

Tietokantasovellusdokumentaatio

Timo J. Kärkkäinen

14. joulukuuta 2017

JOHDANTO

Aihe: Pokémon-tietokanta.

Järjestelmän tarkoituksena on pitää kirjaa Pokémon-pääpelisarjassa omistetuista Pokémoneista. Niiden voimakkuuden määrittävät arvot (Level, Attack, Defense, Speed, HP...) ja hyökkäykset voivat muuttua pelin aikana. Käyttäjä voi lisätä, poistaa ja muuttaa omia Pokémonejaan, perustaa Pokémon-liigaksi kutsutun ryhmän ja toimia sen johtajana. Käyttäjä voi liittyä jäsenenä mihin tahansa liigaan ja erota siitä halutessaan. Liigan johtaja voi erottaa jäsenen liigasta.

Työ toteutetaan PHP-ohjelmointikielellä laitoksen `users2017.cs.helsinki.fi`-palvelimella. Sovelluksen lähdekoodit pidetään ajan tasalla GitHub-sivulla. Työssä ei käytetä JavaScriptiä, poikkeuksena tietojen poistamisen vahvistaminen toteutetaan JavaScriptin JQuery-kirjastoa käyttäen. Tietokanta toteutetaan PostgreSQL:lla. Sivuston jotkin elementit käyttävät HTML5-standardia, ja ulkoasu luodaan Twitter Bootstrap -nimisellä CSS-kirjastolla.

JÄRJESTELMÄN YLEISKUVAUS

KÄYTTÖTAPAUSKAAVIO

Löytyy seuraavalta sivulta, kuvasta 1.

KÄYTTÄJÄRYHMÄT

1. **VIERAILIJA** on kuka tahansa WWW-sivulle saapuva henkilö. Hän pystyy selaamaan Pokémonien, rekisteröityneiden käyttäjien ja liigojen tietoja.
2. **KÄYTTÄJÄ** on henkilö, joka on luonut tunnuksensa järjestelmään ja kirjautunut sisään. Hän voi lisätä, muokata ja poistaa omia Pokémonejaan ja liigojaan. Hän voi poistaa jäseniä liigoista, joissa on johtajana.
3. **YLLÄPITÄJÄ** on henkilö, joka pystyy lisäämään uusia Pokémon-lajeja tietokantaan. Hän voi myös muokata ja poistaa mitä tahansa tietoa. Ylläpitäjiä on vain yksi, ja hänen käyttäjätunnuksensa on `admin`, salasana `pass1`.



Kuva 1: Sovelluksen käyttötapauskaavio.

KÄYTTÖTAPAUKSET

1. Pokémonien tietojen selaaminen

Kaikille sallittu

Voi navigoida Pokémonien listaus- ja esittelysivuille, joilla näkyvät Pokémonien ominaisuudet. Pokémon-lajin esittelysivulla näkyvät kaikki tietokannassa olevat lajin yksilöt.

2. Käyttäjien tietojen selaaminen

Kaikille sallittu

Voi navigoida käyttäjien listaus- ja esittelysivuille, joilla näkyvät käyttäjien omistamat Pokémonit, liigajäsenyydet ja liigajohtajuudet.

3. Liigojen tietojen selaaminen

Kaikille sallittu

Voi navigoida liigojen listaus- ja esittelysivuille, joilla näkyvät liigojen johtajat ja jäsenet.

4. Oman Pokémonin tietojen muuttaminen

Vain ylläpitäjä ja käyttäjä

Voi muokata käyttäjän omaa Pokémonia tai poistaa sen. Käyttäjän on oltava Pokémonin luoja.

5. Pokémon-lajien tietojen muuttaminen

Vain ylläpitäjä

Voi muuttaa Pokémon-lajin tietoja tai poistaa sen.

6. Oman liigan tietojen muuttaminen

Vain ylläpitäjä ja käyttäjä

Voi muokata liigan tietoja, erottaa liigan jäseniä tai poistaa liigan. Käyttäjän on oltava liigan johtaja.

7. Omien tietojen muuttaminen

Vain ylläpitäjä ja käyttäjä

Voi muuttaa omia tietojaan tai poistaa tunnuksensa.

8. Liigaan liittyminen

Vain ylläpitäjä ja käyttäjä

Voi liittyä liigaan jäseneksi, jos ei ole jo valmiiksi jäsen. Liigan johtaja ei voi liittyä omaan liigaansa.

9. Liigasta eroaminen

Vain ylläpitäjä ja käyttäjä

Voi erota liigasta, jos on jo jäsen. Liigan johtaja voi erottaa minkä tahansa jäsenen.

10. Jäsenyyksien selaaminen

Vain ylläpitäjä

Voi navigoida liigajäsenyyksien listaussivulle.

JÄRJESTELMÄN TIETOSISÄLTÖ

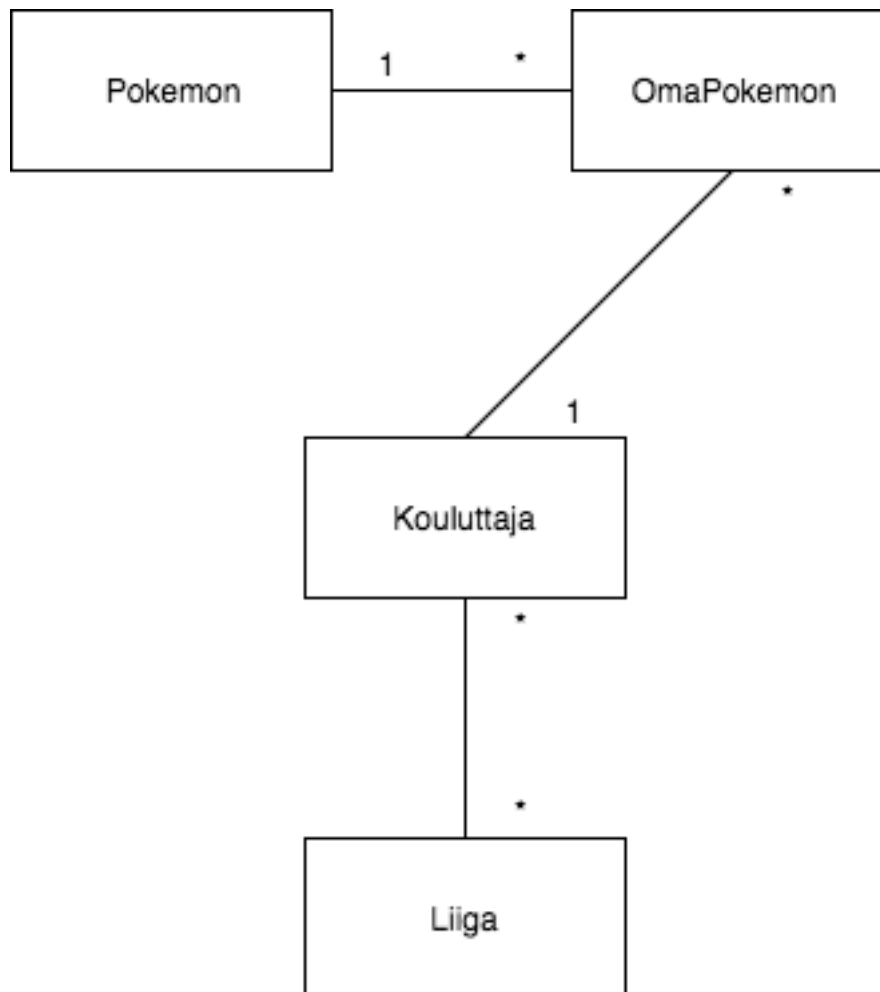
Käsitelkaavio on kuvassa 2.

1. Tietokohde: Pokemon.

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max. 20 merkkiä	Pokémon-lajin nimi, esimerkiksi Pikachu
Tunnusluku	Kokonaisluku (1-9999)	Pokémon-lajin järjestysnumero (pääavain)
Pituus	Kokonaisluku (1-9999)	Pokémon-lajin pituus
Paino	Kokonaisluku (1-9999)	Pokémon-lajin paino
Kuvaus	Merkkijono, max. 1000 merkkiä	Pokémon-lajin vapaa kuvaus

Pokemon-taulun sarakkeet sisältävät kaikille saman lajin yksilöillä olevat yhteiset ominaisuudet (myös pituus ja paino, vaikka tämä onkin intuition vastaista!). Järjestelmään kirjautunut käyttäjä ei voi muuttaa niitä.

2. Tietokohde: OmaPokemon



Kuva 2: Sovelluksen käsitelkaavio

Attribuutti	Arvojoukko	Kuvailu
id	Kokonaisluku	Pääavain
Atk	Kokonaisluku (1-999)	Hyökkäystaso
Def	Kokonaisluku (1-999)	Puolustustaso
Speed	Kokonaisluku (1-999)	Nopeus
SpAtk	Kokonaisluku (1-999)	Erikoishyökkäystaso
SpDef	Kokonaisluku (1-999)	Erikoispuolustustaso
HP	Kokonaisluku (1-999)	Terveystaso
Lvl	Kokonaisluku (1-100)	Koulutustaso
Sukupuoli	Merkki	U (uros), N (naaras) tai E (ei ole)
Lempinimi	Merkkijono, max. 20 merkkiä	Kouluttajan antama lempinimi
Esine	Merkkijono, max. 20 merkkiä	Pokémonin pitämä esine
Laji	Kokonaisluku (1-9999)	Pokémon-lajinumero (vierasavain)
Kouluttaja	Merkkijono, max. 20 merkkiä	Kouluttajan nimi (vierasavain)
Kuvaus	Merkkijono, max. 1000 merkkiä	Pokémonin vapaa kuvaus

OmaPokemon-taulun sarakkeet sisältävät Pokémonin sellaiset yksilölliset tiedot, jotka vaihtelevat lajin sisällä, kuten HP- ja nopeustasot. OmaPokemon on aina jokin tietyn Pokémon-lajin edustaja, ja järjestelmässä voi olla useita saman lajin edustajia. OmaPokemonilla on aina täsmälleen yksi Kouluttaja, ja kouluttajalla voi olla useita OmaPokemoneja. Koska Kouluttajalla voi olla useita saman lajin OmaPokemoneja, on OmaPokemonit yksilöitävä pääavaimen 'id' avulla.

3. Tietokohde: Kouluttaja

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max. 20 merkkiä	Käyttäjätunnus kirjautumista varten (pääavain)
Salasana	Merkkijono, max. 20 merkkiä	Kirjautumissalasana
Kuvaus	Merkkijono, max. 1000 merkkiä	Käyttäjän vapaamuotoinen kuvaus

Kouluttaja on yksinkertaisesti järjestelmään kirjautunut käyttäjä, jolla on käyttäjätunnus ja salasana. Kouluttajalla voi olla useita OmaPokemoneja sekä kuulua useaan Liigaan. Lisäksi Kouluttaja voi olla yhden Liigan johtaja.

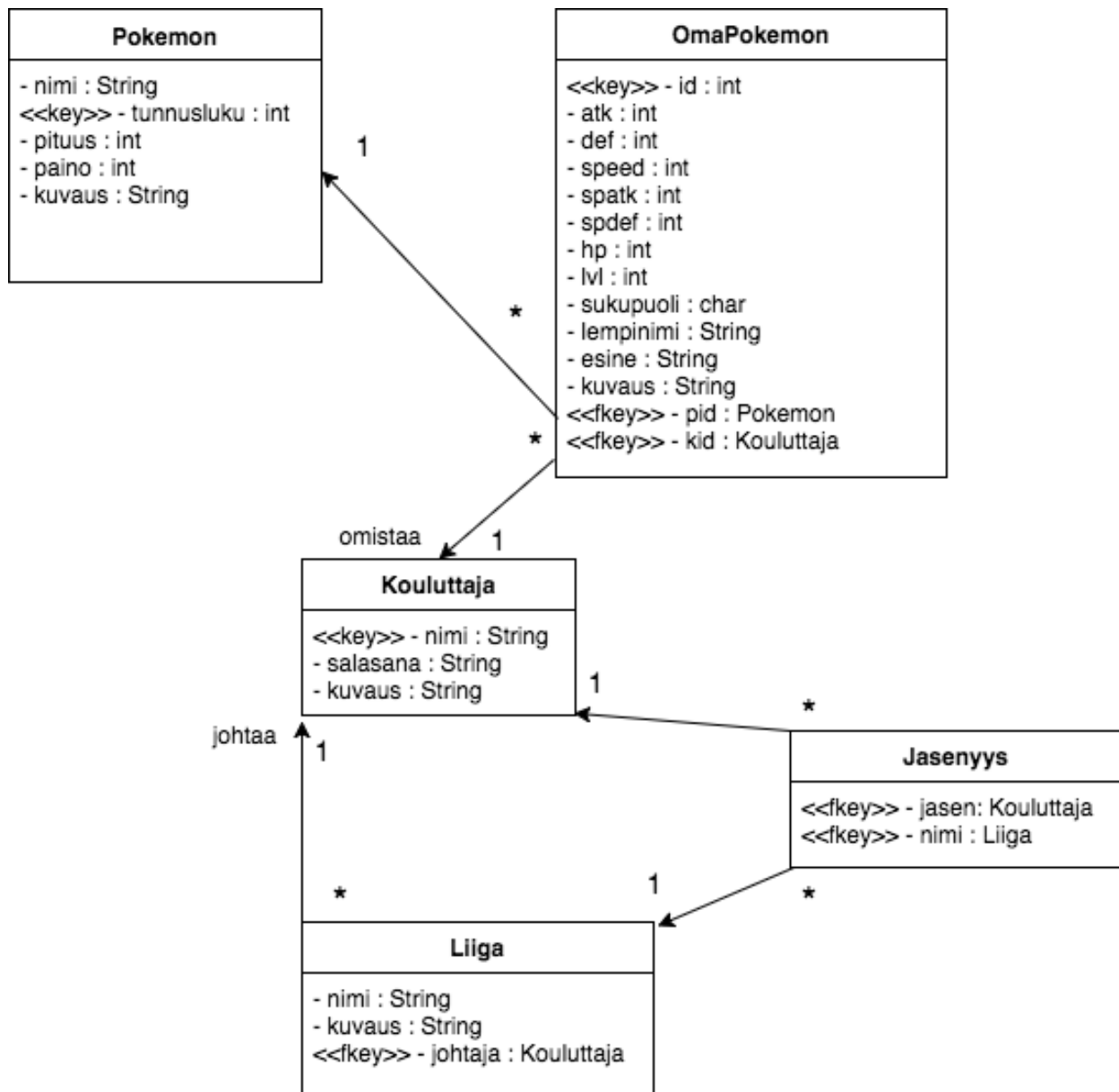
4. Tietokohde: Liiga

Attribuutti	Arvojoukko	Kuvailu
Nimi	Merkkijono, max. 20 merkkiä	Pokemon-liigan nimi
Johtaja	Merkkijono, max. 20 merkkiä	Liiga johtajan nimi (vierasavain)
Kouluttaja	Merkkijonotaulukko	Sisältää liigaan kuuluvien kouluttajien nimet
Kuvaus	Merkkijono, max. 1000 merkkiä	Liigan vapaamuotoinen kuvaus

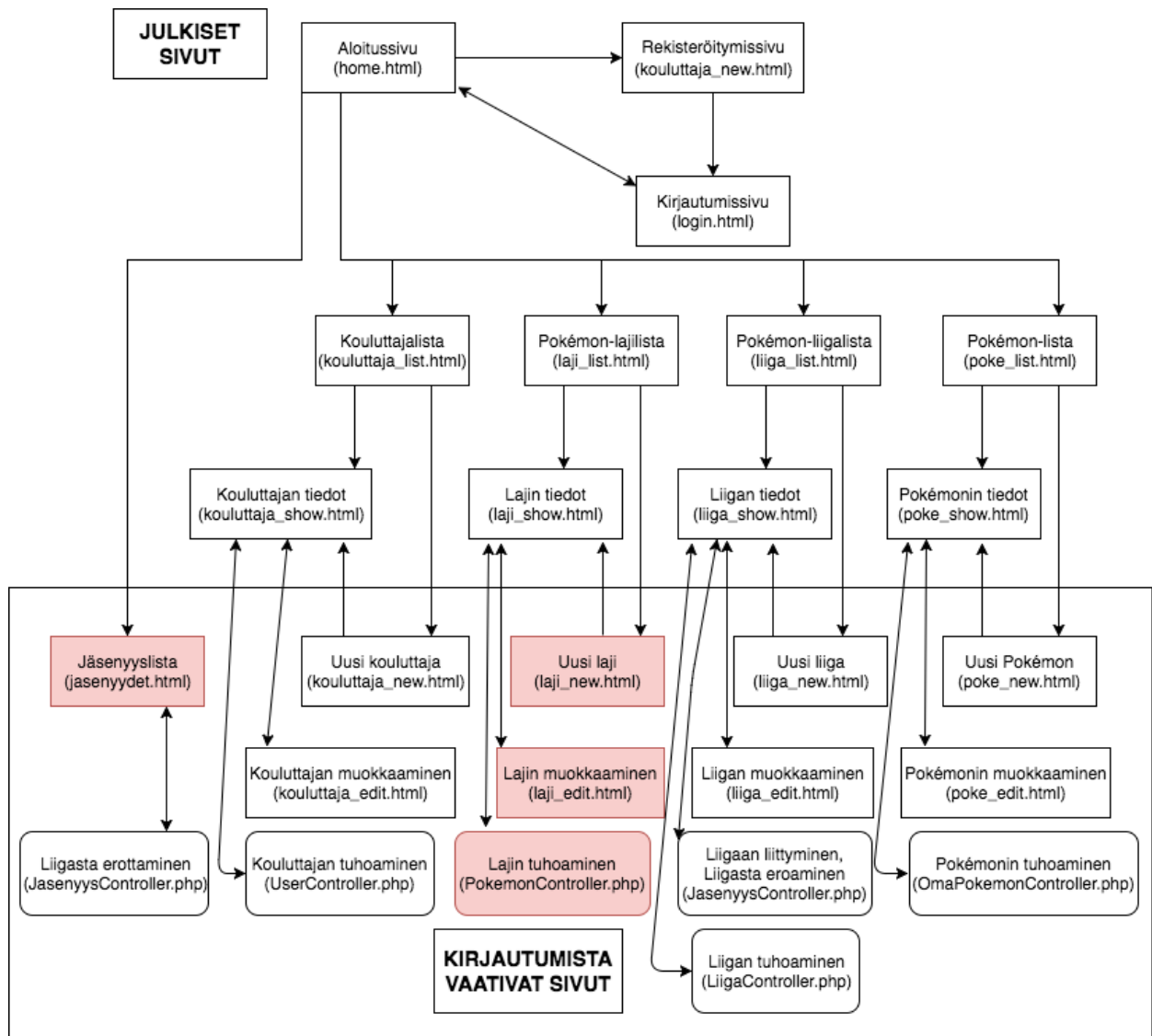
Liiga koostuu yhdestä johtajan roolissa olevasta Kouluttajasta, ja mahdollisesti muista Kouluttajista.

RELAATIOTIETOKANTAKAAVIO

Relaatiotietokantakaavio on kuvassa 3. Siinä on myös yksi monesta moneen -tyyppinen relaatio, liigojen ja kouluttajien välillä. Toisin kuin kuvassa 2, kaaviossa esiintyy liitostaulu Jasenyys. Se on luotu helpottamaan monesta moneen -yhteyksien toimintaa Kouluttaja- ja Liiga-taulujen välillä. Liigalla voi olla useita jäseniä, ja kouluttaja voi olla jäsen useassa liigassa.



Kuva 3: Sovelluksen tietokantakaavio.



Kuva 4: Sovelluksen käyttöliittymä. Punaisella merkityt kentät vaativat, että käyttäjä on kirjautunut sovellukseen ylläpitäjänä. Käyttäjä voi muokata vain omia tietojaan, Pokémonejaan ja liigojaan sekä erottaa vain sellaisista liigoista jäseniä, joissa hän on liigaohjaajana.

JÄRJESTELMÄN YLEISRAKENNE

Tietokantasovellus on tehty MVC-mallin periaatteita noudattaen. Kansiossa **app** on kolme kansiota: **models** (M), **views** (V) ja **controllers** (C). Ne sisältävät vastaavasti malli-, näkymä- ja ajuritiedostot¹ Malli- ja ajuritiedostonimet alkavat isolla alkukirjaimella, mutta noudattavat muutoin camelCase-muotoa, kun taas näkymätiedostot alkavat pienellä alkukirjaimella ja noudattavat snake case -muotoa. Näkymätiedostojen peruskomponentit ovat **views**-kansion juuressa, kun taas kouluttajiin, Pokémon-lajeihin, Pokémon-liigoihin ja omiin Pokémoneihin liittyvät näkymätiedostot on ryhmitelty kansioihin **kouluttaja**, **laji**, **liiga** ja **poke**, vastaavasti **suunnitelmat**-kansion sisällä.

Tietokantataulujen luomis-, tuhoamis- ja testidatanlisäystiedostot sijaitsevat kansiossa **sql**, apukirjastot ovat kansiossa **lib**, ulkoasuun luovat CSS- ja JavaScript-tiedostot kansiossa **assets** ja sovelluksen konfiguraatiotiedostot kansiossa **config** (missä erityisesti **routes.php** sisältää tiedot, mihin palvelin ohjaa selaimen antaman HTTP-pyyynnön). Sovelluksen dokumentaatio (eli tämä dokumentti) on kansiossa **doc**.

KÄYTTÖLIITTYMÄ JA JÄRJESTELMÄN KOMPONENTIT

Käyttöliittymäkaavio on kuvassa 4. Sovelluksessa on yläreunassa navigaatiopalkki, josta pääsee missä tahansa vaiheessa sovelluksen mille tahansa listaussivulle, voi kirjautua ulos tai sisään tai rekisteröityä (jos ei ole kirjautunut vielä sisään).

SOVELLUKSEN ASENTAMINEN

Kopioi tiedostot palvelimelle, kansioon joka on julkisesti näkyvissä Internetissä. Konfiguroi `config/database.php`, aja komentorivillä bash-skriptit `create_tables.sh` ja `add_test_data.sh` (ensimmäinen luo tietokantataulut, jälkimmäinen lisää testidataa). Siirry selaimella osoitteeseen, mihin sovellus asennettiin.

KÄYNNISTYS- JA KÄYTTÖOHJE

Sovellus sijaitsee osoitteessa `http://tjkarkka.users.cs.helsinki.fi/tsoha/`. Pääkäyttäjätunnus on `admin`, ja salasana `pass1`. Sisäänkirjautumisnappula on oikeassa yläkulmassa. Kirjautumisen jälkeen käyttäjä ohjataan etusivulle ja siihen ilmestyy tiedote onnistuneesta kirjautumisesta. Vierailija voi selata sovellusta ilman muokkaus-, lisäys- ja poisto-oikeuksia. Kuka tahansa voi rekisteröidä tunnuksen.

TESTAUS, TUNNETUT BUGIT, PUUTTEET JA SOVELLUKSEN JATKOKEHITYS

En ole tällä hetkellä tietoinen ohjelmassa mahdollisesti olevista bugeista.

Sovellusta voi laajentaa lisäämällä mahdollisuus tykätä toisten käyttäjien Pokémoneista ja antamalla kommentointimahdollisuus liigan esittelysivulla. Lisäksi Pokémoneille pitäisi antaa mahdollisuus kehittyä toisiksi Pokémoneiksi. Ylläpitäjän pitäisi voida estää tiettyä käyttäjää kirjautumasta sisään ilman, että käyttäjän tiedot poistetaan - eli olisi annettava mahdollisuus väliaikaisen porttikiellon antamiseen. Salasanat tallennetaan tekstimuotoisena, mikä on tietoturvan kannalta erittäin vaarallista. Jatkokehityksessä on ne suojattava esimerkiksi *password hashing* -menetelmällä.

OMAT KOKEMUKSET

Tietokantasovelluskurssin materiaali oli ensiluokkainen ja tekstistä huokui asiantuntevuus. Materiaalissa työstettävä esimerkki toimi erinomaisena pohjana sovellukseeni. Kaikkein helpointa oli sovelluksen laajentaminen sen jälkeen, kun prototyypiversio oli valmiina. Lisäksi aiheen valinta oli itselleni helppoa. Vaikeinta oli ymmärtää miten pystyin yhdistämään tietoja eri tietokantatauluista, ja millä tavalla yhdistäminen kannattaisi tehdä: tulisiko yhdistäminen tehdä suoraan SQL-kyselyssä, vai suorittaa useita kyselyjä ja hakea aluksi "liikaa" tietoa ja suodattaa lopuksi tarpeellinen tieto ulos?

¹ *Kontrolleri* on kankeaa anglismia. Suositeltavampaa on puhua ajurista, ohjaimesta tai käsittelijästä. Tässä dokumentissa käytän termiä *ajuri*.