

Quacker -projektiraportti

Tuotekuvaus ja ominaisuudet

Selaimella avattava keskustelupalsta, jossa voi seurata muiden käyttäjien julkaisuja ja julkaista itse viestejä. Julkaisuja voi myös tykätä. Tuotteeseen voi luoda uuden käyttäjätilin, johon voi valita profiilikuvan. Käyttäjät voivat lisätä toisia käyttäjiä kaverilistalle, jolloin heidän julkaisut näkyvät omalla, julkaisujen selaamiseen tarkoitettulla palstalla ensimmäisenä. Käyttäjillä on myös feedi mistä näkyvät kaikkien käyttäjien julkaisut. Julkaisujen sisältö voi olla tekstiä, emojiä, kuvia, linkkejä tai videoita. Omia julkaisuja voi myös poistaa tai muokata.

Toteutetut ominaisuudet

- Sivulle kirjautuminen
- Sivulle rekisteröityminen
- Quackin postaaminen
- Media Quackissa (kuvat, audio, video)
- Käyttäjien seuraaminen
- Quack feedi
- Feedistä pystyy suodattamaan pelkät seurattujen henkilöiden postaukset
- Salasanan vaihto, sähköpostin vaihto
- Quackien tykkääminen
- Omien quackien poistaminen
- Käyttäjätilin poistaminen
- Komponenttien asettelu
- Quackien haku
- GDPR
- Lokalisaatio
- Käyttäjätietojen muokkaaminen
- Käyttäjä saa tiedon uusista quackeista
- Käyttäjien blokkaminen (blokkattujen käyttäjien quackit ei näy feedissä)
- Viestien lähettäminen ylläpitoon

Toteutumattomat ominaisuudet

- Quackien kommentointi
- Quackien jakaminen
- Käyttäjien hakutoiminto frontendiin

Palvelimen asennus

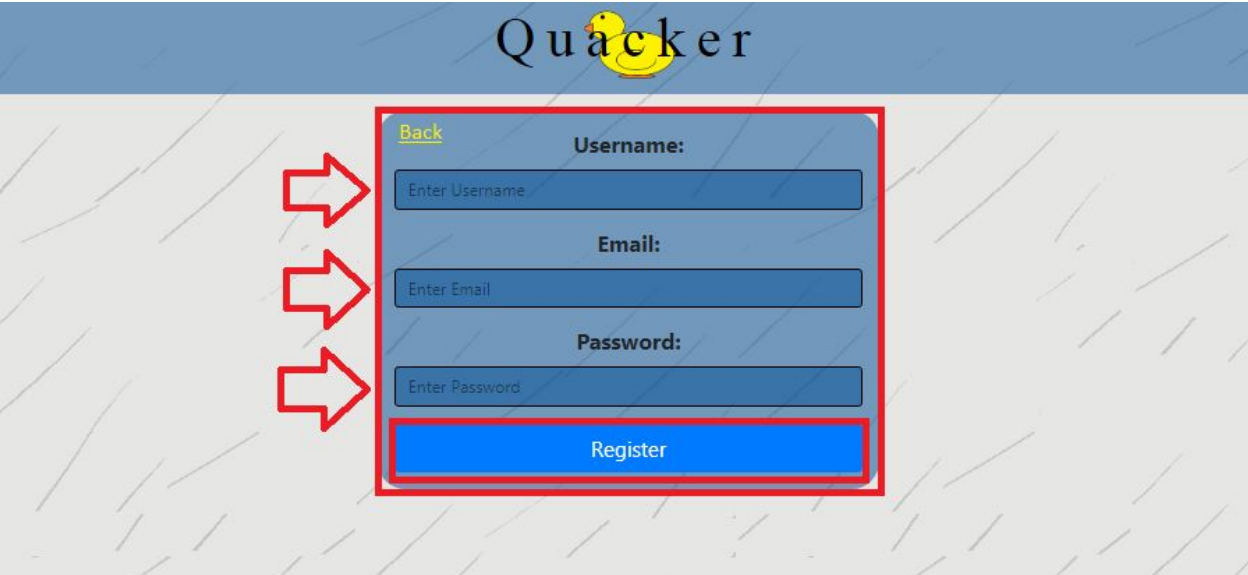
Koska sovellus pohjautuu maveniin, sen koonti lähdekoodista on hyvin suoraviivaista. Linux järjestelmässä vaaditaan riippuvuutena Javan kehitysympäristön versio 1.8 ja maven. Kloonaa koodikanta GitHubista ja luo jonnekin kansio käyttäjien tiedostojen tallentamiseen. Aseta käyttäjien tiedostot sisältävän kansion absoluuttinen polku ympäristömuuttujaksi avaimella QUACKER_FILE_STORAGE.

Seuraavaksi siirry kloonatun koodihakemiston juureen ja aja komento *run mvn clean package*. Kun toiminto on valmis, aja komento *run java -jar target/quacker-0.0.1-SNAPSHOT.jar*. Palvelin on nyt käynnissä. Muista avata portti 8080 palomuurista, jotta käyttäjät voivat yhdistää palveluun.

Käyttäjän ohjeistus

Tässä osiossa on kuvailtu erilaisia projektin käyttöön liittyviä, yleisiä käyttötapauksia, joita suurin osa sovelluksen käyttäjistä tulee toteuttamaan.

1.Rekisteröinti



Käyttäjän täytyy rekisteröityä Quackeriin ennen sen käyttöä. Rekisteröinti sivulle pääsee [sisäänkirjautumis](#) sivulla olevasta rekisteröinti (Register) napista. Syötä käyttäjänimi (Username), sähköpostiosoite (Email) sekä salasana (Password) ja paina rekisteröinti (Register) nappulaa luodaksesi uuden käyttäjän Quackeriin. Jos käyttäjällä on jo tunnukset Quackeriin, paina vasemmalla ylälaidassa olevasta takaisin (Back) -linkistä ja kirjaudu sisään.

2.Kirjautuminen

The image shows a web form for the 'Quacker' application. At the top, the word 'Quacker' is displayed with a yellow duck icon replacing the 'a'. Below the header, there is a login and registration section. It contains two input fields: 'Username:' with a placeholder 'Enter Username' and 'Password:' with a placeholder 'Enter Password'. Below these fields is a blue 'Sign in' button. Underneath the 'Sign in' button is the text 'OR' and a grey 'Register' button. A red rectangular box highlights the 'Sign in' and 'Register' buttons. Two red arrows point from the left towards the 'Sign in' button.

Käyttäjän täytyy kirjautua sisään käyttääkseen Quackeria ja sen ominaisuuksia. Kun Quackerin sisäänkirjautuminen avautuu ensimmäisenä sivuna. Jos käyttäjällä on jo käyttäjätunnukset syötä käyttäjänimi (Username) sekä salasana (Password) niille tarkoitetuille kentille ja paina kirjautumis nappia (Sign in). Jos käyttäjällä ei ole tunnuksia, luo uudet käyttäjätunnukset painamalla rekisteröinti (Register) nappia. Tarkemmat ohjeet [rekisteröinti](#) osiossa.

Testaaminen

Testauksessa keskityttiin integraatio ja yksikkötesteihin. Integraatiotesti varmistaa että sovellus käynnistyy normaalisti ja yksikkötestit testaavat yksittäisten endpointtien toimintaa. Yksikkötesteissä kaikki riippuvuudet mockataan mockitolla ja Spring-framework ohjeistetaan käynnistämään vain se controller jonka endpointteja testataan. Testitulosten varmentamisessa käytettiin hamcrestiä.

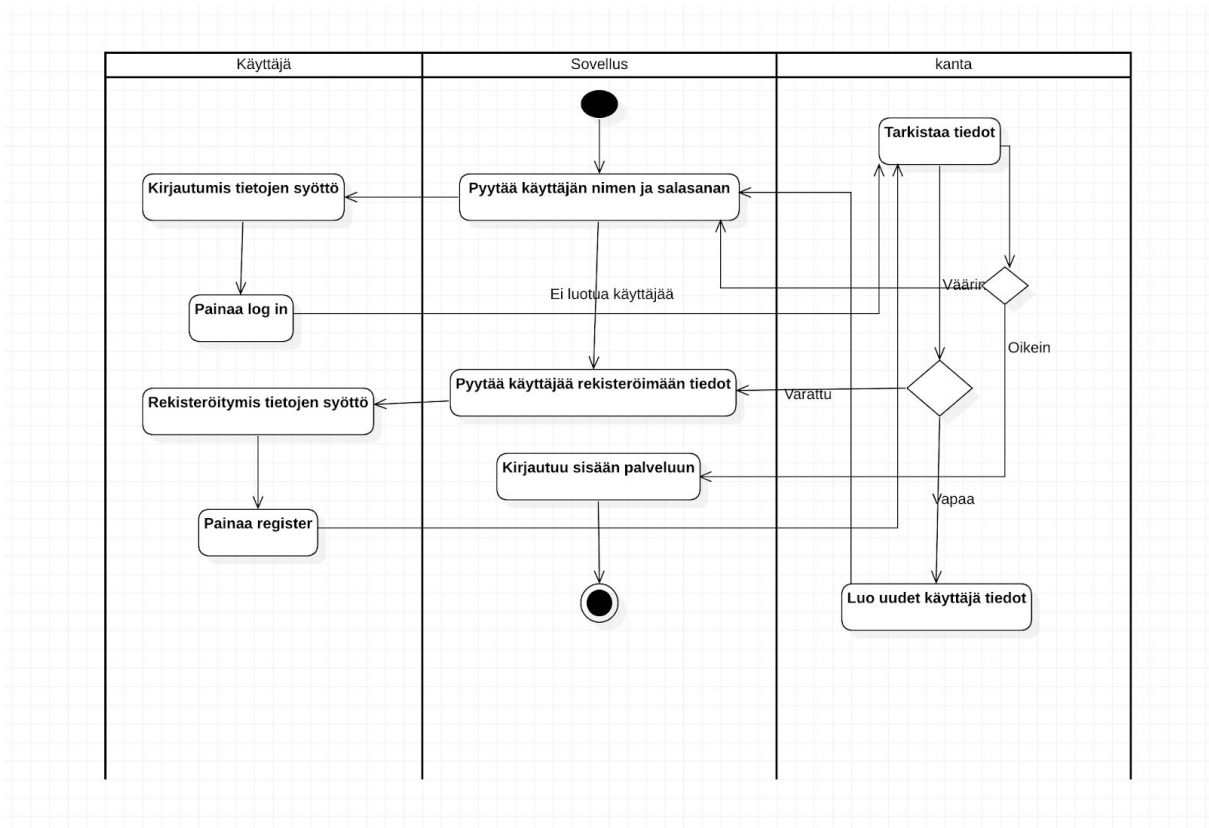
Arkkitehtuurinäkökulma

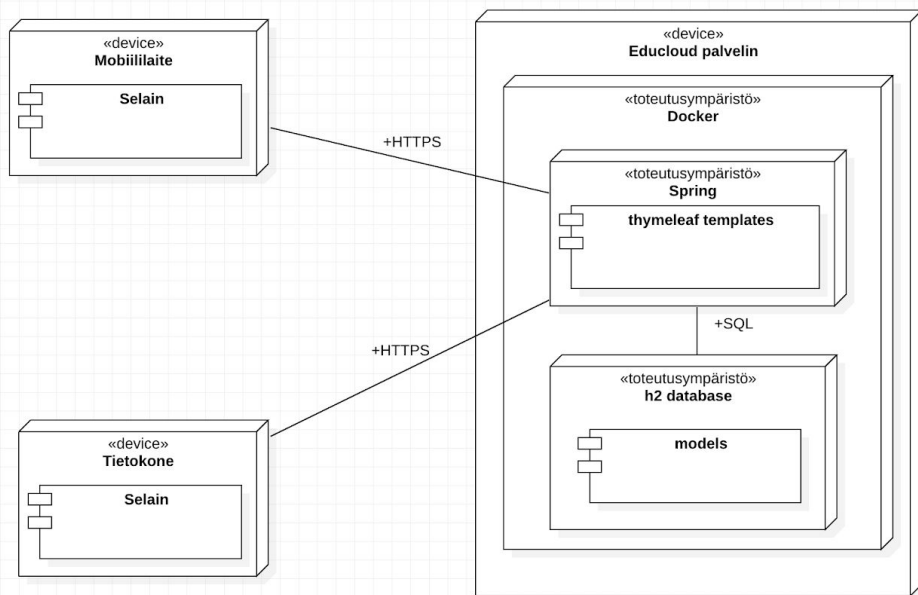
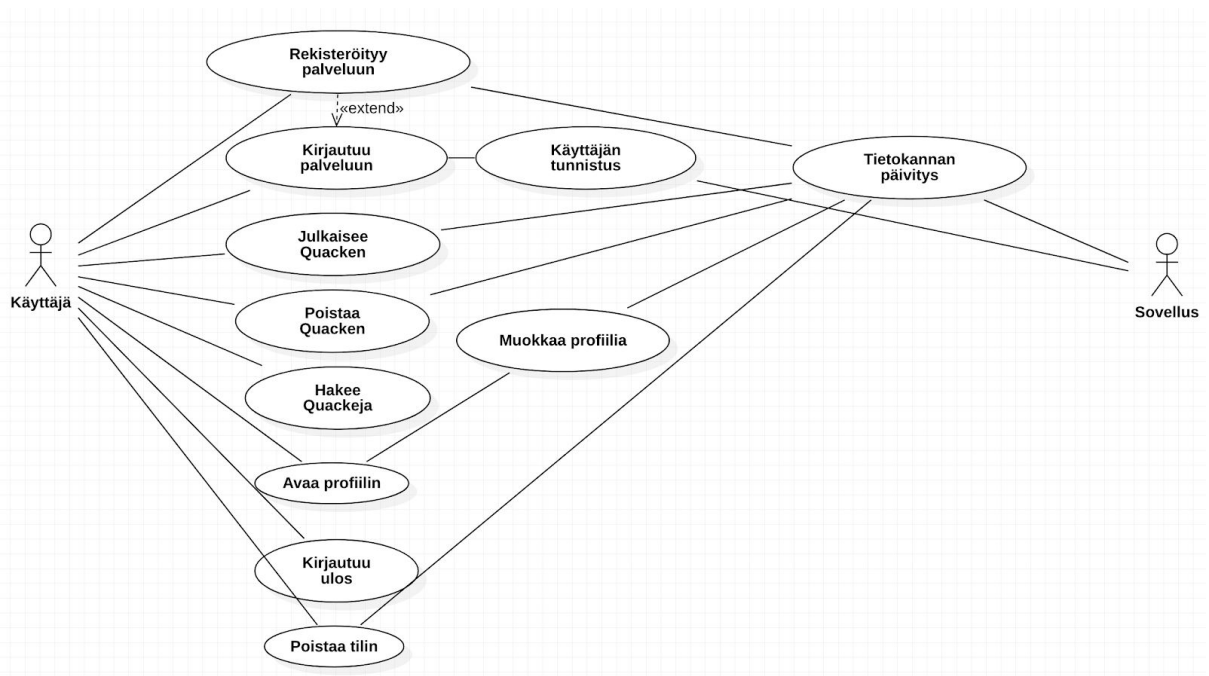
Quacker on niin sanotusti full stack -toteutus. Tällä tarkoitetaan, että kokonaisuus rakentuu tietokannasta, backend-sovelluskoodista sovelluspalvelimen kera ja web-käyttöliittymästä frontendinä. Toteutuksessa on hyödynnetty seuraavia teknologioita:

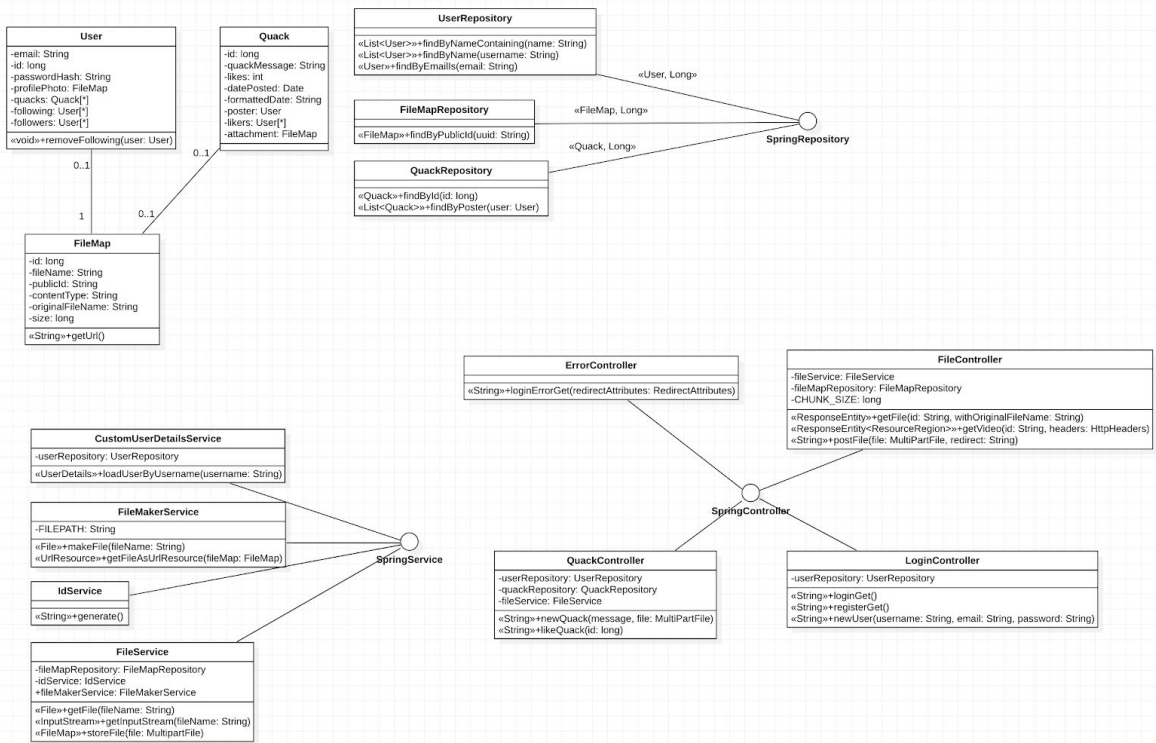
- Java
- Maven
- Thymeleaf
- Spring Boot
- H2
- Jackson

-
- ```
graph TD
 subgraph Swimlanes
 direction TB
 subgraph Käyttäjä
 K1[Kirjautumistietojen syöttö]
 K2[Quack viestin täyttö]
 K3[Submit nappulan painaminen]
 K4[Logout nappin painaminen]
 end
 subgraph Käyttöliittymä
 K5[Kirjautumissivun näyttö]
 K6[Kirjautumistietojen tarkastus]
 K7[Päänäkymän näyttö]
 K8[Päänäkymän näyttö]
 end
 subgraph Palvelin
 K9[Käyttäjän haku]
 K10[Käyttäjän tietojen haku]
 K11[Quack viestin tallennus]
 K12[Käyttäjän kirjaaminen ulos]
 end
 subgraph Tietokanta
 K13[Uuden olion tallentaminen tietokantaan]
 end
 end

 Start(()) --> K5
 K5 --> K1
 K5 --> K9
 K1 --> K6
 K9 --> K10
 K10 --> K6
 K6 -- "[tiedot väärinä]" --> K5
 K6 -- "[tiedot oikein]" --> K7
 K7 --> K2
 K2 --> K3
 K3 --> K11
 K11 --> K13
 K13 --> K8
 K4 --> K12
 K12 --> End((()))
```







#### interaction Sekvenssikaavio rekisteröityminen

