

Aineopintojen harjoitustyö: Tietokantasovellus

Jesper Ruuth

# Tietokantasovellus

Dokumentaatio

# 1 Esittely

Tämän projektin tarkoituksena on toteuttaa tämän vuoden kesä-heinäkuu välisenä aikana Chilessä pelattavien jalkapallon Etelä-Amerikan mestaruuskilpailujen ottelutulokset ja maaliveikkaussovellus.

Rekisteröitymällä sovellukseen käyttäjä voi vaikata kunkin pelattavan ottelun lopputulosta. Veikkaamalla oikein käyttäjät ansaitsevat pisteitä ja heidät sijoitetaan ansaittujen pisteiden perusteella julkiselle rankinglistalle. Lisäksi, käyttäjät voivat muodostaa kaveriryhmiä, joiden sisällä he voivat kilpailla keskinäisestä paremmuudestaan. Veikkaustoimintojen lisäksi, sovellus tulee tarjoamaan koko turnauksen kattavan otteluohjelman, tulospalvelun sekä tiedot kaikista joukkueista ja niiden pelaajista.

Sovellus tulee soveltumaan ensisijaisesti Copa America turnauksen otteluveikkausten kirjaus- ja pisteidenlaskujärjestelmäksi, korvaamaan perinteisemmät kynä-paperi tai solutaulukko menetelmät, mutta sovellusta voidaan käyttää myös otteluohjelman ja -tulosten seurantaan sekä turnauksen joukkueisiin ja pelaajiin tutustumiseen.

Toteutuksen laadullisina tavoitteina on tuottaa toimiva, helppokäyttöinen, mobiilioptimoitu ja tietoturvallinen web-sovellus, joka on rakennettu vakaan ja toimivan arkkitehtuurin päälle noudattamalla ns. clean code periaatteita. Lisäksi, sovelluksen suunnittelussa tullaan ottamaan huomioon sovelluksen jatkokehitettävyyden skaalattavuus esim. tulevia arvokilpailuja ja seurajoukkueiden sarjoja ja turnauksia varten.

Sovellusta tullaan kehityksen aikana ajamaan Helsingin yliopiston TKTL:n users-palvelimella, jossa sovellus pyörii Apache-palvelimen alla. Lähdekoodin versionhallinnasta vastaa git.

Palvelinpuolen lähdekoodi tullaan kirjoittamaan PHP-kielellä, joten sovellusta ylläpitävän palvelimen tulee tukea vähintään PHP:n versioita 5.4. Tämän lisäksi, sovellus tulee käyttämään PostgreSQL tietokantaa, jonka tukea sovellus tulee myös vaatimaan palvelimelta.

Sovellus tulee toimimaan kaikilla nykyaikaisilla web-selaimilla, sekä tietokoneissa että mobiililaitteissa. Sovelluksen front-end toiminnallisuudet tulee kuitenkin jossain määrin vaatimaan JavaScript -tukea selaimelta.

## 2 Käyttötapaukset

### 2.1 Käyttäjärühmät

#### **Vierailija**

Sovelluksen käyttäjä, joka ei omista käyttäjätiliä. Vierailija voi siis olla kuka tahansa sovelluksen sivuille saapuva internetin käyttäjä.

#### **Rekisteröitynyt käyttäjä**

Sovelluksen rekisteröitynyt käyttäjä, jolla on tavallisen käyttäjän oikeudet. Käyttäjällä on oma henkilökohtainen käyttäjätili ja hän voi olla kuka tahansa rekisteröinnin suorittanut internetin käyttäjä.

#### **Ylläpitäjä**

Sovelluksen rekisteröitynyt käyttäjä, jolla on ylläpitäjän oikeudet. Ylläpitäjällä on oikeus sovelluksen ylläpidollisiin toimintoihin ja hän voi olla kuka tahansa käyttäjä, jolle on annettu ylläpitäjän oikeudet.

#### **Ryhmän jäsen**

Sovelluksen rekisteröitynyt käyttäjä, joka on jäsenenä jossain ryhmässä. Ryhmän jäsenellä on pääsy ryhmän ja ryhmän muiden jäsenten sivulle ja hän voi olla kuka tahansa käyttäjä, joka on vastaanottanut ja hyväksynyt ryhmäkutsun.

#### **Ryhmän ylläpitäjä**

Sovelluksen rekisteröitynyt käyttäjä, joka ylläpitäjän jossain ryhmässä. Ryhmän ylläpitäjällä on ylläpitäjän oikeudet ryhmän sivulle ja hän voi olla kuka tahansa ryhmän perustanut käyttäjä tai käyttäjä, jolle on annettu ryhmän ylläpitäjän oikeudet.

### 2.2 Käyttötapauskuvaukset

#### **Käyttäjätilin luominen**

Vierailijan suorittama tapahtuma, jossa tämä rekisteröityy sovellukseen luomalla itselleen uuden käyttäjätilin. Tapahtuman seurauksena, käyttäjän tiedot tallentuvat järjestelmään.

#### **Käyttäjätilin jäädyttäminen**

Ylläpitäjän suorittama tapahtuma, jossa rekisteröityneen käyttäjän käyttäjätili lukitaan esim. epäasiallisen käytöksen seurauksena. Tapahtuman seurauksena, käyttäjältä evätään pääsy järjestelmään, ilman että tämän tietoja poistetaan.

### **Käyttäjätilin muokkaaminen**

Käyttäjän suorittama tapahtuma, jossa tämä muokkaa (esim. muuttaa salasanaa) oman käyttäjätilinsä tietoja. Tapahtuman seurauksena, käyttäjän tiedot päivittyvät.

### **Käyttäjätilin poistaminen**

Käyttäjän suorittama tapahtuma, jossa käyttäjä poistaa oman tilinsä esim. halusta lopettaa sovelluksen käyttö. Tapahtuman seurauksena, kaikki käyttäjään liittyvät tiedot poistetaan järjestelmästä.

### **Käyttäjätilin selaaminen**

Käyttäjän suorittama tapahtuma, jossa tämä selailee oman tai sellaisen käyttäjän tietoja, johon tällä on lukuoikeus.

### **Ottelutulosten kirjaaminen**

Ylläpitäjän suorittama tapahtuma, jossa tämä päivittää jonkin pelatun ottelun lopputuloksen. Tapahtuman seurauksena, ottelun tulos sekä käyttäjien otteluun tekemät veikkaukset pisteytyvät ja päivittyvät.

### **Ryhmään liittyminen**

Käyttäjän suorittama tapahtuma, jossa tämä liittyy avoimeen ryhmään tai hyväksyy vastaanotetun ryhmäkutsun. Tapahtuman seurauksena, käyttäjästä tulee ryhmän jäsen.

### **Ryhmän luominen**

Käyttäjän suorittama tapahtuma, jossa tämä perustaa uuden ryhmän esim. kaveriporukalle. Tapahtuman seurauksena, järjestelmään luodaan uusi ryhmä.

### **Ryhmäkutsun lähettäminen**

Ryhmän ylläpitäjän suorittama tapahtuma, jossa tämä lähettää ryhmäkutsun jollekin sovelluksen käyttäjistä. Tapahtuman seurauksena, sovellus ilmoittaa vastaanottajaa saapuneesta ryhmäkutsusta.

### **Ryhmän muokkaaminen**

Ryhmän ylläpitäjän suorittama tapahtuma, jossa tämä muokkaa ryhmään liittyviä tietoja. Tapahtuman seurauksena, ryhmän tiedot päivittyvät.

### **Ryhmästä poistuminen**

Ryhmän jäsenen tai ylläpitäjän suorittama tapahtuma, jossa käyttäjän jäsenyys poistetaan esim. jäsenen omasta halusta lopettaa jäsenyys tai erottamalla tämä ylläpitäjän toimesta. Tapahtuman seurauksena, jäsenen tiedot poistetaan ryhmän tiedoista, kuitenkin poistamatta tämän käyttäjä- tai veikkaustietoja.

### **Ryhmän selaaminen**

Ryhmän jäsenen tai ylläpitäjän suorittama tapahtuma, jossa tämä selailee ryhmään liittyviä tietoja.

### **Sisäänkirjautuminen**

Käyttäjän suorittama tapahtuma, jossa tämä kirjautuu sovellukseen. Tapahtuman seurauksena, käyttäjää varten perustetaan uusi istunto.

### **Turnaustietojen muokkaaminen**

Ylläpitäjän suorittama tapahtuma, jossa tämä muokkaa jotain turnaukseen liittyvää tietokohdetta esim. pudotuspelivaiheen ottelupareja. Tapahtuman seurauksena, kyseisen tietokohteen tiedot päivittyvät.

### **Turnaustietojen selaaminen**

Käyttäjän suorittama tapahtuma, jossa tämä selailee jotain turnaukseen liittyvää tietokohdetta esim. päivän otteluohjelmaa.

### **Uloskirjautuminen**

Käyttäjän suorittama tapahtuma, jossa tämä kirjautuu ulos sovelluksesta. Tapahtuman seurauksena, käyttäjää varten varattu istunto lopetetaan.

### **Veikkauksen luominen**

Käyttäjän suorittama tapahtuma, jossa tämä veikkaa jonkin tulevaisuudessa pelattavan ottelun (esim. illan ottelun) lopputulosta. Tapahtuman seurauksena, käyttäjän veikkaus tallennetaan järjestelmään.

### **Veikkauksen muokkaaminen**

Käyttäjän suorittama tapahtuma, jossa tämä muokkaa jotain tekemäänsä veikkausta esim. tähtipelaajan loukkaantumisen vuoksi. Veikkaukseen liittyvän ottelun tulee kuitenkin olla tulevaisuudessa pelattava. Tapahtuman seurauksena, veikkauksen tiedot päivittyvät.

### **Veikkauksen poistaminen**

Käyttäjän suorittama tapahtuma, jossa tämä poistaa jonkin tekemänsä veikkausten esim. halusta välttää veikkausten onnistumisprosentin laskemista. Tapahtuman seurauksena, veikkauksen tiedot poistetaan tietokannasta.

### **Veikkausten selaaminen**

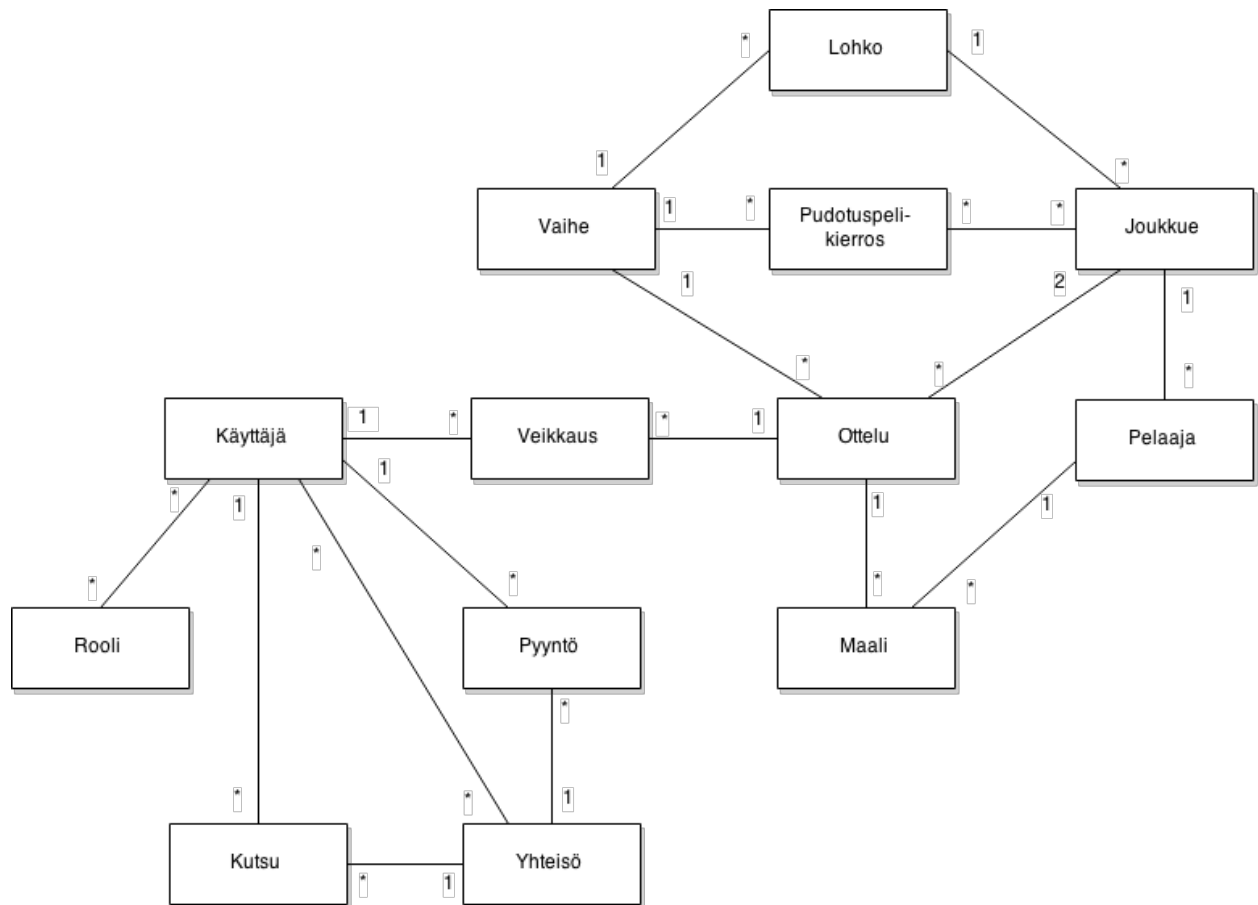
Käyttäjän suorittama tapahtuma, jossa tämä selailee omia veikkauksiaan tai sellaisen käyttäjän (esim. saman ryhmän jäsen) veikkauksia joihin tällä on lukuoikeus.

## 2.3 Käyttötapauskaavio



# 3 Tietosisältö

## 3.1 Käsitekaavio





## 3.2 Tietosisältökuvaukset

### Joukkue

Joukkue (team) on jotakin maata edustava jalkapallomaajoukkue. Joukkueeseen kuuluu monta pelaajaa ja joukkueella voi olla monia otteluita. Lisäksi, joukkue kuuluu yhteen turnauksessa pelattavaan lohkoon.

Attribuutti	Tietotyyppi	Kuvaus
Nimi	varchar(64)	Maan nimi, jota joukkue edustaa esim. "Suomi".
Lyhenne	varchar(3)	Kolmikirjaiminen lyhenne maan nimestä esim. "FIN".

---

### Käyttäjä

Käyttäjä (user) eli käyttäjätiliin liittyvät tiedot. Käyttäjällä voi olla monta veikkausta, jotka kohdistuvat eri otteluihin. Lisäksi, käyttäjä voi myös kuulua moneen yhteisöön, joissa käyttäjä voi olla jäsenenä tai ylläpitäjänä. Yleisellä sovellustasolla, käyttäjä voi olla tavallinen käyttäjä tai ylläpitäjä.

Attribuutti	Tietotyyppi	Kuvaus
Käyttäjänimi	varchar(64)	Käyttäjän määrittämä uniikki käyttäjänimi tai käyttäjätunnus, jonka perusteella käyttäjä voidaan tunnistaa esim. "zesbr".
Salasana	varchar(255)	Käyttäjän salasana, jolla suojataan käyttäjätilin koskemattomuus esim. "Ga2851%01g...".
Sähköposti	varchar(255)	Käyttäjän sähköpostiosoite esim. "admin@google.com". Myös sähköpostiosoite on uniikki, joten myös sen perusteella käyttäjä voidaan tunnistaa.
Etunimi	varchar(64)	Käyttäjän etunimi esim. "Simo".

Sukunimi	varchar(64)	Käyttäjän sukunimi esim. "Siili".
Liittymis-päivämäärä	datetime	Päivämäärä, jolloin käyttäjätili luotiin esim. 2015-01-01 10:00:00.
Viimeinen kirjautuminen	datetime	Aikaleima viimeisestä sisäänkirjautumisesta esim. 2015-01-01 10:12:14.

## Kutsu

Kutsu (invitation) on yhteisön ylläpitäjän lähettämä kutsu yhteisöön liittymisestä. Kutsulla on yksi vastaanottaja, joka on sellainen käyttäjä, joka ei ole jo jäsenenä yhteisössä. Lisäksi, kutsulla on lähettäjä, joka on yhteisön ylläpitäjä. Kutsu liittyy myös johonkin yhteisöön.

Attribuutti	Tietotyyppi	Kuvaus
Luettu	boolean	Kertoo onko kutsu luettu esim. false.
Hyväksytty	boolean	Kertoo onko kutsu hyväksytty esim. true.

## Lohko

Lohko (group) on turnaukseen kuuluva osaryhmä joukkueita. Lohko liittyy turnauksen alkulohkovaiheeseen, jossa ratkaistaan, mitkä joukkueet pääsevät turnauksen pudotuspelivaiheeseen.

Attribuutti	Tietotyyppi	Kuvaus
Nimi	varchar(64)	Lohkon nimi esim. "A-lohko".

## Maali

Jonkun pelaajan tekemä maali (goal), joka on syntynyt ottelussa. Näin ollen maaliin liittyy aina yksi ottelu ja yksi pelaaja.

Attribuutti	Tietotyyppi	Kuvaus
Minuutti	integer	Ottelun kellonaika minuutteina, jolloin maali syntyi esim. 72.
Oma maali	boolean	Kertoo oliko maali oma maali esim. false.
Rangaistuspotkusta	boolean	Kertoo oliko maali tehty rangaistuspotkusta esim. true.

## Ottelu

Kahden joukkueen välinen jalkapallo-ottelu (match). Ottelussa pelaa kaksi joukkuetta, joihin kuulu monta pelaajaa. Otteluun voi kohdistua myös monia veikkauksia. Lisäksi ottelu pelataan, jollain stadionilla. Ottelu on myös osa jotain turnuksen vaihetta.

Attribuutti	Tietotyyppi	Kuvaus
Kotijoukkue	varchar(64)	Ottelussa pelaavan kotijoukkueen nimi esim. Chile.
Vierasjoukkue	varchar(64)	Ottelussa pelaavan vierasjoukkueen nimi esim. Brasilia.
Kotijoukkueen maalit	integer	Kotijoukkueen maalimäärä ottelussa esim. 3. Arvo on tyhjä, niin kauan kunnes ottelu on pelattu.
Vierasjoukkue maalit	integer	Vierasjoukkueen maalimäärä ottelussa esim. 0. Arvo on tyhjä, niin kauan kunnes ottelu on pelattu.
Päivämäärä	datetime	Päivämäärä ja kellonaika jolloin ottelu alkaa esim. 2015-06-06 19:30:00. Aika esitetään GMT+0 muodossa.

---

## Pelaaja

Urheilija joka pelaa jalkapalloa. Pelaaja (player) voi olla mukana yhdessä joukkueessa. Oikeassa elämässä pelaaja voisi pelata myös seurajoukkueessa, joten tämä tieto ilmaistaan omana tietokenttänä.

Attribuutti	Tietotyyppi	Kuvaus
Etunimi	varchar(64)	Pelaajan etunimi esim. "Jari".
Sukunimi	varchar(64)	Pelaajan sukunimi esim. "Litmanen".
Syntymäaika	datetime	Pelaajan syntymäaika esim. 1980-01-01 00:00:00.
Kansallisuus	varchar(64)	Maan nimi, johon pelaajalla on kansallisuus esim. "Suomi". Jos pelaajalla on kaksoiskansallisuus, niin kenttä kuvaa sen kansallisuuden, jonka maajoukkuseen tämä voi kuulua.
Pelinumero	integer	Pelaajan pelipaidan numero, joka on uniikki pelaajan edustaman joukkueen sisällä esim. 10.
Pelipaikka	varchar(3)	Pelaajan pelipaikkaa kuvaava lyhenne esim. "GK" maalivahdille.
Seurajoukkue	varchar(64)	Pelaajan edustaman seurajoukkueen nimi esim. "HJK".

---

## Pudotuspelikierros

Pudotuspelikierros (knock-out round) on turnauksen pudotuspelivaiheeseen kuuluva kierros esim. välierät tai loppuottelu. Pudotuspelikierrokseen kuuluu monta joukkuetta ja joukkue voi kuulua moneen pudotuspelikierrokseen. Lisäksi, pudotuspelikierros kuuluu yhteen vaiheeseen ja vaiheella voi olla monta pudotuspelikierrosta.

Attribuutti	Tietotyyppi	Kuvaus
Nimi	varchar(64)	Pudostuspelikierroksen nimi esim. "Välierä".

## Pyyntö

Pyyntö (request) on käyttäjän tekemä pyyntökutsu yhteisöön liittymisestä. Pyyntön lähettäjänä on yksi käyttäjä ja vastaanottajana yhteisön. Pyyntön hyväksymisestä tai hylkäämisestä vastaa yhteisön ylläpitäjät.

Attribuutti	Tietotyyppi	Kuvaus
Päivämäärä ja kellonaika	datetime	Päivämäärä ja kellonaika, jolloin pyyntö tehtiin esim. 2015-01-01.
Luettu	boolean	Kertoo onko pyyntö luettu esim. false.
Hyväksytty	boolean	Kertoo onko pyyntö hyväksytty esim. true.

## Rooli

Käyttäjän käyttöoikeuksia kuvaava rooli (role). Käyttäjällä voi olla monta roolia esim. ylläpitäjän ja tavallisen käyttäjän oikeudet.

Attribuutti	Tietotyyppi	Kuvaus
Nimi	varchar(64)	Roolia kuvaava nimitys. esim. "Ylläpitäjä".
Ylläpitäjän oikeudet	boolean	Kertoo onko roolilla ylläpitäjän oikeudet esim. true.

---

## Stadion

Urheiluareena eli stadion, jolla pelataan jalkapallo-otteluita. Stadion liittyy aina yhteen otteluun, mutta stadionilla voidaan pelata useita otteluita.

Attribuutti	Tietotyyppi	Kuvaus
Nimi	varchar(64)	Stadionin nimi esim. "Old Trafford".
Kapunki	varchar(64)	Kapungin nimi, jossa stadion sijaitsee esim. "Manchester".

---

## Vaihe

Vaihe (stage) on turnauksen vaihetta kuvaava nimitys esim. alkulohkovaihe. Vaiheeseen voi kuulua monta lohkoa tai pudotuspelikierrosta, mutta lohko ja pudotuspelikierros voi kuulua yhteen vaiheeseen. Lisäksi, jokainen ottelu kuuluu yhteen vaiheeseen ja vaiheella voi olla monta ottelua.

Attribuutti	Tietotyyppi	Kuvaus
Nimi	varchar(64)	Vaihetta kuvaava nimitys esim. "Alkulohkovaihe".

---

## Veto

Veto (bet) on käyttäjän tekemä veikkaus, jonkin ottelun lopputuloksesta. Veto sisältää käyttäjän veikkaamaan maalimäärän ottelun koti- ja vierasjoukkueille. Vedolla on aina yksi käyttäjä ja se voi kohdistua vain yhteen otteluun.

Attribuutti	Tietotyyppi	Kuvaus
-------------	-------------	--------

Kotijoukkueen maalit	integer	Käyttäjän veikkaama maalimäärä ottelun kotijoukkueelle esim. 3.
Vierasjoukkueen maalit	integer	Käyttäjän veikkaama maalimäärä ottelun vierasjoukkueelle esim. 0.
Luomispäivämäärä	datetime	Päivämäärä jolloin veikkaus luotiin. esim. 2015-01-01 10:00:00.

---

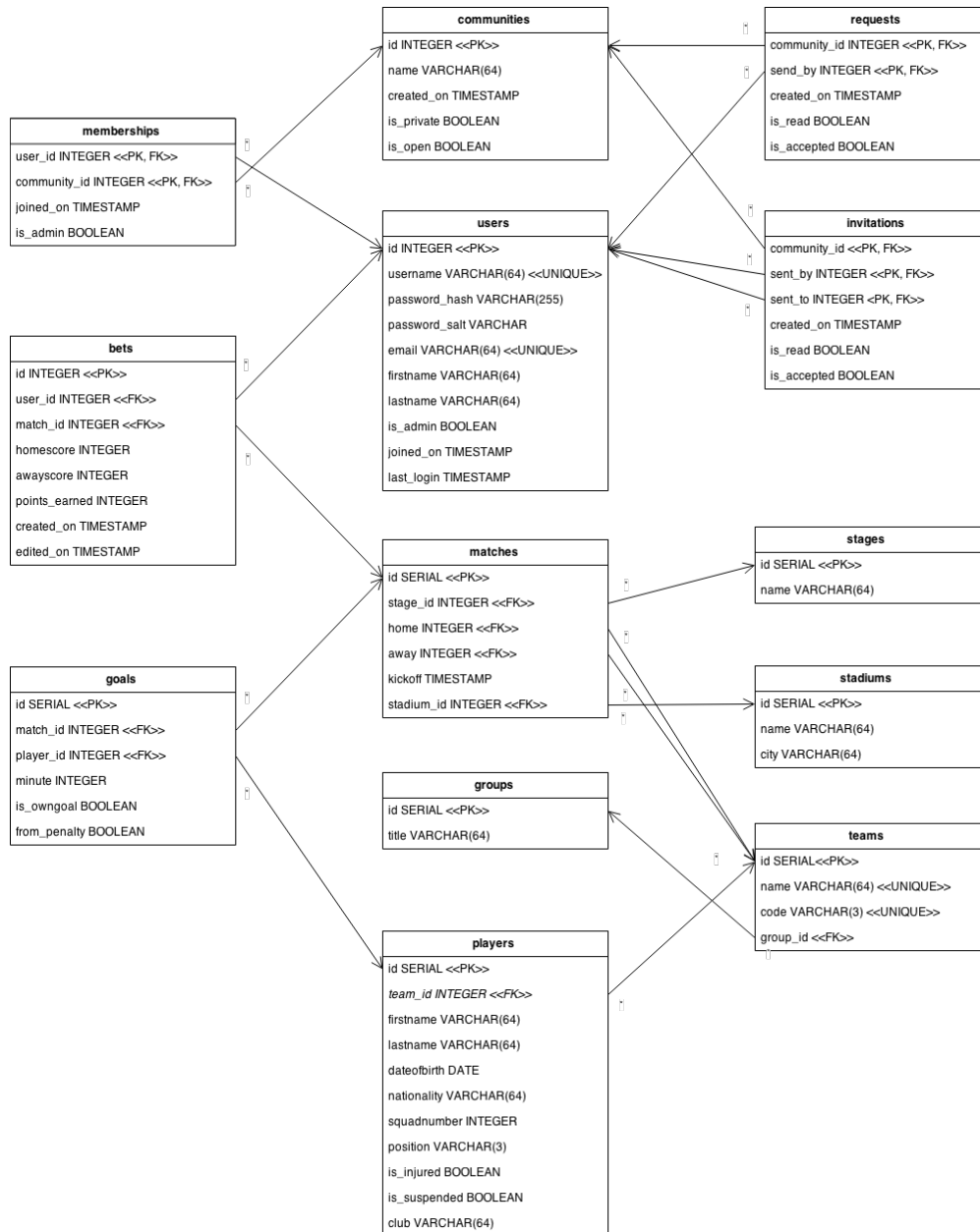
## Yhteistö

Yhteistö (community) on käyttäjistä koostuva ryhmä. Yhteistöllä on siten monta käyttäjää, jotka ovat joko yhteisön jäseniä tai ylläpitäjiä. Lisäksi, käyttäjä voi olla jäsenenä monessa yhteisössä.

Attribuutti	Tietotyyppi	Kuvaus
Nimi	varchar(64)	Yhteisön nimi esim. "Ryhmä räjä".
Luomispäivämäärä	datetime	Päivämäärä, jolloin ryhmä luotiin esim. 2015-01-01 10:00:00.

# 4 Relaatiotietokanta

## 4.1 Tietokantakaavio





## 5 Arkkitehtuuri

Sovellus on toteutetta malli-näkymä-ohjain eli ns. MVC-arkkitehtuuria noudattamalla.

### Reititys

Sovellus käyttää HTTP pyyntöjen reitityksessä Slim frameworkin reitittäjää. Reitittäjän tehtävä on ohjata käyttäjän tai sovelluksen tekemät HTTP pyynnöt oikeille ohjain luokille ja kutsua niiden jotain tiettyä metodia.

HTTP pyynnöt käsitellään RESTful periaatteen mukaisesti eli näkymiä pyydetään GET -pyynnöillä, uuden resurssin luominen tapahtuu POST -pyynnöllä, olemassa olevan resurssin päivittäminen tapahtuu PUT -pyynnöllä ja resurssin poisto tapahtuu DELETE -pyynnöllä. Todellisuudessa kaikki pyynnöt ovat GET ja POST tyyppisiä, sillä PHP ei tue lomakkeiden lähettämistä muiden tyyppien yli. Siksi PUT ja DELETE tapahtumien yhteydessä lähetetään piilotettu POST -muuttuja “\_METHOD”, jonka arvona on joko “PUT” tai “DELETE”, jonka reitittäjä tulkitsee siten “lavastetuksi” PUT tai DELETE pyynnöksi.

Käytössä olevat reitit ja pyyntöjä vastaavat metodikutsut on rekisteröity tiedostoon “routes.php”, joka löytyy “configs” -hakemistosta.

Lisätietoja Slim:stä löytyy osoitteessa: <http://www.slimframework.com/>

### Ohjain eli Controller

Sovelluksen ohjaimet löytyvät hakemistosta “app/controllers”. Ohjainten tehtävä on vastaanottaa ja käsitellä reitittimen kautta tulevat pyynnöt. Käytännössä ne ensiksi pyytävät dataa joltain tietyltä mallilta ja ohjaavat datan sitten jollekin tietylle näkymälle renderöitäväksi. Ohjaimessa suoritetaan myös käyttäjän oikeuksien tarkistus, sekä osin validoidaan pyydettyjen resurssien oikeellisuus. Näin ollen, ohjain voi myös suorittaa uudelleenohjauksia, jos alunperin pyydettyssä kohteessa ilmenee ongelmia.

Ohjaimet on nimetty vastaamaan sovelluksen tietokohteita sekä näkymiä. Suurimmasta osasta ohjaimia, löytyy staattiset metodit index, show, create, edit, save, update, delete.

Index -metodin tehtävä on ohjata sovellus kyseisen tietokohteen päänäkömään, joka on yleensä ns. listausnäkömä. Tällöin mallilta pyydetään kaikkia tietokohteen olioita taulukkona. Index -metodin vastaanottamat pyynnot ovat tyypiltään GET tyyppisiä.

Show -metodin tehtävä on ohjata sovellus yksittäisen tietokohteen näkömään. Show saa aina parametrina halutun kohteen tunnusteen (id), jonka perusteella tietokohdetta vastaava olio haetaan mallin kautta tietokannasta. Show:n vastaanottamat pyynnot ovat tyypiltään GET tyyppisiä.

Create -metodin tehtävä on ohjata sovellus resurssin lisäysnäkömään, joka yleensä sisältää html-lomakkeen. Create -metodin vastaanottamat pyynnot ovat GET tyyppisiä.

Edit -metodin tehtävä on ohjata sovellus olemassa olevan tietokohteen muokkausnäkömään, joka yleensä sisältää html-lomakkeen, jonka tiedot ovat esitetyt tietokohteen aikaisemmillä tiedoilla. Edit -metodin vastaanottamat pyynnot ovat GET tyyppisiä.

Save -metodi luo lomakkeen tai toisen ohjaimen kautta lähetetyistä POST muuttujista uuden jotain tietokohdetta vastaavan olio-intanssin, jonka malli tallentaa tietokantaan. Save -metodin vastaanottamat pyynnot ovat POST tyyppisiä.

Update -metodi päivittää tietyn tietokohteen oliota saatujen POST muuttujien arvoilla, jonka tietokohdetta vastaava malli päivittää tietokantaan. Update:n vastaanottamat pyynnot ovat "lavastettuja" PUT pyyntöjä.

Delete -metodi ohjaa tietokohteen mallille poistettavaksi. Delete:n vastaanottamat kutsut ovat "lavastettuja" DELETE pyyntöjä.

## **Malli eli Model**

Malli on jotain sovelluksen tietokohdetta mallintava luokka, joka vastaa tietokannassa olevaa taulua. Malli tarjoaa erilaisia palveluita tietokohteen hallitsemiselle. Yleisimmät palvelut ovat staattiset all ja find -metodit, julkiset save, update, delete sekä erilaiset laskentaan tai validointiin liittyvät metodit. Lisäksi, jos tietokohteella on viittauksia johonkin toiseen tietokohteeseen, on näiden hakemiseen luotu omat metodinsa. Esim. jos veikkauksella (Bet) on käyttäjä (User), niin luokka "Bet" tarjoaa julkisen metodin "user", joka palauttaa viittausta vastaavan käyttäjä-olion. Sovelluksessa mallit löytyvät hakemistosta "app/models".

## View eli Näkymä

Näkymät ovat tietokohteen tai tietokohteiden esittämiseen tarkoitettuja html sivuja. Näkymiä sovelluksessa pyörittää sivupohjamoottori Twig, joka tarjoaa elegantin ja joustavan tavan tietokohteiden esittämiseen. Twig tarjoaa käytännössä saman toiminnallisuuden kuin html:n sekaan upotettu PHP, mutta sen syntaksi on huomattavasti inhimillisempää, sekä se tarjoaa joustavan ja monipuolisen tavan jakaa uudelleenkäyttää näkymän osioita. Näkymät löytyvät hakemistosta “app/views”.

Lisää Twig:stä löytyy osoitteesta: <http://twig.sensiolabs.org/>

## Muut resussit ja tiedostot

Sovelluksen käyttämät muut resurssit, kuten kuvat, css-dokumentit ja javascript -tiedostot on sijoitettu hakemistoon “assets”. Sovelluksen konfiguraatioihin liittyvät tiedostot sijaitsevat hakemistossa “configs”. Sovelluksen käyttämien kirjastojen alustuksesta vastaavat tiedostot löytyvät hakemistosta “libs”. Sovelluksen tietokannan resetoivat ja alustavat sql -lauseet löytyvät hakemistosta “sql”. Kolmannen osapuolen javascript ja css tiedostot löytyvät hakemistosta “vendor”.

# 6 Ulkoasu ja käyttöliittymä

Sovelluksen käyttöliittymä koostuu näkymien html-sivuista sekä niiden tyyliä ohjaavista css-dokumenteista. Näkymien koostamisessa on hyödynnetty Twig:n tarjoamia ominaisuuksia kuten sivupohjien periyttämistä sekä makroja.

Näkymien ja ohjainten välisiä yhteyksiä kuvaava kaavio löytyy sovelluksen github repositorion hakemistosta docs.

## 6.1 Päävalikon näkymät

### Ottelut

Ottelut näkymä sisältää listan kaikista otteluista ja otteluun liittyvistä tiedoista, kuten maalimääristä, käyttäjän veikkauksista ym. Tämän lisäksi ottelut sisältää omat näkymät yksittäise ottelun esittämiseksi sekä ottelun muokkaamiselle, jossa ottelun tuloksia voidaan muokata.

## **Lohkot**

Lohkot sisältää turnauksen lohkojen sarjataulukot, jossa joukkueiden pelatutu ottelut, voitot, tasapelit, tappiot, tehdyt malit, päästetyt maalit ja pisteet näkyvät listattuna.

## **Veikkaukset**

TODO

## **Joukkueet**

TODO

## **Käyttäjät**

Käyttäjät sisältää listausnäkymän , jossa esitetään luettelo kaikista sovelluksen rekisteröityneistä käyttäjistä. Tämän lisäksi, yksittäisen käyttäjän tarkastelua varten on varattu oma näkymä , joka esittää myös käyttäjän yhteisöt ja veikkaukset käyttämällä community ja bet hakemistojen \_list.html komponenttia.

## **Yhteisöt**

Yhteisöt sisältää listausnäkymän, jossa esitetään luettelo kaikista sovelluksen yhteisöistä. Lisäksi, yhteisöt sisältää näkymän yksittäiselle yhteisölle varatun näkymän sekä näkymät yhteisöjen luontia sekä muokkausta varten.

## **Pistetilanne**

Pistetilanne sisältää listausnäkymän , jossa esitetään kaikkien käyttäjien välinen rankinglista. Näkymä käyttää user hakemiston “\_stats.html” komponenttia, jonka avulla renderöidään kaikki käyttäjät kattavat tulostaulukko.

# **7 Asennustiedot**

Teksti tulee tänne.

# **8 Käyttöohje**

Kirjautuminen sovellukseen tapahtuu osoitteessa:

<http://jesruuth.users.cs.helsinki.fi/tsoha/login>

Sovellukseen on mahdollista rekisteröityä osoitteessa:

<http://jesruuth.users.cs.helsinki.fi/tsoha/registration>

Sovellukseen on myös luotu testausta varten sekä admin että tavallisen käyttäjän oikeuksilla varustettut käyttäjätilit.

#### **Admin käyttäjä:**

käyttäjänimi: admin

salasana: asdf

#### **Tavallinen käyttäjä:**

käyttäjänimi: demo

salasana: asdf

## 9 Testaus, bugit ja puutteet

Osa sovelluksen toiminnallisuuksista on vielä toteuttamatta ja muutama näkymä tarvitsee hiomista ja silottelua. Suurimpana bugina tällä hetkellä lienee piste- ja sarjataulukon lajittelu, sekä mobiilioptimointi. Myös koodin refaktorointi lopullista palautusta varten on kesken.

## 10 Jatkokehitysideat

Sovellukseen ei ehditty toteuttaa kaavoiltuja yhteisökutsu ja -pyyntö toiminnallisuuksia, joten ne siirtyvät mahdolliseen jatkokehitykseen. Tämän lisäksi, sovelluksen mobiilioptimointia voisi parantaa ja sen toimivuutta voisi testata tarkemmin. Testaus ylipäänsä olisi tärkeä jatkokehityksen kohde eli testien tekeminen esim. PHPUnit:lla. Myös käyttöliittimää voisi parantaa monilta osin ja sovelluksen ylläpidolliset toiminnot voisi siirtää erilliseen ylläpidonäkymään tai -sovellukseen. Sovellus toimii myös valitettavan hitaasti tietyissä näkymissä, johtuen epäoptimoidusta tietokantakyselyrakenteesta, joten sen optimoiminen yhdessä välimuistin hyödyntämisen

kanssa olisi järkevä jatkokehityksen kohde. Myös sovelluksen ulkonäköä ja käytettävyyttä parantavat selainpuolen toiminnallisuudet ja tehosten voisivat tuoda sovellukseen käyttökokemukseen lisää arvoa.

Koska kisat loppuvat varsin nopeasti vaihtaa Eventsian sivuilla olevan numeron mahdollisimman pian? Kuittaatko, kun ok. Uusi numero on 020 - 7439 544 easti tulee sovelluksesta nopeasti hyödytön. Kuitenkin koska jalkapallon pelaaminen ei maapallolta lopu ja uusia kilpailuja alkaa tasaiseen tahtiin, voisi sovelluksen suunnitella kattamaan myös muut jalkapallo turnaukset. kuten mm- ja em-kilpailut, kansalliset liigat sekä muut seurajoukkuekilpailut. Tämä vaatisi, jonkin verran räätälöintiä tietokannassa, koska pelaaja voi liittyä saman kauden aikana kahteen eri seurajoukkueeseen ja maajoukkueetkin muuttuvat turnauksittain.

## 11 Omat kokemukset

Kurssi on ollut mielestäni erittäin mielenkiintoinen ja oman sovelluksen toteuttaminen on ollut, vaikkakin työlästä, niin todella antoisaa puuhaa.

Kursilla esiteltyt kirjastot Slim ja Twig ovat olleet itselle mieluisa uusi tuttavuus. Slimin simppelellä toteutettu reititys mekanismi sekä Twigin elegantti syntaksi ovat helpottaneet monen asian tekemistä ja auttamaan clean code -periaatteiden toteuttamisessa.

Vaikkakin Bootstrap on erittäin tehokas ja kehitystyötä nopeuttava kehys, niin itse olen hieman kyllästynyt sen tarjoamiin komponentteihin ja ulkoasu valintoihin. Siksi päätin kokeilla toteuttaa oman CSS-kehityksen sovellukseni tarpeisiin. Kehityksen ulkoasu valintoihin vaikuttivat mm. Googlen material design periaatteen specsit.

Ulkoasusuunnittelu ja selainpuolen toiminnallisuudet eivät kuitenkaan varsinaisesti olleet kurssin agenda, joten niiden toteuttamiseen olen käyttänyt kenties liikaa aikaa.

## 12 Muu dokumentaatio

Teksti tulee tänne.