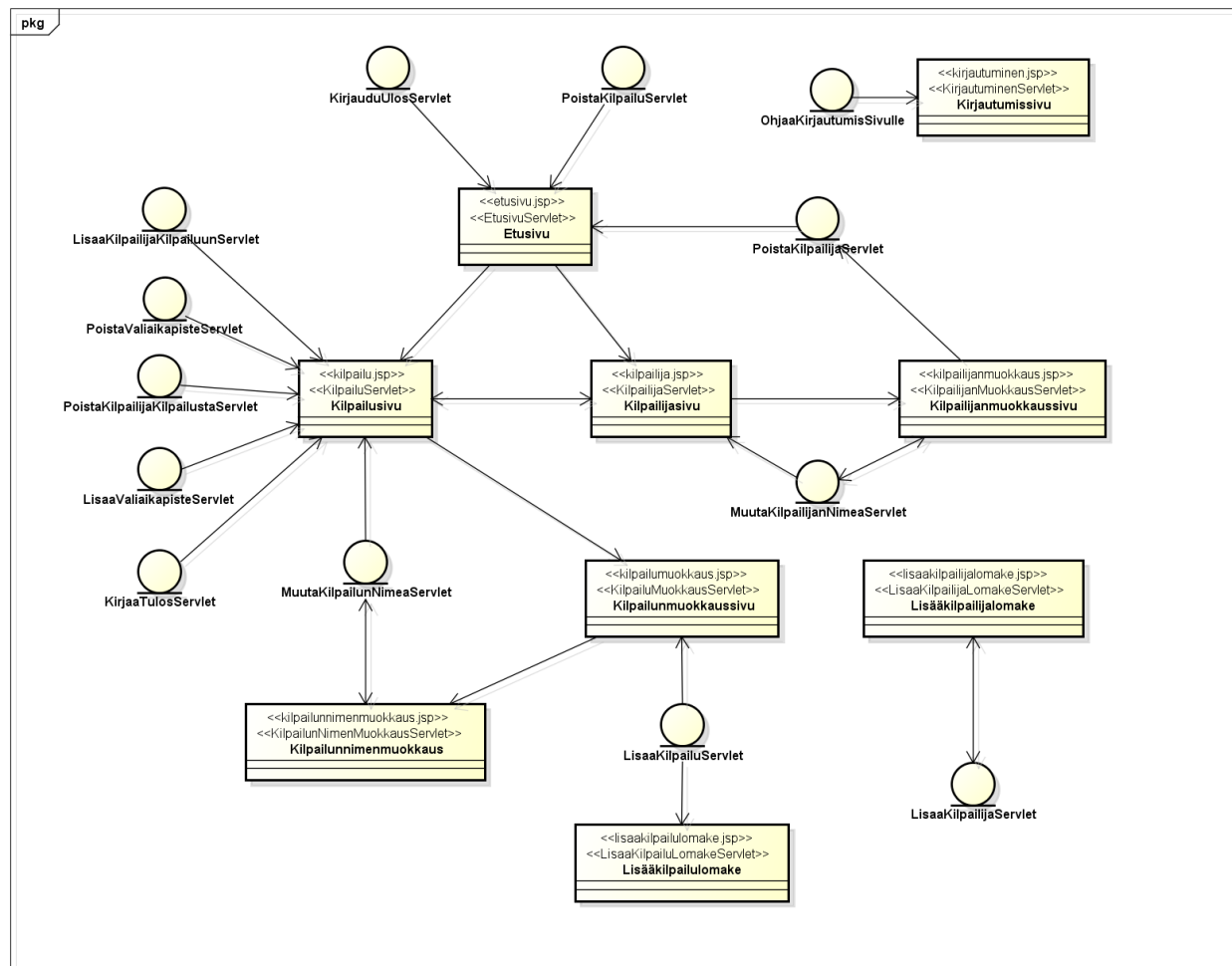
Kaikki näkymät käyttävät lisäksi erillistä tag-tiedostoa "pohja.tag" ja kaikki mallit perivät luokan "KyselyToiminnot", joka sisältää yleisiä toiminnallisuuksia, mitä javalla toteutetut tietokantakyselyt tarvitsevat, kun taas kaikki servletit (= kontrollerit) perivät luokan "YleisServlet", joka sekin sisältää yleisiä toiminnallisuuksia, mitä kontrolletit tarvitsevat.

Projektin näkymät ovat nimetty pienellä ja servletit ja mallit isolla alkukirjaimella käyttäen yleisiä javaohjelmoinnissa käytettyjä nimeämissääntöjä.

Järjestelmä käyttää istuntoa tallentamalla tarpeelliset tiedot HttpServletRequest -olion Session -olioon. Tänne tallennetaan mm. käyttäjän tiedot sekä tarpeen vaatiessa esim. käsitellyssä olevan taulun rivin id-arvo poikkeuksien varalta.

*Liitteenä Javadoc*

## Käyttöliittymä



powered by Astah

Tavallinen käyttäjä pääsee käsiksi ainoastaan sivuihin “Etusivu”, “Kilpailusivu”, “Kilpailijasivu” ja “Kirjautumissivu”. Muiden sivujen selaaminen (kuten myös vastaavien kontrollerien “käyttö” suoraan selaimella) vaatii kirjautumisen.

Sivustolla on navigaatiopalkki, josta kirjautumaton käyttäjä pääsee etusivulle sekä kirjautumissivulle. Kun käyttäjä on kirjautunut sisään, hän näkee navigaatiopalkissa etusivun lisäksi linkit kilpailujen sekä kilpailijoiden lisäämislomakkeisiin sekä napin uloskirjautumista

varten.

## Asennusohjeet

Asenna sovellus kopioimalla sen tiedostot hakemistoon ~/tomcat/webapps. Tämän jälkeen tulee vielä korjata tietokannan yhteystiedot oikeiksi.

Mene siis hakemistoon ~/tomcat/conf/Catalina/localhost, josta tulisi löytyä projektin nimellä kulkeva .xml -tiedosto. Tämän tiedoston sisältö tulee vastata projektin context.xml tiedoston sisältöä. Sisällön voit kopioida tiedostosta context.dist.xml, joka löytyy polusta ~/tomcat/webapps/HiihtoTulosPalvelu/META-INF.

## Käynnistysohje

Web-sovellus löytyy osoitteesta t-mikhuttu.users.cs.helsinki.fi/HiihtoTulosPalvelu/etusivu.

Sisäänkirjautuminen tapahtuu tunnuksella "yllapitaja", salasanalla "qwerty123", jolloin pääsee muokkaamaan järjestelmän tietoja vapaasti.

## Testaus, bugit ja jatkokehitysideat

Sovelluksen kaikkia osa-alueita ja toiminnallisuuksia on testattu käsin Netbeansiä käyttäen. Siinä ei pitäisi olla bugeja; ainoa ongelma johon olen törmännyt on se että tietokantayhteys hajoaa joskus Netbeansillä testaten, mutta tämä saattaa johtua yhtä hyvin myös siitä että tietokantayhteys sulkeutuu satunnaisesti, kun yhteys on avattu Windowsin PuTTY:llä.

Sovellusta voisi kehittää mm. seuraavilla tavoilla:

- mahdollisuus kirjata ainoastaan kilpailijan tulos seuraavalle väliaikapisteelle, johon ko. kilpailija ei saapunut
- mahdollisuus poistaa kilpailijan tulos joltakin väliaikapisteeltä (tällä hetkellä voi vain korvata uudella / poistaa kilpailija kilpailusta kokonaan)
- otetaan ylläpitäjän antama syöte talteen, kun hän haluaa muuttaa kilpailijan tai kilpailun nimeä ja uusi nimi ylittää sallitun kirjainmerkkirajan
- kilpailun viimeisen väliaikapisteen näyttäminen selvemmin "maalina" ja kilpailijoiden sijoitusten näyttäminen kilpailun lopputuloksissa

Omasta mielestäni sovellus ei tällaisia parannuksia toisaalta edes tarvitse, mutta ainahan siihen voisi jotain hienosäätöä tehdä. Näiden lisäksi voitaisiin verkkosivuja tietysti jotenkin koristaa.

## Omat kokemukset

Tietokantasovelluksen tekeminen oli alussa hankalaa ja sen ohjelmoiminen Javan kanssa tuotti päänvaivaa; monet ohjaajatkaan eivät olleet sitä käyttäneet niin apua ei saanut ihan riittävästi. Toinen ongelma itselläni oli se että saatoin sählätä jotain users-palvelimen käyttöönottovaiheessa, minkä takia olen joutunut korjaamaan palvelimella olevan projektiin liittyvän xml-tiedoston tietoja aina manuaalisesti, kun olen uuden version palvelimelle lisännyt.

Kurssin loppuvaiheessa itselläni ei työn tekemisen kanssa kuitenkaan juuri ongelmia ollut, kun projektin tekemiseen oli päässyt paremmin sisälle - ja tietokantayhteyskin toimi, joka oli ollut xml-tiedoston tietojen puuttumisen takia hajalla parikin viikkoa.

Olisi ollut mielenkiintoista koittaa myös php:tä, mutta kyllä Javalla tekiessäkin oppi paljon saamalla painia kaikenlaisten ongelmien kanssa. Mutta kun monet yleistoiminnallisuudet oli kasassa, niin sovelluksen tekeminen sujui suhteellisen helposti. SQL-kyselyjen kirjoittamisen kanssa oli toisaalta joskus vaikeaa, koska Tietokantojen perusteiden suorittamisesta oli jo reilu vuosi, mutta kyllä nekin sai tekemään oikeat asiat, kun jelpiä muilta pyysin.

## Liitteet

Git repositoriossa:

[1] Javadoc, <https://github.com/mikhuttu/TsoHa/tree/master/doc/javadoc>

[2] SQL create-table -lauseet, <https://github.com/mikhuttu/TsoHa/tree/master/sql>