

Aineopintojen harjoitustyö: Tietokantasovellus (Periodi II)

Tekijä: Jaakko Virtanen

1. Johdanto	4
1.1 Järjestelmän tarkoitus	4
1.2 Järjestelmän tavoitteet	4
1.3 Toteutusympäristö	4
2. Käyttötapaukset	5
2.1 Käyttäjäryhmät	5
2.2 Käyttötapauskuvaukset	5
2.3 Käyttötapauskaavio	5
Kuva käyttötapauskaavio	5
2.4 Käyttötapauskuvaukset	6
3. Järjestelmän tietosisältö ja käsitekaavio	7
Kuva käsitekaavio	7
4. Relaatiotietokantakaavio	10
Kuva relaatiotietokantakaavio	10
5. Järjestelmän yleisrakenne	11
5.1 Tietokantataulut vs. PHP-oliot	12
5.1.1 Tietokohteen ja tietokantataulun suhde	12
5.1.2 Tietokannan suunnittelu vs. käyttäjän näkymät	12
Kurssin kannalta oleellinen hakemistopuu kokonaisuudessaan:	14
6. Käyttöliittymä ja järjestelmän komponentit	15
6.1 Yleistä ja käyttäjäprofiilin hallinta	16
6.1.1 Yleistä	16
6.1.1 Käyttäjäprofiilin hallinta	16
6.2 Pelaajien listaus	18
6.3 Pelaajien tarkemmat tiedot ja Käyttäjän seuraamat pelaajat	18
6.4 Ottelulistaus ja ottelutiedot	19
6.5 Kaikki kentällä	21
7. Asennustiedot	22
8. Käynnistys- ja käyttöohje	22
9. Testaus, tunnetut bugit ja puutteet, sekä jatkokehitysideat	22
9.1 Testaus	22
9.2 Tunnetut bugit	22
9.3 Puutteet	23
9.4 Jatkokehitysideat	23
10. Omat kokemukset	23
11. Muu dokumentaatio	23

1. Johdanto

1.1 Järjestelmän tarkoitus

Harjoitustyössä toteutan web-palvelun, josta käyttäjät voivat tarkastella jääkiekko-otteluiden ja sarjojen tietoja. Toteutuksen laajuus rajoittuu ensimmäisessä vaiheessa kotoiseen SM-liigaan (liiga.fi). Tarkoituksena on tallentaa tilastoja automaattisesti tietokantaan, josta niitä on helppo kysellä ja esittää käyttäjän erinäisiin tarpeisiin. Kurssin aikana tavoitteena on ehtiä tehdä toteutus laukauskarttojen piirtämisestä käyttäjän antamin kriteerein. Alussa implementoitavat kriteerit "piirää *pelaajan x* laukaukset *aikajaksolta y*" [kurssin puitteissa aikajakso tarkoittaa kaikkia kannassa olevia laukauksia]. Järjestelmä tarjoaa Käyttäjälle myös mahdollisuuden asettaa pelaajia seurantaan, näitä tietoja käytetään myös yksilöityjen näkymien muodostamiseen. Käyttäjä saa esille kaikki seuraamiensa pelaajat listana josta on helppo klikata pelaajan tiedot lähempään tarkasteluun.

1.2 Järjestelmän tavoitteet

Kurssin puitteissa tavoitteena on luoda sulava web-käyttöliittymä käyttäjän tarpeisiin. Tavoitteena on toteuttaa tarpeeksi laaja kokonaisuus, jotta arvosana voi kaiken onnistuessa olla täysi viisi. Tämä tarkoittaa siis:

- Kirjautumisen toteuttaminen
- Kolme tietokohdetta tai enemmän
- Kaksi täyttä CRUDia tai enemmän
- Vähintään yksi monesta moneen suhde tietokantataulujen välillä

1.3 Toteutusympäristö

Tietokantasovellus toteutetaan suurimmilta osin suositelluissa ympäristöissä users-palvelimella ja PHP / MySQL-yhdistelmällä. Pythonilla (urllib, BeautifulSoup, pymysql) olen tehnyt tietokantaan haettavien tietojen scraping-, sekä tallennustoiminnallisuudet.

2. Käyttötapaukset

2.1 Käyttäjärühmät

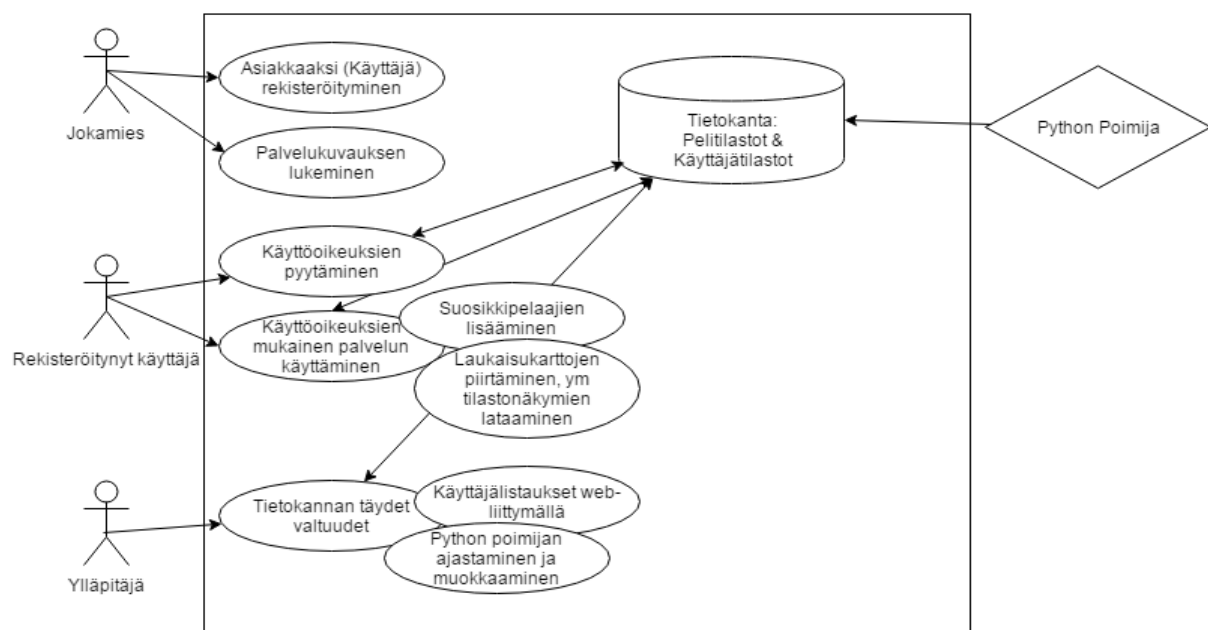
Jokamies	Satunnainen internetin käyttäjä
Käyttäjä	Rekisteröitynyt käyttäjä (aliryhmät esitetty alla)
Ylläpitäjä	Järjestelmän ylläpitäjä

2.2 Käyttötapauskuvaukset

Jokamies voi käydä katsomassa palvelun kuvauksen ja rekisteröityä palveluun. Rekisteröitynyt käyttäjä (Asiakas) voi ostaa palvelusta tarvitsemiaan tietoja, sekä käyttöoikeuksia. Käyttöoikeudet jaetaan luokkiin seuraavasti [käyttöoikeuksien eroavaisuuksia palvelussa ei toteutettu kurssin puitteissa, mutta jätän ne kuitenkin esille mahdollisen jatkojalostuksen vuoksi]:

Asiakas ryhmät	Perus	Hopea	Kulta
Seurattujen pelaajien lkm	5	Rajoittamaton	Rajoittamaton
Viivästetyt tilastot (1 päivä)	X	X	
Reaaliaikainen			X

2.3 Käyttötapauskaavio



Kuva käyttötapauskaavio

2.4 Käyttötapauskuvaukset

Jokamies

Sivustolle päätyvä surffaaja pääsee rekisteröitymään, näkemään palvelun kuvauksen, sekä mahdollisesti hinnaston ja eri asiakastasot [lopullisessa palvelussa asiakastasot toteutettu tietokantaan "*Kayttaja.kayttajaryhma*", mutta ainoastaan 0-, eli Admin-luokan toiminnallisuudet poikkeavat muista web-palvelussa. Rekisteröityminen tuo aina asiakasluokan 1. Tästä olisi kuitenkin helppo laajentaa toiminnallisuuksien poikkeavuuksia tulevaisuudessa, esimerkiksi rajoittaa pelaajien lukumäärää yhtäaikaisessa seurannassa]

Rekisteröitynyt käyttäjä

Kirjautumisen jälkeen käyttäjä pääsee selaamaan pelaaja- ja ottelulistauksia, sekä katsomaan omaa personoitua näkymäänsä, joka pitää sisällään hänen seurattaviksi määrittelemiensä pelaajien listan. Tämän lisäksi hän pääsee tarkastelemaan kaikkia järjestelmässä olevia laukauksia kentälle renderöitynä. Ylläpitäjän ylivertaisuudesta muistuttamiseksi näkyville on jätetty "Käyttäjälistaus", jota kuitenkin ainoastaan Admin-oikeuksien haltija pääsee tarkastelemaan lähemmin. Rekisteröityneitä käyttäjiä on kolme tasoa, jotka voi tulevaisuudessa määrittää eroamaan toisistaan esimerkiksi seurattaviksi asetettavien pelaajien lukumäärärajoituksina, sekä tiedon reaaliaikaisuudessa.

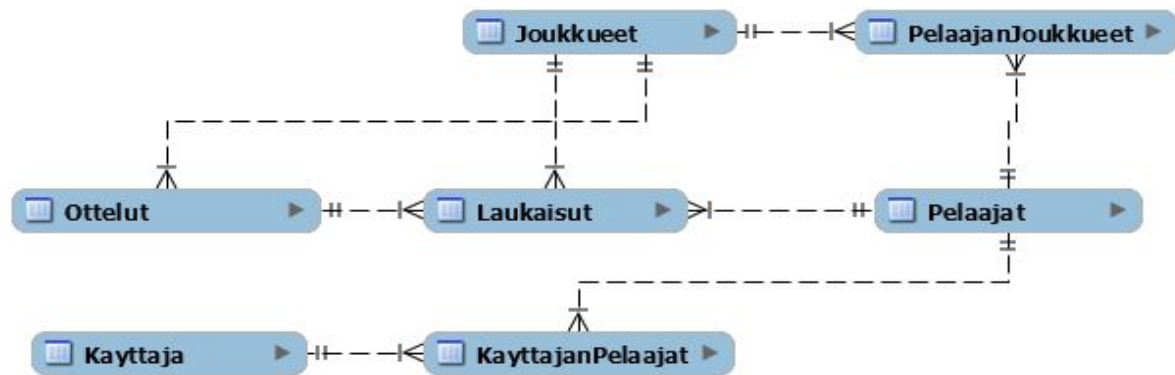
Ylläpitäjä

Ylläpitäjä hallinoi tietokannan eheyttä ja ohjaa Python-poimijaa työssään. Lähinnä muutokset lähdesivustojen koodissa ovat mahdollisia sudenkuoppia. Python-poimijan ajastaminen kuuluu myös ylläpitäjän vastuulle [Python-poimija "otetaan annettuna" kurssin puitteissa]. Ylläpitäjä pääsee muokkaamaan käyttäjäryhmiä, sekä niiden sisältämiä toiminnallisuuksia [ei toteutettu web-palveluun kurssin puitteissa]. Kurssin puitteissa eri käyttäjäryhmien poikkeavuuksia demotaan antamalla ylläpito-käyttäjärhmälle oikeus listata käyttäjätiedot web-palvelun kautta.

Python poimija

Pythonilla kirjoitettu ja serverillä ajettava web-scraper, joka hakee annetuin parametrein tietoja tietokantaan. Otteluohjelman mukaisilla ottelutunnuksilla, sekä päivämäärä / kellonaika tiedoilla tapahtuvaa poimintaa ottelukohtaisilta sivuilta oikeaan aikaan. Kurssin puitteissa Python poimija toimittaa testidatan kauden 2016-2017 alkukauden otteluista (217 kpl), sekä niihin liittyvistä tiedoista tietokantamuodossa.

3. Järjestelmän tietosisältö ja käsitekaavio



Kuva käsitekaavio

Tietokohde: Joukkueet

Attribuutti	Arvojoukko	Kuvaus
idJoukkueet	tinyint(3) UN AI PK	
nimi	varchar(15)	

Tietokohde: Kayttaja

Attribuutti	Arvojoukko	Kuvaus
idKayttaja	int(10) UN AI PK	
kayttajatunnus	varchar(45)	
etunimi	varchar(45)	
sukunimi	varchar(45)	
email	varchar(45)	
kayttajaryhma	tinyint(1)	Käyttöoikeuksien hallinta, kurssin puitteissa ainoastaan 0 (Admin) eroaa web-toiminnallisuuksien suhteen muista

Tietokohde: KayttajanPelaajat

Attribuutti	Arvojoukko	Kuvaus
idKayttajanPelaajat	int(10) UN AI PK	
kayttajaid	int(10) UN	
pelaajaTunnus	int(10) UN	
alkupvm	date	
loppupvm	date	

Tietokohde: Laukaisut

Attribuutti	Arvojoukko	Kuvaus
idLaukaisut	int(10) UN AI PK	
pelaajaTunnus	int(10) UN	
top	double	
lefty	double	
otteluTunnus	int(10) UN	
joukkue	tinyint(4)	
aika	time	
tulos	varchar(45)	
event	varchar(25)	

Tietokohde: Ottelut

Attribuutti	Arvojoukko	Kuvaus
idOttelut	int(10) UN AI PK	
otteluTunnus	int(5) UN	
kotijoukkueid	tinyint(3) UN	
vierasjoukkueid	tinyint(3) UN	
pvm	date	
yleiso	int(5)	

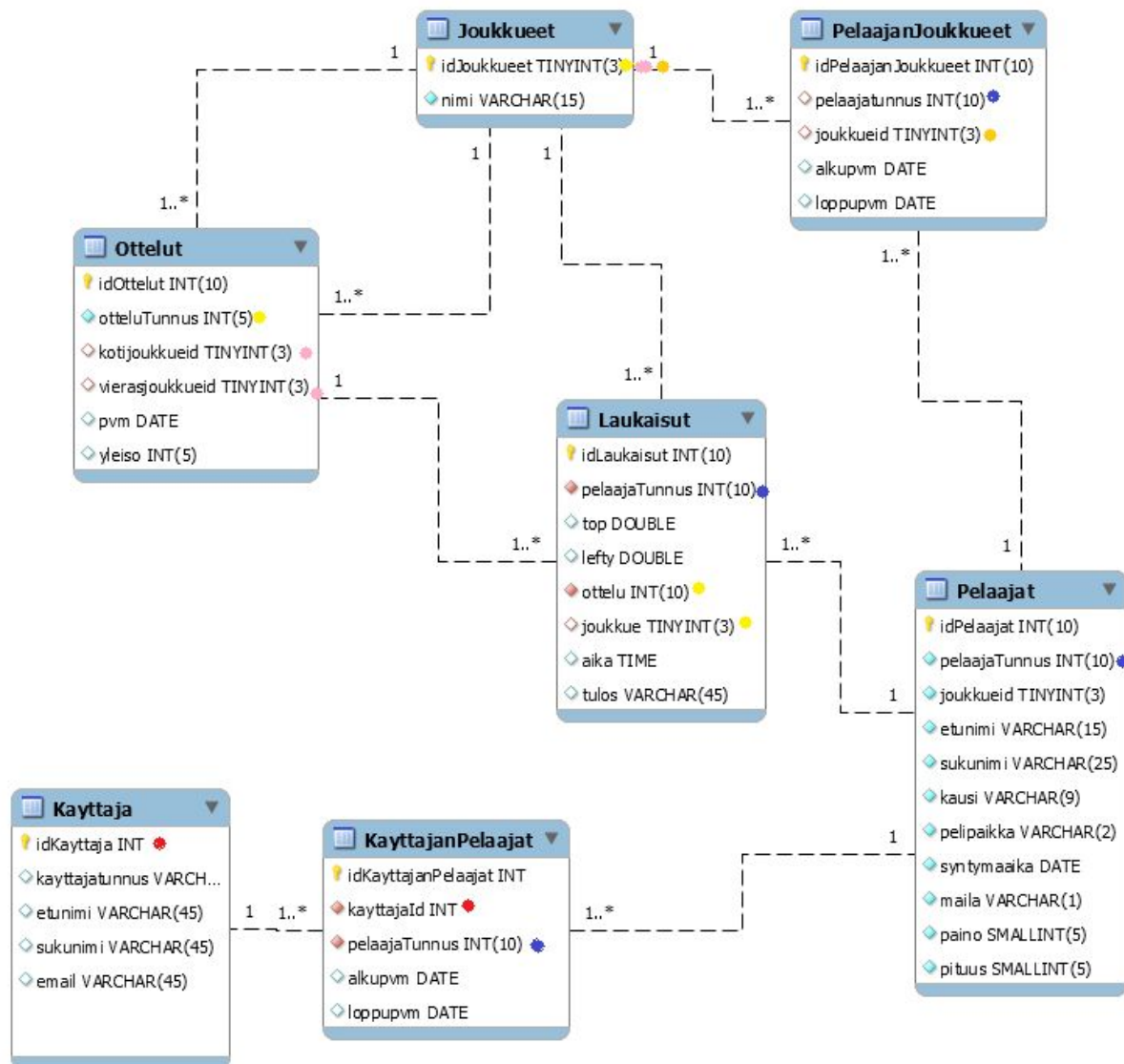
Tietokohde: PelaajanJoukkueet

Attribuutti	Arvojoukko	Kuvaus
idPelaajanJoukkueet	int(10) UN AI PK	
pelaajatunnus	int(10) UN	
joukkueid	tinyint(3) UN	
alkupvm	date	
loppupvm	date	

Tietokohde: Pelaaja

Attribuutti	Arvojoukko	Kuvaus
idPelaajat	int(10) UN AI PK	
pelaajaTunnus	int(10) UN	
joukkueid	tinyint(3) UN	
etunimi	varchar(15)	
sukunimi	varchar(25)	
kausi	varchar(9)	
pelipaikka	varchar(2)	
syntymaika	date	
maila	varchar(1)	
paino	smallint(5) UN	
pituus	smallint(5) UN	

4. Relaatiotietokantakaavio



Kuva relaatiotietokantakaavio

Yllä olevassa kaaviossa on merkattu viiteavaimet väripilkuilla. Käytän jokaisessa taulussa automaattisesti generoituja pääavaimia, mutta myös monet taulujen tunnuksista ovat täysin uniikkeja (vastaavat tietolähteessä - liiga.fi olevia tunnuksia). Näitä tunnuksia (esim. *Pelaajat:pelaajaTunnus*) käytän pääsääntöisesti pääavaimina liitoksien tekemiseen. *Huom. Kayttaja-tauluun on lisätty yksi attribuutti "kayttajaryhma TINYINT(1)", sekä Laukaisut-tauluun lisätty attribuutti "event VARCHAR(25)" kuvan luomisen jälkeen.*

5. Järjestelmän yleisrakenne

Tietokantasovellus noudattaa kurssipohjan hyvin jäsentelemää MVC-mallia, jossa kontrollerit, näkymät ja mallit ovat eroteltuina toisistaan omiin kansioihinsa (controllers, views ja models). Valmiita apukirjastoja hyödynnetään kansioista lib.

Järjestelmä hyödyntää PHP:n superglobaaleista muuttujista `$_SESSION`ia, joka tallentaa käyttäjän kirjautuessa käyttäjäid:n säilöön. Tätä säilöttyä ID:tä hyödynnetään käyttöoikeuksien tarkastelussa (suurin osa sivuista rajattu vain kirjautuneille käyttäjille, osalle sivuista pääsee ainoastaan tietyt käyttäjäryhmät). Myös käyttäjäkohtaisten tietojen muuttaminen onnistuu vain kirjautuneelta käyttäjältä, joka voi muokata omia tietojaan. Palvelu tarjoaa uusille käyttäjille rekisteröitymismahdollisuuden, jonka jälkeen palvelu on käytettävissä kirjautuneen tavoin.

Järjestelmässä on toteutettu tietokannan tauluista viisi tietokohdetta/oliota:

- — Pelaaja.php
- — Ottelu.php
- — Laukaus.php
- — Kayttaja.php # **CRUD**
- — KayttajanPelaaja.php # **CRUD**

Järjestelmä tarjoaa usean näkymän:

- — Pelaajat
 - — pelaajatiedot.html
 - — pelaajalistaus.html
- — Ottelut
 - — ottelutiedot.html
 - — ottelulistaus.html
- — Kentta
 - — kentta.html
- — Kayttaja
 - — rekisteroidy.html
 - — kirjaudu.html
 - — kayttajatiedot.html
 - — kayttajanSeuraamatPelaajat.html
 - — kayttajalistaus.html
- — home.html
- — base.html

Sekä muutaman makron:

- — Makrot
 - — kentta.html
 - — forms.html

Näitä komentaa kontrollerit:

- — YleinenKontrolleri.php
- — PelaajaKontrolleri.php
- — OtteluKontrolleri.php
- — LaukausKontrolleri.php
- — KayttajanPelaajienKontrolleri.php
- — KayttajaKontrolleri.php

5.1 Tietokantataulut vs. PHP-oliot

5.1.1 Tietokohteen ja tietokantataulun suhde


Järjestelmää toteutettaessa pyrin luomaan tietokohteet täsmäämään yksi-yhteen tietokannan taulujen attribuuttien kanssa. Tästä tavoitteesta tuli mielenkiintoinen haaste, jossa jouduin pakolla tekemään kompromisseja.

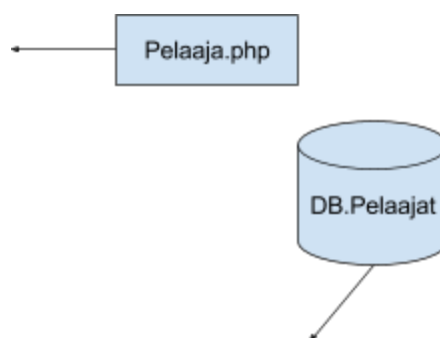
5.1.2 Tietokannan suunnittelu vs. käyttäjän näkymät

Koska tietokannan on tarkoitus tulevaisuudessa pitää sisällään suurta tietomäärää, on suunnittelussa pyritty tehokkaaseen tiedon säilyttämiseen. Tämä tarkoittaa esimerkiksi useassa taulussa tietorakenteiden oleellisuuden kannalta esiintyvän joukkue-tunnuksen säilyttämistä pienenä kokonaislukuna merkkijonon sijasta. Web-palvelussa käyttäjät kuitenkin haluavat nähdä mieluummin joukkueiden selkokiekiset nimet kuin generoidun ID-numeron. Tämä johtaa tilanteeseen, jossa ideaalista(?) PHP-olion oliomuuttujien ja tietokohdetta vastaavan tietokantataulun attribuuttien täydellisestä täsmäämisestä joudutaan poikkeamaan tai vaihtoehtoisesti web-palvelun käyttäjälle joudutaan keksimään toinen keino saada nämä PHP-olion ulkopuolelta tulevat tiedot renderöitävään näkymään.

Esimerkiksi käyttäjälle halutaan näyttää pelaajan perustiedot näkymässä. Nämä tiedot tulevat luonnollisesti tietokohteesta Pelaaja.php, vaikka tietokantataulu ei sisällä Joukkueen nimeä, vaan ainoastaan viiteavaimen Joukkue-tauluun, josta kysellään PHP-oliota luodessa joukkueen nimi oliomuuttujaan \$joukkueNimi. Palvelun rakenteen vuoksi myös laukaustenLkm haetaan Pelaaja-olion perustietoihin luomisvaiheessa ylimääräisten tietokantakyselyiden minimoimiseksi.

Mikko Lehtonen

Joukkue:	KooKoo
Pelipaikka:	VP
Syntymäaika:	1994-01-16
Kuva:	



	idPelaajat	pelaajaTunnus	joukkueid	etunimi	sukunimi	kausi	pelipaikka	syntymaika	maila	paino	pituus
▶	675	22358280	16	Mikko	Lehtonen	2016-2017	VP	1994-01-16	L	89	182
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
class Pelaaja extends BaseModel {  
    public $idPelaajat, $pelaajaTunnus, $joukkueid, $etunimi, $sukunimi, $kausi, $pelipaikka, $dob, $maila, $paino, $pituus, $joukkueNimi, $laukaustenLkm;
```

Tämä mieltäni vaivaava epäloogisuus tietokohteiden ja tietokantataulujen suhteen sai minut pohtimaan eri vaihtoehtoja haasteen ratkaisemiseksi. Listaani niistä muutaman:

Luodaan siis tietokohde ja tietokantataulu täysin yhteneväisiksi attribuuttien ja oliomuuttujien kesken.

→ **Miten ratkaistaan käyttäjien näkymien luonti?**

1. Tehdään PHP-olioihin metodit ja kysellään tietokannan muista tauluista tarpeen mukaan lisää tietoja
 - a. Haittapuolena suuri hitaus, testasin tätä ottelulistauksen esittämisessä suurella määrällä tietoja laukaisutaulusta (8 tietokantakyselyä per rivi → 217 riviä kesti 34 sekuntia ladata)
2. Muutetaan tietokantataulua pitämään sisällään kaikki olion tarvitsemat attribuutit
 - a. Haittapuolena tietokannan paisuminen ja turha toisteisuus
3. Välitetään näkymälle useampi olio, esimerkiksi Pelaaja ja Joukkue, joka on alustettu joukkueId:llä pelaajan oliomuuttujan tiedoilla
 - a. Tämäkin on hidasta käyttäjän kannalta, koska suoritetaan useita tietokantakyselyitä
 - b. Onko järkevää luoda kahden attribuutin tietokohdetta, mielestäni tässä tapauksessa ei (en tehnyt joukkueesta omaa tietokohdetta)
4. Luodaan tietokantanäkymä ja näkymää vastaava tietokohde
 - a. Tämä heräsi vasta myöhemmässä vaiheessa mieleeni, joten en ehtinyt testaamaan

Edellä mainituista mikään ei tuntunut kovin toimivalta ratkaisulta (paitsi ehkä 4. kohta), joten siirryin pohtimaan kysymystä: *Onko tietokohde luotu replikoimaan tietokannan taulua (1=1) vai MVC:n näkymää varten (insinööriratkaisu, tehdään jotain, mikä toimii)?* Päädyin päätelmään, että tietokohde on sopivampaa nähdä näkymää varten olevana mallina. Tämä tietysti tekee järkeä myös äkkiseltään ajateltuna, MVC malli pohjimmaltaan on täysin erillään tietokannasta. Edellä listatuista tapauksista kohta 4 vaikuttaisi lupaavalta tämän dilemman kokonaisvaltaiseen ratkaisemiseen, myös verbaalisella tasolla → joka suunnassa on *näkymä*.

Malleissani on siis luomisvaiheessa käytetty liitoskyselyitä saadakseni tarvittavan laajat tiedot alustettua oliomuuttujiin tietokantakyselyiden minimoiseksi web-palvelun käytössä.

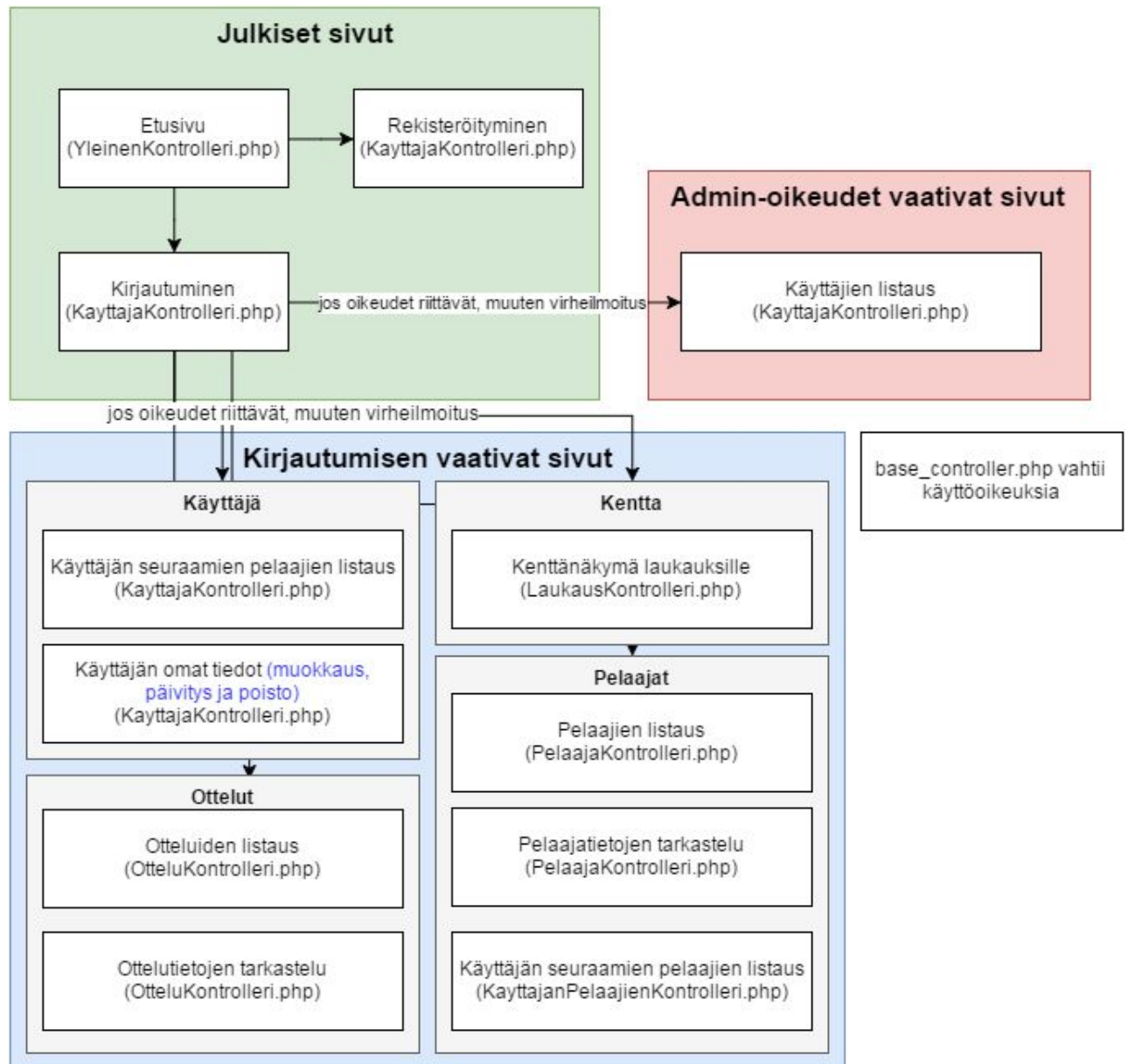
- Pelaaja.php
 - \$joukkueNimi, \$laukaustenLkm (2 liitettyä taulua)
- KayttajanPelaaja.php
 - \$joukkueNimi, kaikki Pelaaja-tilin tiedot (2 liitettyä taulua)
- Laukaus.php
 - Tämä on hieman erilainen, laukausten kentälle renderöintiä varten pitää päätellä koti- ja vierasjoukkueet, sitä varten alustan oliomuuttujan \$isHome tiedolla, jota ei suoraan löydy mistään taulusta, vaan tulee SQL-kyselyn loogisena tulosteena (1/0) *laukausut.joukkue = ottelut.kotijoukkueid as isHome*
- Ottelut.php

- Koti- ja vierasjoukkueen nimet haen tekemällä kaksi liitosta samaan tauluun (Joukkue)

Kurssin kannalta oleellinen hakemistopuu kokonaisuudessaan:

```
├── sql
│   ├── drop_tables.sql
│   ├── create_tables.sql
│   └── add_test_data.sql
├── README.md
├── lib
│   ├── view.php
│   ├── redirect.php
│   ├── database.php
│   ├── base_model.php
│   └── base_controller.php
├── index.php
├── drop_tables.sh
├── doc
│   └── Tietokantasovelluksendokumentaatio.pdf
├── deploy.sh
├── create_tables.sh
├── config
│   ├── routes.php
│   ├── environment.sh
│   └── database.php
├── assets
│   ├── pics
│   ├── vieras.png
│   ├── vierasmaali.png
│   ├── nedry.gif
│   ├── koti.png
│   ├── kotimaali.png
│   └── jaakiekkokaukalo.png
├── app
│   ├── views
│   │   ├── Pelaajat
│   │   │   ├── pelaajatiedot.html
│   │   │   └── pelaajalistaus.html
│   │   ├── Ottelut
│   │   │   ├── ottelutiedot.html
│   │   │   └── ottelulistaus.html
│   │   ├── Makrot
│   │   │   ├── kentta.html
│   │   │   └── forms.html
│   │   ├── Kentta
│   │   │   └── kentta.html
│   │   ├── Kayttaja
│   │   │   ├── rekisteroidy.html
│   │   │   ├── kirjaudu.html
│   │   │   ├── kayttajatiedot.html
│   │   │   ├── kayttajanSeuraamatPelaajat.html
│   │   │   └── kayttajaListaus.html
│   │   ├── home.html
│   │   └── base.html
│   ├── models
│   │   ├── Pelaaja.php
│   │   ├── Ottelu.php
│   │   ├── Laukaus.php
│   │   ├── Kayttaja.php
│   │   └── KayttajanPelaaja.php
│   └── controllers
│       ├── YleinenKontrolleri.php
│       ├── PelaajaKontrolleri.php
│       ├── OtteluKontrolleri.php
│       ├── LaukausKontrolleri.php
│       ├── KayttajanPelaajienKontrolleri.php
│       └── KayttajaKontrolleri.php
└── add_test_data.sh
```

6. Käyttöliittymä ja järjestelmän komponentit



6.1 Yleistä ja käyttäjäprofiilin hallinta

6.1.1 Yleistä

Käyttöliittymä pohjautuu ensin tietojen listaamiseen käyttäjälle, joita klikkaamalla pääsee yksityiskohtaisempiin tietoihin. Tämä takaa nopeahkon vastausajan, kun listausnäykymät perustuvat yksinkertaisempiin tietokantakyselyihin (verrattuna monipuolisempien tietojen näyttämiseen listausnäykymässä) → hyvä käyttäjäkokemus. Rekisteröityminen (CRUD “Create”) ja kirjautuminen ovat suoraviivaisia toimenpiteitä, jonka jälkeen käyttäjä pääsee nauttimaan palvelun sisällöstä.

The image shows two side-by-side screenshots of a web application's registration and login pages. The left page is the registration form, titled 'Tervetuloa rekisteröitymään palveluumme:'. It has a navigation bar with 'LiigaVinkit', 'Etusivu', 'Rekisteröidy' (highlighted), and 'Kirjaudu sisään'. The form includes fields for 'Käyttäjätunnus', 'Etunimi', 'Sukunimi', 'Sähköposti', and 'Salasana', followed by a 'Rekisteröidy' button. The right page is the login page, titled 'Tervetuloa! Katsellaksesi sisältöä, kirjaudu sisään.'. It has a navigation bar with 'LiigaVinkit', 'Etusivu', 'Rekisteröidy', and 'Kirjaudu sisään' (highlighted). The form includes fields for 'Käyttäjätunnus' and 'Salasana', followed by a 'Kirjaudu sisään' button.

6.1.1 Käyttäjäprofiilin hallinta

Käyttäjä pääsee tarkastelemaan ja muokkaamaan oman profiilinsa tietoja kirjautumisensa jälkeen:

The image shows a screenshot of the user profile management page. The navigation bar includes 'LiigaVinkit', 'Etusivu', 'Pelaajalistaus', 'Ottelulistaus', 'Käyttäjän seuraamat pelaajat', 'Kaikki kentät', 'Käyttäjalistaus', 'Omat tiedot' (highlighted), and 'Kirjaudu ulos'. The main content area is titled 'Omien tietojen muokkaus:'. Below the title, it says 'Käyttäjätunnuksesi on: Jarza ja sitä ei voi muuttaa'. The form includes fields for 'Etunimi' (filled with 'Jarzi'), 'Sukunimi' (filled with 'Littanen'), 'Sähköposti' (filled with 'jarzanaama@littanen.com'), and 'Salasana' (filled with '***'). At the bottom, there are two buttons: 'Päivitä tiedot' and 'Poista käyttäjätietosi'.

Palvelu käyttää PHP:n tarjoamaa superglobaaliin muuttujaan `$_SESSION` tallennettua käyttäjälä: käyttäjän muutosten automaattisena parametrina, jotta vain omia tietojään voi muokata. Omat tiedot näkymä tarjoaa mahdollisuuden muokkaamiseen (CRUD “UPDATE”) ja poistoon (CRUD “DELETE”). Käyttäjän syötteiden tarkastamiseksi on toteutettu vain merkkijonon pituus > 3 kaikissa kohdissa, sekä käyttäjätunnuksen vapaus rekisteröintivaiheessa.

Kurssin vaatima neljäs CRUD-ominaisuus “READ” on toteutettu Admin-käyttäjärühmän käyttäjille, jotka pystyvät lataamaan “Käyttäjälistaus”-näkymän:

Kaikki käyttäjät

KäyttäjälD	Etunimi	Sukunimi	Käyttäjätunnus	Sähköposti	Salasana (Heheh)	Käyttäjärühmä
1	Jarilii	Litmanen	Jarza	jarzanaama@litmanen.com	123	3
2	Jarmo "Jamo"	CurRy	Jarcca	jarccanen@kurri.com	321	1
3	Kuningas	Päälikkö	Admin	admin@admin.com		0

Muut käyttäjät saavat yrittäessään ikävän ja alentavan näkymän:

Kirjautunut käyttäjä: Jarza

Tarvitset ylläpitäjän oikeudet tarkastellaksesi tätä sivua



Tämä differointi käyttäjien kesken toivottavasti säilyttää asiaankuuluvan kunnioituksen ylläpitäjiä kohtaan ja vähentää käytöshäiriöitä.

6.2 Pelaajien listaus

Kirjaututtuaan käyttäjä saa näkyviin navigaatiopalkkiin lisää valikkoja, joista ensimmäinen on "Pelaajalistaus":

LiigaVinkit	Etusivu	Pelaajalistaus	Ottelulistaus	Käyttäjän seuraamat pelaajat	Kaikki kentällä	Käyttäjälistaus	Omat tiedot	Kirjaudu ulos
-------------	---------	----------------	---------------	------------------------------	-----------------	-----------------	-------------	---------------

Kaikki järjestelmässä olevat pelaajat

Pelaajatunnus	Pelaajan nimi	Joukkue	Pelipaikka	Syntymäaika	Maila	Paino	Pituus	Laukaukset	Seurannassasi?
22358280	Lehtonen, Mikko	KooKoo	VP	1994-01-16	L	89	182	170	1
23668720	Virtanen, Jesse	Lukko	VP	1991-08-07	L	84	181	158	0
34254136	Bertrand, Charles	Ässät	OL	1991-02-05	R	94	185	153	0
24703147	Lehtonen, Mikko	Kärpät	OL	1987-04-01	R	92	194	151	0
22671530	Nenonen, Markus	Sport	OL	1992-10-29	L	93	187	148	0


Näkymä esittää kaikki järjestelmässä olevat pelaajat tietoineen. Viimeinen sarake tarjoaa bitin, joka ilmaisee, onko käyttäjä liittänyt pelaajaa omien seurattaviensa piiriin (toteutettu pelaaja-olion metodilla `onkoKirjautuneenSeurannassa($pelaajaTunnus)`).

6.3 Pelaajien tarkemmat tiedot ja Käyttäjän seuraamat pelaajat

Pelaajien nimiä klikkaamalla pääsee pelaajan perustietoihin (näkymä on riippuvainen, onko pelaaja käyttäjän seurannassa vai ei):

LiigaVinkit	Etusivu	Pelaajalistaus	Ottelulistaus	Käyttäjän seuraamat pelaajat	Kaikki kentällä	Käyttäjälistaus	Omat tiedot	Kirjaudu ulos
-------------	---------	----------------	---------------	------------------------------	-----------------	-----------------	-------------	---------------

Mikko Lehtonen

Joukkue:	KooKoo
Pelipaikka:	VP
Syntymäaika:	1994-01-16
Kuva:	
Pelaajan Tilastot	
Laukaukset	170
Maalit	5
Torjutut	73
Blockatut	25
Oni	67

Poista pelaaja seurannastasi

Poista Seurannasta

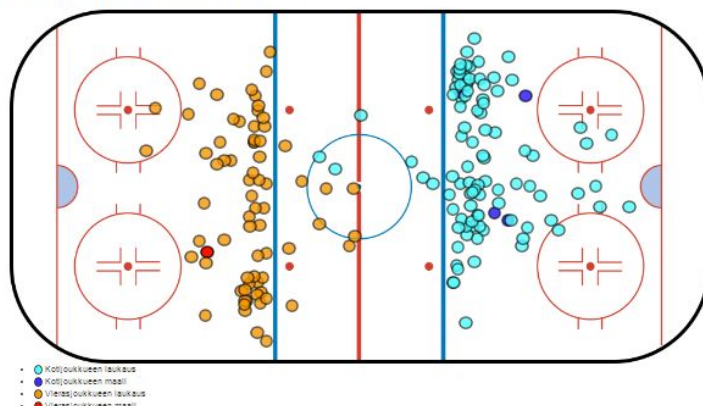
Muokkaa pelaajan seuranta

Aseta alkupäivämäärä

Aseta loppupäivämäärä

Muokkaa

Pelaajan laukaukset järjestelmässä



Pelaajan perustiedot näkymä tarjoaa käyttäjälle yksityiskohtaisempia tietoja pelaajasta, mm. Laukaukset-tietokantataulusta poimittuja. Tällä sivulla käyttäjä pääsee myös

1. Lisäämään pelaajan omien seurattaviensa listalle (CRUD "CREATE")
2. Muokkaamaan seuraamansa pelaajan tietoja seuratuissaan (CRUD "UPDATE")
3. Poistamaan seuraamansa pelaajan seuratuistaan (CRUD "DELETE")

Käyttäjä lisää seuraamiensa pelaajien listalle pelaajan omalta sivulta. Käyttäjä voi valita haluamansa parametrit (alkupvm / loppupvm [kurssin puitteissa nämä eivät vaikuta palvelun toiminnallisuuksiin]) lisäyksen tueksi. Järjestelmä kirjoittaa tämän jälkeen tietokantatauluun annetut tiedot, sekä lisäksi \$_SESSION['kayttaja']:n perusteella käyttäjälid:n, sekä pelaaja-olion perusteella pelaajaTunnuksen. Sarake "idKayttajanPelaajat" on autoincrementti.

idKayttajanPelaajat	kayttajaId	pelaajaTunnus	alkupvm	loppupvm
2	1	24256597	2016-05-05	2100-01-01
3	1	24244826	2016-05-05	2100-01-01
4	1	24260583	2016-05-05	2100-01-01
5	1	24242404	2016-05-05	2100-01-01
7	2	24260583	2016-05-05	2100-01-01
8	2	24256597	2016-05-05	2100-01-01

Käyttäjä saa seuraamansa pelaajat näkyviin palvelussa valitsemalla "Käyttäjän seuraamat pelaajat" (CRUD "READ"):

LiigaVinkit	Etusivu	Pelaajalistaus	Ottelulistaus	Käyttäjän seuraamat pelaajat	Kaikki kentällä	Käyttäjälistaus	Omat tiedot	Kirjaudu ulos
-------------	---------	----------------	---------------	------------------------------	-----------------	-----------------	-------------	---------------

Käyttäjän seuraamat pelaajat

Lisää pelaaja seurantaan

Pelaajatunnus	Etnimi	Joukkue	Pelipaikka	Syntymäaika	Maila	Paino	Pituus	Laukaukset	Alkaen	Päättyen
24242404	Aaltonen, Juhamatti	HIFK	OL	1985-06-04	R	88	184	117	2016-05-05	2100-01-01
24244826	Ahonen, Ari	Ässät	MV	1981-02-06	L	85	186	0	2016-05-05	2100-01-01
24260583	Ahonen, Rony	Lukko	VP	1987-01-17	L	96	181	71	2016-05-05	2100-01-01
24272490	Ahtola, Jerry	TPS	VL	1986-03-25	L	94	192	50	2016-01-01	2100-01-01
22358280	Lehtonen, Mikko	KooKoo	VP	1994-01-16	L	89	182	170	2016-12-16	2016-12-23

Käyttäjä pääsee jälleen pelaajan nimeä klikkaamalla tarkastelemaan pelaajan tietoja, sekä muokkaamaan tai poistamaan seuraamansa pelaajan.

6.4 Ottelulistaus ja ottelutiedot

Käyttäjä voi listata järjestelmässä olevat ottelut näkymään "Ottelulistaus":

LiigaVinkit	Etusivu	Pelaajalistaus	Ottelulistaus	Käyttäjän seuraamat pelaajat	Kaikki kentällä	Käyttäjälistaus	Omat tiedot	Kirjaudu ulos
-------------	---------	----------------	---------------	------------------------------	-----------------	-----------------	-------------	---------------

Kaikki järjestelmässä olevat ottelut

Ottelutunnus	Päivämäärä	Kotijoukkue	Vierasjoukkue	Yleisöä
4147	2016-09-16	Jukurit	Sport	4200
4153	2016-09-16	TPS	Kärpät	7251
4152	2016-09-16	Tappara	Ässät	7300
4151	2016-09-16	KooKoo	SaiPa	4712
4150	2016-09-16	Lukko	HIFK	3783
4149	2016-09-16	KalPa	Pelicans	2876

Sivuttamista harkittiin, mutta ei tämän kurssin puitteissa toteuteta. Näkymää testattiin myös enemmän tietoa sisältävänä versiona (koti- ja vierasjoukkuiden kaikki laukaisut tyypeittäin per ottelu). Informatiivisempi näkymä latautui 34 sekuntia ja piti sisällään 8 tietokantakyselyä jokaista riviä kohden (vaihtoehtona olisi tehdä monimutkaisempi SQL-kysely tai näkymä ja alustaa nämä kentät oliomuuttujaan). Sivuttamistakin mietittiin vaihtoehtona, mutta henkilökohtainen mieltymykseni on ko. ratkaisutapaa vastaan, joten päätin jättää tarkemman sisällön esittämättä listauksessa ja sisällyttää sen yksittäisten Ottelu-olioiden tarkemman tarkastelun näkymään, johon pääsee ottelutunnus-sarakkeen arvoja klikkaamalla:

2016-09-16 : Jukurit - Sport

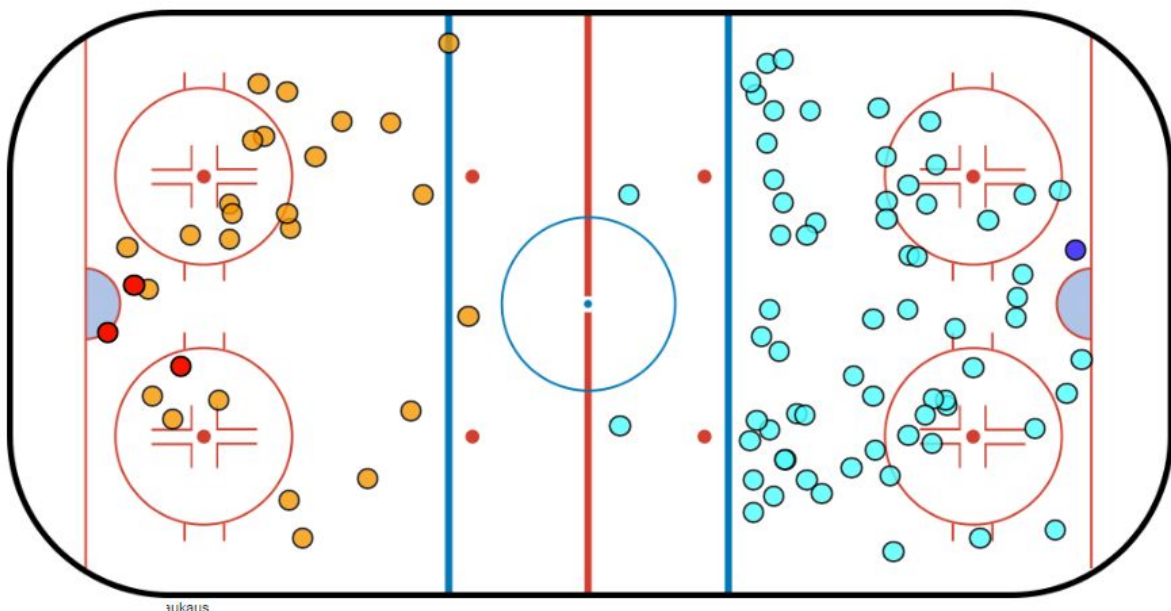
[Palaa ottelulistaukseen](#)

Jukurit		Sport	
Maalit	1	Maalit	3
Torjutut	35	Torjutut	13
Blokatut	2	Blokatut	0
Ohi	29	Ohi	12
Yhteensä	67	Yhteensä	28

Jukurit
Maalit
1
Laukaukset Yhteensä
67

Sport
Maalit
3
Laukaukset Yhteensä
28

Ottelun laukaukset järjestelmässä



Näkymä kokoa ottelun perustiedot, sekä laukaukset tyypeittäin *Laukaisut*-tietokantataulusta ja esittää ne visuaalisessa muodossa.

6.5 Kaikki kentällä

Tämä on Nice-to-have toiminnallisuus palvelussa, joka näyttää kaikki kannassa olevat yli 20000 laukausta pelikentille renderöityinä tyypeittain (Maalit, Torjunnat, Ohi, Blokatut).

Mielestäni on mielenkiintoista nähdä hajontaa eri laukaustyyppien suhteen. Esimerkiksi maalit syntyvät pääsääntöisesti maalien edestä (yllätys yllätys), kun taas siniviivalaukaukset päätyvät usein blokatuiksi toisin kuin keskialueelta lähtevät hutaisut. Tämä osoittaa visuaalisuuden voiman tyypillisten ihmisaivojen päättelyssä tilastojen suhteen, kun taas kone murskaa numeerisia koordinaattipisteitä tehokkaammin. ([extravinkki](#))
Myös pelaajien sivustot näyttävät laukaisukartat heidän laukauksilleen järjestelmässä ja paljastaa eroja esim. pelipaikkojen suhteen)

idLaukaisut	pelaajaTunnus	top	lefty	otteluTunnus	joukkue	aika	tulos	event
1223	21772430	12.3968871	63.1601562	4147	9	30:03:00	Maalivahti torjui	event-save
1224	21600081	64.3424124	79.46875	4147	9	30:23:00	Laukaus ohi maalin	event-miss
1225	22476117	69.2062256	87.0859375	4147	9	30:34:00	Laukaus ohi maalin	event-miss
1226	24644552	33.7976653	83.0820312	4147	9	11:06:00	Laukaus ohi maalin	event-miss
1227	24644552	52.0856031	80.25	4147	9	11:00:00	Maalivahti torjui	event-save
1228	21913668	36.3268482	67.5546875	4147	9	35:44:00	Laukaus ohi maalin	event-miss
1229	21772430	74.4591439	65.6015625	4147	9	35:52:00	Laukaus ohi maalin	event-miss
1230	22051859	15.1206225	67.8476562	4147	9	36:28:00	Laukaus ohi maalin	event-miss
1231	23394612	86.3268482	88.84375	4147	9	36:31:00	Maalivahti torjui	event-save
1232	24421720	10.2568093	62.7695312	4147	9	36:35:00	Laukaus ohi maalin	event-miss
1233	21888612	42.9416342	86.0117187	4147	9	36:59:00	Maalivahti torjui	event-save
1234	24421720	53.4474708	63.6484375	4147	9	38:40:00	Laukaus ohi maalin	event-miss
1235	24644552	90.0233463	74.9765625	4147	9	38:57:00	Maalivahti torjui	event-save
1236	24644552	67.6498054	63.2578125	4147	9	39:18:00	Laukaus ohi maalin	event-miss
1237	23075106	64.1478599	78.3945312	4147	9	44:59:00	Maalivahti torjui	event-save
1238	21888612	63.1750972	89.8203125	4147	9	46:06:00	Laukaus ohi maalin	event-miss
1239	22476117	14.7315175	73.7070312	4147	9	46:16:00	Laukaus ohi maalin	event-miss
1240	22051859	63.5642023	73.21875	4147	9	46:28:00	Maalivahti torjui	event-save
1241	21913668	57.3385214	91.0898437	4147	9	48:37:00	Maalivahti torjui	event-save
1242	22895620	68.622568	51.5390625	4147	9	49:25:00	Maalivahti torjui	event-save
1243	22895620	33.4085603	74.390625	4147	9	49:40:00	Maalivahti torjui	event-save
1244	22476117	66.8715953	77.7109375	4147	9	15:10:00	Maalivahti torjui	event-save
1245	21600081	48.9727626	76.1484375	4147	9	17:10:00	Laukaus ohi maalin	event-miss
1246	22240990	29.3229571	34.6445312	4147	23	22:57:00	Laukaus ohi maalin	event-miss
1247	22064642	36.3268482	14.7226562	4147	23	32:37:00	Laukaus ohi maalin	event-miss
1248	25116157	36.9105058	18.140625	4147	23	33:02:00	Maalivahti torjui	event-save
1249	24947327	66.0933852	33.6679687	4147	23	33:17:00	Maalivahti torjui	event-save
1250	25364666	31.0739299	18.140625	4147	23	33:43:00	Laukaus ohi maalin	event-miss
1251	24993465	77.5719844	29.9570312	4147	23	34:44:00	Maalivahti torjui	event-save
1252	23597163	52.8638132	7.69140625	4147	23	34:44:00	Maali	event-goal
1253	211375764	45.4708171	11.109375	4147	23	25:11:00	Maalivahti torjui	event-save

7. Asennustiedot

Asenna sovellus kopioimalla tiedostot palvelimelle internetissä näkyvään hakemistoon. Aseta sen jälkeen yhteystiedot oikeaksi lib/database.php tiedostossa. GitHub-repo antaa vinkkejä oikeiden parametrien asettamiseksi. Tässä users:lla olevassa toteutuksessa on käytetty mysql-tietokantaa. Muista myös tarkistaa yhtenevät charset ja collation-parametrit tietokannasta ja yhteyksistä. Tässä versiossa on kaikkialla käytetty utf8 / utf8_unicode_ci. Kurssin puitteissa palvelu on ainoastaan staattisen, sql-tiedostoissa annetun tiedon tarkkailua varten. Tuotantoversiossa live-feedia tarjoaa kurssin ulkopuolinen Python-koodi.

8. Käynnistys- ja käyttöohje

Palvelu on sen verran onnistunut, että käynnistys- ja käyttöohjeeksi pitäisi riittää:

1. Mene osoitteeseen: <http://jaakvirt.users.cs.helsinki.fi/liigaVinkit/>
2. Kirjaudu valmiilla peruskäyttäjällä

Käyttäjätunnus	Salasana
Jarza	123
Jarcca	321

3. Tai ylläpitäjänä ilman salasanaa

Käyttäjätunnus	Salasana
Admin	

4. Tai rekisteröi oma tunnuksesi

Mikäli ylläolevat kirjautumistiedot eivät toimi, niin joku on trollannut ja poistanut käyttäjiä. Nämä kuitenkin luodaan aina sql-tiedostossa "add_test_data.sql".

9. Testaus, tunnetut bugit ja puutteet, sekä jatkokehitysideat

9.1 Testaus

Palvelua on testattu monella koneella, laitteella ja selaimella. Esimerkiksi tuore Chrome-selain näyttää kaiken virheettömästi linuxilla, macilla ja winukalla. Kännykällä ja tabletilla iOS näyttää kaiken virheettömästi, joskin leveät taulukot ovat aina hieman haasteellisia.

9.2 Tunnetut bugit

Macin vanhahko safari-selain ei suostunut näyttämään KäyttäjänPelaajan lisäys/muokauspäivämääriä oikein, mikä voi helposti johtaa käyttäjän virheelliseen

päivämääräsyötteeseen [kurssin puitteissa ei merkitystä, koska sitä ei käytetä toiminnallisuuksiin ja virheellisellä syötteellä tallentuu päivämäärä 0000-00-00].

9.3 Puuteet

Päivämääräsyötteiden validointi on selvä puute.

9.4 Jatkokehitysideat

Lopullisesta muodosta saattaisi olla hyvä jättää pelaaja/otteluTunnuksia pois, sillä ne ovat tavallaan ylimääräistä tietoa, vaikkakin valistunut käyttäjä löytää niillä suoraan vastaavat tietorakenteet liiga.fi-sivustolta.

Jatkokehityksessä pääsisi melko pitkälle tilastojen lisäämisessä ja paremmassa crunchaamisessa, esimerkiksi:

- Laukausten käsittely sektoreittain ja tyypeittäin
- Pelaajien ja joukkueiden erinäisten vahvuuslukujen laskenta
- Kaiken yhdistäminen erilaisilla tavoilla

Käyttäjärühmien differointi ja niille lisänäkymien luonti, esimerkiksi:

- Näkymä, joka listaa ja renderöi kentälle seurattujen pelaajien laukaukset / tilastoja

Lisätietolähteiden lisääminen, esimerkiksi:

- Vedonvälittäjien kertoimia

10. Omat kokemukset

Tietokantasovelluksen tekemisessä hankalinta oli alkuvaihe ja työn rajaaminen kurssin tavoitteita noudattavaksi. Sen jälkeen tekeminen oli helpohkoa, koska se oli koko ajan mukavaa. Dokumentaation kirjoittamiseen motivaation löytäminen onnistui aiheeseen paneutumisen myötä.

Vaikeaa oli myös löytää MVC-mallin ja tietokannan yhteensovittaminen mielekkäällä tavalla. Myös MySQL:n konfiguroiminen skandimerkkien ja salasana-tarkistusten suhteen oli hieman hankalaa, mutta onnistui lopulta. Tietokantarakenteen luominen sujui mielestäni hyvin ja web-palvelun tyylyttely onnistui mukavasti aikaisempaan kokemattomuuteen nähden.

11. Muu dokumentaatio

Muu dokumentaation löytyy github-repostani (mm. sql-lauseet):

<https://github.com/JaakkoV/Tsoha-Bootstrap>