

Johdanto:

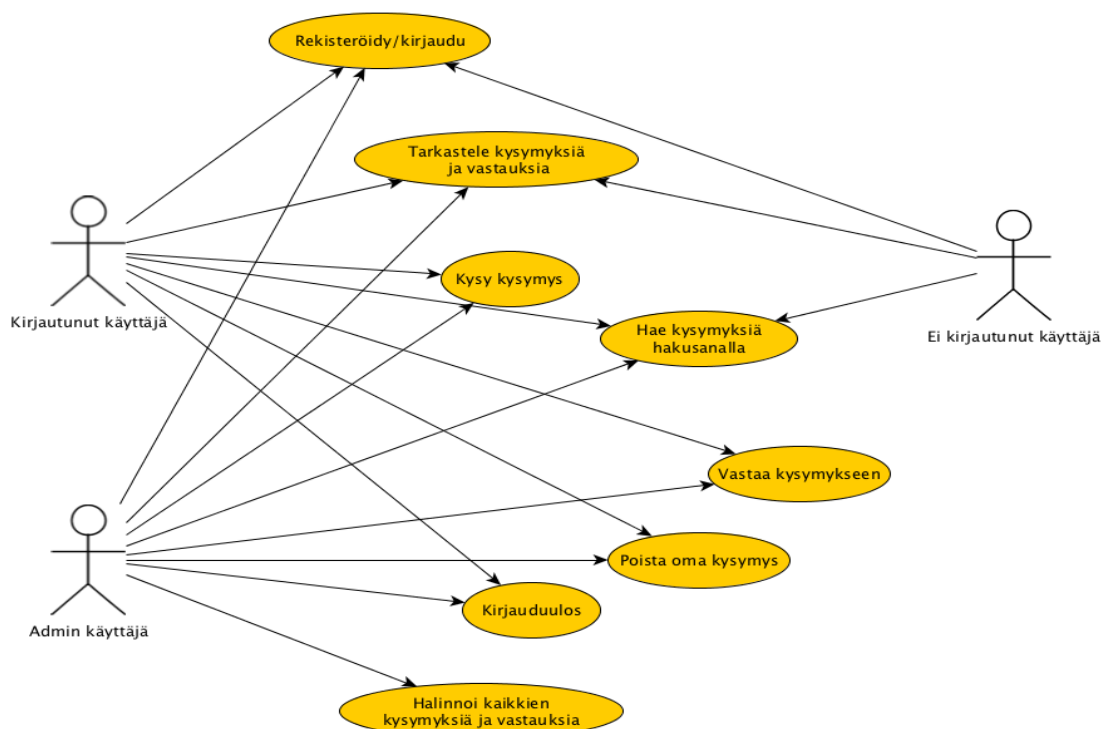
AnswerHub-sivustolla käyttäjä voi kysyä ja vastata kysymyksiin. Sivusto vaatii kysymysten kysymiseen ja niihin vastaamiseen kirjautumisen. Kysymyksiä ja vastauksia voi kuitenkin katsella ilman kirjautumista. Jokaisella käyttäjällä on oma profiilisivu.

Kysymyksillä on oma sivu, jossa näkyy vastaukset kysymykseen. Kysymyksellä ja vastauksilla on tähti "arvosana", joka kuvaa kuinka hyödylliseksi kysymys on yleensä koettu. Jokaiseen kysymykseen voidaan hyväksyä yksi vastaus. Kysymyksien ja vastausten yhteydessä on kysymyksen tai vastauksen kirjoittaja.

Jokaiseen kysymykseen liittyy tagi. Tagillä on mahdollista hakea samaan aiheeseen liittyviä kysymyksiä. Etusivulla näkyy viimeisimmät vastaamattomat kysymykset. Etusivulla on myös hakupalkki, jonka avulla on mahdollista etsiä kysymyksiä hakusanan perusteella.

Ohjelma toteutetaan käyttäen MySQL-tietokantaa, back-end ohjelmointikielenä käytetään php:tä ja front-end toteuttamiseen käytetään javascriptiä. Javascriptin kanssa käytetään kirjastona Angularjs:ää.

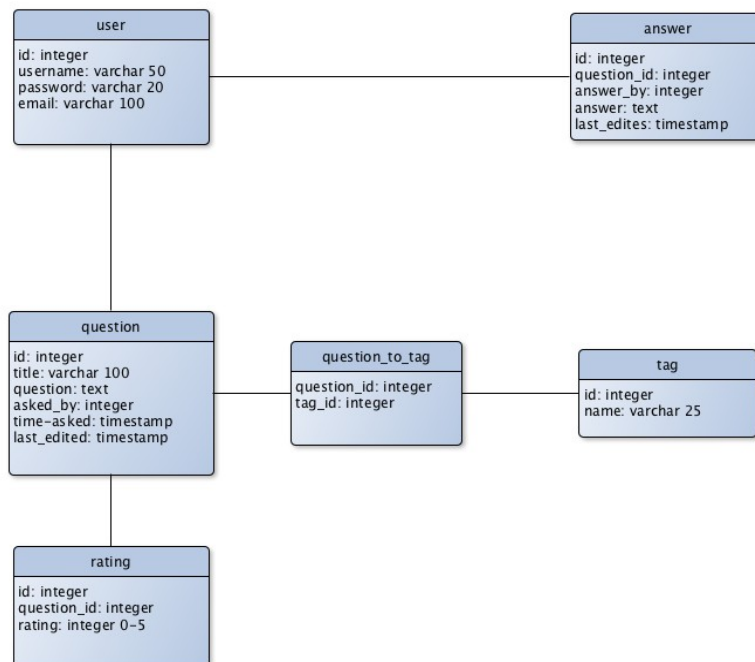
Käyttötapaukset



Kirjautunut käyttäjä voi kysyä ja vastata kysymyksiin. Sivustolla voi tarkastella kysymyksiä ja vastauksia ilman kirjautumista, mutta tällöin ei voi kysyä kysymyksiä tai vastata niihin. Admin-käyttäjä voi lisäksi poistaa ja muokata kysymyksiä. Normaali käyttäjä voi vain poistaa omia kysymyksiään. Kysymystä voi muokata 15 minuuttia sen kysymisestä (paitsi admin käyttäjä). Sivuston käyttäjät voivat hakea kysymyksiä hakupalkin avulla.

Tietokanta

Ohjelma käyttää mysql tietokantaa. Relaatiotietokannan rakenne on seuraavan kuvan mukainen.



Seuraavana on esitetty create-table -lauseet, jotka ajettiin tietokantataulujen luomiseksi. Lauseista käy ilmi taulujen tarkempi rakenne, kuten mitkä rivit ovat avaimia ja mitkä rivit viittaavat mihin.

CREATE TABLE users

```
(
    id int NOT NULL UNIQUE AUTO_INCREMENT,
    username varchar(50) NOT NULL UNIQUE,
    password varchar(20) NOT NULL CHECK(password.length >= 8 AND password.length <=
20),
    email varchar(100) NOT NULL,
    PRIMARY KEY(id)
);
```

CREATE TABLE questions

```
(
    id int NOT NULL UNIQUE AUTO_INCREMENT,
    title varchar(100) NOT NULL,
    question text NOT NULL,
    asked_by int NOT NULL,
    time_asked timestamp NOT NULL,
    last_edited timestamp NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY(asked_by) REFERENCES users(id)
);
```

CREATE TABLE answers

```
(
    id int NOT NULL UNIQUE AUTO_INCREMENT,
    question_id int NOT NULL,
```

```
    answer_by int NOT NULL,  
    answer text NOT NULL,  
    time_answered timestamp NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY(answer_by) REFERENCES users(id),  
    FOREIGN KEY(question_id) REFERENCES questions(id)  
);
```

```
CREATE TABLE tags  
(  
    id int NOT NULL UNIQUE AUTO_INCREMENT,  
    name varchar(25) NOT NULL,  
    PRIMARY KEY(id)  
);
```

```
CREATE TABLE questions_to_tags  
(  
    question_id int NOT NULL,  
    tag_id int NOT NULL,  
    FOREIGN KEY(question_id) REFERENCES questions(id),  
    FOREIGN KEY(tag_id) REFERENCES tags(id)  
);
```

```
CREATE TABLE ratings  
(  
    id int NOT NULL UNIQUE AUTO_INCREMENT,  
    question_id int NOT NULL,  
    rating int NOT NULL CHECK(rating >= 0 AND rating <= 5),  
    PRIMARY KEY(id),  
    FOREIGN KEY(question_id) REFERENCES questions(id)  
);
```