Dokumentaatio

1. Johdanto

Järjestelmä on hieman modifioitu versio ehdotetusta keskustelufoorumi-aiheesta. Järjestelmä on toiminnallisuudeltaan samantyylinen kuin minkälainen foorumi tahansa, eli tukee käyttäjän rekisteröitymistä, tunnistautumista ja osallistumista keskusteluihin. Luonnollisesti rekisteröityneellä käyttäjällä on myös mahdollisuus aloittaa uusi keskustelu (n.s. *thread*).

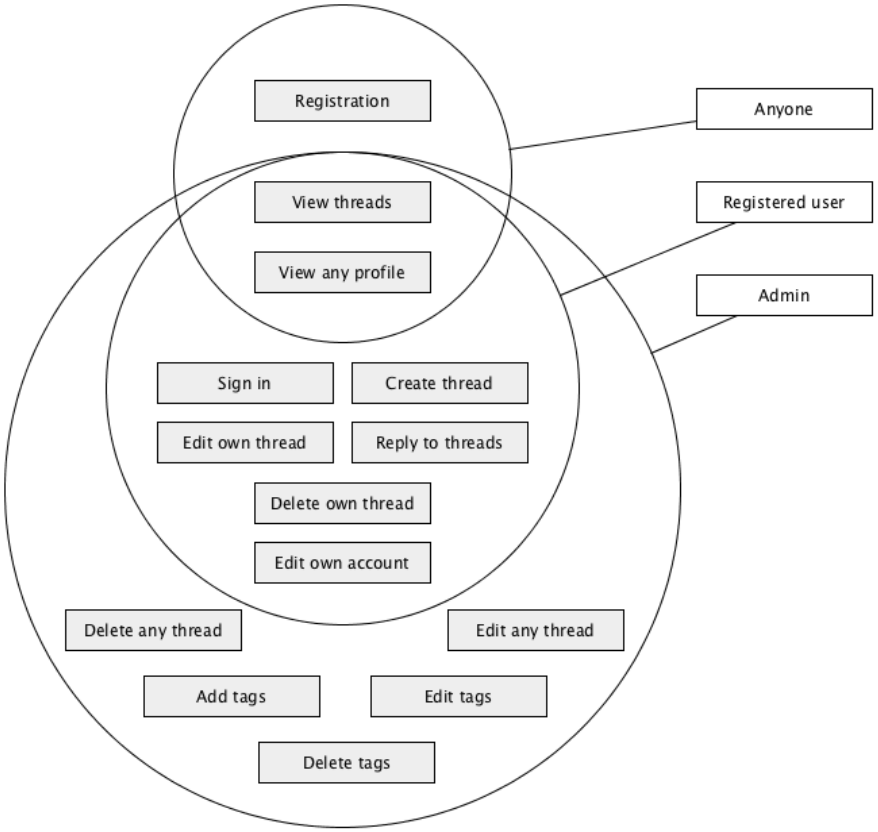
Sovellus toteutetaan seuraavilla teknologioilla:

Deployment OS	**Ubuntu 16.04**
Web server	**nginx + gunicorn**
Database	**PostgreSQL**
Language	**Python 3**
Framework	**Flask**
Templating	**jinja2**
Environment	**virtualenv**
Package manager	**pip**

Käyttöjärjestelmän voi mahdollisesti vaihtaa toiseen, kunhan sillä löytyy tuki muille teknologioille – toisin sanoin melkein mihin tahansa yleiseen *nix järjestelmään. Tietokannan voi käytännössä vaihtaa toiseen, mutta kaikki Postgres-kohtaiset SQL-komennot tulisi kääntää, ja tietokantakyselyitä suorittava abstraktio tulisi myös toteuttaa uudestaan, eli en suosittelisi vaihtamista.

2. Yleiskuvaus

Käyttötapauskaavio:



Käyttäjäryhmien selitykset:

Anyone	**Kuka tahansa.**
Registered user	**Järjestelmään rekisteröitynyt käyttäjä.**
Admin	**Järjestelmänvalvoja tai pääkäyttäjä, eli siis erityisillä oikeuksilla varustettu käyttäjä.**

Kuten kaavio esittää, kuka tahansa voi rekisteröityä käyttäjäksi, lukea keskusteluja (sekä vastauksia) ja tarkastella minkä tahansa käyttäjän julkisia tietoja. Rekisteröitynyt käyttäjä voi lisäksi kirjautua järjestelmään, ja kirjautuneena päivittää omia tietojaan, luoda uusia keskusteluja, vastata mihin tahansa keskusteluun, sekä poistaa käyttäjän itse luotuja keskusteluja. Järjestelmänvalvojalla on kaiken tämän lisäksi mahdollisuus lisätä, muokata ja poistaa aiheita (n.s. *tags*), poistaa mitä tahansa keskusteluja, sekä muokata minkä tahansa käyttäjän julkista profiilia.

Kenen tahansa käyttötapaukset

Käyttäjätunnuksen rekisteröiminen.
Foorumin keskusteluketjujen selaaminen: Kuka tahansa voi selailla keskusteluketjujen listaa sekä tarkastella keskusteluketjujen sisältöä.
Käyttäjien profiilien tarkasteleminen: Kuka tahansa voi tarkastella minkä tahansa käyttäjän profiilia, josta löytyy muun muassa käyttäjän julkiset tiedot ja viimeisimmät käyttäjän aloittamat keskusteluketjut.

Kirjautuneen käyttäjän käyttötapaukset

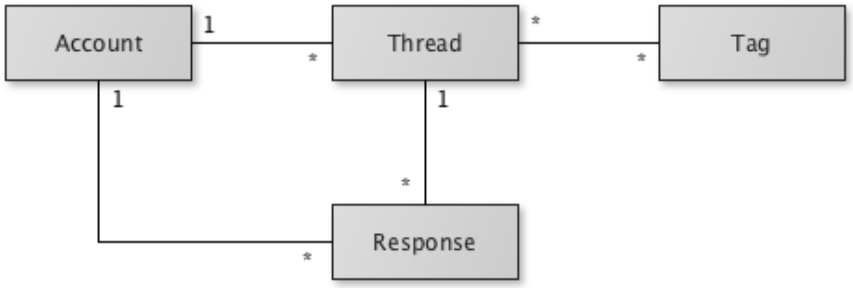
Vuorovaikuttaminen foorumin keskusteluketjuissa: Kirjautunut käyttäjä voi lisätä vastauksia keskusteluihin, ja halutessaan myös aloittaa uusia keskusteluketjuja joihin muut käyttäjät voivat osallistua.
Omien tietojen päivittäminen: Kirjautunut käyttäjä voi päivittää oman julkisen nimensä joka näkyy käyttäjätunnuksen sijasta esimerkiksi keskusteluketjuissa johon käyttäjä on osallistunut.
Omien keskusteluketjujen muokkaaminen: Kirjautunut käyttäjä voi myös päivittää omien keskusteluketjujen otsikkoa ja aiheita, tai kokonaan poistaa omia keskusteluketjuja.

Pääkäyttäjän käyttötapaukset

Minkä tahansa keskusteluketjun muokkaaminen: Pääkäyttäjä voi päivittää kenen tahansa aloittamien keskusteluketjujen otsikkoa ja aiheita, tai kokonaan poistaa minkä tahansa keskusteluketjun.
Keskusteluaiheiden hallinnoiminen: Pääkäyttäjä voi lisätä, muokata sekä poistaa keskusteluaiheita. Keskusteluaiheet ovat julkisia, ja kuka tahansa kirjautunut käyttäjä voi liittää mitä tahansa keskusteluaiheita keskusteluketjuihinsa.

3. Järjestelmän tietosisältö

Käsitekaavio:



Tietokohteiden selitykset:

Account

Attribuutti	Arvojoukko	Kuvailu
id	Kokonaisluku	Käyttäjätilin yksikäsitteinen tunniste.
username	Merkkijono	Käyttäjänimi jota käytetään kirjautuessa.
password	Merkkijono	Salasanasta laskettu hajautusarvo.
display_name	Merkkijono	Julkinen nimi.
admin	Totuusarvo	Totuusarvo joka viittaa käyttäjän oikeuksiin.

Rekisteröidyillä käyttäjillä on yksi käyttäjätili, jonka tietoja käytetään pääasiallisesti kirjautumisen toteuttamiseen.

Tag

Attribuutti	Arvojoukko	Kuvailu
id	Kokonaisluku	Avainsanan yksikäsitteinen tunniste.
title	Merkkijono	Itse avainsana.

Avainsanat ovat olemassa keskustelujen kategorisointia varten. Yksi avainsana voi liittyä moneen keskusteluketjuun, ja yhteen keskusteluketjuun voi liittyä monta avainsanaa.

Thread

Attribuutti	Arvojoukko	Kuvailu
id	Kokonaisluku	Keskusteluketjun yksikäsitteinen tunniste.
author_id	Kokonaisluku	Viiteavain joka liittyy käyttäjätilin tunnisteeseen.
title	Merkkijono	Keskusteluketjun otsikko.
created	Aikaleima	Keskusteluketjun aikaleima.

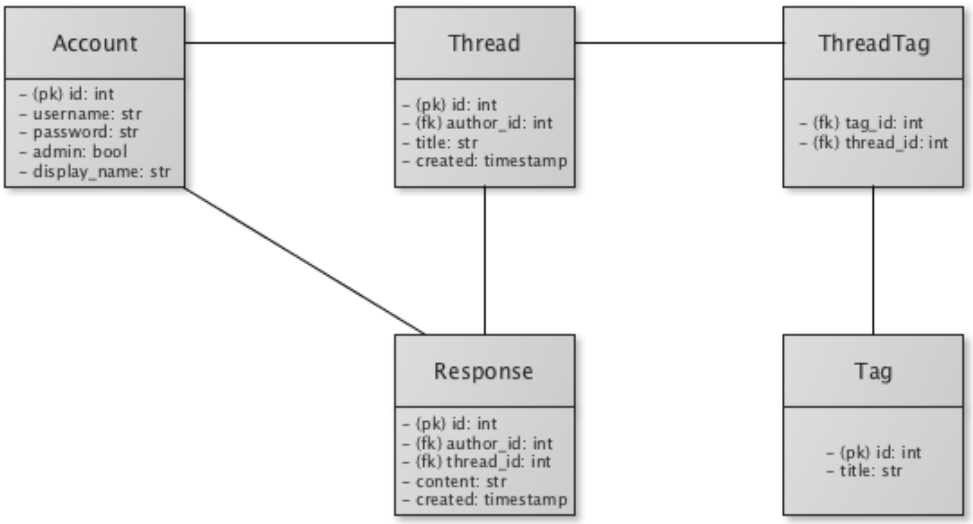
Keskusteluketju liittyy yhteen käyttäjään, mutta yksi käyttäjä voi luoda useita keskusteluketjuja.

Response

Attribuutti	Arvojoukko	Kuvailu
id	Kokonaisluku	Vastauksen yksikäsitteinen id.
author_id	Kokonaisluku	Viiteavain joka liittyy käyttäjätilin tunnisteeseen.
thread_id	Kokonaisluku	Viiteavain joka liittyy keskusteluketjun tunnisteeseen.
content	Merkkijono	Vastauksen tekstillinen sisältö.
created	Aikaleima	Vastauksen aikaleima.

Vastaus liittyy yhteen keskusteluketjuun ja yhteen käyttäjään. Käyttäjällä voi kuitenkin olla useita vastauksia sekä samassa keskusteluketjussa että muissa ketjuissa.

4. Tietokantakaavio



5. Järjestelmän yleisrakenne

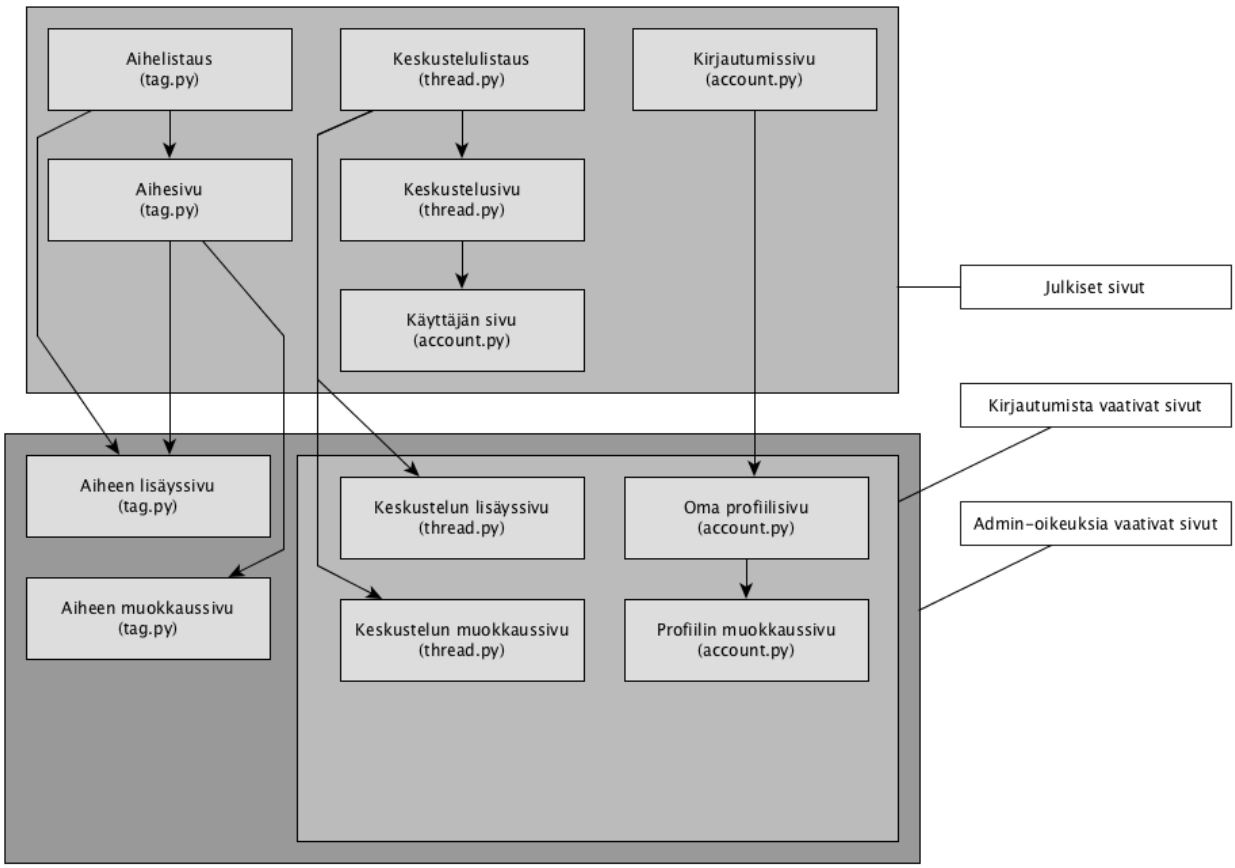
Kaikki ohjelman lähdekoodi on hakemistossa *src*, jonka sisältä löytyy hakemistot MVC-mallin mukaisille tiedostoille: kontrollereille (*controllers*), malleille (*models*) ja näkymille (*templates*). Myös hakemistot staattisille tiedostoille (*static*), apuluokille (*utilities*), sekä itse sovellus (*app.py*) että sovelluksen jaetut instanssit (*shared.py*) löytyvät täältä.

Kaikki hakemistot jotka sisältävät Python-tiedostoja ovat määriteltyjä Python-paketeiksi, josta tiedostot nimeltä *__init__.py* johtuu. Lisätietoja aiheesta voi lukea Pythonin dokumentaatiosta.

Järjestelmän käyttöönotossa alustetaan virtuaaliympäristö (esim. *virtualenv*) jonka vastuulla on ylläpitää projektissa käytettäviä kirjastoja sekä Python-runtime:a joka ajaa itse sovellusta. Tästä aiheesta löytyy lisätietoja kappaleessa 7. Sovelluksessa käytettyjen kirjastojen nimet ja versiot löytyvät tiedostosta *requirements.txt*.

Kaikki reititys ja sessio-validointi on toteutettu suoraan sovellustiedostoon *app.py*, Flask-kirjaston *import*-lauseiden minimoimiseksi. Tiedosto *shared.py* alustaa sovelluksen eri puolilla käytettyjä jaettuja instansseja. Tässä tiedostossa voidaan määritellä tarvittavat asetukset, eli tietokannan tiedot (nimi, käyttäjänimi, salasana, osoite ja portti) ja sovelluksen salainen avain.

6. Käyttöliittymä ja järjestelmän komponentit



7. Asennustiedot

Sovellus on projektin kehityksen aikana ollut asennettuna virtuaalipalvelimella jossa toimintaympäristönä on ollut Ubuntu 16.04 LTS. Web-palvelimena on toiminut gunicorn ja proxy:na nginx, mutta sen syvällisemmin näiden asennusta ei käsitellä tässä dokumentissa. Palvelimella on ollut käytössä Python 3.5, pip3.5 ja virtualenv 15.1.0, joten tämä ohje olettaa kyseisten ohjelmistojen olevan vähintään tällä tasolla.

Aivan ensimmäisenä asennusta varten täytyy ladata Python ja pip, mutta tähän prosessiin löytyy ohjeita useasta lähteestä, joten tämä askel jää lukijan suoritettavaksi. Kun edellä mainitut ohjelmistot on asennettu, asennetaan virtualenv suorittamalla seuraava komento:

```
~$ pip install virtualenv
```

Tämän jälkeen navigoidaan projektihakemiston juureen ja alustetaan virtuaaliympäristö (tässä annetaan ympäristön nimeksi *venv*, mutta tämän voi vaihtaa oman mielen mukaan):

```
~/tsoha$ virtualenv venv
```

Kun virtuaaliympäristö on luotu tarkistetaan että ympäristön Python-versio on oikein:

```
~/tsoha$ find venv/bin/ -name 'python[23].*'
```

Jos versio on 3.5 tai korkeampi niin kaikki on kunnossa, muutoin kannattaa alustaa virtuaaliympäristö määrittelemällä käytettävän kääntäjän polku eksplisiittisesti:

```
~/tsoha$ virtualenv -p /path/to/python3.5 venv
```

Kun kaikki on kunnossa aktivoidaan virtuaaliympäristö, asennetaan kirjastot, ja deaktivoidaan ympäristö:

```
~/tsoha$ source venv/bin/activate
(venv)~/tsoha$ pip install -r requirements.txt
...
(venv)~/tsoha$ deactivate
```

Tässä vaiheessa olisi suotavaa asettaa tietokannan tiedot kuntoon muokkaamalla tiedostoa *shared.py* jossa tietokanta-instanssi luodaan. Esimerkiksi, jos tietokanta on suojaamaton, konfiguroimaton ja nimeltään *tsoha* niin kyseisen rivin tulisi näyttää tältä:

```
db = database.DatabaseManager(database="tsoha")
```

Tämän jälkeen voidaan käynnistää sovellus paikallisesti:

```
~/tsoha$ venv/bin/python wsgi.py
```

Jos käynnistys onnistui ongelmitta, niin seuraavanlainen rivi pitäisi tulostua konsoliin:

```
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Sovellus on siis paikallisesti käynnissä tulostuksessa esiintyvässä osoitteessa, eli sovellusta voi nyt käyttää menemällä kyseiseen osoitteeseen millä tahansa selaimella.

Todelliseen käynnistykseen käytetään gunicorn:ia, kiinnitetään sovellus UNIX-socket:iin ja konfiguroidaan nginx reitittämään kaikki pyynnöt kyseiseen socket:iin. Tällöin käynnistys onnistuu esimerkiksi seuraavaasti:

```
~/tsoha$ venv/bin/gunicorn --workers 2 --bind unix:tsoha.sock wsgi:app
```

Yllä mainittu konfigurointi jää kuitenkin lukijan selvitettäväksi ja pohdittavaksi – vaihtoehtoisesti palvelimet voi myös vaihtaa oman maun mukaan.

8. Käynnistys- ja käyttöohje

Sovellus on saatavilla osoitteessa http://viela.pw ja kirjautuminen onnistuu seuraavilla tiedoilla:

Käyttäjänimi	Salasana	Kuvaus
admin	12345	Pääkäyttäjä.
test	12345	Tavallinen käyttäjä.

Sivu kirjautumista varten löytyy oikeasta yläkulmasta, ja kyseisellä sivulla löytyy myös linkki uuden käyttäjätunnuksen rekisteröintiin. Kirjautumisen jälkeen yläkulman kirjatumispainikkeen tilalle tulee painike jonka takaa löytyy oma profiili.