



Antti Aho, Andreas Lang , Jaakko Kuivasniemi, Jesper Oja

LFG

Lets Fucking Go

Looking For Group

Linkedin For Gamers

Metropolia Ammattikorkeakoulu
Insinööri (AMK)

Tieto- ja viestintäteknologian tutkinto-ohjelma

1 Johdanto

LFG on sosiaalinen verkosto nimenomaan pelaajille, jossa käyttäjät voivat lisätä profiiliinsa pelejä, katsoa mitä muut pelaavat, tehdä postauksia, jakaa pelejään sekä kommentoida ja tykätä tai "dislikettää" näistä kaikista.

Dokumentista saa hyvän kuvauksen koko LFG:n toiminnasta ohjelmana.

Projektin backend on rakennettu microsoftin ASP.NET frameworkillä käyttäen MySQL tietokantaa, ja frontend React.js Frameworkkia hyödyntäen Typescriptiä.

2 Tuotteen vaatimukset

Tärkeimmät tuotetavoitteet: Olla pelaajan apuväline ja sosiaalinen alusta, johon voi luoda profiilin ja tähän merkitä pelejä, joita käyttäjä pelaa. Alustalla pelaajat voivat tutustua ja vertailla sijoituksiaan (rankkeja), etsiä peliseuraa, seurata toisiaan, muodostaa tiimejä sekä etsiä peliseuraa. Mahdollisesti myös striimata (käyttäen ulkopuolista palvelua esim. Twitch.tv:n kautta streamin jakoa) ja jakaa pelejään.

Tuote on kohdistettu kaikenikäisille ja tasoisille pelaajille, eSport-tiimeille sekä turnauksien järjestäjille ym.

Käyttäjä pystyy ottamaan yhteyttä toisiin pelaajiin, sekä alusta helpottaa verkostoitumista peliyhteisöissä.

Juuri vastaavanlaista palvelua ei oikeastaan vielä ole, joten tuote on uniikki. Lähimmät tuotteen kaltaiset ovat Discord, Steam community sekä LinkedIn.

Alustalla käyttäjä voi:

- Luoda profiilin ja ilmoittaa siellä mitä kaikkea ja millä tasolla käyttäjä pelaa.
- Etsiä pelejä ja katsoa ketkä muut kyseistä peliä pelaa, ja lisätä näitä kavereiksi ja jutella heidän kanssa.
- Voi tehdä postauksia, nähdä muiden postauksia, sekä kommentoida niitä.
- Saada ehdotuksia mahdollisesti sopivasta peliseurasta peliesi ja tasosi perusteella.
- Julkaista päivityksiä, kuvia ja videoita.

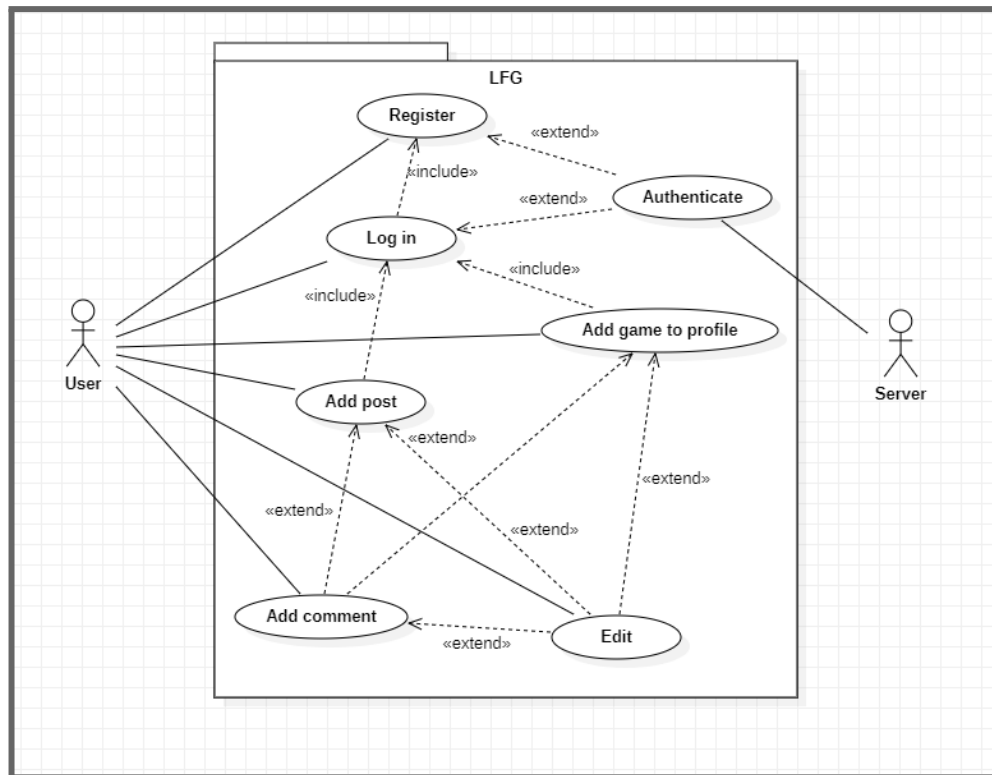
Näillä toiminnoilla pelaaja voi helposti löytää itselleen peliseuraa, löytää kavereita sekä jakaa onnistumisiaan peleissä. Kanavilla voi myös hakea apua tai opettaa muita.

3 Käyttäjäroolit ja käyttötapaukset

Käyttäjärooleja on siis vain yksi, normaali käyttäjä.

Käyttäjä voi rekisteröidä itselleen käyttäjän, jonka jälkeen voi kirjautua sisään. Palvelin tarkistaa rekisteröinnissä ja sisäänkirjautumisessa annetut tiedot valideiksi.

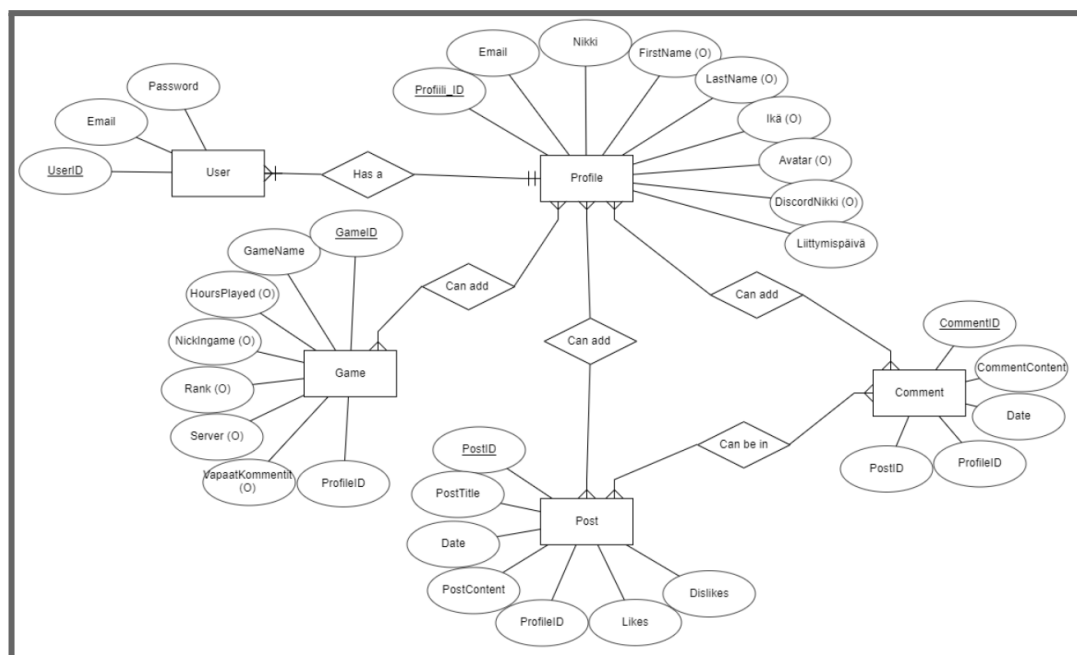
Kun käyttäjä on kirjautunut sisään, hän voi lisätä pelejä omaan profiilinsa, tehdä postauksia, ja kommentoida pelejä ja postauksia. Näitä kaikki voi myös jälkikäteen muokata. Peleillä, postauksilla ja kommenteilla on myös alapeukku (Like) ja yläpeukku (Dislike) nappulat.



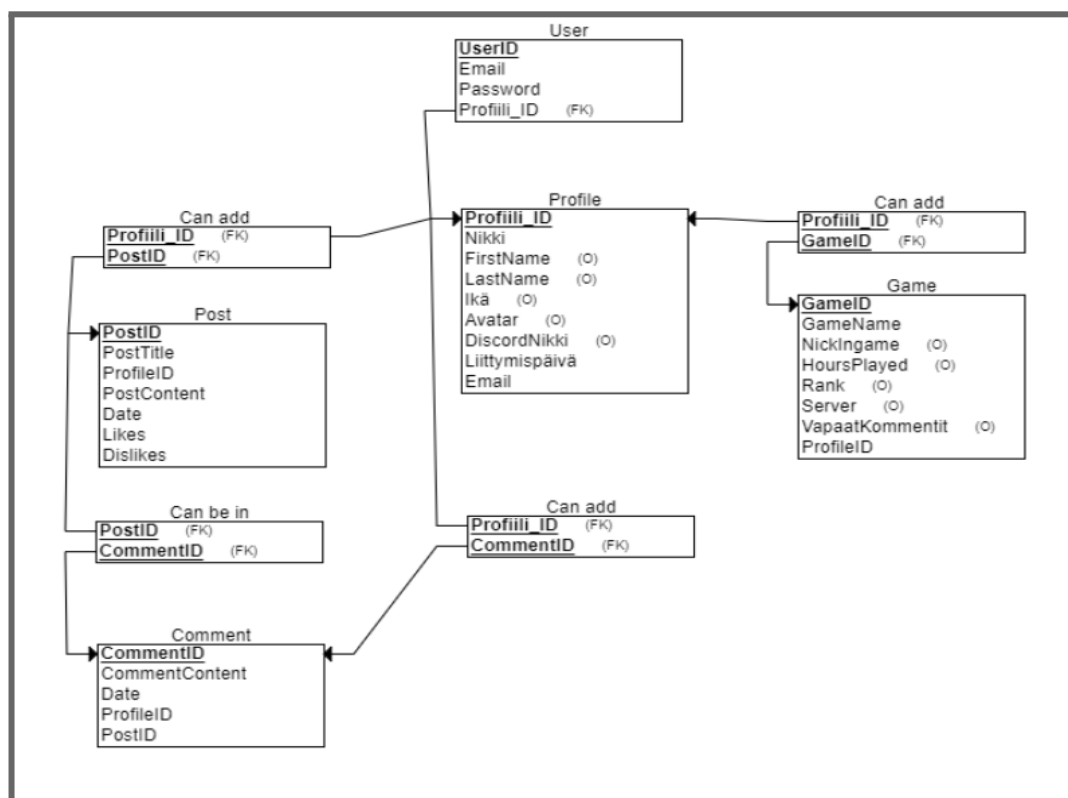
Kuva 1

4 Ohjelmiston tietomalli

Entity-Relation kaavio projektimme tietokannasta (Kuva 2), ja sen alta löytyy ER kaaviosta johdettu projektin tietokannan Relaatiokaavio (Kuva 3).



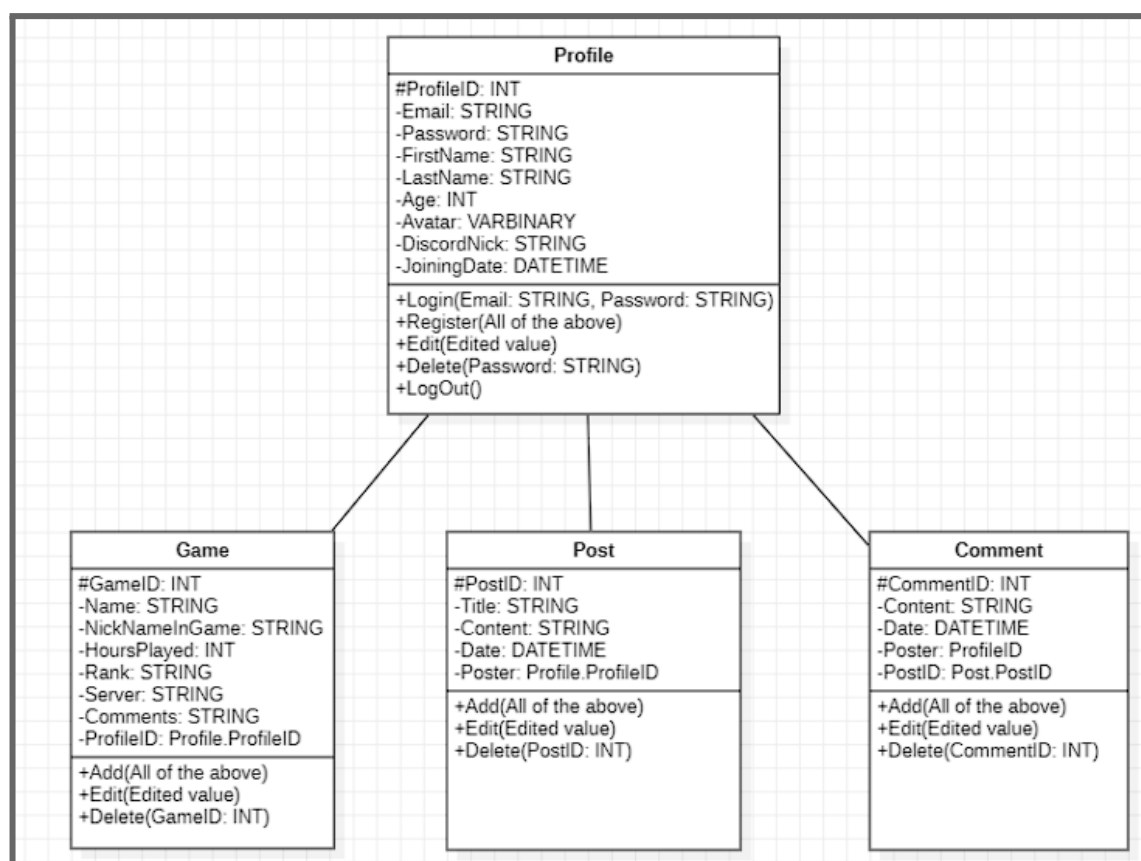
Kuva 2



Kuva 3

5 Ohjelmiston rakenne

Kuvassa LFG:n luokkakaavio. Kaaviosta puuttuu Like ja Dislike muuttujat ja metodit Post ja Comment luokista myöhäisen toteutuksen takia. Projekti toteutettiin ASP.Netillä ja React.js:lla joten varsinaisia pakkauksia ei ole kuten Javassa, mutta Model, Controller ja View namespaces löytyy koska projekti toteutettiin MVC mallin mukaan, tosin View:iä toteuttaa React.js, dataa taas namespace Model ja toiminnallisuutta sekä näiden välistä liikennettä namespace Controller.



Kuva 4

6 Ohjelmiston toiminta

Human - käyttäjä avaa sivun, jolloin website kysyy käyttäjätietoja. Server tarkastaa databasesesta onko kyseistä käyttäjää olemassa:

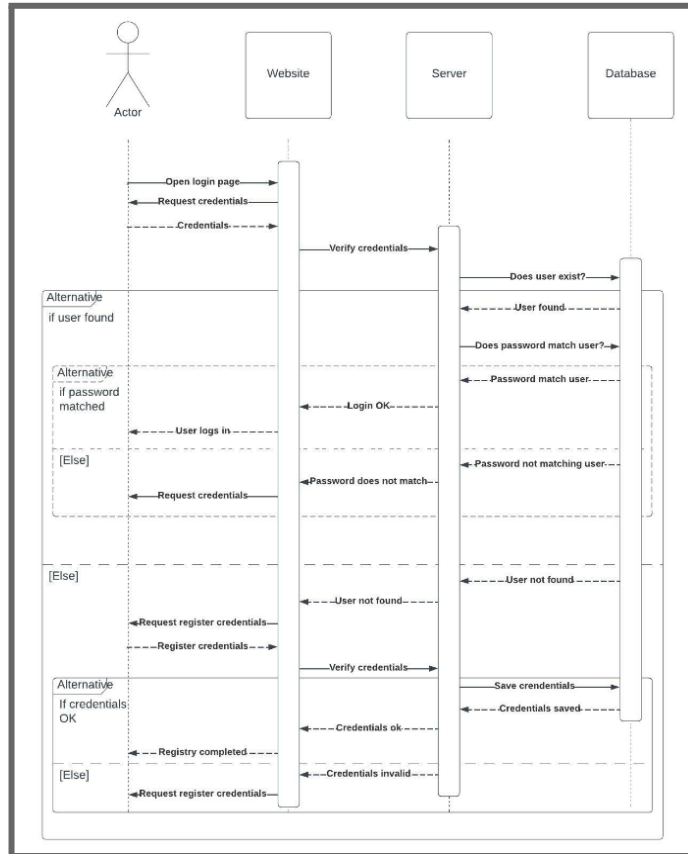
mikäli ei;

käyttäjältä kysytään rekisteröitymistiedot, jotka server validoi.

Rekisteröitymistietojen ollessa oikein ne kirjataan databaseen, muulloin niitä kysytään uudelleen.

mikäli kyllä;

Server vertaa käyttäjän antamaa salasanaa databasessa olevaan salasanaan ja jos salasana on oikein kirjautuminen hyväksytään, muulloin kirjautuminen hylätään ja käyttäjältä kysytään kirjautumistietoja uudelleen.



Kuva 5

7 Kehitysprosessi ja kehitysvaiheen tekniikat

Front-end

- React.js käyttäen Typescript ja tukevia kirjastoja:
- react-dom-router for routing.
- tailwindcss for a css library
- Google Icons for some icons.
- Animate CSS for some css animations.

Back-end

- ASP.NET käyttäen .NET Core 3.1 kirjastoa C#-ohjelmointikielen kanssa.

Tietokanta

- Tietokanta on tehty MySQL-tietokanta mysql.metropolia.fi-palvelimella

8 Yhteenveto

Tavoitteemme ohjelmistotuotantoprojekti ykkösessä oli saada toteutettua Kirjautumis-, Etu- ja Profiilisivut ja niihin sopiva visuaalinen ilme ja jäsenitys, sekä toiminnallisuudet kuten rekisteröityminen, kirjautuminen, postauksen teko, kommentointi sekä pelin lisäys ja muiden postauksien ja profiilien tarkastelu.

Ekstrana saimme tehtyä tykkäykset postauksiin.

Ohjelmistotuotantoprojektin toisessa osassa on tarkoitus saada lisättyä:

API, josta pelit haetaan, ja lisätä sivulle haku, josta voi hakea pelejä, käyttäjiä, jotka pelaavat haettua pelejä sekä käyttäjiä.

Ylläpitäjä-käyttäjä, jolla on oikeuden poistaa kaikkien käyttäjien postauksia, pelejä ja kommentteja, sekä antaa jäähyjä käyttäjille tai poistaa käyttäjiä kokonaan.

Salasanojen hashays ja Json Web Token lisätään.

Myös sivuston ulkonäköä ja toiminnallisuutta parannetaan.