

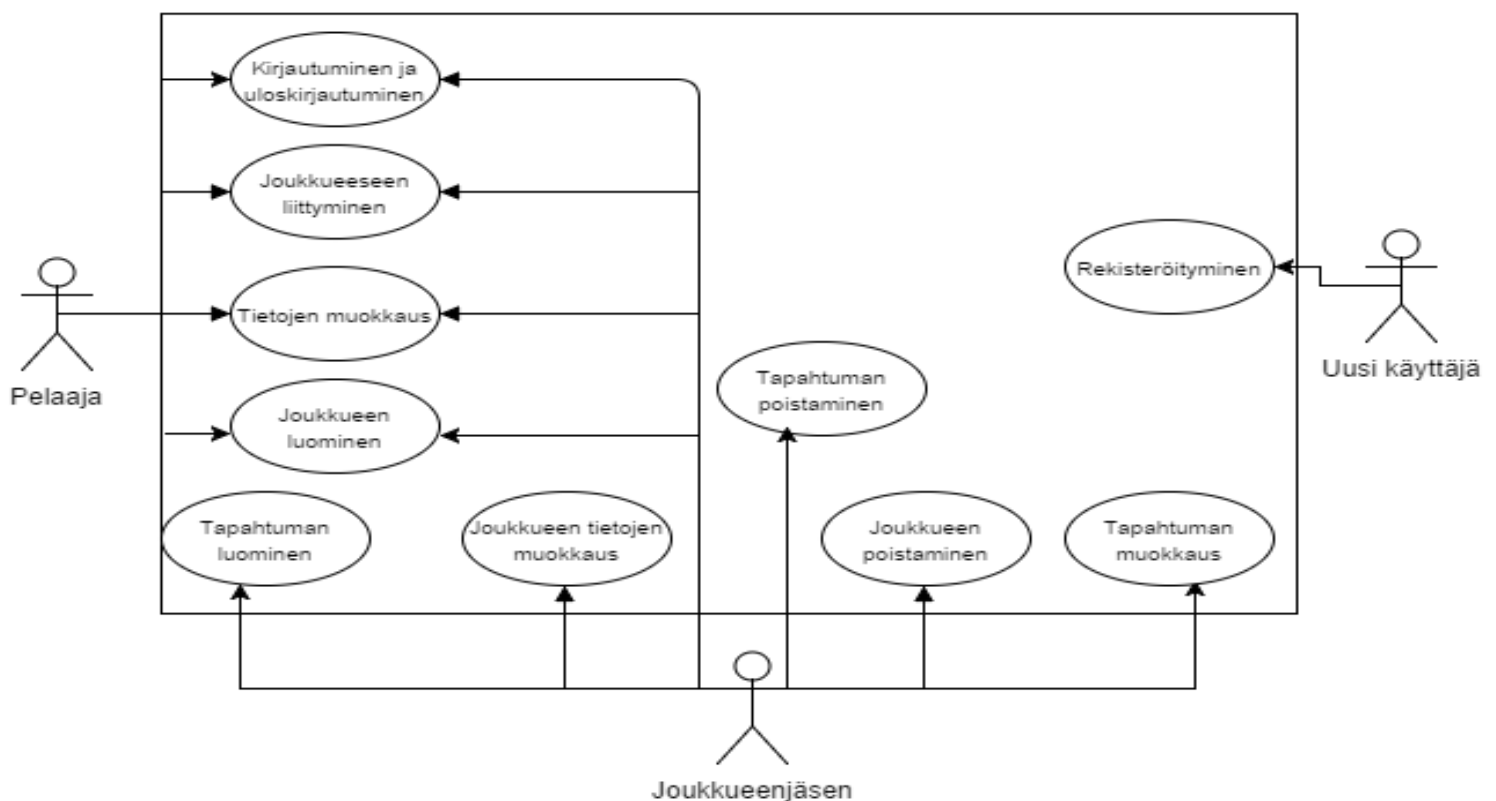
Johdanto

Ohjelman tarkoitus on luoda ympäristö urheilujoukkueille, jossa voidaan luoda ja hallita joukkueita. Järjestelmään voi luoda profiilin pelaajille ja joukkueille. Joukkueet voivat järjestää tapahtumia kuten harjoituksia ja otteluita, joita pelaajat voivat tarkistella sivuilta.

Ohjelma toteutetaan laitoksen users-palvelimelle Php:tä käyttäen.

Yleiskuva järjestelmästä

Käyttötapauskaavio



Käyttäjärühmät

Uusi käyttäjä

Kuka tahansa joka sattuu sivulle, joka ei vielä ole rekisteröitynyt

Pelaaja

Rekisteröitynyt käyttäjä, jolla ei ole joukkueenhallinta oikeuksia

Joukkueenjäsen

Rekisteröitynyt käyttäjä, joka on joukkueen jäsen, ja täten oikeudet muokata joukkuetta

Käyttötapauskuvaukset

Uuden käyttäjän käyttötapaukset

Rekisteröityminen

Uusi käyttäjä luo tunnuksen palveluun

Pelaajan käyttötapaukset

Kirjautuminen

Käyttäjä kirjautuu palveluun

Omien tietojen muokkaus

Pelaaja muokkaa omia tietojaan

Joukkueeseen liittyminen

Pelaaja liittyy joukkueen jäseneksi

Joukkueen luominen

Pelaaja luo uuden joukkueen, ja liittyy samalla sen jäseneksi

Joukkueenjäsenen käyttötapaukset

Tapahtuman lisääminen

Joukkueenjäsen lisää tapahtuman joukkueelle jossa hän on jäsen

Tapahtuman muokkaaminen

Joukkueenjäsen muokkaa tapahtumaa joukkueelta johon hän kuuluu

Tapahtuman poistaminen

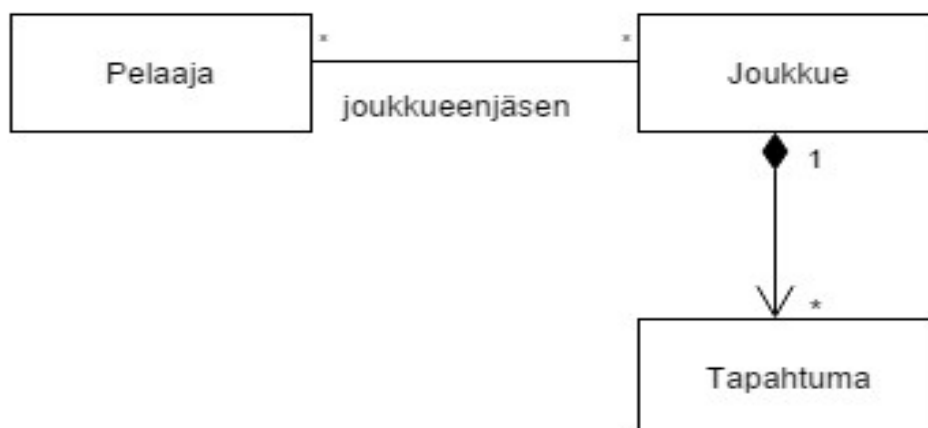
Joukkueenjäsen poistaa tapahtuman joukkueelta jossa hän on jäsen

Joukkueen tietojen muokkaus

Toimihenkilö muokkaa joukkueen tietoja

Järjestelmän tietosisältö

Käsitekaavio



Määrittelyt

Joukkue

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max 30	Joukkueen nimi
Perustamisvuosi	Kokonaisluku	Perustamisvuosi
Paikkakunta	Merkkijono	Paikkakunta

Joukkueeseen voi kuulua yhdestä ylöspäin pelaajia. Tapahtumat ovat joukkuekohtaisia, ja niitä voi olla nollasta ylöspäin.

Pelaaja

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max 30	Pelaajan nimi
Salasana	Merkkijono, max 30	Salasana

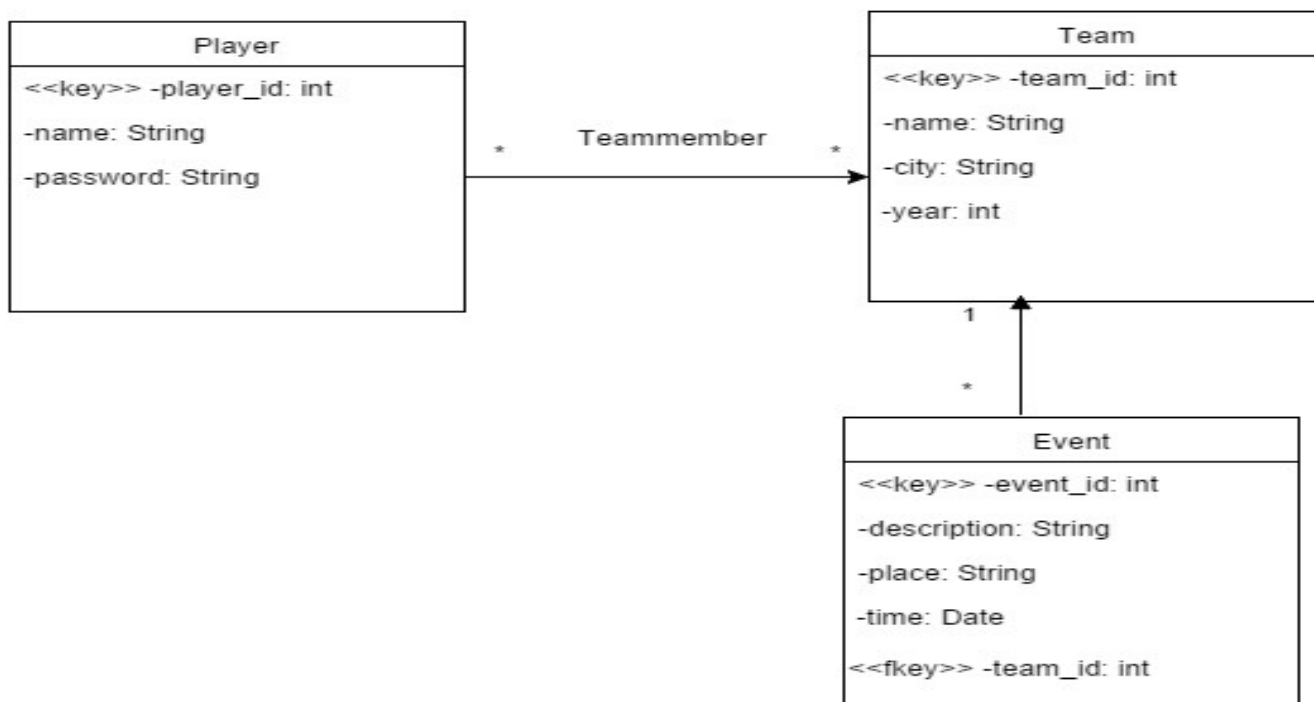
Pelaaja voi kuulua yhteen tai useampaan joukkueeseen, tai ei välttämättä yhteenkään.

Tapahtuma

Attribuutti	Arvojoukko	Kuvailu
Kuvaus	Merkkijono, max 150	Tapahtuman kuvaus
Aika	Päiväys	Tapahtuman ajankohta
Paikka	Merkkijono, max 30	Tapahtuman paikka

Tapahtuma on joukkuekohtainen, eli sillä voi olla vain yksi joukkue. Joukkueella voi toki olla monta tapahtumaa

Relaatiotietokanta



Järjestelmän yleisrakenne

Sovellus on jaoteltu MVC-mallin mukaan kontrolleireihin, näkymiin ja malleihin, jotka löytyvät app kansion sisältä controllers, views ja models kansioiden sisältä. Kirjastot löytyvät hakemistosta lib.

Asennustiedot

Asenna sovellus kopioimalla sen tiedostot palvelimen nettiin näkyvään hakemistoon (esim. usersin htdocs-hakemisto). Aseta sen jälkeen tietokannan yhteystiedot oikeaksi tiedostoon libs/config.php. Voit kopioida asetustiedoston mallin tiedostosta libs/config.php.dist

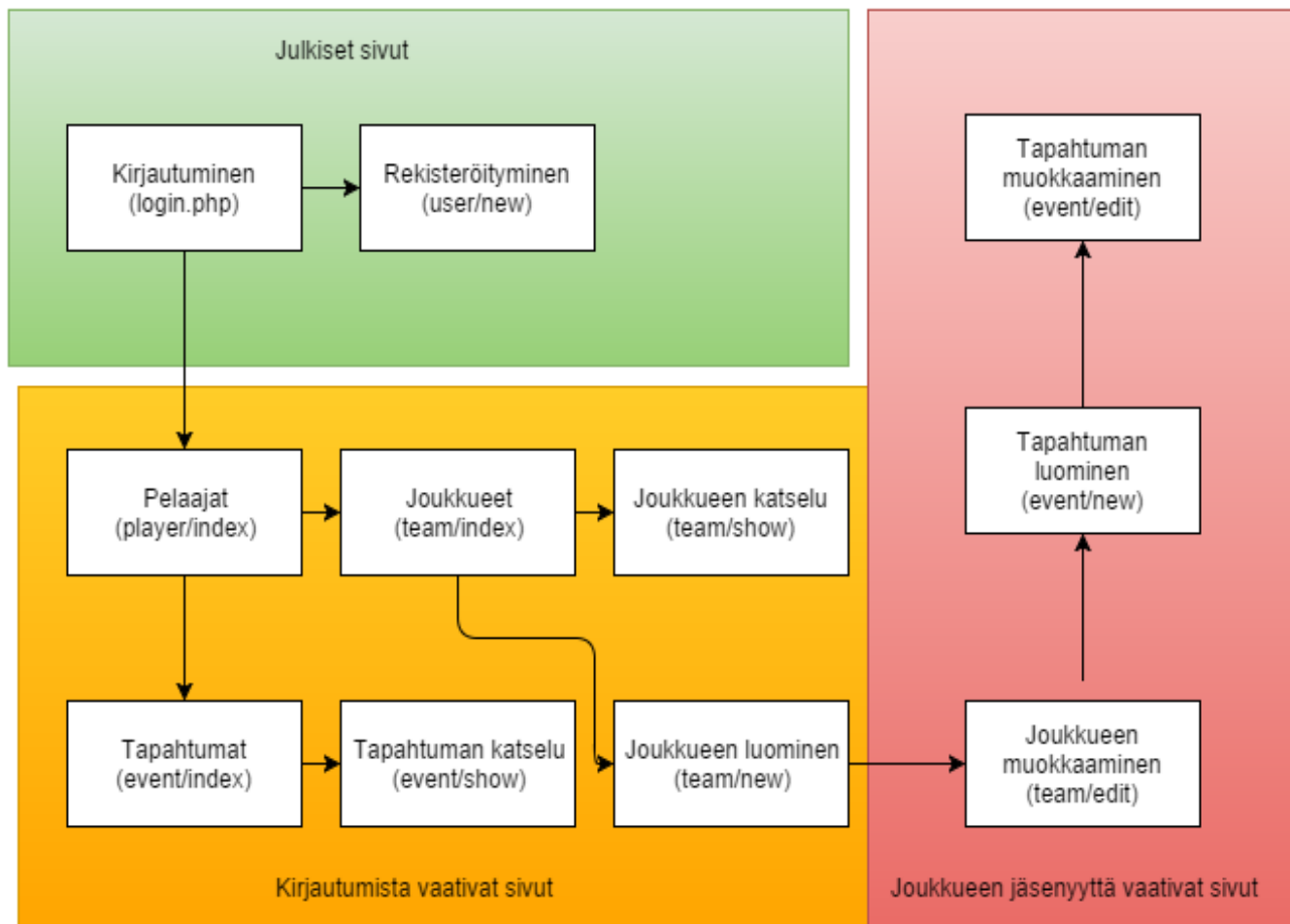
Käynnistys/Käyttöohje

Sovellus löytyy osoitteesta <https://jonharju.users.cs.helsinki.fi/tsoha/login>

Käyttäjänimi on Teemu ja salasana joku123

Sivulla pystyy tällä hetkellä lisäämään joukkueita ja hallinnoimaan niitä(muokkaamaan ja poistamaan). Joukkueisiin pystyy liittymään ja niille pystyy luomaan tapahtumia.

Käyttöliittymä ja järjestelmän komponentit



Testaus, tunnetut bugit ja puutteet & jatkokehitysideat

Ohjelmaa on testattu kokeilemalla kaikkia toimintoja sivuilla ja tutkimalla miten ne reagoivat siihen. Parannusideoina olisi mm. joukkueeseen liittymisen hyväksyminen, tapahtumien ajankohdan hyödyntäminen, tapahtumisiin ilmoittautuminen ja layoutin parantelu. Tällä hetkellä ei ole tunnettuja bugeja, mutta pelaajan ja joukkueen monesta-moneen suhteen päivityksen jälkeen saatta löytyä joukkueista ja sen tapahtumista jäänteitä vanhasta ratkaisusta, jotka sen myötä sen nyt saattavat aiheuttaa ongelmia. Päiväyksen validaattoria en saanut toimimaan, joten siihen pitää antaa syöte muodossa YYYY-MM-DD, muuten ohjelma kaatuu. Validaattori herjaa muistin loppumista.

Omat kokemukset

Tietokantasovelluksessa helppoa oli idean saaminen. Myöskin itse ohjelmointi eteni hyvin sulavasti ja debuggaaminen oli huomattavasti helpompaa kuin mihin tottunut, sillä ohjeet auttoivat yleensä korjaamaan tai päättelämään missä vika lienee. Vaikeaa taas oli keksiä miten on parasta toteuttaa ideat, joten ratkaisuja on ollut monta erilaista ja niitä on saanut muuttaa useampaan otteeseen. Opin ohjelmoimaan php:ta ja miten tietokanta toteutetaan käytännössä, sekä miten sitä voi hyödyntää sovelluksessa.